IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 29, 2020, accepted November 23, 2020, date of publication December 9, 2020,
date of current version December 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3043188

Web2Vec: Phishing Webpage Detection Method Based on Multidimensional Features Driven by Deep Learning

JIAN FENG[1], LIANYANG ZOU[2], OU YE[1], AND JINGZHOU HAN[1]
[1]College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China
[2]Information Technology Department for Head Office of SPD Bank, Application Development Services Sub-center (Xi'an), National Institute of Standards and Technology, Xi'an 710077, China

Corresponding author: Jian Feng (fengjian@xust.edu.cn)

This work was supported in part by the Shaanxi Provincial Natural Science Foundation under Project 2020JM-533 and Project 2018JQ5095, and in part by the Chinese Postdoctoral Science Foundation under Grant 2020M673446.

ABSTRACT Phishing is a kind of online attack that attempts to defraud sensitive information of network users. Current phishing webpage detection methods mainly use manual feature collection, and there are problems that feature extraction is complicated and the possible correlation between features cannot be avoided. To solve the problems, a new phishing webpage detection model is proposed, among which the main components are automatic learning representations from multi-aspects features through representation learning and extracting features by hybrid deep learning network. Firstly, the model treats URL, HTML page content, and DOM (Document Object Model) structure of webpages as character sequences respectively, and uses representation learning technology to automatically learn the representation of the webpages; then, sends multiple representations to a hybrid deep learning network composed of a convolutional neural network and a bidirectional long and short-term memory network through different channels to extract local and global features, and use the attention mechanism to strengthen the influence of important features; finally, the output of multiple channels is fused to realize classification prediction. Through four sets of experiments to verify the detection effect of the model, the results show that the overall classification effect of the model is better than the existing classic phishing webpage detection methods, the accuracy reaches 99.05%, and the false positive rate is only 0.25%. It is proved that the strategies of extracting webpage features from all aspects through representation learning and hybrid deep learning network can effectively improve the detection effect of phishing webpages.

INDEX TERMS Attention mechanism, deep learning, phishing, representation learning.

I. INTRODUCTION

Phishing is a kind of attack that attackers use social engineering and technical disguise and other attack methods to cheat users to visit fake webpages by sending deceptive spam, real-time communication messages, etc., in order to induce users to disclose their personal identity, financial account, and other sensitive information. According to the latest report of the APWG (Anti-Phishing Working Group), the total number of phishing webpages in the second quarter of 2020 increased by 13.9% over the same period in 2019 [1]. The continued growth of phishing attacks has had a huge negative impact on the healthy development of the Internet.

In the offensive and defensive game with phishing, phishing webpage analysis and detection technology have been continuously developed, and the traditional phishing webpage detection methods such as blacklist-based [2], heuristic-based [3], [4], visual similarity-based [5], [6], and machine learning-based [7]–[11] methods are proposed, and emerging detection methods based on deep learning [13]–[21] are also proposed in recent years. Traditional phishing webpage detection methods are mainly based on the analysis and modeling of manually extracted multi-source features such as URL features, page content features, and webpage structural features, which once showed strong resistance to phishing attacks. However, as the iterative update speed of phishing webpages is accelerated and the attacks are more evasive, the traditional methods can only continue to give

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

more detailed analysis and extract more features, resulting in a disaster of feature dimensions, and at the same time cannot avoid the possible correlation between features [11]. In recent years, deep learning has been researched and applied in the detection of phishing webpages with its powerful automatic feature extraction capabilities. However, due to the semi-structured nature of webpages, it is complicated to automatically extract features from URL, page content, webpage structure, and other aspects. Therefore, existing studies usually extract features only from a single aspect such as URL and lack of comprehensive learning of webpage features, so the detection effect still needs to be improved.

Aiming at solving the above problems, a phishing webpage detection method Web2Vec is proposed to automatically extract webpage features in multi-aspects. Firstly, the model takes the URL, page content, and DOM (Document Object Model) structure of the webpages as text, and constructs corpora based on characters, words, and sentences from these texts respectively, and use the representation learning technology to automatically learn the multidimensional representations of webpages; then, these representation vectors are input into different channels for feature extraction. CNN (Convolutional Neural Network) is used to extract local features, following by BiLSTM (Bi-directional Long-short Term Memory) to obtain context semantics and dependency features, and then the attention mechanism is used to strengthen the influence of the important features. Finally, a classifier is used for category prediction. This model automatically learns the characteristics of phishing webpages without prior knowledge, avoids the subjectivity of manually selecting features, and can take advantage of a hybrid deep learning network to extract features, so it achieves ideal detection results. To the best of our knowledge, this is the first research to use representation learning technology in Natural Language Processing (NLP) to automatically extract page content and DOM structural features, and to achieve multidimensional feature extraction together with automatically extracted URL features. Notice that in the paper, the difference between phishing webpages and benign webpages is their fraudulent intention instead of their appearances. Phishing webpages are designed to cheat users of key private information, while benign webpages are dedicated to attracting people to browse repetitively. So what the model wants to find is the latent difference between phishing and benign webpages.

In particular, the key contributions in this work are listed as follows:

- The paper takes the URL, page content, and DOM structure of the webpages as text, and uses representation learning technology to automatically learn the representation of the webpages in all dimensions.
- A hybrid deep learning model that fuses CNN, BiLSTM, and attention is represented.
- Further, four experiments on the Web2Vec are conducted from different aspects. The results show that the classification performance is good.

The paper is organized as follows. In Section II, we present related works on phishing webpage detection. Then, the framework and the detailed process of Web2Vec is described in Section III. In Section IV, the performance of the Web2Vec is evaluated. Finally, we conclude the paper and discuss future works.

II. RELATED WORKS

Researchers have proposed a series of phishing webpage detection methods, including the widely used traditional phishing webpage detection methods and the emerging deep learning-based methods.

