Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 4678746, 9 pages
https://doi.org/10.1155/2018/4678746

Research Article

Web Phishing Detection Using a Deep Learning Framework

Ping Yi,[1] Yuxiang Guan,[1] Futai Zou,[1] Yao Yao,[2] Wei Wang,[2] and Ting Zhu[2]

[1] *School of Cyber Security, Shanghai Jiao Tong University, Shanghai 200240, China*
[2] *Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD 21250, USA*

Correspondence should be addressed to Ping Yi; yiping@sjtu.edu.cn

Received 2 June 2018; Revised 1 August 2018; Accepted 2 September 2018; Published 26 September 2018

Academic Editor: Tony T. Luo

Copyright © 2018 Ping Yi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Web service is one of the key communications software services for the Internet. Web phishing is one of many security threats to web services on the Internet. Web phishing aims to steal private information, such as usernames, passwords, and credit card details, by way of impersonating a legitimate entity. It will lead to information disclosure and property damage. This paper mainly focuses on applying a deep learning framework to detect phishing websites. This paper first designs two types of features for web phishing: original features and interaction features. A detection model based on Deep Belief Networks (DBN) is then presented. The test using real IP flows from ISP (Internet Service Provider) shows that the detecting model based on DBN can achieve an approximately 90% true positive rate and 0.6% false positive rate.

1. Introduction

Web service is a communication protocol and software between two electronic devices over the Internet [1]. Web services extends the World Wide web infrastructure to provide the methods for an electronic device to connect to other electronic devices [2]. Web services are built on top of open communication protocols such as TCP/IP, HTTP, Java, HTML, and XML. Web service is one of the greatest inventions of mankind so far, and it is also the most profound manifestation of computer influence on human beings [3].

With the rapid development of the Internet and the increasing popularity of electronic payment in web service, Internet fraud and web security have gradually been the main concern of the public [4]. Web Phishing is a way of such fraud, which uses social engineering technique through short messages, emails, and WeChat [5] to induce users to visit fake websites to get sensitive information like their private account, token for payment, credit card information, and so on.

The first phishing attack on AOL (America Online) can be traced back to early 1995 [6]. A phisher successfully obtained AOL users personal information. It may lead to not only the abuse of credit card information, but also an attack on the online payment system entirely feasible.

The phishing activity in early 2016 was the highest ever recorded since it began monitoring in 2004. The total number of phishing attacks in 2016 was 1,220,523. This was a 65 percent increase over 2015. In the fourth quarter of 2004, there were 1,609 phishing attacks per month. In the fourth quarter of 2016, there was an average of 92,564 phishing attacks per month, an increase of 5,753% over 12 years [7]. According to the 3rd Microsoft Computing Safer Index Report released in February 2014, the annual worldwide impact of phishing could be as high as $5 billion [8]. With the prevalence of network, phishing has become one of the most serious security threats in modern society, thus making detecting and defending against web phishing an urgent and essential research task. Web phishing detection is crucial for both private users and enterprises [9].

Some possible solutions to combat phishing were created, including specific legislation and technologies. From a technical point of view, the detection of phishing generally includes the following categories: detection based on a black list [10] and white list, detection based on Uniform Resource Locator (URL) features [11], detection based on web content, and detection based on machine learning. The antiphishing way using blacklist may be an easy way, but it cannot find new phishing websites. The detection on URL is to analyze the features of URL. The URL of phishing websites may be very similar to real websites to the human eye, but they are different in IP. The content-based detection usually refers to the detection of phishing sites through the pages of elements, such as form information, field names, and resource reference.

In this paper, we will focus on the detection model using a deep learning framework. The main contributions are as follows:

(i) We present two feature types for web phishing detection: an original feature and an interaction feature. The original feature is the direct feature of URL, including special characters in URL and age of the domain. The interacting feature is the interaction between websites, including in-degree and out-degree of URL.

(ii) We introduce DBN to detect web phishing. We discuss the training process of DBN and get the appropriate parameters to detect web phishing.

(iii) We use real IP flows data from ISP to evaluate the effectiveness of the detection model on DBN. True Positive Rate (TPR) with different parameters is analyzed; our TPR is approximately 90%.

The paper is organized as follows. Related works are discussed in Section 2. The detection model and algorithm are discussed in Section 3. DBN is tested and evaluated in Section 4. The conclusion is drawn in Section 5.

2. Related Works

Researchers have conducted lot of work in security [12–18], including secure routing [19–21], intrusion detection [22–27], intrusion prevention [28], and smart grids security [29]. Different from research problems in wireless networks [30–60] and energy networks [61–64], web phishing is the attempt to acquire sensitive information such as usernames, passwords, and credit card details, often for malicious reasons, by masquerading as a trustworthy website on the Internet. Researchers present some solutions to detect web phishing as follows.

When we judge whether a specific website is web phishing, the direct way is to use a white list or black list. We may search the URL in some database and decide. Pawan Prakash *et al.* [10] presented two ways to detect phishing websites by the blacklist. The first way includes five heuristics to enumerate simple combinations of known phishing sites to discover new phishing URLs. The second way consists of an approximate matching algorithm that dissects a URL into multiple components that are matched individually against entries in the blacklist. Many well-known browser vendors such as Firefox [65] and Chrome [66] also used a self-built or third-party black-white list, to identify whether the URL is a phishing site. This method is very accurate, but its blacklist or whitelist usually relies on manual maintaining and reviewing. Obviously, these methods are not real time and may cost a lot of time and effort.

