Phishing Web Page Detection with Semi-Supervised Deep Anomaly Detection

Linshu Ouyang[1,2(✉)] and Yongzheng Zhang[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{ouyanglinshu,zhangyongzheng}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. Phishing web page is one of the most serious threats to the users of the Internet. Recently, deep learning-based phishing detection methods have achieved significant improvement. However, these supervised deep neural networks require a large number of training samples. They also have difficulties in detecting novel phishing web pages. Using anomaly detection approaches is a possible way out yet is currently less explored, possibly due to two reasons. First, HTML codes lie in high dimensional discrete space which is difficult to handle for existing anomaly detection methods. Second, existing anomaly detection methods may find other types of anomalies that are beyond the scope of phishing.

In this paper, we propose a novel semi-supervised deep anomaly detection-based phishing webpage detection method. We first utilize a multi-head self-attention network to learn feature representation that is suitable for anomaly detection from HTML codes. Then we build a semi-supervised learner with Gaussian prior and contrastive loss to fulfill an end-to-end anomaly detector that is specifically optimized for detecting phishing webpages. Extensive experiments on a real-world dataset demonstrate that the accuracy of our method outperforms other state-of-the-art methods by a large margin.

Keywords: Phishing · Semi-supervised learning · Anomaly detection

1 Introduction

Phishing is a kind of social engineering attack that tricks the victims to perform specific actions by imitating trusted web pages [17]. It's one of the most prevalent attacks due to its effectiveness and low cost [1].

Many phishing detection methods have been proposed in the last decades. Traditional detection methods typically focus on designing informative features and utilize sophisticated classifiers such as SVM [5] or Random Forest [4]. In recent years, several deep learning-based phishing detection methods achieved

J. Garcia-Alfaro et al. (Eds.): SecureComm 2021, LNICST 399, pp. 384–393, 2021.
https://doi.org/10.1007/978-3-030-90022-9_20

significant performance improvement [6,10,16]. These methods typically use URL or HTML as input and utilize CNN or RNN to perform classification.

However, these supervised deep learning-based methods require plenty of training samples to learn accurate decision boundaries. This limits the model's ability to detect unknown phishing webpages. On the contrary, anomaly detection approaches have stronger abilities to detect novel phishing webpages, but applying anomaly detection to phishing webpage detection faces several challenges. Traditional anomaly detection methods require manually designed features, which is less effective compared to deep learning-based methods. Also, there exist several types of anomalies in web pages. Existing anomaly detection methods may detect other types of anomalies beyond phishing webpages.

In this paper, we propose a novel semi-supervised deep anomaly detection-based phishing detection method to address these problems. With HTML codes as inputs, we first employ a multi-head self-attention network [7] to perform feature learning and output the anomaly scores. Then we use a prior distribution of anomaly scores to guide the model to learn a representation of normality. Finally, we utilize a novel contrastive loss function to fulfill end-to-end semi-supervised training to finetune the anomaly scoring network towards phishing webpage detection.

Our method combined the strength of deep learning on automatic feature learning and the ability of anomaly detection in detecting novel anomalies. The adoption of the semi-supervised learning paradigm further reduces the false positive rate of the anomaly detection model. Compared with the existing deep learning-based methods, our method achieves higher detection accuracy with fewer training samples, and better performance in detecting novel unknown phishing web pages. Compared with traditional anomaly detection-based methods, we can automatically extract more comprehensive features, thus achieve higher detection accuracy, and can better adapt to the evolution of phishing attacks (Fig. 1).

2 Methods

In this section, we describe our method in detail. With the raw HTML codes as inputs, our anomaly scoring network directly learns and outputs the anomaly scores. We also use a prior distribution of the normal samples to generate reference labels to guide the learning of the network. Finally, the output of the network and the reference labels are fed into the contrastive loss to fulfill end-to-end semi-supervised learning.

