A Deep Learning Technique for Web Phishing Detection Combined URL Features and Visual Similarity

Saad Al-Ahmadi[1] and Yasser Alharbi[2]

[1]College of Computer and Information Science, Computer Science Department,
King Saud University, Riyadh, Saudi Arabia
[2]College of Computer and Information Science, Computer Engineering Department,
King Saud University, Riyadh, Saudi Arabia

Abstract

The most popular way to deceive online users nowadays is phishing. Consequently, to increase cybersecurity, more efficient web page phishing detection mechanisms are needed. In this paper, we propose an approach that rely on websites image and URL to deals with the issue of phishing website recognition as a classification challenge. Our model uses webpage URLs and images to detect a phishing attack using convolution neural networks (CNNs) to extract the most important features of website images and URLs and then classifies them into benign and phishing pages. The accuracy rate of the results of the experiment was 99.67%, proving the effectiveness of the proposed model in detecting a web phishing attack.

Keywords

Phishing detection, URL, visual similarity, deep learning, convolution neural network.

1. Introduction

The deployment and use of website services has increased rapidly in recent years and this growth has created a high level of risk for internet users globally. Web phishing attacks are the most severe and the most common cybersecurity threat over the internet. Web phishing can be described as a way to deceive and steal users' sensitive personal information through a phished webpage that looks similar to a valid web page. Phishers use a diversity of techniques in order to execute an effective deception.

The lack of awareness of phishing attacks by internet users and innovative phishing methods highlight the need for efficient technology-based detection techniques. The setting up an appropriate mechanism to detect spoofed websites would be vital in reducing the probable enormous damage caused to internet users [1].

In this research, we propose a technique for recognizing web phishing using two convolution neural networks (CNNs) based on websites image and URL. The first CNN is used to extract the website URL features and identify whether the webpage is a phishing webpage or a benign webpage. The second CNN is used to simultaneously extract the visual features of the webpage and classify the website as either legitimate or malicious. The results of the first and the second CNN are combined and on the basis of the outcome it is decided whether the webpage should be reported as a phishing webpage. The motivation to use the URL based approach with visual

DOI: 10.5121/ijcnc.2020.12503

similarity based method is that attackers generally use a fake URL that looks similar to a legitimate URL and deploy webpages that look visually similar to legitimate websites with the goal of deceiving internet users to obtain sensitive information.

This paper is structured as follows: section 2 presents some background information; Section 3 reviews research on detecting phishing attacks; Section 4 describes the method used; Section 5 presents the results of the experiment and comparison with other research; while Section 6 presents the conclusion.

2. BACKGROUND

The impact of a phishing attack on an internet user is significant, causing vital damage and financial loss when their confidential information is stolen. Hence, deploying an appropriate anti-phishing technique is crucial. Web phishing detection mechanisms can be categorised under five approaches: firstly, the whitelist based approach, which is a way to classify phishing websites depending on a predefined list of legitimate websites. This approach is ineffective since it is impossible to have a whitelist that contains all the legitimate websites in the world. Secondly, the Blacklist based approach, which uses a predefined list of anonymous websites. This list needs to be updated very frequently to make it possible to detect all newly created phishing pages, which makes the zero-hour phishing attack a major issue in this approach [2],[3]. Thirdly, the Content based approach that relies on the component of the webpage that needs to be extracted to use in retrieving the corresponding legitimate site and to detect phishing. This method requires more run-time overhead to extract contents, the use ofsearch engines to locate the domains and, finally, a system through which judgements are made on whether the website is legitimate [4]. Fourthly, the URLs based approach that executes by embedding sensitive words or characters in a link that mimic legitimate URLs (spelling mistakes, reliable keywords, redirecting to other websites or shortened URLs). This method relies on the lexical and host-based features of the URLs to detect a website phishing attack. The lexical features includes properties that malicious URLs use to look like legitimate URLs, whereas the host-based features are properties that belong to website hosts. Fifthly, the Visual similarity-based approach, which detects phishing by comparing the visual characteristics of suspicious websites and legitimate websites to identify phishing and non-phishing websites [2],[3].

The phishing detection approach that is based on visual similarity can be categorised into six main types. The first type depends on the document object model (DOM) tree which defines the structure of webpages. This technique judges the phishing attack on the basis of a comparison of the DOM tree of unreliable websites and of reliable websites; if there is a match, the phishing attack is reported. The second type is the visual feature-based technique that focuses on comparing text features such as the font size, background colours and image features such as height, width and position of images on the website, in order to detect a phished webpage. The third type is relays on Cascading Style Sheets (CSS) which is used to create a uniform appearance (text styles, font and colours) across several webpages. In this mechanism, if the result of the matching of the CSS layout of a suspicious website and a legitimate website are identical, then the suspicious page is reported as a phishing page. The fourth type is based on images of websites and works by comparing the similarity of unreliable website images with the images of a legitimate website. If the outcome of comparison is low, then both sites are reliable; otherwise, the unreliable website is considered to be a phishing attack.The fifth type is the visual perception method which relies on the theory that defines how people perceive visual components. This method views websites in a holistic manner rather than as a collection of distinct features, as in the other methods. This method determines the extent of similarity between websites from a reader's point of view. The sixth type is the hybrid method that combines two or more kinds of phishing detection mechanisms with the goal of increasing the accuracy rate [5].