Another phishing detection way is to analyze the features of URL. For example, sometimes a URL looks similar to the famous site URL or contains some special characters in the URL. Samuel Marchal *et al.* [11] used one concept of intra-URL relatedness and evaluate it using features extracted from words that compose a URL based on query data from Google and Yahoo search engines. These features are then used in machine-learning-based classification to detect phishing URLs from a real data set. This method is efficient and economical because it utilizes the preexisting knowledge of the URL, which has a fast detection speed and a lower cost. However, we cannot fully exploit the characteristics of phishing in terms of an URL only because the essence of the scheme is to fraud by means of web content. Phishing attackers are very likely familiar with URLs and easily tailor their URLs to avoid detection; therefore this method will result in a lower detection rate if only the information of the URL is checked.

The content-based detection usually refers to the detection of phishing sites through the pages of elements, such as form information, field names, and resource reference. Anthony Fu *et al.* [67] proposed an approach to detect phishing web page using Earth mover's distance (EMD) to measure web page visual similarity. The accuracy rate of this method is high. But at the same time the downside is a need to collect large amounts of data as a priori knowledge.

With the popularity of machine learning, phishing detection has focused on the use of machine learning algorithms. This method integrates URL text features, domain name features, and web content features into a unified detection basis. W. Chu *et al.* [68] presented a machine learning algorithm based on phishing detection using only lexical and domain features. J. Ma *et al.* [69] described an approach to classifying URLs automatically as either malicious or benign based on supervised learning across both lexical and host-based features. In general, the essence of these methods of machine learning detection is to map all the features of the phishing website into the same space and then to use the machine learning and data mining algorithms to detect phishing.

3. The Phishing Detection Model Based on DBN

3.1. Phishing Feature Extraction and Definition. First, we get real traffic flow from ISP. The data set includes traffic flow for 40 minutes and 24 hours. We construct the graph structure of traffic flow and analyze the characteristics of web phishing from the view of the graph.

Each piece of data contains the following fields.

(i) *AD*: user node number.

(ii) *IP*: user IP address.

(iii) *TS*: access time.

(iv) *URL*: Uniform Resource Locator, access web address.

(v) *REF*: request page source.

(vi) *UA*: user browser type.

(vii) $DST - IP$: server address to access.

(viii) *CKE*: User Cookie.

A graph is mathematical structures used to model pairwise relations between objects. It is also a very direct way to describe the relationship between nodes in a network. The relationship between the nodes on the Internet can also be expressed through the graph structure. Therefore, we construct a graph to store the real traffic flow data and describe the relationship between the nodes in traffic flow.

Give an undirected graph $G = (V, E)$, where V includes two kinds of node:

(i) user node AD;

(ii) access URL and REF. $E \subset V \times V$ denotes an access relationship between REF, AD, and URL.

The vertices of the graph $G = (V, E)$ are as follows:

(i) User node V_{AD} has one attribute: total access times (vertex out-degree).

(ii) User node V_{URL} has two attributes: total accessed times (vertex in-degree) and website registration time.

The edges of the graph $G = (V, E)$ are as follows:

(i) The number of visits: which corresponds to the number of occurrences of the edge, the number of times an AD may have access to a URL, or the number of direct links between two URLs, depending on the corresponding vertex type.

(ii) Cookie: the cookie field in the access record.

(iii) UA: User Agent in the access record.

3.2. Feature Definition. We define two kinds of features to detect web phishing, and they are an original feature and interactive feature.

3.2.1. Original Feature. There are some features in the phishing URL, such as special characters. We definite these features in URL as an original feature as follows:

(i) O_1: there are special characters in URL, such as @, Unicode, and so on. Those special characters are not allowed in a normal URL.

(ii) O_2: there are too many dots or less than four dots in normal URL.

(iii) O_3: the age of the domain is too short. For example, the age of the normal domain is more than 3 months.

In order to quantify the above characteristics, all the characteristic values are binary, that is, one of 0 or 1. Intuitively, the more of the 1 appear in the feature, the higher the likelihood that the site will be a phishing site.

3.2.2. Interaction Feature. There are some features in graph $G = (V, E)$, such as access frequency. We define these features through a node relationship as interaction feature as follows:

(i) I_1: in-degree of URL node from REF is very small. In general, the normal websites do not link to phishing sites. The phishing sites are directly accessed.

(ii) I_2: out-degree of URL node is very small. In order to get personal private information, the phishing sites are usually terminal websites and do not link to the other sites.

(iii) I_3: the frequency of URL from AD is one. In general, one user accesses the phishing site only one time and the user cannot access the phishing site more than one time.

(iv) I_4: when AD accesses URL, user browser type UA is not the main browser. Well-known browser vendors often have a built-in filtering phishing site plug-in. A user who uses unknown browsers is more likely to access the phishing sites.

(v) I_5: there is no cookie in user. The phishing site does not leave its cookie in user.

```
Require: Visible Layer $V = \{v_1, \dots, v_m\}$, Hidden Layer
    $H = \{h_1, \dots, h_n\}$
Ensure: Gradient Approximation $\Delta\theta \longleftarrow \Delta w_{ij}, \Delta a_i, \Delta b_j$ for
    i in $\{1 \dots n\}$, j in $\{1 \dots m\}$
for $i$ in $\{1 \dots n\}$, $j$ in $\{1 \dots m\}$ do
    Initialize $\Delta w_{ij} = \Delta a_i = \Delta b_j = 0$
end for
for Each $v$ in V do
    $v^0 \longleftarrow v$
    for $t$ in $\{0 \dots k-1\}$ do
        for $i$ in $\{1 \dots n\}$ do
            Sample $h_i^t \sim p(h_i|v^t)$
        end for
        for $j$ in $\{1 \dots m\}$ do
            Sample $v_j^t \sim p(v_j|h^t)$
        end for
    end for
end for
for $i$ in $\{1 \dots n\}$, $j$ in $\{1 \dots m\}$ do
    $\Delta w_{ij} \longleftarrow \Delta w_{ij} + p(h_i|v^0)v_j^0 - p(h_i|v^k)v_j^k$
    $\Delta a_i \longleftarrow \Delta a_i + p(h_i|v^0) - p(h_i|v^k)$
    $\Delta b_j \longleftarrow \Delta b_j + v_j^0 - v_j^k$
end for
```

Algorithm 1: k-step CD-k.

