Efficient Detection of Phishing Attacks with Hybrid Neural Networks

Xiaoqing Zhang[1], Dongge Shi[1], Hongpo Zhang[1], Wei Liu[2], Runzhi Li[1]
[1] Collaborative Innovation Center of Internet Healthcare, Zhengzhou University, Zhengzhou 10459, China
[2] Software and Applied Science and Technology Institute of Zhengzhou University
e-mail: xqzhang@ha.edu.cn, dgshi@ha.edu.cn, zhp@zzu.edu.cn, wliu@ha.edu.cn, rzli@ha.edu.cn

***Abstract*—Many machine learning techniques and social engineering methods have been adopted and devised to combat phishing threats. In this paper, a novel hybrid deep learning model is proposed to identify phishing attacks. It incorporates two components: an autoencoder (AE) and a convolutional neural network (CNN). The AE is adopted to reconstruct features that enhances correlation relationship among the features explicitly. The results from the experiments show that the model is able to detect phishing attacks with a mean accuracy over 97.68%, yet it has high generalization ability and can detect phishing attacks in the receivable time scale.**

Keywords-phishing attack; convolutional neural network; autoencoder; reconstruct

I. INTRODUCTION

Internet has become an indispensable part in our daily life, where Internet users can exchange their private information like username, password, and bank account etc. Internet users are exposed to various types of web-threats. Phishing is considered as one type of the web-threats that is loosely defined as the act of persuading users into letting out essential personal information [1]. Phishing attacks often involve social engineering methods and technical tricks. It can be sent in different ways, such as message, e-mail and so on. Phishing attacks behavior are predicted to be more changeable and easy in the future. Accordingly, a promising method need be dynamically adapt to evolution in phishing attacks and detect phishing websites at good time scale.

With the development of deep learning, plenty of variable deep neural network models are applied different fields and get good results.

In this work, a hybrid deep neural network model is proposed to classify phishing websites that can detect potential phishing attacks. It is accommodative and dynamic with regards to new coming phishing attacks. The main contribution of this work can be summarized as follows:

Firstly, we propose a hybrid deep neural network model to identify phishing attacks, which consists of two parts: an autoencoder (AE) and a convolutional neural network. The classic convolution filters only get local feature combinations rather than global feature combinations, AE is adopted to construct features that enhances correlation relationship in all features.

It always play a very important role for the operation of weight parameters initialization in deep learning and can improve the performance of the algorithm. In this work, we apply two different weight initializer functions in the stage of AE and CNN respectively. The results from the experiments show that there is an increase of 0.5% for accuracy.

The remainder of this paper is organized as follows. Section Ⅱ introduces related work. The detailed description of datasets is given in section Ⅲ. The hybrid deep neural networks model is presented in section Ⅳ. Section Ⅴ provides our experiments and results. The conclusion and the future works are described in section Ⅵ.

II. RELATED WORK

Phishing is a serious security issue. It has a tremendous effect on the budgetary and web retailing segments. In phishing research literatures, some machine learning (ML) and deep learning techniques are adopted as the binary classifiers to classify websites into either legitimate or fake websites [2]. Rami M. Mohammad et al. [3] proposed a deep learning model to predict phishing websites. They mainly used a modified back-propagation neural network and achieved an accuracy with 92.5%. Sonowal et al. [4] developed a multi-filter approach to detect phishing attacks. They highlight several tricks: upgrade whitelist automatically, URL features selection, lexical signature, string matching and comparison of accessibility score. The experiments showed the multi-filter approach has an accuracy of 92.72%.

Abutair et al [5] used a Case-Based Reasoning methodology to detect phishing websites. The proposed methodology was highly adaptive and dynamic, because it combined offline experiences and online experiences. The result proved that it enhance the classification accuracy with a small set.

Justin Ma et al [6] proposed a Large-Scale Online Learning application to identify suspicious URLs. They gathered URL features that involved lexical and host-based features of the correlated URLs. The experiments showed that online algorithms achieved excellent phishing detection result performance. Researchers [7] introduced an efficient method to detect phishing sites by computing the similarities among web pages through mining the websites, make comparisons by matching the HTML source codes and also computing the cosine similarity of their textual contents. The result was obtained by this method showed that the true positive rate was comparatively high as it utilizes Google page ranking information and detection rate was as well as high compared with other existing mechanisms.

