POSTER: A PU Learning based System for Potential Malicious URL Detection

Ya-Lin Zhang[†,‡], Longfei Li[‡], Jun Zhou[‡], Xiaolong Li[‡], Yujiang Liu[‡], Yuanchao Zhang[‡],Zhi-Hua Zhou[†]
[†]National Key Lab for Novel Software Technology, Nanjing University, China
[†]{zhangyl, zhouzh}@lamda.nju.edu.cn
[‡]Ant Financial Services Group, China
[‡]{longyao.llf, jun.zhoujun, xl.li, yujiang.lyj, yuanchao.zhang}@antfin.com

ABSTRACT

This paper describes a PU learning (Positive and Unlabeled learning) based system for potential URL attack detection. Previous machine learning based solutions for this task mainly formalize it as a supervised learning problem. However, in some scenarios, the data obtained always contains only a handful of known attack URLs, along with a large number of unlabeled instances, making the supervised learning paradigms infeasible. In this work, we formalize this setting as a PU learning problem, and solve it by combining two different strategies (two-stage strategy and cost-sensitive strategy). Experimental results show that the developed system can effectively find potential URL attacks. This system can either be deployed as an assistance for existing system or be employed to help cyber-security engineers to effectively discover potential attack mode so that they can improve the existing system with significantly less efforts.

KEYWORDS

URL Attack Detection, Machine Learning, PU Learning

1 INTRODUCTION

With the rapid development of internet, more and more kinds of URL attacks have arisen, becoming a serious threat to cyber-security. Traditional URL attack detection systems are mainly constructed through the use of blacklists or rule lists. These lists will gradually become much longer, and it is impracticable to cover all attacks by these ways. More severely, these kinds of methods lack the ability of detecting potential attacks, making it awkward for cyber-security engineers to efficiently discover newly generated URL attacks.

To provide better generalization performance, machine learning based approaches have been employed to this task. These approaches mainly fall into two categories: most formalize it as a supervised learning problem, in which labeled data are needed [6], while the rest try to solve the problem in an unsupervised manner, e.g., by anomaly detection techniques [5], with no label information required. When the labeled data can be obtained, supervised learning methods can always provide better generality. However, in some conditions, the exact label information is difficult to acquire. For example, we may only get a small set of malicious URLs and a large amount of unlabeled URL records, which means that the aforementioned supervised learning methods can not be directly employed, as we have no labeled negative instances. On the other hand, if we simply solve it in an unsupervised manner, the label information of the known malicious URLs will be terribly wasted, and the performance may be extremely unsatisfactory.

In this paper, we formalize the aforementioned setting as a PU learning (Positive and Unlabeled learning) problem [3], which can naturally make better use of the detected malicious URLs, along with the unlabeled URLs, and provide better performance. Furthermore, we develop a potential URL attack detection system based on PU learning methods. There are many strategies which can be employed to handle PU learning problem, such as two-step strategy [4], cost-sensitive strategy [2], etc. In this work, we combine the models of two-step strategy and cost-sensitive strategy to construct our system. We empirically evaluate the developed system, and the results show that it can effectively find potential URL attacks and significantly reduce the efforts of cyber-security engineers, making it useful in real scenarios of URL attack detection.

The rest of this paper is organized as follows. In Section 2 we describe the developed system. In Section 3 we empirically evaluate the developed system based on the scenario we encountered in Ant Financial[1]. Finally, in Section 4 we conclude the work.

2 THE SYSTEM ARCHITECTURE

In this section, we present the architecture of the developed system. As shown in Figure 1, our system mainly contains 3 modules: (i) **Feature Extraction**, which transforms original URLs into numerical feature vectors; (ii) **Model Training**, which trains PU learning models using the extracted features of training URLs; (iii) **Prediction**, which makes prediction for the new-coming URLs, and a candidate malicious URL set will be outputted by the system.

2.1 Feature Extraction

The original URLs are first transformed to numerical feature vector representations, which can be conveniently used in the subsequent machine learning algorithms. Below, we briefly explain the points of focus of our developed system and present the details of feature extraction process that we use in the system.

[1]Ant Financial is a technology company that brings inclusive financial services to the world. It operates Alipay, the world's largest mobile and online payments platform.