3.3. Detection Based on DBN. DBN is one of the deep learning models, each of which is a restricted type of Boltzmann machine that contains a layer of visible units that representing the data [70].

DBN can extract phishing features from a data set. The key to training a DBN is how to determine some parameters. According to Hinton and Salakhutdinov [71], we select Contrastive Divergence (CD) as training algorithm, which calculates the gradient through k times of Gibbs Sampling [72]. The pseudocode of k-step CD-k is in Algorithm 1.

$w_{i,j}$ is the weight matrix of all edges, a_i and b_j are, respectively, the offset vector of the visible and hidden layers, and Sample is Gibbs Sampling [72]. We can get a set of

```
Require: Period $T$, Learning Rate $\eta$, Momentum $\rho$, Visible Layer $V$, Hidden Layer $H$, Number of visible and hidden layer units $n_v, n_h$, Offset Vector $a, b$, Weight Matrix $W$
Ensure: $\theta = \{W, a, b\}$
Initialize $W, a, b$
for $i \in \{1 \ldots T\}$ do
    Calling CD-$k$ to generate $\Delta\theta = \{\Delta W, \Delta a, \Delta b\}$
    $W \longleftarrow \rho W + \eta((1/n_v)\Delta W)$
    $a \longleftarrow \rho a + \eta((1/n_v)\Delta a)$
    $b \longleftarrow \rho b + \eta((1/n_v)\Delta b)$
end for
```

Algorithm 2: Training process.

parameters $\theta = w_{ij}, a_i, b_j$ by this algorithm. The gradient of formula is as

$$CD_k\left(\theta, v^0\right) = -\sum_h p\left(h \mid v^0\right) \frac{\partial E\left(v^0, h\right)}{\partial\theta} + \sum_h p\left(h \mid v^k\right) \frac{\partial E\left(v^k, h\right)}{\partial\theta} \tag{1}$$

First, we set initialization parameters. The weight matrix obeys the normal distribution (0,0.01). We set visible layer offset a_i as

$$a_i = \ln \frac{p\left(v_i\right)}{1 - p\left(v_i\right)} \tag{2}$$

where $p(v_i)$ is the probability of the i in the active state. For the original feature, we can determine the characteristics of nonphishing sites and then calculate the ratio of nonphishing sites to take the back, that is, a_i. We set the offset vector of hidden layers as 0. After initialization, we start the training process, and pseudocode is in Algorithm 2.

The iteration period T and k of CD-k do not have to select a large number. Hinton [71] discussed that the algorithm can get to good result even if $k = 1$. The parameter η is related to the concept of gradient ascent in Maximum likelihood Approximation in Restricted Boltzmann Machine (RBM).

$$L_\theta = \ln\left(L\left(\theta \mid V\right)\right) = \ln \prod_{i=1}^{n} p\left(v_i \mid \theta\right) = \sum_{i=1}^{n} \ln\left(p\left(v_i \mid \theta\right)\right) \tag{3}$$

In order to maximize L_θ, we use the iterative (4).

$$\theta \longleftarrow \theta + \eta \frac{\partial \ln\left(L\left(\theta\right)\right)}{\partial\theta} \tag{4}$$

The learning rate η is related to the convergence speed of the algorithm. The larger the learning rate η, the faster the convergence. But there is no guarantee that the algorithm always has a good result. That is to say, the algorithm stability is not high. If the learning rate η selects a smaller value, the algorithm can guarantee the stability, but at the same time it leads to slower convergence speed. The algorithm will run for a long time. To solve this problem, the algorithm introduces a momentum ρ associated with the direction of the last parameter change in the algorithm to avoid premature convergence of the algorithm. The iterative formula is as follows:

$$\theta \longleftarrow \rho\theta + \eta \frac{\partial \ln\left(L\left(\theta\right)\right)}{\partial\theta} \tag{5}$$

The number of nodes on the hidden layers is entirely determined by the training effect and experience. The classic training process of DBN is in Hinton's paper [71]. We present a training process as follows:

(i) Step 1: to initialize set O of original features and set I of interaction features, we use set $V_0 = \{O, I\}$ as input of the bottom layer. Then, the DBN trains the first layer and gets the result H_0 of the hidden layer.

(ii) Step 2: the output from the previous layer is used as the input feature of the next layer $V_i = H_{i-1}$, and DBN gets the output H_i.

(iii) Step 3: do Step 2 until getting to the top layer.

(iv) Step 4: fine-tune weight matrix $W = \{w_{ij}\}$.

The fine-tuning step is key to the training process of DBN, in order to get better features from the data set. There are an unsupervised way and a supervised way in the process of fine-tuning. The Backpropagation is a supervised way [73]. The wake-sleep algorithm is an unsupervised way [74]. We use the supervised way to fine-tune, for we can calibrate the data by some blacklists in advance.