Recently, several machine learning approaches have been used by researchers as an assistant tool to detect web phishing attacks. Naive Bayes, support vector machine, random forest and convolution neural network are commonly used algorithms [3].

In this research, we used the convolution neural network (CNN) to detect web phishing attacks based on the URLs and the screenshot of websites. The CNN is one of the most popular types of deep learning mechanisms in particular for high dimensional data, such as videos and images. It is suitable to use to extract the local features of images and it uses these features to distinguish between different images. It has also been tremendously successful in extracting textual features and recognizing sentences. A general CNN architecture typically consists of alternating convolution and pooling layers, followed by one or more fully connected layers. These layers automatically represent high-level features of inputs and enable the CNN to carry out theclassification task. The basic three layers of CNN can be described as follows: First there is the Convolution layer which is responsible for extracting features of input and consists of a series of convolution kernels that divide the input into a small block, referred to as receptive fields. Convolving the input with a kernel creates a feature map which highlights the existence of a specific feature in the input. Second there is the Pooling layer that usually helps to reduce the features map's dimensionality. Several types of pooling functions can be performed, such as max pooling, average pooling and sum pooling. Third is a fully connected layer (FCL), which is a typical neural network layer used mostly for classification. It receives input from the previous extraction features step and analyses the output of all the preceding layers in order to create a non-linear combination of selected features. Consequently, it can calculate the class scores and produce an 1-D size array match number of classes [6].

3. LITERATURE REVIEW

The need to secure web applications has increased significantly given the fact that web applications are commonly used as a key interface between users and platforms. However, users have an issue in remembering all reliable URL pages on the web which puts them at risk for a wide variety of attacks. The phishing attack is one of the most common threats that users experience while browsing online websites.The attacker deceives users by using a phished website to acquire sensitive user credentials, for example, usernames and passwords. A variety of anti-phishing strategies have been introduced in an effort to address this problem based on the URL ,the Blacklist, the DNS, the Whitelist or the Visual appearance. The visual similarity anti-phishing technique relieson the fact that the attacker always uses web pages that look like genuine websites to trick internet users into inputting their sensitive information [7].

A number of researchers have investigated the use of visual similarity as an approach to detect phishing web pages. The authors in [8] analysed images in webpages by measuring similarity scores using an image processing mechanism. The snapshot of legitimate and suspected pages was divided into blocks and matched using Earth Mover's Distance algorithm. Their results show a high detection rate (99.6%) and they concluded that the detection of a phishing attack is more robust when the image-based approach is used than when the HTML-based technique is used. The authors in [9] used a compression algorithm called Normalised Compression Distance (NCD) to compute similarities based on distance between the image of a requested website and the image of a cached benign website. The results were submitted to a classifier (C4.5, Ripper, Logistic and SVM) which set off an alarm if the webpage was marked as a phish. The researchers' results showed a high true positive rate of 99.99%; however, their method of calculating visual similarity is complex and unpractical and cannot be implemented with real-time browsing.

Recently, computer vision based mechanisms have been used to analyse and evaluate website similarities and detect phishing attacks. The authors in [10] used a Histogram of Oriented Gradients (HOG) descriptor to extract website features and to compute the similarity metric between legitimate and suspicious webpages. The results highlighted the effectiveness of using the HOG descriptor for webpage phishing detection, especially for zero-hour attacks. An extension to the research study in [10] was presented by the authors of [11].They conducted a comparative study on the performance of five compact visual descriptors (SCD, FCTH, JCD, CEDD and CLD) with two machine learning techniques (random forest and SVM) to detect web phishing attacks from screenshots of legitimate and phished websites. According to their results, the SCD with RF delivered the highest F1 score at 0.895 for the analysis of the whole website image.

Instead of investigating the similarity of the whole webpage, the authors in [12] proposed focusing on the logo of a website to detect phishing. Their study encompassed two processes: logo-extraction and web-site identity verification. They used a machine learning algorithm (SVM) to detect and extract the correct logo image from all downloaded web site images. Furthermore, to obtain the corresponding domains of the right logo image, they used the Google image search engine to find the domains of reliable websites and compare with the domains of the suspected webpages. The results revealed the effectiveness of using a website logo to detect phishing websites. Moreover, the authors in [13]used a similar method to analyse a favicon image instead of the whole webpage image to detect phishing websites. The authors extract favicon photos from the webpage and used the Google engine to acquire features that can be used to distinguish between suspect and legal websites. However, since both research [12] and [13] rely on a search engine, the threat of DNS spoofing is of concern.