2.1 Anomaly Scoring Network

Consider a raw HTML code x, we first split it into a sequence of tokens by spaces and punctuations: $(w_1, w_2, \cdots, w_m)$. Then we embed these tokens with an embedding layer E that maps each token w_m to a k-dimensional learnable vector $E(c) \in R^k$:

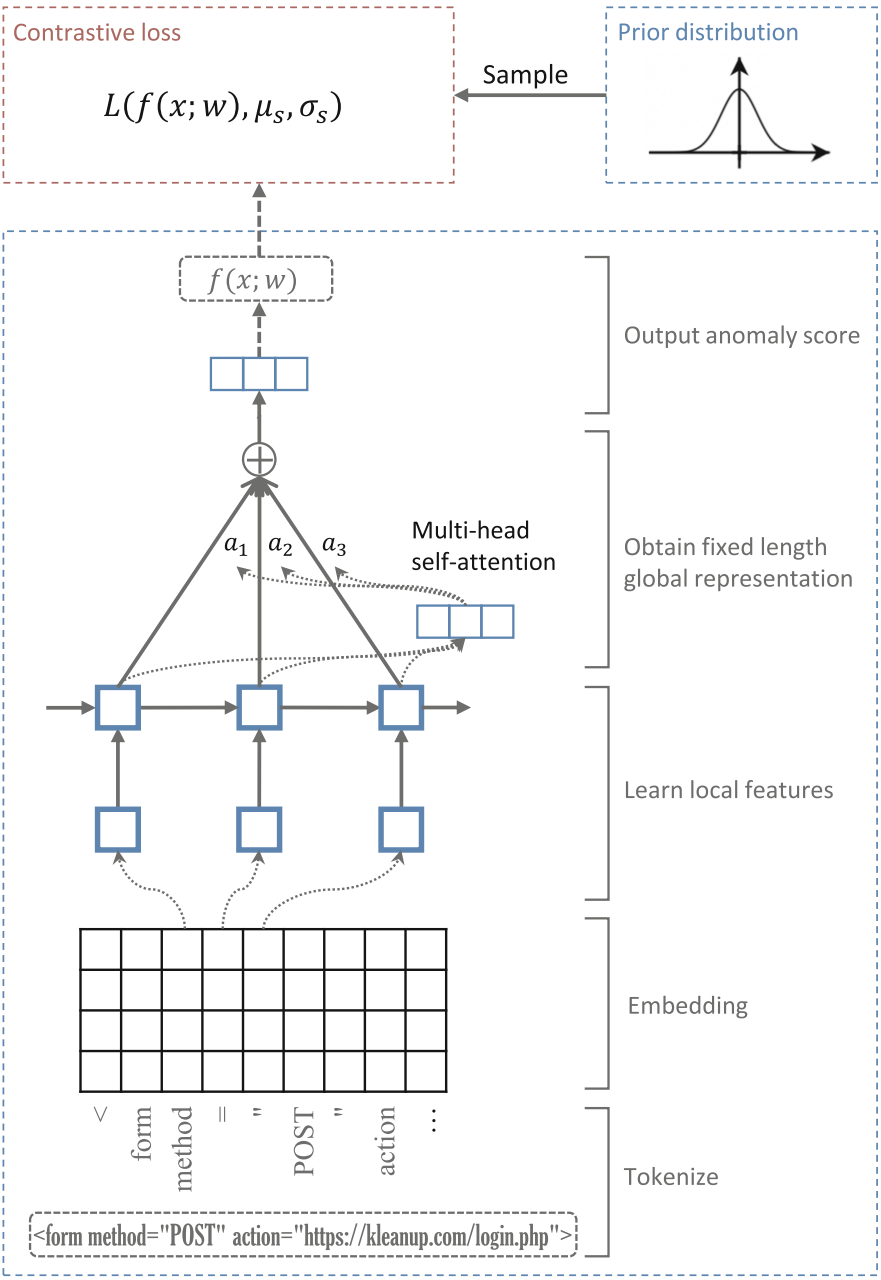

Fig. 1. The architecture of our proposed semi-supervised deep anomaly detection-based phishing detection model. There are three main components: 1) Anomaly scoring network that learns the feature and outputs the anomaly scores. 2) Gaussian prior that generate pseudo labels to guide the learning of the network. 3) Contrastive loss function that fulfill end-to-end semi-supervised training of the network.