Since the entire DBN can be seen as a feature extraction process, the output of the top RBM can be seen as a feature in a space. At this point these features can be used as a common machine learning algorithm input. Although we can do the processing of the top RBM directly as an input to a classifier without any processing, it is clear that the error return can be obtained with fine-grained features under supervised conditions. Y. Tang [75] describes a case in the top classifier using Support Vector Machine (SVM). It is not difficult to speculate that other binary classifiers are also feasible. In addition, it should be noted that the practice of the top classifier found that the characteristics of the original input and DBN extracted after the characteristics of the classification will play a better classification effect. This paper chooses SVM as a binary classifier and classifies the DBN features together with the original features as SVM input.

According to H. Wang and B. Raj [76], the time complexity of deep learning model including DBN is $O(logn)$. S. Bahrampour *et al.* [77] do a comparative study of five deep learning frameworks, namely, Caffe, Neon, TensorFlow, Theano, and Torch. The experimental results show the gradient computation time of TensorFlow increases from 14ms to 23ms while batch size increases from 32 to 1024.

Table 1: Raw data statistics.

	Traffic in 40min	Traffic in 24 hours
Record Sum	882,103	9,774,545
Unique IP	13,754	842,601
Unique AD	8,533	467,343
Unique URL	36,729	1,982,005

4. Test and Analysis

4.1. Test Data and Evaluation Criterion. The test data come from ISP and are composed of two data sets. The small data set includes real traffic flow for 40 minutes. The big data set includes real traffic flow for 24 hours. After pretreatment, we get record sum, unique IP, unique AD, and unique URL as in Table 1.

This paper belongs to a classical binary classification model application. In the binary classification model, the results are usually marked as Positive (P) or Negative (N). In this paper, the corresponding node is either a phishing site or not a phishing site. Then with the classification results with a priori facts, there will be the following four categories:

(i) True Positive (TP): is actually P and the classification is also P

(ii) False Positive (FP): is actually N and the classification is also P

(iii) True Negative (TN): is actually N and the classification is also N

(iv) False Negative (FN): is actually P and the classification is also N

The above classification data can generate four categories of evaluation criterions with details as follows:

(i) Accuracy (ACC): $ACC = (TP+TN)/(TP+TN+FP+FN)$

(ii) True Positive Rate (TPR, Recall): $TPR = TP/(TP+FN)$

(iii) False Positive Rate (FPR, Fall-Out): $FPR = FP/(FP+TN)$

(iv) Positive Predictive Value (PPV, Precision): $PPV = TP/(TP+FP)$

In this paper, we use TPR as evaluation criterion.

4.2. Experimental Environment and Parameter Setup. In this paper, DBN experiments are conducted in stand-alone mode. The hardware environment includes CPU processor Intel i5-4570 quad-core, 16G memory, and the Nvidia GeForce series GTX760 graphics card. Deep learning algorithms often require high computational performance. Many popular deep learning libraries use the GPU to increase computation speed.

GPUMLib [78] is a GPU machine learning library. It may use C++ and Compute Unified Device Architecture (CUDA) and has support for Backpropagation (BP), Multiple Backpropagation (MBP), Autonomous Training System (ATS) for creating BP and MBP networks, Neural Selective Input Model (NSIM) for BP and MPB, RBM, SVM, and other computationally machine learning algorithms.

SVM model can be seen as a shallow feature extraction (with a hidden layer). DBN selects at least two layers in order to relatively enhance the feature selection effect, and too many layers will lead to overfitting. DBN main module declaration is as in Listing 1.

Some parameters are explained as follows:

(i) layers: the number of nodes per layer. Here, as the visible layer has a total of 10 different variables as a set of features, select 10 as the number of visible layer nodes.

(ii) inputs: the matrix to be trained.

(iii) initialLearningRate: learning rate.

(iv) momentum: learning rate correction momentum. Select the default value.

(v) useBinaryValuesVisibleReconstruction: whether to use the binary value to reconstruct the visible layer. Select the initial value false.

(vi) stdWeights: the upper and lower bounds of the weight matrix are initialized.

The number N of DBN layer is one of the key parameters of the DBN algorithm. In this paper, we do not specify a fixed value for N, because N is regarded as change parameter to test the DBN. We set the number of each layer to 10. The learning rate η is in $[0.01, 0.1]$ and sets as 0.1 for faster learning rate. The momentum ρ sets as the default value.

4.3. Experiment and Analysis. There are three parameters to affect the accuracy. They are the number N of DBN layer, the number T of iterations per layer, and the number of nodes in hidden layers. L. McAfee [79] shows that when the number of iterations and the number of hidden layer nodes exceed a certain threshold, the precision of the algorithm will reach a higher level. With the number of iterations or hidden layer nodes increase, the detection rate will be a small drop. The reason may be overfitting. Therefore, we first set the larger number of iterations $T = 1000$ and hidden layer nodes, such as $layers = \{top = 100, hidden = 50, \ldots, 50, visible = 10\}$.

Figure 1 shows that TPR is related to the number of layers. When the number of layers is 2, TPR gets the top level at about 89%. With the number of layers increase, TPR decreases a little. The reason is that too many layers will lead to overfitting. Therefore, the best number of layers is two layers.

Figure 2 shows that TPR is related to the number of iterations. The results show that when the number of iterations is at 200, the detection rate is above 80%. The highest detection rate achieves at about 250 iterations. After that, the accuracy of the algorithm decreases with the increase of the number of iterations. Moreover, the more iterations of each layer are, the longer the algorithm overall run time. Therefore, the best number of iterations is 250.