The authors in [14] combined the global-image (snapshot image) feature as in [8],[9] and local-image (logo) feature as in [12], but relied purely on the image level to detect phishing webpages. The authors used a novel technique to detect a logo and a modified EMD algorithm to calculate the similarity score for a snapshot of websites. If suspected webpages have a logo as a legitimate page and the global similarity score exceeds the defined threshold, then that page is marked as a phishing webpage. The results proved the effectiveness of the proposed method with up to 90% true positive rate and 97% true negative rate.

Other research focused on page layout similarity to detect phishing pages. A phishing webpage often has a similar appearance to the legitimate webpage, and hence, the page layouts of the two webpages are expected to be identical. In [15], the authors conducted research based on the Document Object Model (DOM) tree in order to detect malicious pages. In their proposed technique, when a user reuses the same information (user credentials) on a website that has a different domain from that of previously visited pages, the DOM tree of the first webpage where the user credentials were initially entered is compared with that of the recent webpage to detect a phishing attack. Their results show a high positive rate, but their approach is ineffective if the attacker changes the DOM tree for phishing pages.

Similar to [15] in terms of using page layout similarity to recognise a phishing website, but focusing on the similarity of Cascading Style Sheets, the authors in [16] presented an algorithm to compare the similarity of Cascading Style Sheets of malicious and genuine pages. In their proposed approach, if the CSS layout of both a suspicious page and an original page exceeds a defined threshold and the two pages have a different URL, the suspicious page is reported as phishing. The results revealed a high detection rate. However, depending solely on the CSS to detect phishing attacks is insufficient since not all websites have a CSS layout.

The authors in [17] combined page layout similarity and a snapshot of websites to detect phishing attacks. They conducted research to propose a phishing detection approach based on image and CSS layout. They used a database to store the CSS layout and screenshots of legitimate web pages. Their approach helps detect suspicious websites that match the visual appearance or CSS layout of original websites. The results revealed that 22% of phishing webpages had been classified incorrectly because many phishing websites do not have a CSS layout.

In addition, several researchers have used a hybrid approach with a visual similarity mechanism in order to increase the effectiveness of detecting a webpage phishing attack. The authors in [18]proposed an approach that depends on the visual and textual features of pages. Their approach included words that appear in a webpage and a set of visual features such as images, page layout and logos. The research was conducted using a naïve Bayes classifier to extract textual items from webpages and Earth Mover's Distance classifier to deal with the image of the webpage. Moreover, they used a Bayesian method as a fusion algorithm to aggregate the outcomes of the two classifiers and distinguish between phishing and legitimate pages. Their results illustrated the capability of the proposed mechanism to increase the accuracy of detection of a phishing attack. Furthermore, in [19], the authors computed a single similarity score called the signature, which consists of website text pieces, images and visual appearance, to detect a phishing attack. The authors matched between the signature of a suspect website and the saved signatures of legal webpages. If the two signatures had a high rate of similarity a phishing attack will be identified. The findings showed the efficacy of the suggested strategy, with a 0% false positive rate and a 0%false negative rate. Table 1 presents a summary of web phishing detection approaches based on visual similarity.

Table 1. Summary of web phishing detection approaches based on visual similarity.

Paper	Features	Search engine dependence	Advantages	Drawbacks
[8]	Image (EMD)	No	High detection rate (99.6%)	Image processing complex. Can’t detect new phishing.
[9]	Image (NCD)	No	High true positive rate of 99.99%	Complex method and can't implement with real time browsing.
[10]	Image (HOG)	No	Detect zero-hour phishing attack	Define the similarity threshold value for phishing alarm
[11]	Image (compact visual descriptors)	No	Detect zero-hour phishing attack	It is time-consuming.
[12]	Logo	Yes	Can detect new phishing webpage	High false negative rate of 13%
[13]	Favicon	Yes	Detect new phishing webpage	Not all websites have Favicon
[14]	Image and logo	Yes	90% true positive rate, 97% true negative rate.	FN increase if phisher page does not contain an official logo

[15]	DOM layout	No	High true positive rate (100%)	Fail if the attacker changed DOM. Fail if the phisher page uses images only.
[16]	CSS layout	Yes	High true positive rate 99%	Can't detect new phishing webpage
[17]	image and CSS layout	No	Using image and CSS layout	22% of phishing webpages classified incorrectly
[18]	Hybrid (Textual and image)	No	Used the Bayesian theory to define threshold	Can't detect new phishing webpage time consuming
[19]	Hybrid (text, image & visual appearance)	No	Detect embedded objects	Consume a long time to measure signature of the websites

