Phishing URL Detection Via Capsule-Based Neural Network

Yongjie Huang, Jinghui Qin, Wushao Wen*
School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China
huangyj45@mail2.sysu.edu.cn; qinjingh@mail2.sysu.edu.cn; wenwsh@mail.sysu.edu.cn

***Abstract*—As a cyber attack which leverages social engineering and other sophisticated techniques to steal sensitive information from users, phishing attack has been a critical threat to cyber security for a long time. Although researchers have proposed lots of countermeasures, phishing criminals figure out circumventions eventually since such countermeasures require substantial manual feature engineering and can not detect newly emerging phishing attacks well enough, which makes developing an efficient and effective phishing detection method an urgent need. In this work, we propose a novel phishing website detection approach by detecting the Uniform Resource Locator (URL) of a website, which is proved to be an effective and efficient detection approach. To be specific, our novel capsule-based neural network mainly includes several parallel branches wherein one convolutional layer extracts shallow features from URLs and the subsequent two capsule layers generate accurate feature representations of URLs from the shallow features and discriminate the legitimacy of URLs. The final output of our approach is obtained by averaging the outputs of all branches. Extensive experiments on a validated dataset collected from the Internet demonstrate that our approach can achieve competitive performance against other state-of-the-art detection methods while maintaining a tolerable time overhead.**

Keywords—phishing detection; cyber security; deep learning; capsule network

I. INTRODUCTION

According to Anti-Phishing Working Group (APWG)[1], phishing is a cyber attack that employs both social engineering and sophisticated technical subterfuge to steal users' private information like financial data. Usually, criminals use spoofed e-mails or other messages to lead users to counterfeit websites which are designed to lure users into divulging their private information like financial data. Recent decades have witnessed a dramatic growth of phishing attacks. As reported by APWG[2], the number of phishing websites detected in the first quarter 2019 was 180,768, which was up remarkably from the 138,328 seen in the fourth quarter 2018, and from the 151,014 seen in the third quarter 2018. Phishing has caused severe damage to many industries, e.g., Software-as-a-Service (SaaS) and webmail services, payment, financial institution, etc. According to the Federal Bureau of Investigation (FBI)'s latest report[3], there was a 136% increase in identified exposed losses from December 2016 to May 2018, and the loss due to phishing attacks has reached 12.5 billion dollars worldwide.

To mitigate the threat of phishing attacks, researchers in academia and industry have been dedicated to improving the performance of phishing detectors. Current phishing detection methods can be simply divided into list-based and machine learning-based methods which the latter one leverages handcrafted features of URLs, host information and website contents to build a classifier[4]. However, anti-phishing remains an open challenge with no authoritative solution because of the constant evolution of phishing attacks, which makes developing techniques that can detect phishing websites efficiently and effectively continues to be an urgent need.

To thwart phishing attacks, we propose a capsule-based deep learning neural network, which turns out to be efficient and effective for phishing URL detection. The reason why we detect phishing by exploiting patterns of URLs can be threefold. First, it has been proved that URL-based methods[5-7] can achieve comparable performance against content-based methods. Second, since URL-based detection does not require assistance from third-party or the webpage content, we can achieve higher detection speed and realize real-time detection. Third, there is no security risk involved in URL-based approaches since they detect phishing by utilizing URLs only rather than downloading webpage contents which may carry crimeware. Because of the focus on URLs only, our approach can be applied to everywhere that a URL might exist, e.g., e-mails, websites, other text messages, etc.