Phishing attacks are expected to be more mutable and low-cost in the future. It is more challenging for both

978-1-5386-7635-6/18/$31.00 ©2018 IEEE

conventional methods and deep learning methods to detect phishing attacks timely. Deep learning have good self-learning ability to generate unseen features and is heavily supported by industry.

III. Dataset and Feature selection

A. Dataset

In this work, we collect the phishing URLs from phishtank [8] which is publicly available. Legitimate URLs were gathered from the Open Directory Project (DMOZ) [9], which is the largest, most versatile human-edited directory of the Websites and Phishload [10], which is a public dataset. The dataset from UCI Machine Learning Repository are also used [11]. Overall, the total data set contains 7943 records. In this experiments, we adopt the 10-fold cross- validation method.

B. Feature Selection

Feature selection is a very challenging task. Good representation of features that can help to identify the phishing from legitimate ones. In this study, 30 features from website URLs, source codes and textual contents are collected [11]. In the preprocessing, they are transformed to either a binary value or a ternary value. Here binary value features represent either "Phishing" or "Legitimate". Yet one more value has been added (this is "Suspicious") in ternary value features. All features are mainly categorized as in Table I.

IV. Hybrid Neural Networks

A. Autoencoder

AE is one of deep learning models that mainly is used for unsupervised representation learning. The task of an AE is to reconstruct an approximated representation for a set of data. Many variants are proposed to enhance the performance of an AE. In this work, the architecture of the AE is shown in Figure 1. It includes three layers which are an input layer, a hidden layer and an output layer. The input and output layers have the same number of neurons 30, whereas the hidden layer has a special number of neurons 20. The encoding and decoding processes are denoted as follows:

$$\mathrm{Y} = \mathrm{f}\left(W_y X + b_y\right) \tag{1}$$

$$\mathrm{Z} = \mathrm{f}(W_z Y + b_z) \tag{2}$$

where Y and Z represent encoder and decoder respectively, W_y and W_z represent weight function of encoder and decoder respectively, as well as b_y and b_z are the biases of hidden layer and output layer. We can apply uniform distribution function to both W_y and W_z, that is, $\mathrm{W} = W_y = W_z$. Leaky rectifier linear unit (LeakyRelu) function is used as the activation function for the AE.

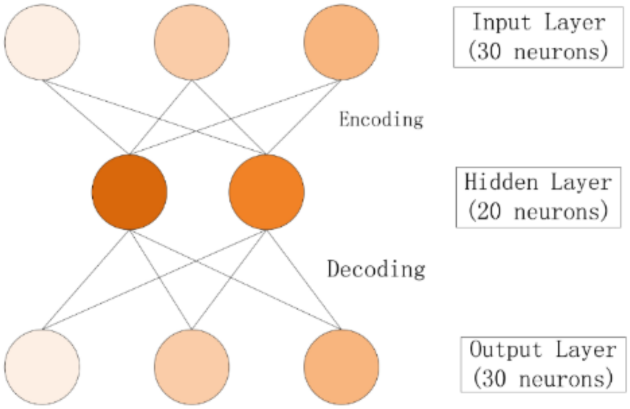

Figure 1. The architecture of the autoencoder.

TABLE I. All Available Features

Aspects	***Features***
Address Bar based Features	IP address ; Long URL; TinyURL; URL's having "@" Symbol; Redirecting using "//"; Adding Prefixes and Suffixes to URL; Sub Domain; HTTPS; Domain Registration Length; Favicon; Non-Standard Port; The Existence of "HTTPS" Token in the Domain Part of the URL
Abnormal Based Features	Request URL;URL of Anchor; Links; SFH; Submitting Information to Email; Abnormal URL
HTML and JavaScript based Features	Redirect Page; Status Bar Customization; Disabling Right Click; Using Pop-up Window; Iframe Redirection
Domain based Features	Age of Domain; DNS Record; Website Traffic; PageRank; Google Index; Number of Links Pointing to Page; Statistical-Reports Based Feature;

B. Convolutional Neural Networks

In the past years, convolutional neural networks (CNNs) have been successfully applied in many fields, such as image classification, speech and natural language processing, scene labeling and object tracking, and so on. Convolutional neural networks incorporate cascaded convolutional layers and pooling layers. Convolution layers take inner product of the convolution filter and the underlying receptive field followed by a nonlinear activation function at every local connectivity of the input. In this work, we take one dimensional convolution operation. The equation for one dimensional convolution is shown in Equation 3,

$$C_{1d} = \sum_{a=-\infty}^{\infty} x(a)\omega(t-a) \tag{3}$$

where x is the input, ω is the the filter and a is the sliding window size. Pooling layer is a vital component of CNN. It reduces the computational cost by cutting connections between convolution layers.