Approaches to detect web phishing attacks based on URLs mainly differ in three things: type of URL features, method deployed to extract features and the classification algorithms used. Several researchers focus on URL string analysis and external information to define the feature vectors of the URLs. The authors of [20]employed a method using a stacked restricted Boltzmann machine to select a feature from the vector format of URL lexical and host-based features and then fed these features into four classification algorithms: the artificial neural network (ANN), the deep belief network (DBN), the Naïve Bayes (NB) and the support vector machine (SVM) in order to detect malicious URLs. Their results demonstrate that the performance of the deep DBN exceeds the NB, SVM and ANN in terms of accuracy, the true positive rate and the false positive rate. The merit of their model is the ability to detect each class of malicious URLs: spam, phishing and malware attacks. Furthermore, the authors of [21] analysed suspicious websites using a method referred to as SHLR, which consists of three parts: first, recognising a benign website solely by using the title tag content of the website as the main entry to a search engine; second, if the website could not be identified as un-phishing, extracting the URLs features (word lists) and using seven heuristic rules to detect phishing webpages; and, finally, using three classifiers to recognise the residual websites, namely: a logistic regression, Naive Bayes and SVM, and extracting URL features from Who is, HTML, DNS, lexical features and similarity with phishing vocabulary. Their results show a high accuracy rate of 98.8%, but their approach depends on a search engine and a third party. Using web host characteristics and the features of the URL string, the authors of [22] proposed a method that uses a bidirectional LSTM algorithm based on a recurrent neural network and the convolution neural network that encompasses three features: the URL static vocabulary feature, the texture fingerprint feature, and the URL word vector feature. These three features are combined and applied to the model in order to detect a phished website. Their results show a high accuracy rate of 99.45%.

Unlike the studies outlined above, which detected malicious URLs based on groups of the URL lexical and host-based features, some researchers use a maximum of one or two URL external features with lexical features. The authors of [23] proposed an approach that uses a support vector machine algorithm with five lexical URL features and a similarity index feature to detect a web phishing attack. Their results show that using the similarity index feature increased the system detection rate by 21.8% to reach the highest recognition rate of 95.80%. Moreover, the researchers in [24] proposed a method to detect phishing attacks based on a random forest algorithm and by using eight features from the URL string and the webpage (page rank and Google index). They evaluated the classifier in terms of various metrics using the random forest algorithm with an accuracy rate of 95%.However,their technique relies on a third party service.

Most recent studies focused solely on the lexical features extracted from the URL string since this can be extracted quickly, does not require any execution for the URL and has a good performance [25]. The authors of [26] proposed a method that uses the URL string as a raw short character which inputs directly to a convolution neural network that extracts and aggregates locally detected features and then classifies the malicious URLs. They compared their results, based on a method that extracted features automatically, with two manual feature extraction methods and concluded the effectiveness of their approach based on the convolution neural networks. The authors in [27] proposed a method that extracted a byte values vector from URL characters and fed them into a neural network algorithm to classify the input URL as a benign or phishing URL. They investigated their model with three optimizers: SGD, Adam and AdaDelta, and declared that the best optimizer for the deployed neural network model was Adam, with an accuracy rate of 94.18%.

In [28] the authors extended what was done in [27] by using the character-level text features of the URL string and investigating the performance of five deep learning algorithms: long short-term memory (LSTM), convolutional neural network-long short-term memory (CNN-LSTM), convolution neural network (CNN), identity-recurrent neural network (I-RNN) and recurrent neural network (RNN) to differentiate between phishing and un-phishing webpages. Their experimental results proved that extracting URL features by using any deep learning algorithm outperformed the manual feature extraction methods in particular with the LSTM and the CNN-LSTM model, which obtained the best accuracy rates at 99.96% and 99.95% respectively.

In contrast to the research by [27] and [28], the authors in [29] focused on NLP and word vector features extracted from the URL string. They investigated the performance of seven machine-learning algorithms (Adaboost, Naive Bayes, SMO, Random Forest, K-star, KNN and Decision Tree) with three different types of URL string feature sets: Word Vectors, NLP and Hybrid (NLP and word vectors) to detect phishing attacks. It is obvious from their results that using NLP or hybrid features delivers higher performance compared to the use of Word Vectors features and the best accuracy rate (97.98%) acquired by combing the RF classifier with NLP features. Their method is independent of third-party services and can be executed as a real time anti-phishing detection model.

A mix of the lexical URL features (characters and words features) has been used by the authors in [25] to propose an URLNet technique that uses convolution neural networks as a way to automatically learn features and do the classification task in order to distinguish malicious URLs from benign URLs. Their results show that the performance of the URLNet model with characters and words features significantly outperforms the URLNet model based on character features only and the URLNet model based on word features only. Furthermore, the author of [30] relies on structure and semantic features of the URL string to propose a method using a bidirectional LSTM network based on the recurrent neural network (RNN) to extract the global features of the URL string and, after that, using the convulsion neural network to extract the local features of the URL string, then merging the extracted characteristics into a fixed length vector which is used to classify the URLs into legitimate or phishing webpages. Their results show that their approach achieved a high accuracy rate of 97%. Table 2 provides a summary of web phishing detection approaches based on URLs.

Table 2. A summary of web phishing detection approaches based on URLs.