A. TRADITIONAL PHISHING WEBPAGE DETECTION METHODS

Traditional phishing webpage detection methods mainly include four categories: ① Methods based on a blacklist. These methods detect phishing webpages simply based on the blacklist by matching URL and other information without false positives. But they cannot correctly identify phishing webpages not listed on the blacklist. Representative applications are Google Chrome and other projects [2]. ② Methods based on heuristic rules. These methods design and implement heuristic rules based on the similarities existing between phishing webpages. Typical researches include CANTINA+ [3], PhishDetector [4], etc. Heuristic rules can detect most unreported phishing webpages in real-time, but the premise is that the statistical characteristics of phishing webpages are unique and fuzzy matching technologies are used, so the False Positive Rate (FPR) is high. ③ Methods based on visual similarity. These methods convert the webpages to be detected into images, and then compare the feature vectors of the webpages to be tested and the target webpages through image processing technologies [5]. A typical method is proposed in [6]. Although there are some new researches proposed in recent years [7], such methods are still powerless to phishing webpages which are not visually similar to the target webpages. ④ Methods based on machine learning. They treat phishing webpage detection as a classification or clustering problem, and use the corresponding machine learning algorithms to build detection models [8]. Among them, the clustering methods first divide the webpages into several clusters, and then distinguish the phishing webpages from the benign webpages by marking the clusters [9]. On the other hand, the classification methods construct classifiers according to the characteristics of the labeled samples, and then map the unlabeled samples to phishing or benign [10], [11]. Among existed researches, PCA-RF (Principal Component Analysis Random Forest) achieved state-of-the-art performance with an accuracy of 99.55% [12]. Due to the superior adaptability, scalability and accuracy, the machine learning-based methods became mainstream among the above four types of methods.

The effectiveness of machine learning-based methods usually depends on the quality of the extracted features, so the methods focus on how to extract and select more effective

features. Common features extracted from phishing webpages usually include URL statistical characteristics (length of URL, number of special characters, etc.), identity characteristics of webpages (Whois, DNS information, etc.), page content characteristics (page layout, theme, etc.) [8], etc. In order to resist evasion attacks from attackers, the number of extracted features is increasing. For example, Google Chrome has extracted 2130-dimensional features for phishing detection [9], which greatly increases the complexity of modeling, but leaves the detection efficiency to be improved. At the same time, these techniques are bypassed by the attackers once the algorithms or features are known to the phisher.

B. DEEP LEARNING BASED PHISHING WEBPAGE DETECTION METHODS

In recent years, deep learning has been used in various fields as an alternative to traditional machine learning methods and has achieved great success. Some researchers have also applied it to phishing webpage detection [13]–[21]. According to whether the deep neural network is used to extract features automatically, these studies are divided into three categories: ① Methods based on artificial feature engineering. These methods follow the idea of traditional phishing webpage detection research to artificially extract features as the input of deep neural networks. The differences between them include which features are extracted and which deep neural networks are used for learning. For example, literature [13] extracts classic features, such as URL features, domain features, webpage content and encoding features as inputs, and uses deep feedforward neural network for detection, while [14] extracts 56-dimensional features from the URL, page content, and DOM structure, and uses an Auto-Encoder to detect phishing webpages. But these methods still cannot avoid the bias caused by human experience. ② Methods based on automatic feature learning. These methods first reorganize the original data from the webpages into a form that can be learned by the neural networks, and then extract features automatically by the deep neural networks, finally use a traditional machine learning classifier to establish a classification model. According to different sources of the original data, this type of method can be divided into two sub-types: URL-based methods and page content-based methods. Among them, a lot of researches has been done on the URL-based methods because URLs are easy to obtain and deal with. For example, literature [15]–[18] take URLs as text, and uses LSTM [15], [16], DAE (Denoising Autoencoder) [17], CNN [18] respectively to characterize URLs as feature vectors with fixed-length. The page content-based methods regard page content as text instead, and attempt to automatically learn the characteristic representation of the webpages from the page content [19], [20]. For example, literature [20] extracts a series of semantic features of phishing webpages adopting the Word2Vec model. Because they can avoid human bias, the methods based on automatic feature learning have strong generalization ability and are more suitable for the short life cycle and rapid iteration of phishing webpages. However, this kind of method generally uses original, single input, such as URL or page content. Compared with the traditional multi-faceted features, it lacks comprehensive analysis of the webpages, so the detection accuracy needs to be improved. ③ Hybrid methods. The artificial features and automatic features are used simultaneously as the input of the classification model. For example, literature [21] first uses the hybrid CNN-LSTM model to extract URL features automatically, and then combines it with traditional artificial features such as page content features to form a multi-dimensional feature and put it into XGBoost for classification. Although the method makes full use of the advantages of the above two types of methods, it cannot avoid the influence of human experience.

In order to solve the problem that existing methods cannot learn the representation of webpages from multi-aspects automatically, Web2Vec proposed in this paper is designed to learn features automatically from three aspects, including URL, page content, and DOM structure, which do not require prior knowledge about phishing. Then a hybrid deep learning network based on CNN and BiLSTM is used, which can take advantage of CNN to extract the local features, and then take advantage of BiLSTM to extract the global semantic features, and adopting an attention mechanism after BiLSTM to strengthen the learning effect.

III. PROPOSED METHOD

In this section, the formal statement of phishing detection is given firstly, and then the overall framework of Web2Vec and its key technologies are gone into detail.

A. PROBLEM STATEMENT

The goal of Web2Vec is to classify the webpages to be tested as phishing or benign, so the problem is regarded as a binary classification problem. Consider a set of webpages N, $N = \{p_1, \cdots, p_i, \cdots, p_n\}$, where p_i is the i-th webpage, $i = 1, 2, \ldots, n$. $Y_i \in \{-1, +1\}$ represents the label of webpage i, where $Y_i = +1$ represents i is a benign webpage, otherwise i is a phishing webpage. Each webpage consists of three parts, $p_i = \{u_i, h_i, d_i\}$, where u_i represents the URL of the webpage, h_i is the page content, and d_i is the DOM structure. The key is to automatically obtain the representation of the webpage $p_i \rightarrow X_i$, where X_i is the feature matrix of p_i, and then learn the discriminant model $f: X \rightarrow Y$. The discriminant model is used to classify the webpage to be tested.

B. THE OVERALL FRAMEWORK

The Web2Vec consists of five parts as shown in Fig.1: ① Data collection. Obtain phishing webpages and benign webpages from PhishTank and Alexa websites respectively, to form a dataset; ② Webpage parsing. Extract the original data of URL, page content, and DOM structure of each webpage to construct the corresponding corpus; ③ Webpage representation. Use the word embedding technology in NLP to learn corresponding representations of the URL, page content,