$$(\mathbf{e_1}, \mathbf{e_2}, \cdots, \mathbf{e_m}) = (E(w_1), E(w_2), \cdots, E(w_m)) \tag{1}$$

Next, we fed the above sequence of embedded vectors into an RNN to extract their local context features:

$$Q = (\mathbf{q_1}, \mathbf{q_2}, \cdots, \mathbf{q_m}) = \text{RNN}(\mathbf{e_1}, \mathbf{e_2}, \cdots, \mathbf{e_m}) \tag{2}$$

The output of the above RNN is a sequence of vectors. Since the lengths of the HTML codes are variable, the lengths of the outputs are also variable. To obtain a fixed-length vector representation for each HTML code, a common method is using the attention mechanism. However, vanilla attention only focuses on one type of information. This is insufficient for phishing detection since it is crucial to synthesize multiple types of information from different locations of the HTML codes to accurately understand the anomalous behavior of the webpage. Therefore, we adopt the multi-head self-attention mechanism on the output of the RNN.

We first use a two-layer fully connect layer to obtain a set of attention weights:

$$A = \text{softmax}(W_2 \text{tanh}(W_1 Q^T)) \tag{3}$$

Then we use these attention weights to sum the output of the RNN to obtain the fixed-length sample representation:

$$\mathbf{m} = \text{flatten}(AQ) \tag{4}$$

This vector is then passed through a fully connected layer and an activation function, outputting the anomaly score:

$$f(x; \phi) = \text{sigmoid}(\mathbf{w_f m} + \mathbf{b}) \tag{5}$$

where ϕ represents all of the learnable parameters of the network.

2.2 Prior Distribution

To guide the training of the above anomaly scoring network to find good representations of normal HTML codes, a crucial task is to define appropriate learning targets. A naive solution is to use 0 as the learning target for all of the normal HTML codes. However, this may lead the network to converge to a degenerate solution where all the data points are transformed to a single point. To avoid this problem, we employ a Gaussian distribution as the prior distribution of the anomaly scores of the normal samples to generate pseudo labels. Using these pseudo labels as the learning targets of the anomaly scoring network provides a normalization effect and makes the model more robust to unclean training datasets. Specifically, we sample a small set of scores from the standard Gaussian distribution for each batch of samples in the training phase:

$$S = \{s_1, s_2, \cdots, s_n\} \sim \mathcal{N}(0, 1^2) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \tag{6}$$

We call this set a reference set. Then we calculate the mean μ_s and variance σ_s of this reference set.

2.3 Contrastive Loss

The output of the anomaly scoring network and the μ_s, σ_s of the reference set are fed into the contrastive loss to fulfill end-to-end learning. We first calculate the distance between the output of the anomaly scoring network and the center of the reference set:

$$D = \frac{f(x;\phi) - \mu_s}{\sigma_s} \tag{7}$$

D represents the probability that the sample x is anomalous.

Then, we define the constrastive loss:

$$\mathbf{L}(f(x;\phi), \mu_s, \sigma_s) = (1-Y)\frac{1}{2}D^2 + (Y)\frac{1}{2}\max(0, m-D)^2 \tag{8}$$

Minimizing this loss will push the anomaly scoring network to learn a meaningful representation of HTML codes and assign small scores for normal samples. In this way, we achieve end-to-end semi-supervised anomaly scoring training. Starting from the loss function, the gradient is first passed back to the attention layer, then to the RNN layer, finally to the embedding layer.