Paper	Features	Search engine or third-party dependence	Advantages	Drawbacks
[20]	lexical and host-based	Yes	Ability to detect each class of malicious URLs	Complex method
[21]	lexical and host-based	Yes	High accuracy rate 98.8%,	Consumes a time to extract features
[22]	lexical and host-based	Yes	High accuracy rate of 99.45%.	Consumes a time to extract features
[23]	lexical and one external information	Yes	Recognition rate 95.80%.	Small dataset size
[24]	lexical and two external information	Yes	high accuracy rate 95%	Low performance
[26]	lexical features (characters)	No	Automatic way to extract features	Ignores extracting features from words in the URLs.
[27]	lexical features (characters)	No	High accuracy 95.17%	When dataset changed, the results changed
[28]	lexical features (characters)	No	High accuracy 99.96%	Complex architecture
[29]	lexical features (NLP and words)	No	A real time anti-phishing detection model. High accuracy rate 97.98	NLP can't directly handle symbols in the URLs.
[25]	lexical features (characters and words)	No	Automatic way to extract features	Requires large data sets to works in an end-to-end manner
[30]	lexical features (structure and semantic features)	No	Fast High accuracy rate of 97%.	If the URL lacks the relevant semantics, it may cause the wrong classification.

4. Proposed Method

Since the goal of our proposed method is to detect phished websites by using the URL and the screenshots of suspicious webpages, we formulated this issue as a binary classification problem with the URL and images of websites as input leading to their classification as either legitimate websites or phished websites. The classification task is done by training a network to produces 0 if the input URL or website image is classified as a legitimate webpage or 1 if the URL or the screenshot of the website is classified as a phishing webpage.

Our proposed method can be divided into three parts, as shown in Figure 1. In the first part, the pre-processing task is done with the URL string representing each character individually as a vector. The entire URL (a series of characters) is transformed into a matrix representation, whereas for the website image, the pre-processing task is completed by creating a matrix of pixel values. In the second part, CNN1 receives the website image as input, then extracts its features. The output is the classification of the webpage as either legitimate or malicious. Simultaneously, the CNN2 receives the URL of the website as an entry and then extracts its features. The outcome is identification of the webpage as phishing or legitimate. Finally, the two results are combined to decide whether there has been a phishing attack. This is the case if the classification outcomes of the first or the second CNN is that it is a phishing webpage; otherwise, the website is considered to be legitimate.

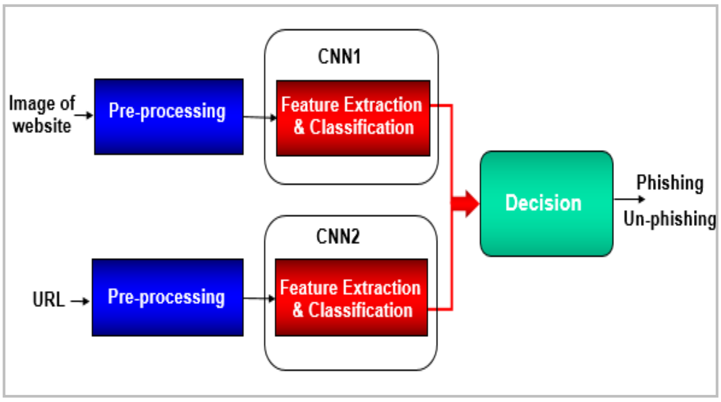

Figure 1. Structure of our model

For building blocks of CCN1 and CCN2, we created a convolution neural network (CNN) containing two convolutional layers and two max pooling layers to extract the most important website image features and URL features. These features are fed into a fully connected layer to classify websites as phishing websites or legitimate webpages.

The design of the convolution neural network that we deployed in each block was identical, but CNN1 deals with the website image, whereas the CNN2 works with the URLs. The operations that are executed in each block can be split into three phases: In the initial phase, website's images and URLs will pass over a set of convolution layers with kernels (filters) one by one in order to extract the local features for each image or URL. Using a collection of weights, the filter convolves with the input images or URLs in order to compute a new feature map. The outcomes of the convolution are then passed on via a nonlinear activation function that is used to accelerate the learning process. More formally, the convolution procedure to compute a feature map can be defined as follows:

$$Y_i = f\,(W_i * m) \qquad (1)$$

Where *m* refers to input, *Wi* refers to the convolution filter associated with the *i*th feature map; the multiplication sign refers to the operation of computing a new feature map by the *i*th filter; *f* refers to the activation function and Y_i refers to the *i*th output feature map.

In this research, we used two convolutional layers with filter size 5x5 and rectified linear units as an activation function. The second phase was done by applying a pooling layer to minimise the feature dimension and extract the most significant features. We used two pooling layers with a max pooling. The reciprocal between convolution layers and pooling layers makes it possible to combine the outputs of the two convolution filters to generate the final features vector which is a one-dimensional vector representing the outputs of the two coevolution layers and pooling layers [6],[31].