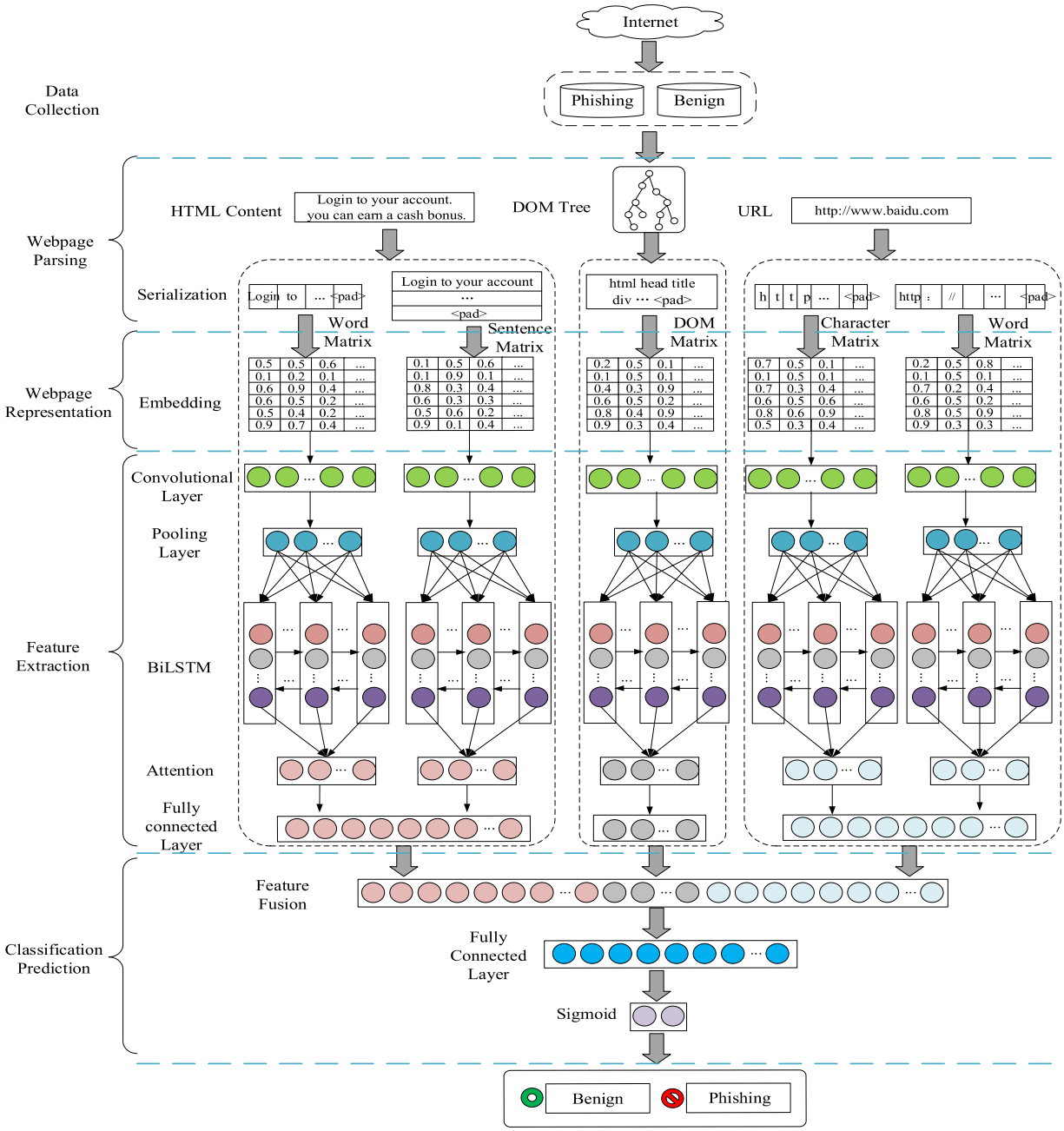

FIGURE 1. Model framework.

and DOM structure; ④ Feature extraction. Input various representation vectors to different hybrid CNN-BiLSTM networks to extract local and global features, and then combine the attention mechanism to strengthen important features; ⑤ Classification prediction. Concatenate the multi-channel output vectors, and use the classifier to determine the category of the tested webpage. Because the main feature of the model is to represent webpages and extract features in all aspects of the webpages, it is named Web2Vec.

The key technologies in the Web2Vec model are described below.

C. WEBPAGE CORPORA CONSTRUCTION

Most related researches have focused on automatically learning webpage features from URLs [15]–[18], [21], because URLs are natural character sequences and can be easily vectorized without preprocessing. But the URL itself does not cover all the structure and semantic information of the phishing webpage. Therefore, Web2Vec learns the comprehensive feature representation of the webpages from the three aspects of URL, page content, and DOM structure. In order to learn these representations automatically, it is necessary to extract the corresponding corpus from the webpages.

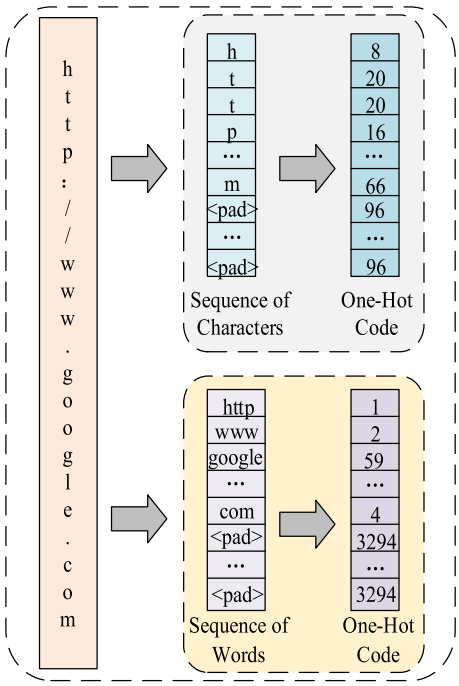

FIGURE 2. Example of construction of URL character corpus and word corpus.

1) URL CORPUS

Drawing on the data processing method of the literature [18], the URLs are processed in units of characters and words respectively.

a: CHARACTER-LEVEL CORPUS

When constructing a character-level corpus, each URL is regarded as a character sequence, and each character sequence is normalized to a fixed-length by the interception or zero paddings, and all character sequences form a sequence set. A total of 96 letters, numbers and special characters with high frequency in the sequence set are selected to form a character vocabulary, including a special symbol <UNK> for replacing infrequent characters and placeholder <PAD>, and then a unique number is assigned to each character in the vocabulary. Finally, each URL character sequence in the sequence set is encoded, that is, the corresponding number is used to replace the original character one by one, so that the one-dimensional digital vector named One-Hot code corresponding to each URL character sequence is obtained. This constitutes the character-level corpus of URLs.

b: WORD-LEVEL CORPUS

If regarding URL as a combination of words, the essence of the URL can be understood from a higher level. The construction of a word-level corpus is similar to that of the character-level corpus. The difference is that URLs are segmented into word sequences instead of character sequences. The word-level splitting of URLs divides each URL into the protocol, hostname, path, file name, and parameter parts according to the structural characteristics of the URL, by separators '.', ':', '//', etc, to highlight the sequence relationship between the parts. Fig.2 shows an example of the construction of character corpus and word corpus for URL.