Figure 3 shows that TPR is related to the number of hidden units. The results show that TPR increases significantly

```
DBN(
HostArray<int> & layers,
HostMatrix<cudafloat> & inputs,
cudafloat initialLearningRate,
cudafloat momentum = DEFAULT_MOMENTUM,
bool useBinaryValuesVisibleReconstruction = false,
cudafloat stdWeights = STD_WEIGHTS
);
```

Listing 1: DBN main module declaration.

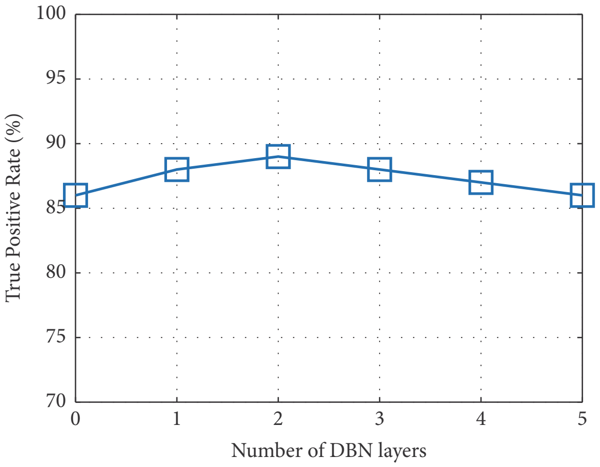


Figure 1: The relationship between the number of layers and True TPR.

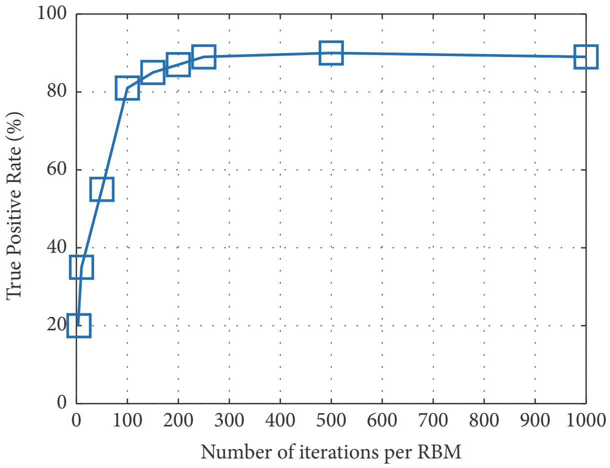


Figure 2: The relationship between number of iterations and TPR.

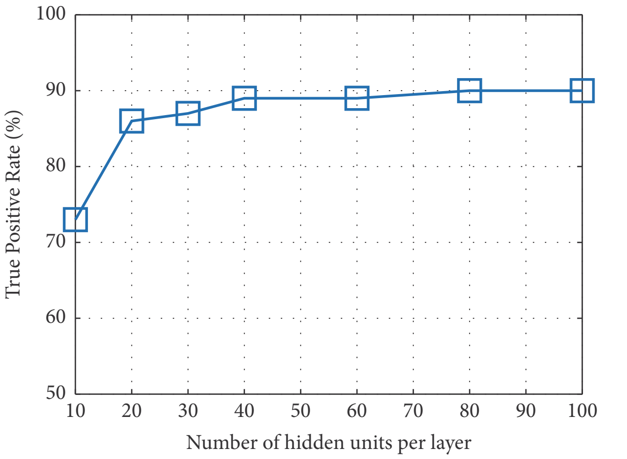


Figure 3: The relationship between number of hidden units and TPR.

Table 2: The TPR between BP and no BP.

	BP	no BP
Accuracy	89.6%	89.1%
TPR	89.2%	87.2%

to above 85%, when the number of hidden units gets 20. The detection rate does not change much under 30 hidden units. And when it gets to 40 hidden nodes, the detection rate again significantly increases and reaches nearly 90%. Since then, as the number of nodes increases, the detection rate under 80 hidden units is slightly higher than 90%. But the overall detection rate does not significantly change, after more than 40 hidden nodes. As the number of hidden layer nodes increase, the running time also significantly increases. Therefore, the number of hidden units should be 40.

Table 2 shows TPR between BP and no BP. We find that fine-tuning in BP does not improve the TPR but reduces detection rate and increases running time. The possible reason is that BP results in a degree of overfitting in the case of small input latitudes. It is also possible that the parameters of the BP algorithm are not appropriate. Therefore, we do not use BP in detection.

After training and getting the parameters in the small data set, we use DBN to detect the phishing websites in the big data set. The results show that there were 17672 nodes in phishing websites, and the detection rate was 89.2%. The FPR was 0.6%. Because the big data set cannot be fully calibrated, the results are only reference significance.

5. Conclusions

In this paper, we analyze the features of phishing websites and present two types of feature for web phishing detection:

original feature and interaction feature. Then we introduce DBN to detect phishing websites and discuss the detection model and algorithm for DBN. We train DBN and get the appropriate parameters for detection in the small data set. In the end, we use the big data set to test DBN and TPR is approximately 90%.

Data Availability

The test data used to support the findings of this study have not been made available because these data belong to the ISP (Internet Service Provider).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by National Natural Science Foundation of China (61571290, 61831007, and 61431008), the NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informationization under Grant U1509219, and Shanghai Municipal Science and Technology Project under Grants 16511102605 and 16DZ1200702 and NSF Grants 1652669 and 1539047.

References

[1] https://en.wikipedia.org/wiki/Web_service.