In the third phase, since the feature vector was extracted in the previous phase, we used a fully connected layer with a sigmoid function in order to create a non-linear combination of extracted features and classify the input as legitimate or phishing webpages. In order to evaluate the performance of our model, we used accuracy as the main metric. This is the most commonly used experimental criteria and is given by the following equation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

In addition, we evaluated the model using the three famous metrics used in website phishing detection research: precision, recall and F1 score, the descriptions of which are given by the following equations [32]:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1 = \frac{2 * Recall * Precision}{Recall + Precision} \tag{5}$$

5. Experiment Results and Comparison

To implement our approach we used Python Programming Language and a dataset containing2000 screenshots and URLs of legitimate and phishing websites. We measured the performance of the proposed method in terms of accuracy, F1 score, precision and recall. The experiment result is shown in Table 3. The results reflect the effectiveness of using CNNs as an automatic way to extract features and carry out a classification task to distinguish between reliable and phishing webpages.

Table 3. The results of our model

Metric	Value
Accuracy	99.67 %
Precision	99.43 %
F1 score	99.28 %
Recall	99.47 %

Moreover, since several variables have a significant role in the performance of the convolution neural network structure, the effects of changing the batch size on the proposed model have been investigated. As shown in Figure 2, the accuracy of our model decreased as the batch size increased and our proposed model obtained the highest accuracy when the batch size was16.

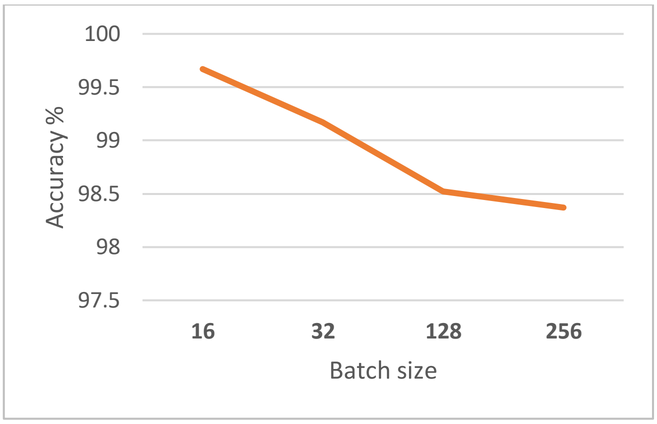

Figure 2. Batch sizes impact on the model

A great deal of research has been conducted using the convolution neural networks (CNNs) on its own or combined with other algorithms to detect a web phishing attack based on URLs or content of websites. The novelty of our proposed method is its use of an URL and a screenshot of a website to detect a phishing attack using only CNNs. We compared our results with [33], [34], [30] and [28] where only CNNs were used as shown in Figures 3,4, 5 and 6.