2) PAGE CONTENT CORPUS

The page content corpus is divided into a word-level corpus and a sentence-level corpus. The construction process is similar to the URL corpus, so it will not be repeated here. It should be noted that the sentence is separated by the character ".''; in order to facilitate processing, the multimedia content, HTML tags, CSS styles, and some other information of the HTML document are removed, only the text information is retained.

3) DOM STRUCTURE CORPUS

An HTML document is a typical semi-structured document. HTML tags in it have a nested relationship and reflect the hierarchical structure of the webpage, which can be characterized by the DOM. For simplicity, when forming a DOM corpus, only the tags that make up the DOM are considered, and their attributes, text, and comment nodes are ignored. The construction of the DOM corpus is divided into two steps: constructing a tag sequence and constructing a corpus from the tag sequence.

First, construct the DOM tag sequence for each webpage. The process is as follows: ① Parse the HTML document and obtain the root node of the DOM tree, and use it as the current layer and the first element of the tag sequence; ② Starting from the current layer, use breadth-first strategy to traverse layer by layer. The specific method is to traverse the child nodes of the nodes in the current layer from left to right, and save them in sequence; ③ Repeat ② until all layers are scanned, and return to the tag sequence. After completing the above steps, the webpage will be converted into a sequence of DOM tags. Fig.3 is an HTML document, and Fig.4 shows how it is converted into the sequence of DOM tags.

Then, the DOM tag sequences formed by all webpages are assembled, and the HTML tags are regarded as the words constituting the sequences, thereby constructing a word-level corpus.

D. WEBPAGE REPRESENTATION

The multiple corpora constructed in the last section have actually completed the vocabulary mapping of each corpus. These mappings are kinds of One-Hot encoding. One-Hot is also known as one-bit effective encoding. Its principle is to use N-bit status registers to encode N states. Each state corresponds to an independent effective register bit. One-Hot cannot reflect the association and semantic information of the corpora, and the encoding result is relatively sparse, so it can only be used for preliminary quantization.

In recent years, the word vector technologies in NLP have been extensively studied and applied, which are for mapping characters or words from a dictionary to low-dimensional vectors. They can not only reduce the dimension, but also capture the context information of the current characters or words in the sequence, so are often used to learn the representation

```
<!DOCTYPE HTML>
1.  <html lang= "en-US">
2.    <head>
3.      <meta charset= "UTF-8">
4.      <title>DOMtree</title>
5.    </head>
6.    <body>
7.      <div>
8.        <ul>
9.          <li>one</li>
10.         <li>two</li>
11.       </ul>
12.       <p>para</p>
13.       <div>
14.         <p>three</p>
15.         <p>four</p>
16.       </div>
17.     </div>
18.   </body>
19. </html>
```

FIGURE 3. HTML document.

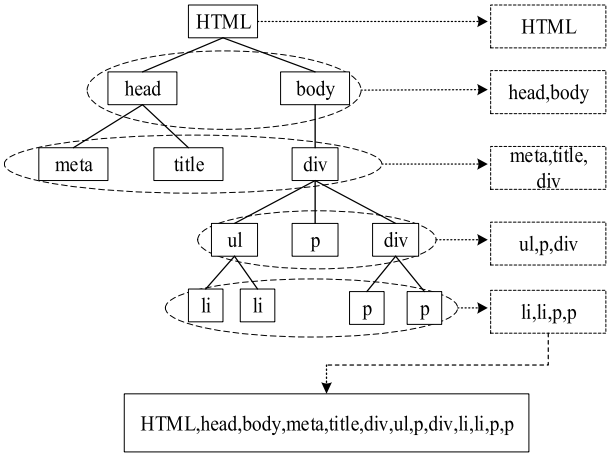

FIGURE 4. DOM tag sequence of Fig.3.

of text-like corpus. However, typical word vector generation methods such as Word2Vec are pre-trained models [22], and their complicated pre-training process will bring a huge burden to the detection of phishing webpages. To this end, a simplified word vector construction method is used in Web2Vec, and the One-Hot matrix after preliminary quantization is embedded as word vectors by a single-layer neural network. The embedding layer and the subsequent feature extraction and classification parts are jointly optimized through back-propagation to gradually enhance the semantic representation ability of the model.

The following takes the URL character representation learning process as an example to explain the principle of representation learning in Web2Vec. Consider the i-th URL u_i in the URL character-level corpus U, and encode each character in u_i. If the j-th character is expressed as g_j after One-Hot encoding, then $g_j = (g_{j1}, g_{j2}, \cdots, g_{jm})^T$ is a vector with a dimension of $m = 96$, m is the size of the URL character dictionary. Each URL forms a One-Hot matrix $G = G_{m\times n} = (g_1, g_2, \cdots, g_n)$. Next, G is mapped into its representation matrix S via single-layer neural network embedding. Where $W \in R^{p\times m}$ is the weight matrix of the embedding layer, and $p = 128$ is the embedding dimension. The calculation process of URL character embedding is:

$$S^c = WG = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ w_{p1} & w_{p2} & \cdots & w_{pm} \end{bmatrix} \times \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ g_{m1} & g_{m2} & \cdots & g_{mn} \end{bmatrix} \quad (1)$$

For the webpage p_i, after representation learning, the representation of the webpage is $X_i = (S_i^c, S_i^w, C_i^w, C_i^s, D_i)$, where S_i^c and S_i^w are representation vectors learned from u_i, C_i^w and C_i^s are vectors learned from the page content, and D_i is learned from the DOM structure. The five vectors are all fixed-length with the same dimensions, where c, w and s represent character-level, word-level and sentence-level respectively. Sequence vectorization makes feature-based mathematical calculations possible. In the next step, feature extraction is further performed based on these five vectors.

E. FEATURE EXTRACTION

In the research and application of deep learning, CNN is the most widely used one because it is good at extracting local features from data, but it lacks the ability to learn contextual information; and on the other hand, LSTM, as a time-recursive neural network, is just suitable for processing sequence information, so in recent years, the combination of CNN and LSTM applied to various types of research has emerged and succeeded [23], [24]. Inspired by this, Web2Vec intends to use a hybrid CNN-LSTM scheme for feature extraction. At the same time, in order to overcome the shortcomings that LSTM only considers the forward information and ignores the backward information, BiLSTM with a bidirectional sequential structure is used instead of LSTM to include all context information into the model. Furthermore, in order to strengthen the influence of important features, the attention mechanism that has been extensively studied and used in recent years is applied to the output of BiLSTM, so that the detection ability of the model can be improved.