In this paper, we focus on building a phishing URL detector with deep learning. Specifically, inspired by Sabour et al.[8], we designed a capsule-based neural network which proved to be efficient and effective for phishing URL detection. Considering that a phishing website might imitate the URL of a legitimate one by modifying the legitimate URL slightly and the advantages of character-level Convolutional Neural Network (CNN) shown by Kim et al.[9] and Xiang et al.[10], we propose a capsule-based neural network which leverages CNN to extract features from URLs. Our neural network contains several parallel branches wherein one convolutional layer is used to extract shallow features from URLs and the subsequent two capsule layers are utilized for generating accurate feature representations of URLs from the shallow features and determining whether a URL is a phishing URL or a legitimate one. More detailedly, we use four branches to extract the feature representations of URLs and determine the legitimacy of URLs, and we average their outputs as the final output to improve the generalization ability of our approach. To evaluate the performance of our approach, we conducted extensive experiments on a dataset comprised of 4,820,940 URLs, among

978-1-7281-2458-2/19/$31.00 ©2019 IEEE

which 241,047 are phishing URLs collected from Phishtank[11] and Openphish[12] while others are crawled based on the Alexa[13] top one million sites. As baselines, we consider seven state-of-the-art methods which can be simply divided into three categories, namely heuristic-based, machine learning-based and deep learning-based detection respectively. Because we have built an imbalanced dataset where the ratio of legitimate URLs to phishing ones is 95:5 to represent the real world distribution as precise as possible, and Area Under the Receiver Operating Characteristic Curve (AUC)[14] is insensitive to such imbalanced dataset, we consider AUC together with F1 score, true positive rate, false positive rate, accuracy, recall rate, and precision as our evaluation metrics to fully evaluate the performances of different methods in phishing URLs detection.

In summary, our primary contributions include:

- We propose a novel capsule-based neural network consisting of four branches wherein one convolutional layer and two capsule layers are utilized to discriminate the legitimacy of a URL.
- Extensive experiments are conducted and the results demonstrate the efficiency and effectiveness of our approach, indicating that the performance and generalization ability of our approach outperform the compared state-of-the-art methods.

The remainder of this paper is organized as follows: In Section 2, we will review the representative works on phishing detection. Next, we shed light on the details of our proposed approach in Section 3. Then, in Section 4, we will describe our experimental setup and analyze the experimental results. Finally, conclusion will be provided in Section 5.

II. RELATED WORK

The anti-phishing approaches can be categorized into four classes generally, i.e., list-based approaches, heuristic-based approaches, machine learning-based approaches, and deep learning-based approaches.

A. *List-Based Approaches*

List-based phishing detection approaches mainly include whitelist-based schemes and blacklist-based schemes. Wang et al.[15] proposed a lightweight and efficient whitelist-based technique to detect phishing websites. However, blacklist-based approaches are used more widely in publicly available anti-phishing toolbars such as Google Safe Browsing[16], a browser plugin which checks URLs against Google's constantly updated database of phishing URLs on browsers and gives warnings to users once it judges a URL as a phishing one. Besides, Prakash et al.[17] proposed an improved list-based method to prevent phishers from evading the blacklist-based phishing detection approaches by expanding the blacklist of phishing URLs through some specific heuristics rules and performing an approximate match rather than an exact match. Nevertheless, list-based approaches suffer from an inability to defend against zero-hour attacks, i.e., they cannot detect the newly emerging phishing URLs until developers update the phishing database.

B. *Heuristic-Based Approaches*

Heuristic-based approaches depend on the features generated by cyber experts, e.g., URL-based lexical features, URL-based host information, webpage contents, etc. Zhang et al.[18] proposed an approach named Cantina for phishing websites detection based on webpage contents. Cantina first extracts key terms of a webpage using Term Frequency-Inverse Document Frequency (TF-IDF) to provide a unique signature of the webpage and then searches the signature via Google to infer the legitimacy of that webpage. As an enhanced version of Cantina, Xiang et al.[19] proposed Cantina+ that takes 15 heuristic features to train a phishing detector. Yet, despite the improvement of generalization ability, unfortunately, such feature engineering not only is time consuming but also needs constant adaptation since circumvention techniques evolve constantly, which imposes a restriction on the achievable performances of the detectors.