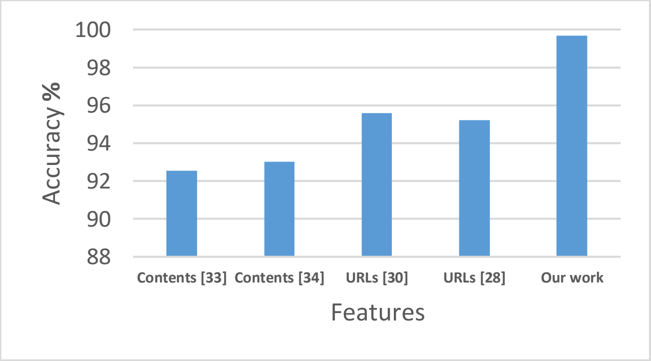

Figure 3. The accuracy of our work with other methods

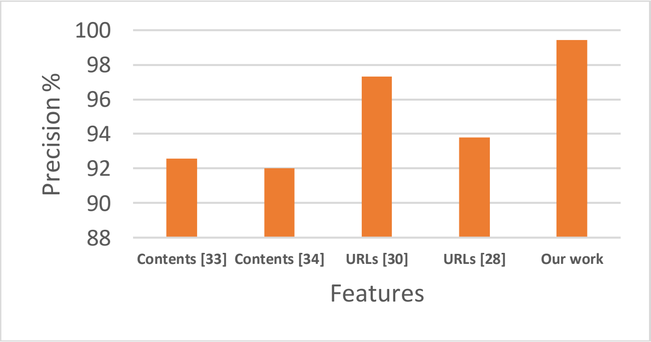

Figure 4. The precision of our work with other methods

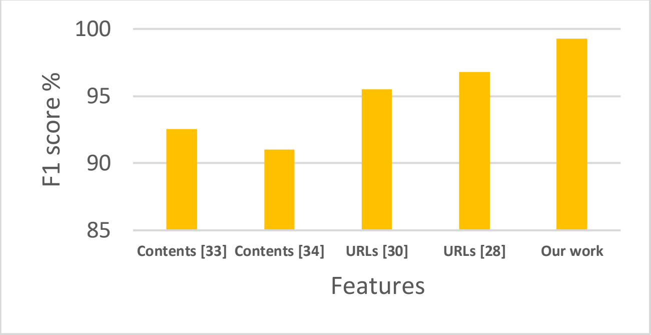

Figure 5. The F1 score of our work with other methods

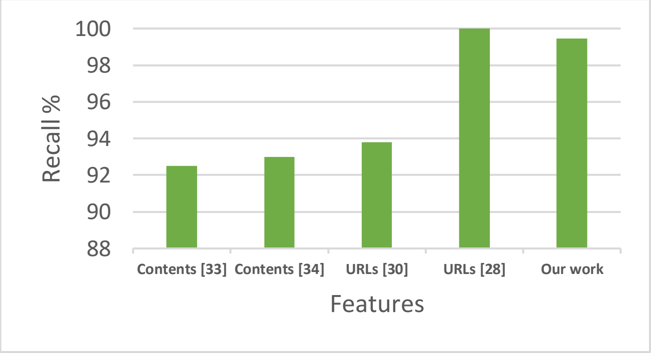

Figure 6. The Recall of our work with other methods

It is clear from Figures 3, 4, 5 and6 that our proposed model is the optimal one. It performed better than the models used in [33], [34], [30] and [28] in almost all aspects. Furthermore, our model relies only on an URL and a screenshot of a website, which makes it independent of the website language and contents. It can detect phishing even if the URLs are shortened or hidden.

6. Conclusions

This study explored the possibility of detecting a phishing attack by distinguishing between the URLs and images of legitimate websites and the URL and images of phishing websites using CNNs. We proposed a way that can be used to detect newly created phishing webpages based only on the URL and the screenshot of suspicious websites. The proposed model shows a classification accuracy of 99.67%.

Based on the obtained results, it can be concluded that combining URLs features and visual similarity using convolution neural networks is a highly effective approach which is superior to other approaches when it comes to performing automatic features extraction and the classification of websites into phishing or legitimate websites. It is clear, furthermore, that increasing batch sizes leads to the lowering of the accuracy of the model.

For future work, we suggest improving the model by finding a way to automatically identify the lowest URL length and the smallest screenshot size of webpages thus contributing to the optimum performance of the proposed method.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] H. Thakur, "Available Online at www.ijarcs.info A Survey Paper On Phishing Detection," vol. 7, no. 4, pp. 64–68, 2016.

[2] G. Varshney, M. Misra, and P. K. Atrey, "A survey and classification of web phishing detection schemes," Security and Communication Networks. 2016, doi: 10.1002/sec.1674.

[3] E. S. Aung, T. Zan, and H. Yamana, "A Survey of URL-based Phishing Detection," pp. 1–8, 2019, [Online]. Available: https://db-event.jpn.org/deim2019/post/papers/201.pdf.

[4] S. Nakayama, H. Yoshiura, and I. Echizen, "Preventing false positives in content-based phishing detection," in IIH-MSP 2009 - 2009 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009, doi: 10.1109/IIH-MSP.2009.147.

[5] A. K. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity based approaches," Security and Communication Networks. 2017, doi: 10.1155/2017/5421046.

[6] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks," pp. 1–68, 2019, doi: 10.1007/s10462-020-09825-6.

[7] J. Mao et al., "Phishing page detection via learning classifiers from page layout feature," Eurasip J. Wirel. Commun. Netw., 2019, doi: 10.1186/s13638-019-1361-0.

[8] I. F. Lam, W. C. Xiao, S. C. Wang, and K. T. Chen, "Counteracting phishing page polymorphism: An image layout analysis approach," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2009, doi: 10.1007/978-3-642-02617-1_28.

[9] T. C. Chen, T. Stepan, S. Dick, and J. Miller, "An anti-phishing system employing diffused information," ACM Trans. Inf. Syst. Secur., vol. 16, no. 4, 2014, doi: 10.1145/2584680.

[10] A. S. Bozkir and E. A. Sezer, "Use of HOG descriptors in phishing detection," in 4th International Symposium on Digital Forensics and Security, ISDFS 2016 - Proceeding, 2016, doi: 10.1109/ISDFS.2016.7473534.

[11] F. C. Dalgic, A. S. Bozkir, and M. Aydos, "Phish-IRIS: A New Approach for Vision Based Brand Prediction of Phishing Web Pages via Compact Visual Descriptors," ISMSIT 2018 - 2nd Int. Symp. Multidiscip. Stud. Innov. Technol. Proc., 2018, doi: 10.1109/ISMSIT.2018.8567299.

[12] K. L. Chiew, E. H. Chang, S. N. Sze, and W. K. Tiong, "Utilisation of website logo for phishing detection," Comput. Secur., 2015, doi: 10.1016/j.cose.2015.07.006.

[13] K. L. Chiew, J. S. F. Choo, S. N. Sze, and K. S. C. Yong, "Leverage Website Favicon to Detect Phishing Websites," Secur. Commun. Networks, 2018, doi: 10.1155/2018/7251750.