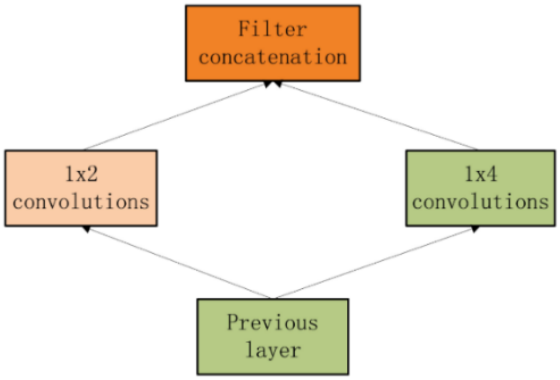

Figure 2. The architecture of modified Inception module.

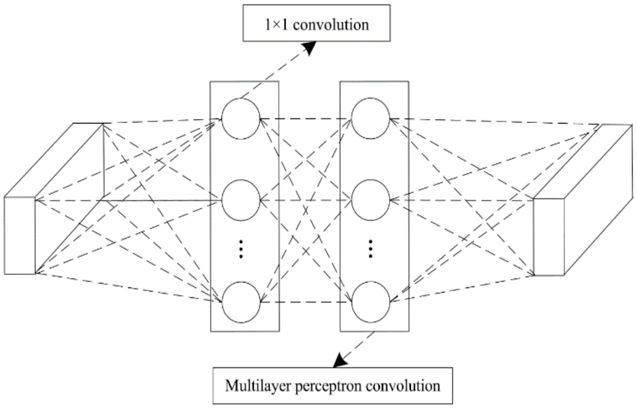

Figure 3. The architecture of Mlpconv layer

We get the inspiration from GoogleNet [12] and Network in Network [13]. We adopt a modified Inception module [13] and combine it with and the multilayer perceptron convolution (mlpconv, 1 × 1 convolution) in our convolutional neural network. Two convolution modules (Inception module and mlpconv layers) are given in Figure 2 and Figure 3 respectively. The advantages are listed as following:

- The mlpconv layers are used to enhance the abstraction ability of the model. Meanwhile they can reduce computation cost by cutting down number of the training convolution parameters.
- The Inception module contains heterogeneous size of filters that contributes to get better feature combinations and state-of-the-art model.

C. *Hybrid Neural Networks Architecture*

We propose a hybrid deep neural network to classify and predict the phishing attack. It comprises AE and the CNN modules. The high-level architecture of the model is shown in Figure 4. There are 5 convolutional layers in our CNN as you can see Figure 4: three normal convolution layers (one standard convolution layers and two modified inception modules) and two mlpconv layers. First normal convolution layer convolve with kernel size of 1x3 and have 4 filters. Last normal two convolution layers (two modified inception modules) convolved with the same kernel size of (1x2, 1x4) and have 8 and 10 filters respectively. Two mlpconv layers both have 10 filters and kernel size of 1x1, then following by a fully connected (dense) layer. Two pooling operations are used in the CNN: a max-pooling operation and an average pooling. Max-pooling operation and average pooling operation are used to reduce the size of the feature map. We adopt softmax function as classifier and leaky rectifier linear unit (LeakyRelu) [14]activation function for both the AE and the CNN.

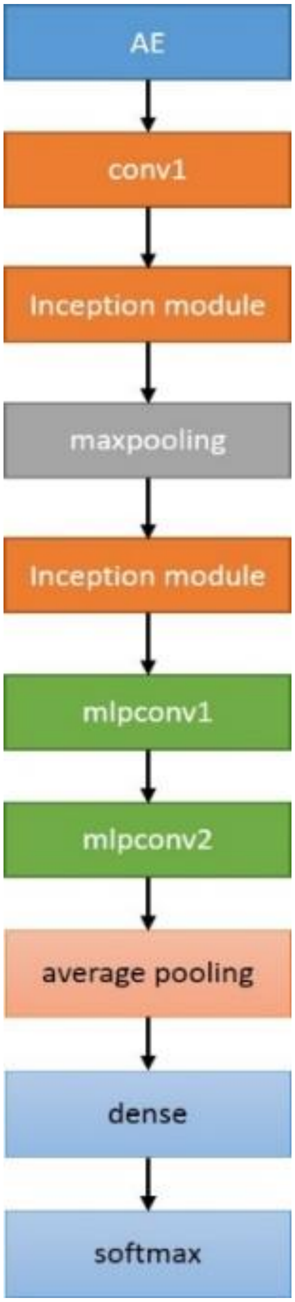

Figure 4. The architecture of the hybrid neural network.