C. *Machine Learning-Based Approaches*

Recent years have witnessed several efforts to detect phishing websites with machine learning techniques. Cui et al.[20] proposed to detect phishing websites via a hierarchical clustering approach. Apart from clustering, researchers also focus on training a classifier to detect phishing websites. Zhang et al.[21] trained a phishing classifier from bag-of-words features of URLs via online learning. Similarly, Sahingoz et al.[22] trained seven phishing classifiers by applying different machine learning algorithms on distributed representations of words within a given URL.

D. *Deep Learning-Based Approaches*

As more and more success achieved by deep learning in different areas, more and more phishing detection methods embrace deep learning techniques. While some researches[5-7, 23] focus on detecting phishing via exploiting the patterns embedded in URLs, other studies[24-26] learn a classifier using webpage contents.

III. PROPOSED APPROACH

A. *Problem Overview*

Given a URL u, we suppose a capsule-based neural network $\mathcal{F}$ parameterized by weights and biases is used to learn an end-to-end mapping function between u and {0, 1} where 0 indicates that u is classified as a legitimate URL and 1 otherwise.

B. *Capsule-Based Neural Network*

As depicted in Fig. 1, our proposed neural network first transforms a URL into a sequence of L_c characters through padding or trimming. Then we look up a k-dimensional character-level embedding table, which converts a URL into a distributed representation. After that, the distributed representation of a URL is input into four parallel branches which mainly composed of one convolutional layer and two capsule layers. The convolutional layer consisting of one convolution operation and one spatial dropout operation. The convolution operation is used to extract shallow features from

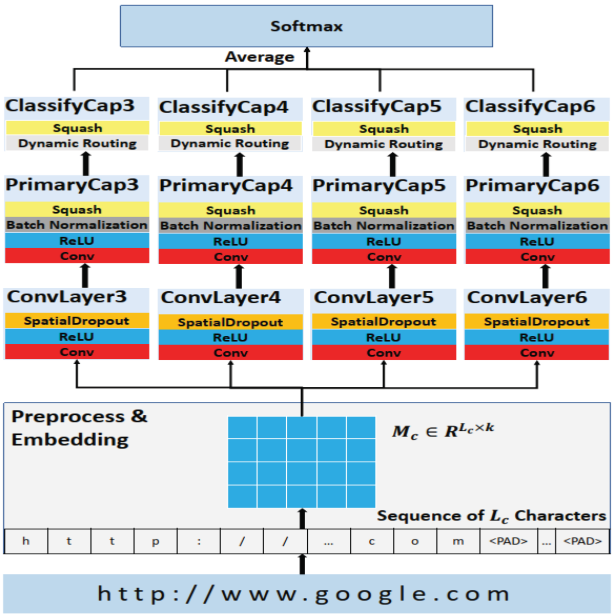

Figure 1. Capsule-based neural network architecture.

Algorithm 1: Dynamic Routing

```
Input:
        Input $\hat{\mathbf{u}_{j|i}}$, number of iteration $r$, layer depth $l$
Output:
        Squashed vector $v_j$
for all capsule $i$ in layer $l$ and
        capsule $j$ in layer $l+1$: $b_{ij} \leftarrow 0$;
for $r$ iterations do
    | for all capsule $i$ in layer $l$: $\mathbf{c}_i \leftarrow \mathrm{softmax}(\mathbf{b}_i)$;
    | for all capsule $j$ in layer $l+1$: $\mathbf{s}_j \leftarrow \sum_i c_{ij}\hat{\mathbf{u}_{j|i}}$;
    | for all capsule $j$ in layer $l+1$: $\mathbf{v}_j \leftarrow \mathrm{squash}(\mathbf{s}_j)$;
    | for all capsule $i$ in layer $l$ and
    |         capsule $j$ in layer $l+1$: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}_{j|i}}\mathbf{v}_j$;
End
```

the distributed representation, and the spatial dropout is for the improvement of the network generalization ability. In the primary capsule layer, i.e., the first capsule layer, we use one convolution operation to extract accurate feature representation from the shallow features generated by the former convolutional layer. Besides, we utilize batch normalization to improve the performance of our network, and then we leverage a squashing function to make sure that the output vector gets shrunk to between 0 and 1.