[14] Y. Zhou, Y. Zhang, J. Xiao, Y. Wang, and W. Lin, "Visual similarity based anti-phishing with the combination of local and global features," in Proceedings - 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2014, 2015, doi: 10.1109/TrustCom.2014.28.

[15] A. P. E. Rosiello, E. Kirda, C. Kruegel, and F. Ferrandi, "A layout-similarity-based approach for detecting phishing pages," in Proceedings of the 3rd International Conference on Security and Privacy in Communication Networks, SecureComm, 2007, doi: 10.1109/SECCOM.2007.4550367.

[16] J. Mao, P. Li, K. Li, T. Wei, and Z. Liang, "BaitAlarm: Detecting phishing sites using similarity in fundamental visual features," in Proceedings - 5th International Conference on Intelligent Networking and Collaborative Systems, INCoS 2013, 2013, doi: 10.1109/INCoS.2013.151.

[17] S. Haruta, H. Asahina, and I. Sasase, "Visual Similarity-Based Phishing Detection Scheme Using Image and CSS with Target Website Finder," in 2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings, 2017, doi: 10.1109/GLOCOM.2017.8254506.

[18] H. Zhang, G. Liu, T. W. S. Chow, and W. Liu, "Textual and visual content-based anti-phishing: A Bayesian approach," IEEE Trans. Neural Networks, 2011, doi: 10.1109/TNN.2011.2161999.

[19] E. Medvet, E. Kirda, and C. Kruegel, "Visual-similarity-based phishing detection," Proc. 4th Int. Conf. Secur. Priv. Commun. Networks, Secur., no. September 2008, 2008, doi: 10.1145/1460877.1460905.

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

[21] Y. Ding, N. Luktarhan, K. Li, and W. Slamu, “A keyword-based combination approach for detecting phishing webpages,” Comput. Secur., vol. 84, pp. 256–275, 2019, doi: 10.1016/j.cose.2019.03.018.

[22] H. huan Wang, L. Yu, S. wei Tian, Y. fang Peng, and X. jun Pei, “Bidirectional LSTM Malicious webpages detection algorithm based on convolutional neural network and independent recurrent neural network,” Appl. Intell., vol. 49, no. 8, pp. 3016–3026, 2019, doi: 10.1007/s10489-019-01433-4.

[23] M. Zouina and B. Outtaj, “A novel lightweight URL phishing detection system using SVM and similarity index,” Human-centric Comput. Inf. Sci., vol. 7, no. 1, pp. 1–13, 2017, doi: 10.1186/s13673-017-0098-1.

[24] S. Parekh, D. Parikh, S. Kotak, and S. Sankhe, “A New Method for Detection of Phishing Websites: URL Detection,” Proc. Int. Conf. Inven. Commun. Comput. Technol. ICICCT 2018, no. Icicct, pp. 949–952, 2018, doi: 10.1109/ICICCT.2018.8473085.

[25] H. Le, Q. Pham, D. Sahoo, and S. C. H. Hoi, “URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection,” no. i, 2018, [Online]. Available: http://arxiv.org/abs/1802.03162.

[26] J. Saxe and K. Berlin, “eXpose: A Character-Level Convolutional Neural Network with Embeddings For Detecting Malicious URLs, File Paths and Registry Keys,” 2017, [Online]. Available: http://arxiv.org/abs/1702.08568.

[27] K. Shima et al., “Classification of URL bitstreams using bag of bytes,” 21st Conf. Innov. Clouds, Internet Networks, ICIN 2018, pp. 1–5, 2018, doi: 10.1109/ICIN.2018.8401597.

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

[29] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, “Machine learning based phishing detection from URLs,” Expert Syst. Appl., vol. 117, pp. 345–357, 2019, doi: 10.1016/j.eswa.2018.09.029.

[30] W. Wang, F. Zhang, X. Luo, and S. Zhang, “PDRCNN: Precise Phishing Detection with Recurrent Convolutional Neural Networks,” Secur. Commun. Networks, 2019, doi: 10.1155/2019/2595794.

[31] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, “A Guide to Convolutional Neural Networks for Computer Vision,” Synth. Lect. Comput. Vis., 2018, doi: 10.2200/s00822ed1v01y201712cov015.

[32] V. Karthikeyani and S. Nagarajan, “Machine Learning Classification Algorithms to Recognize Chart Types in Portable Document Format (PDF) Files,” Int. J. Comput. Appl., 2012, doi: 10.5120/4789-6997.

[33] M. A. Adebowale, K. T. Lwin, and M. A. Hossain, “Deep learning with convolutional neural network and long short-term memory for phishing detection,” 2019 13th Int. Conf. Software, Knowledge, Inf. Manag. Appl. Ski. 2019, no. March 2019, doi: 10.1109/SKIMA47702.2019.8982427.

[34] C. Opara, B. Wei, and Y. Chen, “HTMLPhish: Enabling Phishing Web Page Detection by Applying Deep Learning Techniques on HTML Analysis,” no. October 2018, 2019, [Online]. Available: http://arxiv.org/abs/1909.01135.