Phishing Websites Detection Based on Hybrid Model of Deep Belief Network and Support Vector Machine

Xuqiao Yu

Department of Data Science, University of Melbourne, 3000 Melbourne, VIC Australia

E-mail: xuqiaoy@student.unimelb.edu.au

Abstract. The boosting of financial crimes that employ technical methods has become a critical issue that is urgent to be solved. However, the performance of most of the traditional classification methods are dependent on the quality of the prior knowledge of features. To address these problems, this paper proposed a hybrid model that combines the advantages of deep learning neural network of Deep Belief Network and machine learning method of Support Vector Machines. Firstly, the unidentified URLs from blacklist filtering are processed to have the URLs features extracted, the features are including statistical features, webpage code features and webpage text features. Secondly, deep features are extracted by the quick classification of deep learning model. Lastly, the resulting feature vectors combining with URL statistical features, webpage code features, webpage text features are fed into SVM model for classification. The model was tested on a dataset containing millions of phishing URLs and legitimate URLs, and have achieved the accuracy of 99.96%, the precision rate of 99.94% and the false positive rate of 51.32% which showed better performance than other comparison models.

Keywords. Phishing website detection, Hybrid model, Deep learning, Machine learning

1. Introduction
As the Internet is becoming an indispensable infrastructure that brings the era of information exploding. It is now an intermediate for people to perform various activities including searching for information, conducting business and enjoying entertainment[1]. However, one consequence that was being brought is that the Internet has also become the main platform for attackers to conduct cybercrime. One form of such crime is the phishing websites. Anti Phishing Working Group defines phishing as “A criminal mechanism employing both social engineering and technical subterfuge to steal consumers’ personal identity data and financial account credentials.”[2] The various adverse impacts that can be brought by phishing including privacy disclosure, identity theft, property loss and so on. According to statistics that provided by Kaspersky Lab, in 2017, 29.4% of users suffered at least one malicious network attack, and 199,455,606 unique URLs were identified as malicious by web antivirus components[3]. Consequently, cyber fraud and cyber security have gradually become the main issues of public concern.

With the rapid progress of Internet, phishing has become diversified and hidden. Much work had also been done regarding this concern. Blacklists and whitelists are widely used in phishing website detection. However, the list needs to be constantly updated in the accordance with the emergence of new URLs. Once it is not updated in time, the attackers could break through the limitations of the blacklist method and the detection method becomes invalid. In addition to that, machine learning methods are also widely used in phishing website detection. Nonetheless, the quality of features is critical in affecting the performance of classifiers. Deep learning is widely applied method and have
showed excellent performance in feature extraction. It is flexible with respect to the input and can extract complex features through adaptive learning. However, deep learning methods also tend to have overfitting problems in classification.

To address the above concerns, this paper proposes a hybrid model based on Deep Belief Network (DBN) and Support Vector Machine (SVM) for phishing URL detection, which combines the advantages of the deep learning and traditional machine learning methods.

In summary, the contributions in this paper are as follows:

Firstly, with the large input data, the detection efficiency and false negative rate are improved by incorporating URL filter before applying the hybrid model. Secondly, deep features extracted from DBN increase the complexity of features for training. Thirdly, the dimensionality of the data used for classification is ensured by using multidimensional feature vectors, which combines URL statistical features, webpage code features, webpage text features and features from DBN ensures. Lastly, SVM has showed great performance in phishing classification, by integrating it with DBN, the overall performance has improved comparing with base methods in terms of accuracy, precision and false negative rate.

The rest of the paper is organized as follows. In Section II, the related work on phishing website detection is presented. Then, in Section III, the proposed architecture is introduced. In Section IV, the detailed process of experiment is described. The performance of the proposed approach is evaluated in Section V. Finally, in Section VI, the paper is summarized, and the future work is discussed.

2. Related Work

The research of URL detection has gone through three main stages over the years: blacklist method, traditional machine learning method and deep learning-based method.