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2}\frac{s_j}{\|s_j\|}, \tag{1}$$

where s_j is the total input of capsule j and v_j is its vector output.

Next, we introduce our classification capsule layer wherein the dynamic routing algorithm and the squashing function are used to calculate the output of the corresponding branch. The total input s_j to capsule j in our classification capsule layer is a weighted sum over all $\hat{u_{j|i}}$ from capsules j in the primary capsule layer.

$$\hat{u_{j|i}} = W_{ij}u_i, \tag{2}$$

where u_i is the output of a capsule j in the primary capsule layer and W_{ij} is a weight matrix.

$$s_j = \sum_i c_{ij}\hat{u_{j|i}}, \tag{3}$$

where c_{ij} is the coupling coefficients calculated by the dynamic routing algorithm.

$$c_{ij} = \frac{exp(b_{ij})}{\sum_k exp(b_{ik})}, \tag{4}$$

where b_{ij} represents the probability that capsule j should be coupled to capsule j.

Finally, we average the outputs from four branches and get the final output through a softmax function. All convolution operations are equipped with ReLU activation function.

To train our network, we adopt a separate margin loss:

$$L_k = T_k \max(0, m^+ - \|\mathbf{v}_k\|)^2 + \lambda(1-T_k)\max(0, \|\mathbf{v}_k\| - m^-)^2, \tag{5}$$

where $T_k = 1$ if and only if a URL of class k is present, and λ is used for numerical stability during training.

IV. Experiments

A. Dataset

We built a dataset containing legitimate URLs crawled from Alexa top one million sites and phishing URLs collected from Phishtank and Openphish. The collection process lasted for three months from August 2018 to November 2018, resulting in a dataset of over 4.8 million URLs, among which 5% are phishing ones while the other 95% are legitimate. To get rid of mislabeled data, we leveraged a third-party platform called VirusTotal[27] for validation. In order to fully evaluate the performance of our proposed capsule-based neural network and avoid the "look-ahead" time bias, we sorted the verified dataset by collection time in ascending order and split it into three parts, namely training set, validation set, and test set respectively. Other details about the dataset can be found in TABLE I.

Since we have built a neural network which takes characters as input, publicly available word embeddings are not suitable for this work. Therefore, we managed to apply the skip-gram algorithm to building a k-dimensional character-level embedding trained on 3,295,473,093 unclassified URLs crawled from Common Crawl[28].

TABLE I. URL DATASET

	Phishing	Legitimate	Total
Training	167,346	3,179,574	3,346,920
Validation	9,905	188,195	198,100
Testing	63,796	1,212,124	1,275,920
Total	241,047	4,579,893	4,820,940

TABLE II. PERFORMANCE ON TEST DATA. THE BEST PERFORMANCES HAVE BEEN BOLDED.

Model	AUC	F1	TPR	FPR	Acc	Recall	Precision
RF on Heuristic Features	0.9220	0.6252	0.5202	0.0072	0.9689	0.5202	0.7836
LR on Heuristic Features	0.8380	0.4403	0.2870	0.0009	0.9636	0.2870	0.9459
SVM on Heuristic Features	0.8323	0.8189	0.7673	0.0029	0.9678	0.7673	0.8779
RF on Bag of Words	0.9551	0.6261	0.9873	0.0583	0.9410	0.9873	0.4585
LR on Bag of Words	0.9447	0.5956	0.9872	0.0667	0.9329	0.9872	0.4265
SVM on Bag of Words	0.9472	0.8389	0.7813	0.0043	0.9770	0.7813	0.9059
URLNet	0.9407	0.4063	**0.9933**	0.1201	0.8844	0.9933	0.2554
Ours	**0.9966**	**0.9616**	0.9349	**0.0005**	**0.9963**	**0.9349**	**0.9898**

B. Baseline

As baselines, we consider seven state-of-the-art models which can be simply divided into heuristic-based, machine learning-based and deep learning-based models. For the heuristic-based baselines, we select 16 handcrafted features from prior works[5, 29] and then apply three prevalent machine learning algorithms implemented by scikit-learn[30] to train three different classifiers, including Random Forest (RF), Logistic Regression (LR) and Support Vector Machine (SVM). For the machine learning-based baselines, we train another three different classifiers using the same three algorithms that take bag-of-words of URLs as input. We select URLNet[6] as the deep learning-based baseline.