1) CNN

The CNN designed in Web2Vec consists of a convolutional layer and a pooling layer. At the convolutional layer, multiple convolution kernels perform convolution operations on the input vectors to generate multiple feature maps; at the pooling layer, the dimension of the feature map is reduced by

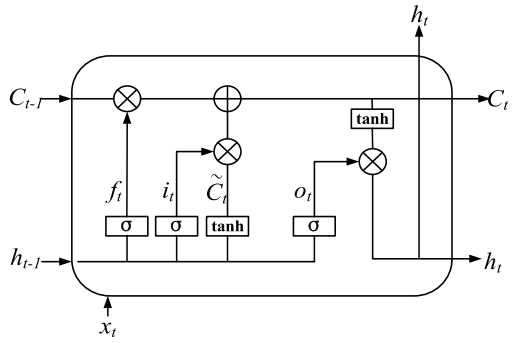

FIGURE 5. LSTM cell structure.

maximum pooling. The input of the convolutional layer is the representation X_i of the webpage i. For a certain convolution kernel W, the matrix R_i after convolution is:

$$R_j = f(W \otimes V_{j:j+h-1} + b) \tag{2}$$

where b is the bias, $\otimes$ refers to the convolution operation, and $f(\cdot)$ is the activation function. In order to reduce network parameters and extract the most important features, 1-Maxpooling is used for the feature map after the convolution operation:

$$X_j = max(R_j) \tag{3}$$

CNN extracts the local features by convolution and pooling of input data. Its ability to tolerate noise and deformation can better deal with typical obfuscation techniques that do not change malicious attacks.

2) BiLSTM

The output vector of CNN is used as the input of BiLSTM, which is formed by linking LSTM in two directions, forward and reverse.

LSTM uses a memory cell structure to replace the hidden layer of general neural networks. Its cell structure is shown in Fig. 5 [25].

The memory cell of LSTM is mainly composed of the input gate, output gate, and forget gate. The update process of the t-th cell is given below.

$$\begin{Bmatrix} i_t \\ f_t \\ o_t \end{Bmatrix} = \sigma\left(\begin{Bmatrix} W_i \\ W_f \\ W_o \end{Bmatrix} h_{t-1} + \begin{Bmatrix} U_i \\ U_f \\ U_o \end{Bmatrix} x_t + \begin{Bmatrix} b_i \\ b_f \\ b_o \end{Bmatrix}\right) \tag{4}$$

$$\tilde{C} = tanh(W_c h_{t-1} + U_c x_t + b_c) \tag{5}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{6}$$

$$h_t = o_t \cdot tanh(C_t) \tag{7}$$

Among them, x_t is the current input, $i, o, f, \tilde{C}, C, h$ represent input gate, output gate, forget gate, temporary memory cell state, memory cell state, hidden layer output value; W_f, U_f, W_i, U_i, W_o, U_o, W_c, U_c are weight matrices, and b_f, b_i, b_c, b_c are biases.

On the t-th cell, BiLSTM finally synthesizes the feature vectors obtained in both directions as output.

There are complex logical relationships among elements of webpages. BiLSTM can effectively learn a variety of time scales and long-distance dependence in the webpages, and further extract the potential semantic information on the basis of local features learned by CNN.

3) ATTENTION MECHANISM

The attention mechanism realizes the extraction of significantly fine-grained features in sequence data by paying attention to specific information. In the feature vector output by BiLSTM, each feature item does not affect the detection result of the phishing webpage evenly. By giving different features different attention, the model can obtain higher decision-making ability for important features. The calculation formulae are as follows:

$$a = tanh(h) \tag{8}$$

$$\alpha = softmax(w^T a) \tag{9}$$

$$x = \sum_t \alpha h \tag{10}$$

where w is the weight, h is the output of BiLSTM, α represents the attention of h, and x is the feature vector after weighted summation.

F. CLASSIFICATION PREDICTION

After extracting the features, the feature vectors output from each channel are concatenated to form the fusion vector X_i of the webpage i, and the category prediction of the webpage is achieved through the fully connected layer and the sigmoid function. During the training process, a cross-entropy loss function is used to calculate the error between the true value and the predicted one. Let y_i' be the predicted value and y_i the true category, then the loss function is:

$$J(Y', Y) = -\frac{1}{n}\sum_{i=1}^{n} [y_i \log y_i' + (1 - y_i)\log(1 - y_i')] \tag{11}$$

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In order to verify the effectiveness of the Web2Vec model, four sets of experiments are designed to try to answer the following questions:

1) Question 1: Compared with the classic phishing webpage detection methods, how effective is the detection of the Web2Vec model?

2) Question 2: Does the multifaceted representation learned from the original information of the webpages using the representation learning method effectively improve the detection result?

3) Question 3: Does the hybrid CNN-BiLSTM network for feature extraction in the Web2Vec model have advantages over other typical deep learning networks?

4) Question 4: Does using the attention mechanism in the Web2Vec model improve performance?

A. EXPERIMENT PREPARATION

1) EXPERIMENTAL ENVIRONMENT AND DATASET

The experimental development environment is shown in Table 1.

TABLE 1. Development environment.

Operating System	Processor	Memory	Development Environment	Development Language
Windows7	Intel Core i5.3470 CPU	16GB	pyCharm	Python3.5

TABLE 2. Evaluation indicators.

Evaluation indicator	Calculation formula
Accuracy	(TP+TN) / (TP+FP+TN+FN)
Precision	TP / (TP + FP)
TPR(Recall)	TP / (TP + FN)
F1	(2×Precision×Recall) / (Precision + Recall)
FPR	FP / (TN + FP)

The webpages used in the experiments come from the real network environment. The benign webpages collection is from Alexa. Alexa is a website maintained by Amazon that publishes the world rankings of websites. It has a huge number of URLs and detailed website ranking information. We collect webpages in the top list provided by Alexa which are considered as benign webpages. After filtering out some invalid, error, and duplicate pages, we collected 24,800 normal webpages from Alexa.

The phishing webpage collection comes from PhishTank.com. PhishTank is an internationally well-known phishing webpage collection website that provides a timely and authoritative list of phishing webpages. PhishTank collects a suspected phish submitted by anyone and then verifies it according to whether it has a fraudulent attempt or not before publishing. Due to the short survival time of phishing webpages, we collected 21,303 phishing webpages listed on PhishTank from September 2019 to November 2019, and preprocessed the webpages that did not meet the grammar rules. The ratio of the training set to the test set is 0.75:0.25.