Prakash et al.[4] improved the method of blacklist and created PhishNet system. It analyses the properties of URLs including structure and similarity in the blacklist. However, this method is affected by the size of the original blacklist. As the blacklist becomes larger, the new rules need to be manually updated.

Garera et al.[5] studied the structure of phishing URLs and proposed features of phishing URL including lexical features and external features. Cao et al.[6] and Jeeva et al.[7] proposed that the statistical features can be extracted from URL. Li et al.[8] provided the method that uses the features extracted from web content and IP address. However, these URL features alone could not completely represent the characteristics of phishing websites.

Moghimi et al.[9] proposed the supervised machine learning methods for phishing detection based on SVM. And the experiment showed high accuracy of 0.9865. However, this method completely relies on the webpage content feature, the performance could be harmed if the attackers redesign the content.

Hung Le[10] proposed a CNN-based malicious URL detection deep neural network. And proposed Character level CNN and Word level CNN jointly optimized the network. Their approach works in an end-to-end manner, but this way is prone to overfitting if there are no large data sets to make the end-to-end approach.

Taylor et al.[11] have showed that deep learning networks can be used to learn deep features in high dimensional data in various domains such as dynamic vision.

3. Proposed Architecture

Based on the aforementioned problem discussion, the DBN-SVM model is proposed for phishing websites detection. The architecture of the model is displayed in Figure 1.
There are seven components: the input URL, URL filter, feature extraction, pre-processing, DBN feature extraction, SVM classification and the output. Before the output of the DBN, the backward propagation is used to adjust the weights.

### 3.1. URL filter

The input URL is generally filtered through the blacklist database at first in order to maximise the classification time efficiency in the overall model. This step quickly compares the hash value of URL with hash value in blacklist, which are obtained through crawling from PhishTank website. And the matched URLs are outputted as malicious URLs directly, while the unmatched URLs are fed into the next step for further classification.

### 3.2. Feature extraction

After the filtering, the remaining unidentified URLs are collected and have the feature extracted. The features considered here include statistical features, webpage code features as in Table 1.

**Table 1.** Webpage code features and URL statistical features.

| Webpage code features | URL statistical features                                      |
|-----------------------|--------------------------------------------------------------|
| html_len              | Number of URL path                                            |
| div                   | Number of dots                                                |
| embed                 | URL is encoded as octal or hexadecimal                        |
| applet                | Length of URL                                                 |
| open                  | IP address                                                    |
| pop                   | Information entropy                                           |
| exec                  | Euclidean distance                                            |
| eval                  | Number of sensitive words                                     |
| externalLinks         | Number of key words                                           |
| attachment            | Host name position                                            |
| dispatchment          | Length of the longest word in the host name                   |
| script                | Containing “@”                                                |
| onload                | URL contains special character                                 |
| timeout               | Domain names with pure number                                 |
| input                 | Number of top-level domains in paths                          |
| exec                  | Number of domain name servers                                 |

And webpage content features extracted through natural language processing techniques as showed in Figure 2.
After applying TFIDF, logistic regression is used to train text vectors and so that to the probability of text belonging to a phishing website can be generated. And then the probability is used to represent the webpage text features.

The features are combined and pre-processed into unified vectors through normalization. The processed vectors are then be used as input to DBN model for further training.

3.3. Deep Belief Neural Network

As it is defined by Hinton[11], Deep belief network is a probabilistic generative model composing of multiple layers of stochastic and latent variables. Latent variables contain binary values and are referred as hidden units. The structure of DBN as shown in Figure 3, has an undirected, symmetrical connection between the top two layers, forming an associative memory. In addition, each layer of hidden units in DBN represents more abstract features in the input data. More specifically, these layers are the stacks of trained restricted Boltzmann machines (RBMs).