CCS '17, October 30-November 3, 2017, Dallas, TX, USA

ACM ISBN 978-1-4503-4946-8/17/10.
https://doi.org/10.1145/3133956.3138825

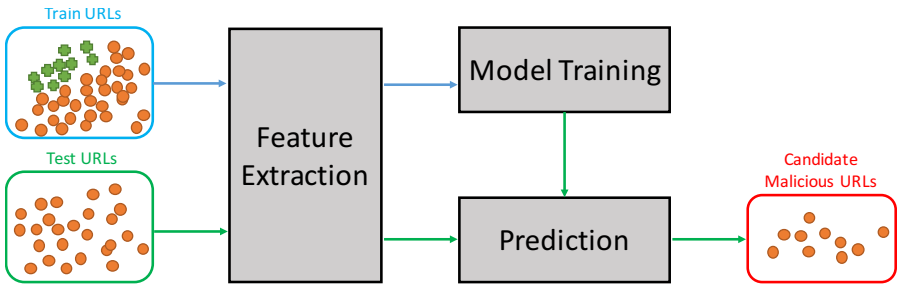

Figure 1: Overview of the proposed system

scheme:[//[user[:password]@]host[:port]][/path][?query][#fragment]

Figure 2: The generic syntax of URLs

Generally speaking, a URL can be separated into several parts, including the scheme part, the authority part, the path part, the query part and fragment part, as shown in Figure 2. A vicious user can modify any of these parts for malicious purpose. In our scenario, the first few parts are restricted, and the attacks mainly come from the fragment part, so we mainly focus on the situation that the attack is executed based on the malicious modification of the fragment parts. Specifically, the fragments are always formed as '$key_1 = value_1 \& \cdots \& key_n = value_n$', and the value may be arbitrarily modified by the attackers to make an attack. Thus, our system mainly deals with this setting, and the feature extraction process directly extracts features from the key-value pairs of the fragment parts.

To be more specific, given a set of URLs, we first separate each of them into the aforementioned parts, and key-value pairs are extracted from the fragments of each URL. Second, since we are focused on discovering the trait of *malicious* URLs, we filter the key-value pairs and only keep the top-N keys that appear mostly in the *malicious* URLs, while the rest of the key-value pairs for each URL are collected together as one key-value pair, thus there will be at most $(N + 1)$ key-value pairs extracted from each URL. Finally, we heuristically extract 8 different statistical information from each filtered value, including the count of *all* characters, letters, numbers, punctuations in the value, and the count of *different* characters, letters, numbers, punctuations in the value. Thus each URL will be described by a $(N + 1) * 8$ dimensional feature vector.

2.2 Model Training

Note that traditional supervised learning techniques can not be directly used in our scenario, as negative labels are unaccessible. In this work, we formalize this problem as a PU learning (Positive and Unlabeled learning) problem.

PU learning [3] is a special case of semi-supervised learning [1, 7], which deals with the tasks where only positive and unlabeled instances are provided, while no negative instance is given. Many strategies have been proposed to solve it. To bypass the embarrassment of lacking labeled negative instances, two-step strategy attempts to first excavate some reliable negative instance, and then transforms the problem into a traditional supervised or semi-supervised learning problem. On the other hand, cost-sensitive learning techniques for binary classification with unequal misclassification cost are readily available for handling PU learning problem [2]. In our developed system, these two strategies are both employed and combined further to form the final prediction model.

Two-Stage Strategy: We select reliable negative instances from unlabeled instances in the first stage, the details of the algorithm are showed in Algorithm 1. In stage two, with the positive instances and selected reliable negative instances, a traditional supervised model is trained and will be further used for predicting new instances.

In this work, with the consideration of efficiency, we employ logistic regression to train the classification model.

Algorithm 1 Reliable Negative Instances Selection

Input: Positive Instance Set P, Unlabeled Instance Set U, Sample Ratio s.
Output: Reliable Negative Instance Set RN.

```
Set RN = ∅
Sample s% of the instances from P as S
Set P_s = P − S with label 1, U_s = U ∪ S with label -1
Train a classifier g with P_s and U_s
Classify instances in U using g, output the class-conditional-probability
Select a threshold θ according to the class-conditional-probability of instances in S
for d ∈ U do
    if Pr(1|d) ≤ θ, RN = RN ∪ d
end for
Output RN.
```

Cost-Sensitive Strategy: We assume that there are very few positive instances in unlabeled instances. By attaching all unlabeled instances with negative labels, the following objective function is minimized:

$$C_+ \sum_{y_i=1} l(y_i, f(x_i)) + C_- \sum_{y_i=-1} l(y_i, f(x_i)) + \lambda R(w)\,, \tag{1}$$

in which C_+ and C_- denote the penalty factor for misclassification of positive and negative instances, respectively. $l(y_i, f(x_i))$ is the loss, such as log-loss and hinge-loss. λ is the regularization coefficient and $R(w)$ is the regularization term, like $L1-$norm and $L2-$norm. In this work, we set the loss to be log-loss and the regularization term to be $L2-$norm. The specific objective function is as:

$$C_+ \sum_{y_i=1} LL(y_i f(x_i)) + C_- \sum_{y_i=-1} LL(y_i f(x_i)) + \lambda \|w\|^2\,, \tag{2}$$

in which $LL(z) = \log(1 + \exp(-z))$ is the log-loss. In practice, C_+ and C_- are selected via a validation set, and C_+ is always much bigger than C_-, which means that the penalty of misclassifying a positive instance is much bigger than misclassifying a negative one. The learned model will pay more attention on correctly classifying malicious URLs.