2) EVALUATION STANDARD

To summarize various evaluation indicators in the literatures, the most commonly used are the following: Accuracy, Precision, True Positive Rate (TPR), which is equivalent to Recall, FPR, F1-measure, and their calculation formulas are shown in Table 2.

Among them, TP (True Positive) denotes the number of benign webpages correctly classified as benign webpages, FP (False Positive) denotes the number of phishing webpages classified as benign webpages, TN (True Negative) denotes the number of phishing webpages classified as phishing webpages, and FN (False Negative) denotes the number of benign webpages classified as phishing webpages. F1 is the harmonic mean of Precision and Recall, which can comprehensively reflect the performance of the method.

3) BASELINES

Classic phishing webpage detection methods compared with Web2Vec include PCA-RF [12], CANTINA+ [3], URLNet [18], and MPURNN [15]. Among them, PCA-RF is a typical machine learning approach based on artificial and heuristic features, which has the state-of-the-art performance. On the other hand, CANTINA+ is the most recognized heuristic method. Both methods manually extract features from all aspects of URL, page content, and DOM structure; On the other hand, URLNet and MPURNN are deep learning methods, which both automatically learn features from URL. The difference is that URLNet performs feature extraction by CNN after character-level and word-level embedding, while MPURNN only embeds characters, and then extracts features through LSTM.

When comparing the feature extraction methods in the Web2Vec, the classic single deep learning networks CNN, RNN, LSTM, as well as the hybrid network CNN-LSTM and CNN-BiLSTM without adding attention were selected.

Notice that the traditional supervised machine learning methods such as Sequential Minimal Optimization (SMO), Bayesian Network (BN), Support Vector Machine (SVM), and AdaBoost are not compared in our experiments, because from the experimental comparison in [12], PCA-RF performed the best out of all these baselines.

TABLE 3. Parameters of Web2Vec.

Parameter	Value
CNN pool_size	3
CNN kernel_size	128
Strides	1
BiLSTM cell number	128
Batch	64
Dropout	0.5
Epoch	10
Learning rate	0.001

4) PARAMETER SETTING

The parameter settings of the Web2Vec are shown in Table 3.

In addition, the lengths of the URLs, the HTML content, and the DOM structure of each webpage are inconsistent, and must be set to a fixed length during the calculation. According to the distribution statistics of different lengths, set the URL length, HTML length, and DOM structure length to 200, 1000 and 2000 respectively.

The source code and dataset used in the experiments are listed on https://github.com/Hanjingzhou/Web2vec.

B. RESULTS EVALUATION

1) EXPERIMENT 1: Web2Vec DETECTION EFFECT

Question 1 aims to access the detection effect of Web2Vec. In order to answer Question 1, experiment 1 compares Web2Vec with classic phishing webpage detection methods PCA-RF, CANTINA+, URLNet, and MPURNN. The results are shown in Table 4.

As can be seen from Table 4, generally speaking, the detection effect of the PCA-RF is the best, and Web2Vec shows sub-optimal results and CANTINA+ has the third place. This is because these three methods have carried out multi-aspects learning on the webpages, and Web2Vec achieves lower performance than PCA-RF because lacking enough data when

TABLE 4. Comparison with classic phishing webpage detection methods.

Model	Accuracy	Precision	TPR	F1	FPR
Web2Vec	0.9905	0.9869	0.9826	0.9908	**0.0025**
PCA-RF	**0.9921**	**0.9915**	**0.9828**	**0.9913**	0.0035
CANTINA+	0.9754	0.9911	0.9754	0.9751	0.0086
URLNet	0.9633	0.9620	0.8503	0.9027	0.0031
MPURNN	0.9326	0.9541	0.8980	0.9252	0.0374

TABLE 5. Detection effects of different feature combinations.

Feature combination	Accuracy	Precision	TPR	F1	FPR
DOM	0.9171	**0.9972**	0.9623	0.9150	0.1221
URL	0.9472	0.9638	0.9162	0.9416	0.0259
HTML	0.9756	0.9788	0.9684	0.9736	0.0181
DOM+URL	0.9639	0.9651	0.9568	0.9609	0.0029
DOM+HTML	0.9770	0.9682	**0.9828**	0.9754	0.0280
URL+HTML	0.9834	0.9829	0.9813	0.9821	0.0147
URL+HTML+DOM	**0.9905**	0.9869	0.9826	**0.9908**	**0.0025**

using deep learning networks, but its performance is close to PCA-RF and still better than CANTINA+ for learning more latent information through deep learning and representation learning. URLNet and MPURNN only learn from URLs and use a single deep learning network for feature extraction, the essential features of webpages are not learned enough, so the classification effect is not ideal.

The computational complexity of the methods depends on the extraction and computing the features from webpages. In experiment 1, CANTINA+ spend the shortest operation time because after obtaining features manually, it only needs to use a simple heuristic rule to make a decision, while deep learning-based methods such as Web2Vec and URLNet need to run many epochs to get the best result. Because PCA-RF uses the ensemble method, it is also slower than the heuristic method. Web2Vec spent the longest running time, because it needs to realize representation learning before deep leaning.

2) EXPERIMENT 2: EFFECTS OF MULTI-FACETED FEATURE LEARNING

Question 2 aims to consider the necessity and effect of learning webpage features from multiple aspects. In order to answer Question 2, experiment 2 combines different features learned from URL, page content, and DOM structure to form multiple sets of different inputs to examine the impact of various features on the detection results. Table 5 shows the detection results obtained by combining different features.

It can be seen from Table 5 that, the best results have been achieved from comprehensive learning features through the combination of URL, page content, and DOM structure. In the existing research, although there are not many researches on phishing webpages from all three aspects of URL, page content, and DOM structure, it is common to study one or a combination of the two as the research object. This shows that the information in different parts of the webpage can reflect some characteristics of the phishing webpage, for example, the content of the page can reflect semantic characteristics, the DOM structure can reflect structural characteristics, etc., and the combination of them will definitely improve the detection effect. This also explains that the method of learning two aspects of information (such as URL + HTML) in Table 5 is better than the method of learning features from only one aspect.

TABLE 6. Detection effects of different feature extraction models.

Model	Accuracy	Precision	TPR	F1	FPR
CNN-BiLSTM	**0.9905**	0.9869	**0.9826**	**0.9908**	**0.0025**
CNN-LSTM	0.9560	**0.9995**	0.9756	0.9803	0.0032
LSTM	0.9654	0.9924	0.9325	0.9616	0.0061
RNN	0.9665	0.9922	0.9351	0.9628	0.0063
CNN	0.9786	0.9844	0.9756	0.9800	0.0085

TABLE 7. Detection effect of attention mechanism.