RBMs are unsupervised networks with the hidden layer of each sub-network serves as the visible layer for the next network. The structure of RBM is showed in Figure 4.
In detail, the visible units are represented by V and the hidden units are represented by H. The weights between the visible and hidden units is represented by W. The probability of the binary vectors V of the observed data units is defined by the weight of the connections and biases of the visible and hidden units using an energy function[12]. The energy function is given by:

\[ E(v,h;\theta) = -\sum_{i=1}^{V} \sum_{j=1}^{H} w_{ij} v_i h_j - \sum_{i=1}^{V} b_i v_i - \sum_{j=1}^{H} a_j h_j \]  

(1)

where \( \theta = \{ w, b, a \} \) is the model parameter, \( w_{ij} \) is the weight associated between the connections, and \( a_i \) and \( b_i \) are the biases of the hidden and visible units, respectively. The dimensions of the visible units and hidden units are represented by \( V \) and \( H \), respectively[12]. The probability distribution that the network sets to every visible-hidden vector with an energy function can be defined by

\[ p(v,h) = z^{-1} e^{-E(v,h)} \]  

(2)

where \( z \) is the normalization constant, which is obtained by adding all visible and hidden vectors[12].

To a visible vector \( v \), the model assigns a probability

\[ p(v) = z^{-1} \sum_h e^{-E(v,h)} \]  

(3)

The partial derivative of \( \log(p(v)) \) with reference to \( w \) is computed as

\[ \frac{\partial \log(p(v))}{\partial w_{ij}} = \langle v_i h_j \rangle - \langle v_i \rangle \langle h_j \rangle \]  

(4)

The expectations under the distribution described by the respective subscripts is denoted by the angle brackets. The upgrade rule for \( w \) is obtained as follows:

\[ \Delta w_{ij} = \varepsilon ( \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} ) \]  

(5)

where \( \varepsilon \) is the learning rate. For a given training image \( v \), the conditional probability distribution of hidden units is obtained by

\[ p(h_j = 1|v) = \sigma ( \sum_i v_i w_{ij} + a_j ) \]  

(6)

where \( \sigma \) is the sigmoid function \( \sigma(x) = (1+\exp(-x))^{-1} \). For a given hidden vector \( h \), an unbiased state of visible data unit is obtained by

\[ p(v_j = 1|h) = \sigma ( b_j + \sum_i h_i w_{ij} ) \]  

(7)

In RBM training, the visible vector and the binary label of the class label are connected to obtain the joint probability distribution of the visible data and the class label. Then the energy function is

\[ E(v,h;\theta) = -\sum_i \sum_j w_{ij} v_i h_j - \sum_i \sum_j w_{ij} h_i v_j - \sum_j a_j h_j - \sum_i c_i v_i - \sum_j b_j h_j \]  

(8)

\[ p(l_j = 1|h;\theta) = \operatorname{softmax}( \sum_h h_j w_{ij} + C_j ) \]  

(9)

Subsequently, \( p(1|v) \) is calculated by

\[ p(l_j|v) = \sum_h e^{-E(v,l,h)} \]  

(10)

Due to the characteristics of DBN structure, the network is initialized by training stacked RBMs so that the model parameter \( \theta \) can explain \( p(h|\theta) \) and \( p(v|h,\theta) \). Consequently, \( p(v) \) can be computed using the following equation:

\[ p(v) = \sum_h p(h|\theta) p(v|h,\theta) \]  

(11)

It is to note that, \( p(v|h,\theta) \) remains consistent when the value of \( \theta \) is obtained. While \( p(h|\theta) \) can be substituted by a superior model, which is determined by the hidden vectors that created by the training data. And then they become training vectors that are required by the next layer of RBMs.
To make the training process faster[13,14], Contrastive Divergence (CD) algorithm is used to train each layer of RBM. So, the upgraded rule used here is

$$\Delta w_{ij} = \epsilon \left( \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{recon}} \right)$$  \hspace{1cm} (12)

In this method, computing $\langle v_i h_j \rangle_{\text{recon}}$, the visible units are initialised through a training vector. Then, eqn. (6), is used to calculate hidden units. Subsequently, eqn. (7), is used to recalculate the visible unit vi. These units are referred to as ‘recon’ in eqn. (12). Finally, the hidden unit states are calculated based on the reconstructed visible state.