[2] O. Adam, Y. C. Lee, and A. Y. Zomaya, “Stochastic resource provisioning for containerized multi-tier web services in clouds,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 7, pp. 2060–2073, 2017.

[3] T. Bujlow, V. Carela-Espanol, J. Sole-Pareta, and P. Barlet-Ros, “A survey on web tracking: Mechanisms, implications, and defenses,” *Proceedings of the IEEE*, vol. 105, no. 8, pp. 1476–1510, 2017.

[4] H.-C. Huang, Z.-K. Zhang, H.-W. Cheng, and S. W. Shieh, “Web application security: Threats, countermeasures, and pitfalls,” *The Computer Journal*, vol. 50, no. 6, pp. 81–85, 2017.

[5] https://en.wikipedia.org/wiki/WeChat.

[6] K. Rekouche, *Early phishing*, 2011.

[7] http://www.antiphishing.org/.

[8] Microsoft, “20% Indians are victims of online phishing attacks: Microsoft,” *IANS*, 2014, http://news.biharprabha.com/.

[9] L. Wu, X. Du, and J. Wu, “Effective defense schemes for phishing attacks on mobile computing platforms,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6678–6691, 2016.

[10] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, “PhishNet: Predictive blacklisting to detect phishing attacks,” in *Proceedings of the 2017 IEEE Conference on Computer Communications (IEEE INFOCOM 2010)*, San Diego, USA, March 2010.

[11] S. Marchal, J. Francois, R. State, and T. Engel, “Phish storm: Detecting phishing with streaming analytics,” *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 458–471, 2014.

[12] P. Yi, T. Zhu, Q. Zhang, Y. Wu, and L. Pan, “Puppet attack: A denial of service attack in advanced metering infrastructure network,” *Journal of Network and Computer Applications*, vol. 59, no. 1, pp. 325–332, 2016.

[13] P. Yi, T. Zhu, Q. Zhang, Y. Wu, and J. Li, “A denial of service attack in advanced metering infrastructure network,” in *Proceedings of the 2014 IEEE International Conference on Communications (IEEE ICC 2014)*, pp. 1029–1034, IEEE, Sydney, Australia, June 2014.

[14] S. Xiao, W. Gong, D. Towsley, Q. Zhang, and T. Zhu, “Reliability analysis for cryptographic key management,” in *Proceedings of the IEEE International Conference on Communications (IEEE ICC 2014)*, Sydney, Austrailia, June 2014.

[15] D. Jiang, Z. Yuan, P. Zhang, L. Miao, and T. Zhu, “A traffic anomaly detection approach in communication networks for applications of multimedia medical devices,” *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 14281–14305, 2016.

[16] Z. Huang, T. Zhu, Y. Gu, and Y. Li, “Shepherd: Sharing energy for privacy preserving in hybrid AC-DC microgrids,” in *Proceedings of the Seventh ACM International Conference on Future Energy Systems (ACM e-Energy 2016)*, Canada, 2016.

[17] Y. Li and T. Zhu, “Gait-Based Wi-Fi signatures for privacy-preserving,” in *Proceedings of the 2016 ACM Symposium on InformAtion, Computer, and Communications Security (ASIACCS 2016)*, Xi’an, China, 2016.

[18] Y. Yao, Y. Li, X. Liu et al., “Aegis: An interference-negligible RF sensing shield,” in *Proceedings of the 37th Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2018)*, Honolulu, HI, Hawaii, USA, April 2018.

[19] T. Zhu, S. Xiao, P. Yi, D. Towsley, and W. Gong, “A secure energy routing mechanism for sharing renewable energy in smart microgrid,” in *Proceedings of the 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm 2011)*, Brussels, Belgium, 2011.

[20] T. Zhu and M. Yu, “A secure quality of service routing protocol for wireless Ad Hoc Networks,” in *Proceedings of the IEEE Global Communication Conference (IEEE GLOBECOM 2006)*, San Francisco, CA, USA, November 2006.

[21] T. Zhu and M. Yu, “A dynamic secure QoS routing protocol for wireless Ad Hoc networks,” in *Proceedings of the 29th IEEE Sarnoff Symposium (IEEE Sarnoff ’06)*, Princeton, NJ, USA, April 2006.

[22] P. Yi, T. Zhu, J. Ma, and Y. Wu, “An intrusion prevention mechanism in mobile ad hoc networks,” *Ad-Hoc & Sensor Wireless Networks*, vol. 17, no. 3-4, pp. 269–292, 2013.

[23] P. Yi, T. Zhu, N. Liu, Y. Wu, and J. Li, “Cross-layer detection for black hole attack in wireless network,” *Journal of Computational Information Systems*, vol. 8, no. 10, pp. 4101–4109, 2012.

[24] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, “A new intrusion detection system based on KNN classification algorithm in wireless sensor network,” *Journal of Electrical and Computer Engineering*, vol. 2014, Article ID 240217, 8 pages, 2014.

[25] P. Yi, Y. Wu, and J. Chen, “Towards an artificial immune system for detecting anomalies in wireless mesh networks,” *China Communications*, vol. 8, no. 3, pp. 107–117, 2011.

[26] P. Yi, Y. Wu, N. Liu, and Z. Wang, “Intrusion detection for wireless mesh networks using finite state Machine,” *China Communications*, vol. 7, no. 5, pp. 40–48, 2010.

[27] P. Yi, X. Jiang, and Y. Wu, “Distributed intrusion detection for mobile ad hoc networks,” *Journal of Systems Engineering and Electronics*, vol. 19, no. 3, pp. 851–859, 2008.