C. Network Configuration

We set the number of characters in a URL L_c to 200 and the embedding dimension k to 128 empirically. For the primary capsule layer, we set the number of capsules to 32 and each capsule consists of 8 neurons. Besides, the convolution kernel size in the primary capsule layer is set to 9 and the stride is 2 without padding. As for the classification capsule layer, the number of capsules is set to 2 since our goal is to build a binary classification network, and each capsule in this layer consists of 16 neurons. Parameters in (5) are set as follows: m^+ is set to 0.9, m^- is set to 0.1, and λ is set to 0.5, empirically. Convolution kernel sizes in the four convolutional layers are set to 3, 4, 5, 6, respectively. In convolutional layers, we set convolution stride to 1, and the number of filters is 64 for each branch. For the Adam optimizer, the learning rate starts from 0.0001 and halved at every epoch. The dropout rate is set to 0.5 while the weight decay coefficient is 0.001, and we set the number of iterations during dynamic routing to 3. We train our network for 5 epochs with a batch size of 128.

D. Evaluation Metrics

Since the number of phishing URLs compared to legitimate ones is much smaller, and we have built an imbalanced dataset subject to the real-world distribution, we consider AUC together with F1 score, true positive rate (TPR), false positive rate (FPR), accuracy (Acc), recall rate, and precision as our evaluation metrics to fully evaluate the performances of different methods in phishing URLs detection.

E. Results

From TABLE II, we can see that our approach outperforms the baselines across most evaluation metrics by a large margin on the dataset. As shown by the results, classifiers trained on bag-of-words achieve better performance than that trained on heuristic features generally, indicating that heuristic features are becoming easier and easier to be circumvented by phishing criminals. According to the experimental results, while random forest achieves the highest AUC among the selected three machine learning algorithms, SVM gets the best F1 score and the classifier trained on heuristic features with logistic regression has the lowest false positive rate. Surprisingly, albeit URLNet achieves excellent performance on AUC and true positive rate, it has the highest false positive rate and the lowest precision, which might be due to that URLNet could not handle the rare words in our dataset well enough and thus fails to extract accurate feature representations of URLs. Our approach achieves the best AUC, F1 score, FPR, accuracy, and precision among all the models by a significant margin. It should be mentioned that we would like to choose a classifier with lower false positive rate rather than that with lower false negative rate since misclassification of a legitimate website might lead to serious consequences such as lawsuits and financial loss. Although the true positive rate and recall rate of our approach are not the best among these methods, our approach has the lowest false positive rate and ranks the first place on the other evaluation metrics. In addition, our approach can process over 2800 URLs in one second, resulting in the time overhead of each URL less than 0.35 ms. Therefore, our experimental results demonstrate that our capsule-based neural network can build accurate feature representations of URLs and achieve the superior generalization ability over state-of-the-art models.

CONCLUSION

In this work, we proposed an efficient and effective capsule-based neural network and demonstrated the superior performance of our approach with extensive experiments against seven state-of-the-art models on a large URL dataset we built. According to the experimental results, our capsule-based neural network outperforms the state-of-the-art methods significantly. However, phishing attacks is so complicated that no authoritative detection solution exists to thwart all the vulnerabilities effectively; thus developing efficient and effective phishing detection techniques is a long-term demand.

ACKNOWLEDGMENT

This research has been supported by grant 2017YFB1001603 in the National Key R&D of China.