Web2Vec	Accuracy	Precision	TPR	F1	FPR
With attention	**0.9905**	0.9869	**0.9826**	**0.9908**	**0.0025**
Without attention	0.9835	**0.9890**	0.9757	0.9823	0.0050

3) EXPERIMENT 3: THE EFFECTIVENESS OF HYBRID CNN-BiLSTM DEEP LEARNING NETWORK

Question 3 aims to consider the effect of feature extraction using the hybrid deep learning network CNN-BiLSTM in the Web2Vec model. In order to answer Question 3, experiment 3 replaced CNN-BiLSTM in the Web2Vec with CNN-LSTM, LSTM, RNN, and CNN respectively. The comparison results are shown in Table 6.

It can be seen from Table 6 that, the detection efficiency of a single network model is significantly lower than that of a hybrid one. Compared with a single model, a hybrid model can extract the latent features of phishing webpages from multiple levels, which is worthy of in-depth research and application. It also shows that the CNN-BiLSTM network has a better classification detection effect than the CNN-LSTM, which means that it is meaningful to perform bidirectional feature extraction.

4) EXPERIMENT 4: EFFECTIVENESS OF ATTENTION MECHANISM

Question 4 examines the effectiveness of using the attention mechanism in the Web2Vec. Therefore, the detection effect of the Web2Vec with and without attention mechanism is compared. The results are shown in Table 7.

It can be seen from Table 7 that the effect of the Web2Vec model with the attention mechanism is significantly better than that without the attention mechanism. This shows that increased attention to the output of BiLSTM can highlight important feature information and effectively improve the classification detection effect.

In order to better illustrate the improvement of the classification effect of the attention mechanism, Fig.6 shows the changes in Accuracy and Loss during training and testing with or without the attention. It can be seen from Fig.6 that the model with attention converges faster and the training and testing process is more stable.

The above four groups of experiments show that the Web2Vec model can represent webpages in multi-aspects

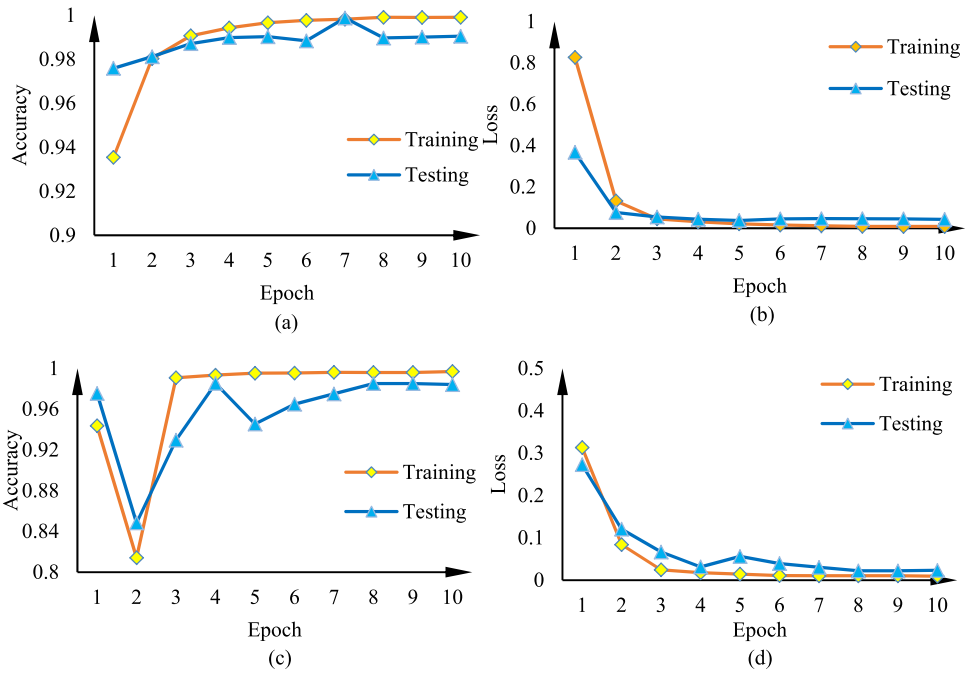

FIGURE 6. The effect of the attention mechanism.

through representation learning, use the hybrid deep learning network CNN-BiLSTM for feature extraction, and use attention mechanism to further improve classification performance. These strategies are all feasible and effective. The prediction effect of the Web2Vec model is ideal.

V. CONCLUSION

A phishing webpage detection model Web2Vec based on representation learning and deep learning is proposed in the paper. The model uses the representation learning technology in NLP to comprehensively learn the representation of webpages from the URL, page content, and DOM structure; then construct a multi-channel hybrid deep learning network to extract the deep hidden features of the webpages and then use the attention mechanism to strengthen the influence of important features; finally, the feature extraction results of different channels are fused for classification prediction. Four sets of experiments verified the classification results of the Web2Vec model from different angles.

With the rapid development of representation learning technology, the deep representation learning on the graph, namely Graph Neural Network (GNN), has been extensively studied. The research object of this paper, webpages, are linked to each other, naturally forming a graph. How to dig deep into the link characteristics of phishing webpages, to construct a large graph structure that can reflect the phishing characteristics, and use powerful analysis and processing capabilities of GNN to find more differentiated phishing webpage detection method is our further research direction.

REFERENCES

[1] APWG. *Phishing Activity Trends Report, 2th Quarter 2020*. Accessed: Aug. 27, 2020. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q2_2020.pdf

[2] P. M, S. L., and C. Thomas, "A static approach to detect drive-by-download attacks on Webpages," in *Proc. Int. Conf. Control Commun. Comput. (ICCC)*, Thiruvananthapuram, India, Dec. 2013, pp. 298–303.

[3] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "CANTINA+: A feature-rich machine learning framework for detecting phishing Web sites," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 2, pp. 1–21, Sep. 2011, doi: 10.1145/2019599.2019606.

[4] M. Moghimi and A. Y. Varjani, "New rule-based phishing detection method," *Expert Syst. Appl.*, vol. 53, pp. 231–242, Jul. 2016, doi: 10.1016/j.eswa.2016.01.028.

[5] M. N. Raj and P. J. Vithalpura, "A survey on phishing detection based on visual similarity of Web pages," *Int. J. Sci. Res. Sci., Eng. Technol.*, vol. 4, no. 2, pp. 81–86, Jul. 2018, doi: 10.32628/IJSRSET.