[28] P. Yi, T. Zhu, Q. Zhang, Y. Wu, and J. Li, “Green firewall: An energy-efficient intrusion prevention mechanism in wireless

sensor network," in *Proceedings of the 2012 IEEE Global Communications Conference (GLOBECOM 2012)*, Anaheim, California, USA, December 2012.

[29] X. D. Wang and P. Yi, "Security framework for wireless communications in smart distribution grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 809–818, 2011.

[30] C. Zhou and T. Zhu, "Highly spatial reusable MAC for wireless sensor networks," in *Proceedings of the 2007 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2007*, IEEE, China, September 2007.

[31] Z. Zhong, T. Zhu, T. He, and Z. Zhang, "Demo: Leakage-aware energy synchronization on twin-star nodes," in *ACM SenSys*, 2008.

[32] Z. Chang and Z. Ting, "Thorough analysis of MAC protocols in wireless sensor networks," in *Proceedings of the 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, IEEE, China, October 2008.

[33] C. Zhou and T. Zhu, "A spatial reusable MAC protocol for stable wireless sensor networks," in *Proceedings of the 2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008*, China, October 2008.

[34] Y. Gu, T. Zhu, and T. He, "ESC: energy synchronized communication in sustainable sensor networks," in *Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP '09)*, Princeton, NJ, USA, October 2009.

[35] Z. Zhong, T. Zhu, D. Wang, and T. He, "Tracking with unreliable node sequences," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, April 2009.

[36] T. Zhu, Z. Zhong, Y. Gu, T. He, and Z.-L. Zhang, "Leakage-aware energy synchronization for wireless sensor networks," in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'09)*, Poland, June 2009.

[37] T. Zhu, Y. Gu, T. He, and Z.-L. Zhang, "EShare: a capacitor-driven energy storage and sharing network for long-term operation," in *Proceedings of the 8th ACM International Conference on Embedded Networked Sensor Systems (SenSys '10)*, pp. 239–252, Zürich, Switzerland, November 2010.

[38] T. Zhu and D. Towsley, "E^{2}R: Energy efficient routing for multi-hop green wireless networks," in *Proceedings of the 2011 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2011*, China, April 2011.

[39] S. Guo, S. M. Kim, T. Zhu, Y. Gu, and T. He, "Correlated flooding in low-duty-cycle wireless sensor networks," in *Proceedings of the 19th IEEE International Conference on Network Protocols (ICNP '11)*, IEEE, Vancouver, BC, Canada, October 2011.

[40] T. Zhu, Y. Gu, T. He, and Z. Zhang, "Achieving long-term operation with a capacitor-driven energy storage and sharing network," *ACM Transactions on Sensor Networks*, vol. 8, no. 4, article 32, 2012.

[41] Q. Zhang, T. Zhu, Y. Ping, and Y. Gu, "Cooperative data reduction in wireless sensor network," in *Proceedings of the 2012 IEEE Global Communications Conference (GLOBECOM '12)*, IEEE, Anaheim, CA, USA, December 2012.

[42] T. Zhu, A. Mohaisen, Y. Ping, and D. Towsley, "DEOS: Dynamic energy-oriented scheduling for sustainable wireless sensor networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, Orlando, Fla, USA, March 2012.

[43] T. Zhu, Z. Zhong, T. He, and Z. Zhang, "Achieving efficient flooding by utilizing link correlation in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 21, no. 1, pp. 121–134, 2013.

[44] Y. Gu, L. He, T. Zhu, and T. He, "Achieving energy-synchronized communication in energy-harvesting wireless sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 13, no. 2, 2014.

[45] L. He, L. Kong, Y. Gu, J. Pan, and T. Zhu, "Evaluating the on-demand mobile charging in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1861–1875, 2014.

[46] S. Ren, P. Yi, D. Hong, Y. Wu, and T. Zhu, "Distributed construction of connected dominating sets optimized by minimum-weight spanning tree in wireless Ad-Hoc sensor networks," in *Proceedings of the 2014 IEEE 17th International Conference on Computational Science and Engineering (CSE)*, IEEE, Chengdu, China, December 2014.

[47] S. Ren, P. Yi, T. Zhu, Y. Wu, and J. Li, "A 3-hop message relay algorithm for connected dominating sets in wireless ad-hoc sensor networks," in *Proceedings of the 2014 IEEE/CIC International Conference on Communications in China, ICCC 2014*, pp. 829–834, China, October 2014.

[48] Z. Zhou, M. Xie, T. Zhu et al., "EEP2P: An energy-efficient and economy-efficient P2P network protocol," in *Proceedings of the 2014 International Green Computing Conference, IGCC 2014*, IEEE, Dallas, TX, USA, November 2014.

[49] L. He, P. Cheng, Y. Gu, J. Pan, T. Zhu, and C. Liu, "Mobile-to-mobile energy replenishment in mission-critical robotic sensor networks," in *Proceedings of the 33rd IEEE Conference on Computer Communications, IEEE INFOCOM 2014*, pp. 1195–1203, Canada, May 2014.

[50] J. Jun, L. Cheng, L. He, Y. Gu, and T. Zhu, "Exploiting sender-based link correlation in wireless sensor networks," in *Proceedings of the 22nd IEEE International Conference on Network Protocols, ICNP 2014*, pp. 445–455, USA, October 2014.

[51] Z. Huang, D. Corrigan, S. Narayanan, T. Zhu, E. Bentley, and M. Medley, "Distributed and dynamic spectrum management in airborne networks," in *Proceedings of the 34th Annual IEEE Military Communications Conference, MILCOM 2015*, pp. 786–791, USA, October 2015.