2.3 Prediction

In prediction phase, a new-coming URL will be firstly delivered to the feature extraction module to transfer the original URL into a $(N + 1) * 8$ dimensional vector. Then the extracted feature vector is fed into the two obtained models (by using two-stage strategy and cost-sensitive strategy) described before, and each model will output a score to denote the probability of the URL being malicious. The higher the score is, the more likely the URL is a malicious one. We simply average the two scores as the final score of an URL. The URLs with higher scores are selected to construct the candidate malicious URL set.

In practice, we will filter a set of K URLs based on the candidate malicious URL set, and the filtered URLs will be manually checked by the cyber-security engineers to verify the result.

3 EMPIRICAL EVALUATION

3.1 Dataset and Setup

The dataset is sampled from the daily-arrived URL requests in Ant Financial. Note that the data mainly contains two parts: a large set of unlabeled URLs and a handful of malicious URLs which have been already marked by the existing system, and different attack types may appear among the malicious ones, including XXE (XML External Entity Injection), XSS (Cross SiteScript) and SQL injection, etc. We simply regard all these types as malicious URLs with no subdivision. Since the total dataset is too large, we sample 1 billions of URLs from each day's requests, in which the number of detected malicious URLs by the existing system varies from tens of thousands to hundreds of thousands. The model is trained using data collected in 7 consecutive days, and will be used to predict the scores of each day's new-coming *unlabeled* URLs.

When extracting key-value pairs, N is set to be 99, so that each URL is described by a 800 dimensional vector. Min-max normalization is used to process the features to the same scale. As we explained in the model training section, logistic regression based methods are employed to train the PU learning models. The parameters such as C_+, C_- and λ are selected via a validation set.

3.2 Empirical Results

Since we have no label information of the daily-arrived unlabeled URLs, we use the help of the cyber-security engineers to check the results and verify the effectiveness of our system.

It is very time-consuming to check the results, so we set the size K of the candidate malicious URL set to be at most 150, and cyber-security engineers will manually check whether the selected URLs are malicious or benign. Table 1 summarizes the details of three day's results. As we can see from the table, the accuracy of the filtered candidate set can be up to 90%, indicating that the proposed system can effectively discover the potential malicious URLs, which can not be captured by the existing system. What should be specially mentioned is that new attack modes have been discovered based on the candidate malicious URL set, and the cyber-security engineers in Ant Financial have already improve the existing system with this help. At the same time, the developed system can also be used together with the existing system to improve the ability of the whole system.

Table 1: Evaluation results of the candidate malicious URLs. The number of selected candidate URLs, the number of confirmed malicious URLs and the accuracy (%) are presented.

	Date A	Date B	Date C
# Candidate Ins.	113	103	141
# Malicious Ins.	91	97	130
Accuracy	80.5	94.2	92.2

4 CONCLUSIONS

In this work, we develop a potential URL attack detection system based on PU learning. Compared to supervised learning based approaches, our method only needs a handful of malicious URLs, along with the unlabeled URLs, which is suitable for the real situation that we encounter.

The developed system mainly contains three parts: firstly, a feature extraction process is executed to transfer the original URLs into numerical feature vectors; Secondly, two-stage strategy and cost-sensitive strategy are employed to train the classification models; Finally, each new-coming URL will be first transformed into numerical feature vector and then be fed into the learned models, those URLs with high scores will be regarded as potential malicious URLs with high probability.

Empirical results show that our developed system can effectively discover potential URL attacks. This system can either be deployed as an assistance for the existing system or be employed to help cyber-security engineers to effectively discover potential attack mode.

ACKNOWLEDGMENTS

Partially supported by NSFC (61333014) and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

[1] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-Supervised Learning. *IEEE Transactions on Neural Networks* 20, 3 (2009), 542–542.
[2] Marthinus C du Plessis, Gang Niu, and Masashi Sugiyama. 2014. Analysis of Learning from Positive and Unlabeled Data. In *Advances in Neural Information Processing Systems 27*. 703–711.
[3] Charles Elkan and Keith Noto. 2008. Learning Classifiers from Only Positive and Unlabeled Data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 213–220.
[4] Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S Yu. 2003. Building Text Classifiers Using Positive and Unlabeled Examples. In *Proceeding of the 3rd IEEE International Conference on Data Mining*. 179–186.
[5] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *Proceeding ot the 8th IEEE International Conference on Data Mining*. 413–422.
[6] Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker. 2009. 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*. 1245–1254.
[7] Zhi-Hua Zhou and Ming Li. 2010. Semi-Supervised Learning by Disagreement. *Knowledge and Information Systems* 24, 3 (2010), 415–439.