References

[1] Anti-Phishing Working Group, https://www.antiphishing.org/

[2] G. Aaron, "Phishing activity trends report, 1st quarter 2019." [Online]. Available: http://docs.apwg.org/reports/apwg trends report q1 2019.pdf

[3] FBI, "Business e-mail compromise the 12 billion dollar scam." [Online]. Available: https://www.ic3.gov/media/2018/180712.aspx

[4] Z. Dou, I. Khalil, A. Khreishah, A. Al-Fuqaha, and M. Guizani, "Systematization of knowledge (sok): A systematic review of softwarebased web phishing detection," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2797–2819, 2017.

[5] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. Gonzalez, "Classifying phishing urls using recurrent neural networks," in Electronic Crime Research (eCrime), 2017 APWG Symposium on. IEEE, pp. 1–8, 2017.

[6] H. Le, Q. Pham, D. Sahoo, and S. C. Hoi, "Urlnet: Learning a url representation with deep learning for malicious url detection," arXiv preprint arXiv:1802.03162, 2018.

[7] P. Yang, G. Zhao, and P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," IEEE Access, vol. 7, pp. 15196–15209, 2019.

[8] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in Advances in neural information processing systems, pp. 3856–3866, 2017

[9] Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint arXiv:1408.5882, 2014.

[10] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in Advances in neural information processing systems, pp. 649–657, 2015.

[11] Phishtank, http://www.phishtank.com/

[12] Openphish, https://openphish.com/

[13] Alexa, https://www.alexa.com/

[14] J. A. Hanley and B. J. McNeil, "A method of comparing the areas under receiver operating characteristic curves derived from the same cases." Radiology, vol. 148, no. 3, pp. 839–843, 1983.

[15] Y. Wang, R. Agrawal, and B.-Y. Choi, "Light weight anti-phishing with user whitelisting in a web browser," in Region 5 Conference, 2008 IEEE. IEEE, pp. 1–4, 2008.

[16] Google Safe Browsing, https://developers.google.com/safe-browsing/

[17] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "Phishnet: predictive blacklisting to detect phishing attacks," in INFOCOM, 2010 Proceedings IEEE. Citeseer, pp. 1–5, 2010.

[18] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: a content-based approach to detecting phishing web sites," in Proceedings of the 16th international conference on World Wide Web. ACM, pp. 639–648, 2007.

[19] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "Cantina+: A featurerich machine learning framework for detecting phishing web sites," ACM Transactions on Information and System Security (TISSEC), vol. 14, no. 2, pp. 21, 2011.

[20] Q. Cui, G.-V. Jourdan, G. V. Bochmann, R. Couturier, and I.-V. Onut, " Tracking phishing attacks over time," in Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 667–676, 2017.

[21] W. Zhang, Y.-X. Ding, Y. Tang, and B. Zhao, "Malicious web page detection based on on-line learning algorithm," in Machine Learning and Cybernetics (ICMLC), 2011 International Conference on, vol. 4. IEEE, pp. 1914–1919, 2011.

[22] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from urls," Expert Systems with Applications, vol. 117, pp. 345–357, 2019.

[23] J. Saxe and K. Berlin, "expose: A character-level convolutional neural network with embeddings for detecting malicious urls, file paths and registry keys," arXiv preprint arXiv:1702.08568, 2017.

[24] B. Athiwaratkun and J. W. Stokes, "Malware classification with lstm and gru language models and a character-level cnn," in Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE, pp. 2482–2486, 2017.

[25] M. Nguyen, T. Nguyen, and T. H. Nguyen, "A deep learning model with hierarchical lstms and supervised attention for anti-phishing," arXiv preprint arXiv:1805.01554, 2018.

[26] J. Saxe, R. Harang, C. Wild, and H. Sanders, "A deep learning approach to fast, format-agnostic detection of malicious web content," arXiv preprint arXiv:1804.05020, 2018.

[27] VirusTotal, https://www.virustotal.com/

[28] Common Crawl, https://commoncrawl.org/

[29] 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 ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 1245–1254, 2009.

[30] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.