After the RBM pre-training process is completed, the weight of the neural network is set using the bottom layers of weights of the subsequent DBN. And the neural network is adjusted to an adaptive mode through the back propagations.

By encoding statistical dependencies in each layer among the units in the layer below it, more complex features could be detected.

3.4. SVM classification
The weighted feature vectors are then gained from DBN network are then used as input to SVM classifier. The algorithm of SVM is to find the optimal separating hyperplane between two classes by maximizing the margin between the classes closest points through supervised training. As the structure showed in Figure 5.

![Support Vector Machine](image)

Figure 5. Support Vector Machine

A discriminating hyperplane will satisfy:

$$\omega' x_i + \omega_0 \geq 0 \hspace{0.5cm} \text{if} \hspace{0.3cm} t_i = +1;$$

$$\omega' x_i + \omega_0 < 0 \hspace{0.5cm} \text{if} \hspace{0.3cm} t_i = -1$$  \hspace{1cm} (13)

Now the distance of any point $x$ to a hyperplane is $|w'x + w_0|/||w||$ and the distance to the origin is $|w_0|/||w||$.

4. Experiments
This section starts with describing the necessary details of the URL dataset, followed by the experimental setup, the comparison methods and finally the evaluation metrics.

4.1. Description of the dataset
The labelled URL dataset in this paper includes benign URLs and malicious URLs, a total of 1089012 URLs were crawled. First, historical data confirmed as phishing websites were collected from the PhishTank, and a total of 864753 URLs were used as positive samples of phishing. Then, benign 224259 URLs were collected from DMOZ[15], which is known as a comprehensive volunteer-edited directory of the Web.

4.2. Experimental setup
In this experiment, python is used as software framework here. By constantly adjusting and optimizing the parameters in the experiments, the most effective hyperparameters are set as follows:

- Max epoch for RBM set to 140
- Learning rate for RBM set to 0.5
- Max epoch for backpropagation set to 100
- Learning rate for backpropagation to 0.05
- Set RBF as kernel function for SVM

4.3. Comparison with baseline methods
Several baseline methods were set, including feature based LR model, SVM model and CNN model.
1) LR model
Firstly, TF-IDF is used to extract features, and then use the classic model Logistic Regression in machine learning to do the classification.
2) SVM model
Same five-folds cross validation and RBF kernel is used in the SVM model based on the extracted combination of features.
3) CNN model
As a comparison model, CNN model adopts world-level embedding layer and CNN layer.

4.4. Experiment Design
The experiment is performed with 5-fold cross-validation. Four sets are used as training sets and one is used as test set. This process uses each fold for validation once and is repeated five times. Finally, all the metrics on validation folds are average.

As for the evaluation metrics, accuracy, precision and false negative rate are used to evaluate the performance of suspicious URL detection. The formula of the metrics can be defined as Equations (14)-(16):

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}
\]

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{15}
\]

\[
\text{False negative rate} = \frac{FP}{TP + FP} \tag{16}
\]

Where TP is true positives, TN is true negatives, FN is false negatives and FP is false positives respectively. The malicious URLs are labelled as +1 and the benign URLs are labeled as -1.

5. Experimental Results and Analysis
In this section, the evaluation results of each of the model are presented and analyzed.