We adopt back propagation [15] method to train the hybrid neural network model. The hyper parameters include: the regularization (λ), learning rate. The loss function (L) incorporates two parts: feature self-reconstruction error loss (L_1) and phishing detection error loss (L_2) . They are computed as follows:

$$\mathrm{L} = \alpha L_1 + (1 - \alpha) L_2 \qquad (4)$$

$\alpha \in [0,1]$ is a hyper-parameter that determines importance between L_1 and L_2. Here the cross-entropy loss function is adopted, which is defined as follows:

$$\text{Loss} = \frac{1}{n}\sum_{i=1}^{n} -(y_i . log y_{pre,i} + (1 - y_i) . \log(1 - y_{pred,i})) \qquad (5)$$

There n represents the size of mini-batch, y_i and $y_{pre,i}$ represent real labels and predictive labels respectively.

D. Optimization

In this work, we take several strategies to improve the performance of our proposed hybrid deep neural network named ACNN. They are presented as following.

- Because the AE and the CNN are two different deep learning models, and weight initializer functions play an important role in model performance, the uniform distribution function and a modified Gaussian function are adopted for the AE and the CNN respectively. The experiments show that it actually improves accuracy for about 0.5%, which is considered to be great.
- In order to decrease the shortcoming of internal covariate shift, which is loosely defined as the change in the distribution of model activations due to the change in model parameters during training. Before each layer we apply Batch Normalization (BN) [16] and a LeakyRelu activation function. BN allows us to set higher learning rates as well as be less careful about parameter initializations.
- Dropout [17] technique is adopted to prevent overfitting and we also apply dropout layer before softmax layer. We use the Adam [18]optimizer as an optimizer and reduce the learning rate by a factor of 5.
- For training, we adopt the training tricks used by Krizhevsky et al [19]. That is to say, we set appropriate initializations for learning rate and the weights manually. We experimented with batch sizes ranging from 20 to 100, and our default batch size is 64. The training process starts from maximum learning rate, it continues until the accuracy stops improving, and then the learning rate is lowered by a scale of 2. This procedure is repeated as far as the final learning rate reaches the minimum learning rate we set. Furthermore, every learning rate is trained for 10 times.

TABLE II. COMPARISON OF DIFFERENT MODELS ON PHISHING DATA (%)

	Accuracy	*Precision*	*Recall*
M1	97.87	98.69	97.20
M2	93.35	92.25	96.14
M3	89.63	91.53	89.67
M4	91.23	90.1	93.74

Table Ⅱ. The best result in each column is highlighted (the larger value, the better). It is obvious that M1 has the best results out of the four models compared.

V. EXPERIMENTS

Phishing website detection is considered as a binary classification problem, in which two possible outputs can be drawn for a certain website: phishing or legitimate. In this section, we compare the proposed hybrid deep neural networks (M1) with three traditional classification algorithms, which include SVM (M2), decision tree (M3) and LinearSVC (M4). We implement the experiments in Tensorflow platform which is an open source deep learning framework and scikit-learn, a machine learning library. There are common metrics such as precision, recall and accuracy are adopted in this experiments.

We assess performance of the different models on the datasets. In Table, the results indicate that M1 achieves the better performance than other models among metrics of accuracy, recall and precision in our dataset. To understand the hybrid deep neural networks model perform better, the confusion matrix of the model is shown in Figure 5. Here, 1 represents legitimate websites and 0 represents phishing websites.