3 Experiments

In this section, we conduct extensive experiments to evaluate the proposed method. First, we introduce the settings of the experiments. Then we compare the performance of our proposed method with other state-of-the-art methods.

3.1 Dataset

To reliably evaluate the performance of the methods, we collect a large set of real web pages.

Data Collection. The phishing webpages are collected from PhishTank and OpenPhish, which are widely used as data sources in previous phishing webpage detection researches. For normal web pages, we build a crawler that treats TrancoTop1M as the start point. Our crawler pays special attention to the login pages in normal web pages since these pages are the common impersonate target.

Dataset Construction. Most existing phishing detection researches use balanced datasets, in which the number of phishing and normal samples are nearly equal. However, these datasets can't reliably evaluate the phishing detection methods in the real world where the number of phishing web pages is significantly less than normal web pages. In this work, we build a highly unbalanced dataset to simulate the real scenario. We divide the dataset into three parts: training set, validation set, and test set. To evaluate the ability of our proposed method to detect unknown phishing web pages, we adopt the group split method that divides the dataset according to the domain of each sample to ensure that samples of the same domain will not appear in the training set and test set at the same time. Table 1 shows the statistics of the final dataset.

Table 1. The statistics of the dataset.

Class	Train	Validation	Test	All
Phishing	132	38	137	307
Benign	7306	1822	5953	15081
All	7438	1860	6090	15388

Table 2. Phishing web page detection performance comparison.

Type	Method	AUC-ROC	AUC-PR	F1	Precision	Recall	Accuracy
Classification	SVM	0.8104	0.2472	0.3260	0.3237	0.3284	0.9694
	RNN	0.8170	0.2835	0.4014	0.3877	0.4160	0.9720
Anomaly detection	OC-SVM	0.8657	0.1980	0.1189	0.064	**0.8102**	0.7298
	PCA	0.7291	0.0625	0.1203	0.070	0.4233	0.8607
	IsolationForest	0.8337	0.1950	0.1883	0.1120	0.5912	0.8853
	Our method	**0.8788**	**0.4745**	**0.5044**	**0.6404**	0.4160	**0.9816**

3.2 Performance Metric

Due to the highly imbalanced nature of the dataset, it's not appropriate to use accuracy alone as the metric. Instead, we report multiple metrics, including AUC-ROC, AUC-PR, F1, Precision, Recall, and Accuracy. F1 and accuracy require a previously specified classification threshold. AUC-ROC and AUC-PR are thresholds-invariant metrics that aggregate the performance of the model across all possible thresholds. Among them, AUC-PR is considered the most important metric in our experiments since it focuses on the anomaly class.

3.3 Compared Methods

The competing methods can be grouped into two categories: supervised classification-based methods and anomaly detection-based methods.

The first group of methods is supervised classification-based methods.

- **Manual features+SVM**: Das et al. [4] extracts several features for phishing web page detection and utilizes the support vector machine (SVM) to perform classification.
- **HTMLPhish**: HTMLPhish [10] is the state-of-the-art supervised deep learning-based phishing HTML detection method that combines a character-level CNN and a word-level CNN into an end-to-end classifier.

The second group of methods is anomaly detection-based methods. We extract features as Das et al. [4] and utilize these anomaly detection methods to perform phishing web page detection.

- **One-Class SVM (OC-SVM)**: OC-SVM [13] is a classic anomaly detection method.

- **IsolationForest**: IsolationForest [8] is the state-of-the-art anomaly detection method before the decade of deep learning.
- **Deep Support Vector Data Description (Deep SVDD)**: Deep SVDD [12] is the state-of-the-art deep learning-based anomaly detection method.