[52] Q. Zhang, Z. Zhou, W. Xu et al., "Fingerprint-free tracking with dynamic enhanced field division," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks, IEEE INFOCOM 2015*, pp. 2785–2793, Kowloon, Hong Kong, May 2015.

[53] F. Chai, T. Zhu, and K.-D. Kang, "A link-correlation-aware cross-layer protocol for IoT devices," in *Proceedings of the 2016 IEEE International Conference on Communications, ICC 2016*, Malaysia, May 2016.

[54] Y. Li and T. Zhu, "Using Wi-Fi signals to characterize human gait for identification and activity monitoring," in *Proceedings of the 2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pp. 238–247, Washington, DC, USA, June 2016.

[55] L. Cheng, Y. Gu, J. Niu et al., "Taming collisions for delay reduction in low-duty-cycle wireless sensor networks," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM 2016*, USA, April 2016.

[56] Z. Chi, Y. Yao, T. Xie, Z. Huang, M. Hammond, and T. Zhu, "Harmony: Exploiting coarse-grained received signal strength from IoT devices for human activity recognition," in *Proceedings*

of the 24th IEEE International Conference on Network Protocols, ICNP 2016, Singapore, November 2016.

[57] Z. Chi, Y. Li, H. Sun, Y. Yao, Z. Lu, and T. Zhu, "B2W2: N-way concurrent communication for IoT devices," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems*, pp. 245–258, Stanford, CA, USA, 2016.

[58] Z. Chi, Y. Li, Y. Yao, and T. Zhu, "PMC: Parallel multi-protocol communication to heterogeneous IoT radios within a single WiFi channel," in *Proceedings of the 25th IEEE International Conference on Network Protocols, ICNP 2017*, Canada, October 2017.

[59] Z. Chi, Z. Huang, Y. Yao, T. Xie, H. Sun, and T. Zhu, "EMF: Embedding multiple flows of information in existing traffic for concurrent communication among heterogeneous IoT devices," in *Proceedings of the 2017 IEEE Conference on Computer Communications, INFOCOM 2017*, USA, May 2017.

[60] Y. Li, Z. Chi, X. Liu, and T. Zhu, "Chiron: Concurrent high throughput communication for iot devices," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '18)*, pp. 204–216, ACM, New York, NY, USA, June 2018.

[61] P. Yi, T. Zhu, B. Jiang, B. Wang, and D. Towsley, "An energy transmission and distribution network using electric vehicles," in *Proceedings of the 2012 IEEE International Conference on Communications (ICC '12)*, Ottawa, ON, Canada, June 2012.

[62] A. Mishra, D. Irwin, P. Shenoy, J. Kurose, and T. Zhu, "Green-Charge: Managing renewableenergy in smart buildings," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1281–1293, 2013.

[63] P. Yi, T. Zhu, G. Lin et al., "Energy scheduling and allocation in electric vehicle energy distribution networks," in *Proceedings of the 2013 IEEE PES Innovative Smart Grid Technologies Conference, ISGT 2013*, USA, February 2013.

[64] T. Zhu, Z. Huang, A. Sharma et al., "Sharing renewable energy in smart microgrids," in *Proceedings of the 2013 ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS 2013*, USA, April 2013.

[65] https://www.mozilla.org/en-US/.

[66] https://www.google.com/chrome/browser/index.html.

[67] A. Y. Fu, W. Liu, and X. Deng, "Detecting phishing web pages with visual similarity assessment based on Earth Mover's Distance (EMD)," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 4, pp. 301–311, 2006.

[68] W. Chu, B. B. Zhu, F. Xue, X. Guan, and Z. Cai, "Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing URLs," in *Proceedings of the 2013 IEEE International Conference on Communications (IEEE ICC 2013)*, Budapest, Hungary, 2013.

[69] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious URLs," in *Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining (ACM KDD09)*, Paris, France, 2009.

[70] http://www.scholarpedia.org/article/Deep_belief_networks.

[71] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *The American Association for the Advancement of Science: Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[72] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[73] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[74] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal, "The "wake-sleep" algorithm for unsupervised neural networks," *Science*, vol. 268, no. 5214, pp. 1158–1161, 1995.

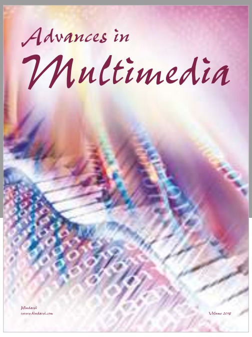
[75] Y. Tang, *Deep Learning Using Support Vector Machines*, vol. abs/1306.0239, CoRR, 2013.

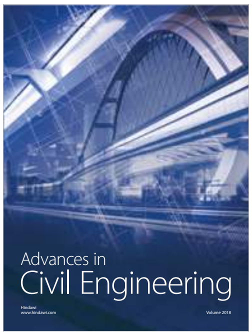
[76] H. Wang and B. Raj, *A survey: Time Travel in Deep Learning Space: An Introduction to Deep Learning Models And How Deep Learning Models Evolved from The Initial Ideas*, 2015.

[77] S. Bahrampour, N. Ramakrishnan, L. Schott, and M. Shah, *Comparative Study of Deep Learning Software Frameworks*, 2015.

[78] http://sourceforge.net/projects/gpumlib.

[79] L. McAfee, "Document classification using deep belief nets," in *CS224n*, Sprint, 2008.





Hindawi

Submit your manuscripts at
www.hindawi.com