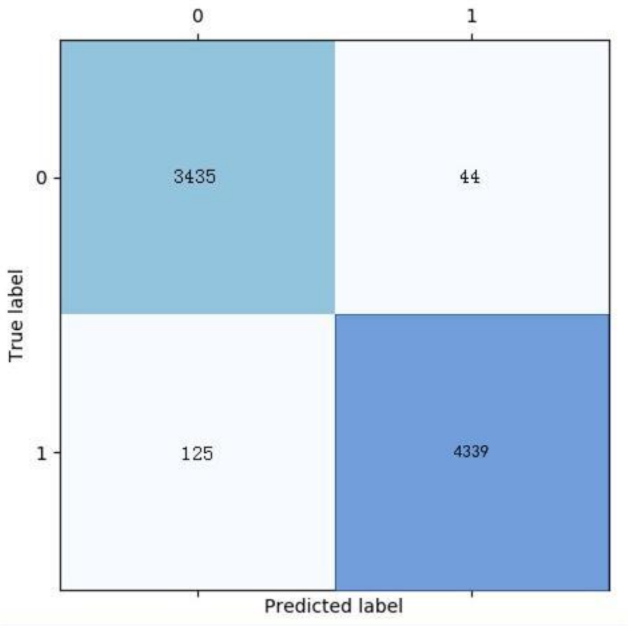

Figure 5. The confusion matrix of the hybrid neural networks model, 1 represents legitimate websites and 0 represents phishing websites.

A. Analysis

According to the absolving results, they show that we adopt an AE to cover the shortage of classic convolution filter, which helps to build correlation among the whole features. Meanwhile we adopt some suitable strategies to optimize the hybrid deep neural network model.

VI. Conclusion and Future Work

Detecting Phishing attacks will be a remarkable challenge in the future. Because malicious features evolve continually and unknown features are introduced daily. We argue that a comprehensive anti-phishing tool to detect phishing sites at a receivable time scale that increases the proportion of predicting the websites. Hence, a hybrid deep neural network model is proposed for predicting phishing websites. The model not only can detect phishing websites in a receivable time, but also has a good self-learning ability that can explore the new features combination without professional feature extraction. In the experiment, the best phishing websites classification accuracy of the hybrid neural networks reaches up to 99%. Comparing to other models, the model can increase the average accuracy of 4% at least.

Acknowledgment

The project is supported by the Program of Scientific and Technological Research of Henan Province, China, under Contract 152102210249, and the program of Integration of Cloud Computing and Big Data, Innovation of Science and Education, Contract FII 2017A11017. The project is also funded by National Natural Science Foundation of China (61602422), Natural Science Foundation of Henan Province (152300410047).

References

[1] The Anti-phishing Working Group. http://www.Antiphishing.org.

[2] Dunham, K. (2008). Mobile Malware Attacks and Defense. Syngress Publishing.

[3] Mohammad, R. M., Thabtah, F., & Mccluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. Neural Computing & Applications, 25(2), 443-458.

[4] Sonowal, G., & Kuppusamy, K. S. (2017). Phidma - a phishing detection model with multi-filter approach. Journal of King Saud University - Computer and Information Sciences.

[5] Abutair, H. Y. A., & Belghith, A. (2017). Using case-based reasoning for phishing detection. Procedia Computer Science, 109, 281-288.

[6] Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009). Identifying suspicious URLs: an application of large-scale online learning. International Conference on Machine Learning (pp.681-688). ACM.

[7] Roopak, S., & Thomas, T. (2014). A Novel Phishing Page Detection Mechanism Using HTML Source Code Comparison and Cosine Similarity. Fourth International Conference on Advances in Computing and Communications (pp.167-170). IEEE.

[8] PhishTank (2016) Anti-phishing community. URL: https://www.phishtank.com/ accessed on June 1-30.

[9] Open Directory Project, community (2016) URL: https://www.phishtank.com/ accessed on July 2-30.

[10] Phishload (2016) Legitimate url dataset. URL: http://www.medien.ifi.lmu.de/team/max.maurer/files/phishload/accessed.

[11] Rami Mustafa A Mohammad. Lee McCluskey (2015) Fadi Thabtah. UCI repository of machine learning databases.

[12] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., & Anguelov, D., et al. (2014). Going deeper with convolutions. 1-9.

[13] Lin, M., Chen, Q., & Yan, S. (2013). Network in network. Computer Science.

[14] A. L. Maas, A. Y. Hannun, A. Y. Ng. (2013). Rectifier nonlinearities improve neural network acoustic models, in: ICML.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner (1998) Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324.

[16] Sergey Ioffe, & Christian Szegedy. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. 448-456.

[17] Srivastava, Nitish, Hinton, Geoffrey E, Krizhevsky, Alex, Sutskever, Ilya, et al .(2014). Dropout: a simple way to prevent neural networks from overfit-ting. Journal of Machine Learning Research, 15(1): 1929–1958.

[18] Kingma, D. P., & Ba, J. (2014). Adam: a method for stochastic optimization. Computer Science.

[19] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Communications of the Acm, 60(2), 2012.