3.4 Implementation Details

We implement the proposed method with PyTorch and run the experiments on a GPU with 11GB memory. The hyperparameters of our model include the embedding dimension, the hidden layer dimension, and the size of the reference set. We adjust these hyperparameters based on the performance of the model on the validation set. Both the embedding dimension and the hidden layer dimension are set to 64, the size of the reference set is set to 5. We use Adam as the optimizer with a learning rate of 0.01. When the loss stops decreasing for 3 consecutive epochs, we reduce the learning rate to 0.001 and then learn until the loss stops falling for another 3 consecutive rounds.

3.5 Performance Comparison

In this section, we compare our proposed method with other state-of-the-art phishing webpage detection methods. There are two major questions we aim to address:

- Is our method more accurate than the unsupervised anomaly detection-based methods?
- Is our method more accurate than the supervised deep learning-based method?

The performance of our proposed method and five competing methods are shown in Table 2. Our method outperforms other methods by a large margin in terms of all metrics except recall. Although IsolationForest achieves better recall than our method, its precision is significantly lower. The better performance of our method comparing with the supervised classification-based methods demonstrates that our semi-supervised anomaly detection approach can utilize the labeled training data more effectively. Also notice that the unsupervised anomaly detection-based methods have significantly worse detection accuracy than supervised classification-based methods, but our method achieves better performance than supervised methods.

4 Related Work

The method we proposed is conceptually related to previous phishing web page detection methods, and anomaly detection methods.

4.1 Traditional Phishing Web Page Detection

Traditional phishing web page detection methods can be broadly divided into three categories.

The first group is rule-based approaches. The blacklist/whitelist methods rely on a constantly maintained list of known phishing web pages [2]. The heuristic-based approaches [11] rely on experts designed rules to detect phishing web pages. These approaches are simple, precise, and fast, but have low recall and cannot detect zero-day phishing attacks[14].

The second group is visual similarity-based approaches [3,9], which detect phishing web pages by examining the visual similarity between phishing web pages and well-known non-phishing web pages. These methods can detect zero-day phishing web pages, but at cost of high computation. Besides, they can only detect the phishing web pages that impersonate well-known targets.

The final group is machine learning-based approaches [15,18,19] . These methods typically rely on features designed by experts [1,4,5] and classification algorithms such as SVM. This group of methods can detect zero-day phishing attacks, but the performance is limited by the expressive power of manually designed features.

4.2 Deep Learning Based Phishing Web Page Detection

Recently, with the advancement of deep learning, several phishing web page detection methods based on deep text classification models have been proposed.

Wang et al. [16] proposed PDRCNN, an improved deep learning phishing web page detection method that uses URL as input only. Their method treats URL as sequences of characters and utilize a fused neural network architecture that combines recurrent neural network and convolutional neural network.

Huang et al. [6] proposed to use attention-based hierarchical RNN to improve the phishing detection accuracy. Their method also uses URL as input only, but their hierarchical RNN structure learns the features from both character level and word level. The Attention mechanism was adopted to let the neural network focus on the important area of the URL.

Recently, Opara et al. [10] made the first attempt to apply the deep learning method to HTML input. They combined the character-level CNN and word-level CNN to improve the detection accuracy. They achieved significant improvement compared to traditional machine learning-based approaches. However, they ignore the inherent structure information of HTML, and CNN has difficulties in capturing long-range semantics in HTML.

4.3 Anomaly Detection

OC-SVM [13] conduct anomaly detection by learning an optimal hyper-plane that separates the normal samples from the origin. IsolationForest [8] utilizes the random forest to conduct anomaly detection. Anomalies tend to have a short path in the partitioning tree. Deep SVDD [12] is one of the first works in deep

anomaly detection. It utilizes neural networks to learn the feature representation that transforms the normal samples into a small hypersphere.

5 Conclusions

We have presented our semi-supervised deep anomaly detection approach for phishing web detection. Our method combined the strength of feature learning from deep learning with the ability to detect novel phishing samples from anomaly detection. The experimental results demonstrate that our approach achieves better performance than existing deep learning-based and anomaly detection-based phishing webpage detection methods.