The comparisons of accuracy, precision and false negative rate of the methods are displayed in Figure 6. For the accuracy, DBN-SVM model was 4% higher than the feature-engineer model with Logistic Regression. As for the precision and false negative rate, DBN-SVM model also showed better result. The possible reason could be that, feature based model with Logistic Regression greatly relies on the performance of natural language processing tools, the accumulated errors could affect the effectiveness largely. Furthermore, external semantics such as word frequency have limited impacts on the classification, while neural networks can encode semantic information into high-dimensional hidden feature space and extract more features.
6. Conclusion

In this paper, by addressing the drawbacks of current methods, a hybrid DBN-SVM model is proposed for phishing detection. This method assembles blacklist filter, multidimensional features and deep learning feature extraction with traditional machine learning classification. The blacklist filtering improves the efficiency of traditional heuristically classification methods, and DBN improves the problems of insufficient feature selection, finally SVM classifier combined with the multidimensional features significantly improves the performance with respect to accuracy, precision and false positive rate. And the results of the experiments demonstrate that the hybrid model outperforms traditional machine learning methods such as feature based LR model, SVM and neural networks such as CNN. The future development of this approach will be incorporating features of webpage images. In addition to that, some structure modifications will be made in specific to malicious classes and would try to conduct multiple classifications according to categories of malicious URLs.

References

[1] K.-K.R. Choo, H. Kaur, X. Tao, "A Conceptual Interdisciplinary Plug- and-Play Cyber Security Framework" 2014 in ICTs and the Millennium Development Goals, Springer, pp. 81-99.
[2] Anti-Phishing Working Group(APWG). Phishing activity trends report-second quarter 2016[EB/OL]. https://docs.apwg.org/reports/apwgreportq22016.pdf.

[3] Kaspersky Security Bulletin: Overall Statisticals For. 2017. Available: https://securelist.com/ksb-overall-statistics-2017/83453/.

[4] P. Prakash, M. Kumar, R. Kompella, and M. Gupta, 2010 "PhishNet: Predictive blacklisting to detect phishing attacks", Proc. IEEE INFOCOM, pp. 1-5.

[5] S. Garera, N. Provos, M. Chew, and A. D. Rubin. 2007 A Framework for Detection and Measurement of Phishing Attacks. In Proc. of WORM.

[6] J. Cao, D. Dong, B. Mao, and T. Wang, 2013 "Phishing detection method based on URL features," J. Southeast Univ.-Engl. Ed., vol. 29, no. 2, pp. 134–138, Jun.

[7] S. C. Jeeva and E. B. Rajsingh, 2017 "Phishing URL detection-based feature selection to classifiers," Int. J. Electron. Secur. Digit. Forensics, vol. 9, no. 2, pp. 116–131, Jan.

[8] A. Le, A. Markopoulou, and M. Faloutsos, "PhishDef: URLnamessayit all," 2010 in Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM), pp. 191–195.

[9] MOGHIMI M, VARIJANI A Y. New rule-based phishing detection method[J]. Expert Systems with Applications, 2016, 53: 231-242.

[10] Le, Hung, et al. 2018 "URLNet: learning a URL representation with deep learning for malicious URL detection." arXiv preprint arXiv:1802.03162.

[11] Hinton, G. E., & Salakhutdinov, R. 2006. Reducing the dimensionality of data with neural networks. Science, 313, 504–507.

[12] Bibin, D., Nair, M.S. and Punitha, P., 2017. Malaria parasite detection from peripheral blood smear images using deep belief networks. IEEE Access, 5, pp.9099-9108.

[13] M. Welling, M. Rosen-Zvi, and G. E. Hinton, 2005 “Exponential family of harmoniums with an application to information retrieval,” in Proc. Adv. Neural Inf. Process. Syst., pp. 1481–1488.

[14] A. Mohamed, G. Dahl, and G. E. Hinton, 2009 “Deepbeliefsnetworksforphone recognition,” in Proc. NIPS Workshop Deep Learn. Speech Recognit. Related Appl, pp. 1–9.

[15] DMOZ. [online] Available:https://dmoztools.net/.

[16] Dhanya Bibin, Madhu S. Nair, P. Punitha. "Malaria Parasite Detection From Peripheral Blood Smear Images Using Deep Belief Networks", IEEE Access, 2017