[6] J.-X. Cao, B. Mao, J.-Z. Luo, and B. Liu, "A phishing Web pages detection algorithm based on nested structure of Earth Mover's distance (Nested-EMD)," *Chin. J. Comput.*, vol. 32, no. 5, pp. 922–929, Aug. 2009, doi: 10.3724/SP.J.1016.2009.00922.

[7] R. S. Rao and A. R. Pais, "Two level filtering mechanism to detect phishing sites using lightweight visual similarity approach," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 9, pp. 3853–3872, Sep. 2020, doi: 10.1007/s12652-019-01637-z.

[8] A. Aleroud and L. Zhou, "Phishing environments, techniques, and countermeasures: A survey," *Comput. Secur.*, vol. 68, pp. 160–196, Jul. 2017, doi: 10.1016/j.cose.2017.04.006.

[9] K. Sahu and S. K. Shrivastava, "Kernel K-means clustering for phishing Website and malware categorization," *Int. J. Comput. Appl.*, vol. 111, no. 9, pp. 20–25, Feb. 2015, doi: 10.5120/19565-1326.

[10] S. Lee and J. Kim, "WarningBird: A near real-time detection system for suspicious URLs in Twitter stream," *IEEE Trans. Dependable Secure Comput.*, vol. 10, no. 3, pp. 183–195, May 2013, doi: 10.1109/TDSC.2013.3.

[11] B. Liang, M. Su, W. You, W. Shi, and G. Yang, "Cracking classifiers for evasion: A case study on the Google's phishing pages filter," in *Proc. 25th Int. Conf. World Wide Web*, Montréal, QC, Canada, Apr. 2016, pp. 345–356.

[12] R. S. Rao and A. R. Pais, "Detection of phishing Websites using an efficient feature-based machine learning framework," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 3851–3873, Aug. 2019, doi: 10.1007/s00521-017-3305-0.

[13] G. Vrbancic, I. Fister, and V. Podgorelec, "Swarm intelligence approaches for parameter setting of deep learning neural network: Case study on phishing Websites classification," in *Proc. 8th Int. Conf. Web Intell., Mining Semantics (WIMS)*, New York, NY, USA, 2018, pp. 1–8.

[14] J. Feng, L. Y. Zou, and T. Z. Nan, "A phishing Webpage detection method based on stacked autoencoder and correlation coefficients," *J. Comput. Inf. Technol.*, vol. 27, no. 2, pp. 41–54, Nov. 2019, doi: 10.20532/cit.2019.1004702.
[15] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. Gonzalez, "Classifying phishing URLs using recurrent neural networks," in *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, Scottsdale, AZ, USA, Apr. 2017, pp. 1–8.
[16] W. Chen, W. Zhang, and Y. Su, "Phishing detection research based on LSTM recurrent neural network," in *Proc. Int. Conf. Pioneering Comput. Sci.*, Zhengzhou, China, 2018, pp. 638–645.
[17] S. Douzi, M. Amar, and B. El Ouahidi, "Advanced phishing filter using autoencoder and denoising autoencoder," in *Proc. Int. Conf. Big Data Internet Thing (BDIOT)*, London, U.K., 2017, pp. 125–129.
[18] H. Le, Q. Pham, D. Sahoo, and S. C. Hoi, "URLNet: Learning a URL representation with deep learning for malicious URL detection," *CoRR*, vol. abs/1802.03162, pp. 1–13, Feb. 2018.
[19] P. Yi, Y. Guan, F. Zou, Y. Yao, W. Wang, and T. Zhu, "Web phishing detection using a deep learning framework," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–9, Sep. 2018, doi: 10.1155/2018/4678746.
[20] X. Zhang, Y. Zeng, X.-B. Jin, Z.-W. Yan, and G.-G. Geng, "Boosting the phishing detection performance by semantic analysis," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Boston, MA, USA, Dec. 2017, pp. 1063–1070.
[21] P. Yang, G. Zhao, and P. Zeng, "Phishing Website detection based on multidimensional features driven by deep learning," *IEEE Access*, vol. 7, pp. 15196–15209, 2019, doi: 10.1109/ACCESS.2019.2892066.
[22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, New York, NY, USA, 2013, pp. 3111–3119.
[23] P. Sun, P. Liu, Q. Li, C. Liu, X. Lu, R. Hao, and J. Chen, "DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system," *Secur. Commun. Netw.*, vol. 2020, pp. 1–11, Aug. 2020, doi: 10.1155/2020/8890306.
[24] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B.-W. On, "Fake news stance detection using deep learning architecture (CNN-LSTM)," *IEEE Access*, vol. 8, pp. 156695–156706, 2020, doi: 10.1109/ACCESS.2020.3019735.
[25] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019, doi: 10.1109/TSG.2017.2753802.

JIAN FENG was born in Xi'an, Shaanxi, China, in 1973. She received the Ph.D. degree in computer software and theory from Northwest University, Xi'an, in 2008.

Since 2010, she has been an Assistant Professor with the College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an. She is the author of two books and more than 30 articles. She holds three patents. Her research interests include computer network and communication, network security, and distributed computing.

LIANYANG ZOU was born in Hunan, China. He received the B.S. degree in computer science and technology from Xinxiang University, Xinxiang, in 2017, and the M.S. degree in computer technology from the Xi'an University of Science and Technology, Xi'an, in 2020.

From 2017 to 2020, he engaged in research on phishing webpage detection. He is currently working with the Head Office Information Technology Department of SPD Bank, Application Development Service Sub-center (Xi'an). His research interests include network security and Internet data mining.

Mr. Zou's awards and honors include the Academy Scholarship and the Second Prize for the China Graduate Electronic Design Competition (Business Plan Special Competition).

OU YE received the B.S. degree in computer science and engineer and the M.S. and Ph.D. degrees in computer software and theory and mechanical engineering from the Xi'an University of Technology, in 2007, 2010, and 2014, respectively.

He is currently an Associate Professor with the School of Computer Science and Technology, Xi'an University of Science and Technology, China. His current research interests include data cleansing, video retrieval, and image processing.

JINGZHOU HAN was born in Lianyungang, Jiangsu, China. He received the B.S. degree in applied chemistry from the Xuzhou University of Technology, in 2019. He is currently pursuing the M.S. degree in computer science and technology with the Xi'an University of Science and Technology, China.

Since 2020, he has been engaged in the research of knowledge graph. His research interests include network security and recommendation systems.

Mr. Han's awards and honors include the First Prize for the China Undergraduate Mathematical Contest in Modeling.