References

1. AlEroud, A., Zhou, L.: Phishing environments, techniques, and countermeasures: a survey. Comput. Secur. **68**, 160–196 (2017)
2. Cao, Y., Han, W., Le, Y.: Anti-phishing based on automated individual white-list. In: Proceedings of the 4th Workshop on Digital Identity Management, pp. 51–60. ACM (2008)
3. Chiew, K., Fatt, J.C.S., Sze, S., Yong, K.S.C.: Leverage website favicon to detect phishing websites. Secur. Commun. Netw **2018**, 7251750:1-7251750:11 (2018)
4. Das, A., Baki, S., Aassal, A.E., Verma, R.M., Dunbar, A.: Sok: a comprehensive reexamination of phishing research from the security perspective. IEEE Commun. Surv. Tutor. **22**(1), 671–708 (2020)
5. Dou, Z., Khalil, I., Khreishah, A., Al-Fuqaha, A.I., Guizani, M.: Systematization of knowledge (sok): a systematic review of software-based web phishing detection. IEEE Commun. Surv. Tutor. **19**(4), 2797–2819 (2017)
6. Huang, Y., Yang, Q., Qin, J., Wen, W.: Phishing URL detection via CNN and attention-based hierarchical RNN. In: 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering, TrustCom/BigDataSE, pp. 112–119. IEEE (2019)
7. Lin, Z., et al.: A structured self-attentive sentence embedding. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017, Conference Track Proceedings. OpenReview.net (2017)
8. Liu, F.T., Ting, K.M., Zhou, Z.: Isolation forest. In: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), Pisa, Italy, 15–19 December 2008, pp. 413–422. IEEE Computer Society (2008). https://doi.org/10.1109/ICDM.2008.17
9. Mao, J., Li, P., Li, K., Wei, T., Liang, Z.: Baitalarm: detecting phishing sites using similarity in fundamental visual features. In: 2013 5th International Conference on Intelligent Networking and Collaborative Systems, pp. 790–795. IEEE (2013)
10. Opara, C., Wei, B., Chen, Y.: Htmlphish: enabling phishing web page detection by applying deep learning techniques on HTML analysis. In: 2020 International Joint Conference on Neural Networks, pp. 1–8. IEEE (2020)
11. Ramesh, G., Krishnamurthi, I., Kumar, K.S.S.: An efficacious method for detecting phishing webpages through target domain identification. Decis. Supp. Syst. **61**, 12–22 (2014)

12. Ruff, L., et al.: Deep one-class classification. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018, vol. 80, pp. 4390–4399. Proceedings of Machine Learning Research, PMLR (2018). http://proceedings.mlr.press/v80/ruff18a.html
13. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Comput. **13**(7), 1443–1471 (2001). https://doi.org/10.1162/089976601750264965
14. Sheng, S., Wardman, B., Warner, G., Cranor, L., Hong, J., Zhang, C.: An empirical analysis of phishing blacklists. In: CEAS 2009 (2009)
15. Stobbs, J., Issac, B., Jacob, S.M.: Phishing web page detection using optimised machine learning. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 483–490 (2020)
16. Wang, W., Zhang, F., Luo, X., Zhang, S.: PDRCNN: precise phishing detection with recurrent convolutional neural networks. Secur. Commun. Netw **2019**, 2595794:1-2595794:15 (2019)
17. Whittaker, C., Ryner, B., Nazif, M.: Large-scale automatic classification of phishing pages. In: Proceedings of the Network and Distributed System Security Symposium, NDSS 2010. The Internet Society (2010)
18. Xiang, G., Hong, J.I., Rosé, C.P., Cranor, L.F.: CANTINA+: a feature-rich machine learning framework for detecting phishing web sites. ACM Trans. Inf. Syst. Secur. **14**(2), 21:1-21:28 (2011)
19. Zhao, P., Hoi, S.C.H.: Cost-sensitive online active learning with application to malicious URL detection. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 919–927. ACM (2013)