A Prior-based Transfer Learning Method for the Phishing Detection

Jianyi Zhang[1,2,3], Yangxi Ou[2,3], Dan Li[2,3], Yang Xin[2,3]
[1]Beijing Electronic Science and Technology Institute, Beijing, China
[2]Information Security Center, Beijing University of Posts and Telecommunications, Beijing, China
[3]Beijing Safe-Code Technology Co.,Ltd
Email: nese@ bupt.edu.cn, ouyangxi@ bupt.edu.cn, bylolo@ bupt.edu.cn, yangxin@safe-code.com

***Abstract*—In this paper, we introduce a prior-based transfer learning method for our statistical machine learning classifier which based on the logistic regression to detect the phishing sites that relies on our selected features of the URLs. Because of the mismatched distributions of the features in different phishing domains, we employ multiple models for different regions. Since it is impossible for us to collect enough data from a new region to rebuild the detection model, we adjust the existing models by the transfer learning algorithm to solve these problems. The proposed algorithm was evaluated on a real-world task of detecting the phishing websites. After a number of experiments, our proposed transfer learning algorithm achieves more than 97% accuracy. The result demonstrates the use of this algorithm in the anti-phishing scenario is feasible and ready for our large scale detection engine.**

***Index Terms*—network security, phishing detection, transfer learning, model transfer, logistic regression**

I. INTRODUCTION

Phishing is a significant form of criminally fraudulent attempt in electronic communication. It generally masquerades as a trusted website to obtain sensitive or financial information such as usernames, passwords and credit card details [1]. Despite the difference between the particular ways of dissemination, the common aim is the same, which is to deceive the victim into clicking the phishing Uniform Resource Locators (URLs). After the recipient posts his sensitive information to the phishing website where the URL points to, the attacker can utilize the acquired data to steal the victim's money or the personal reputation and identities.

In order to help the users avoid the phishing attacks, many software vendors, research institutions and security companies have released a variety of defense mechanisms. Clearly, the blacklist-based method is by far the most popular technique. After the phishing URLs were labeled by human-verification [2] or by heuristic analysis based on features or keywords [3, 4], the vendors distribute their results in a form of blacklist to the terminals like web browsers to achieve a high-efficiency blocking mechanism. However, since registering a new domain has become easier, no thorough blacklist can keep a perfect up-to-date database. It is slow in responding to newly emerged phishing attacks [5] [6]. Furthermore, some strategies are based on page content inspection to overcome the false negative problems and complement the vulnerability of the nonfresh lists. Although, heuristics approach may detect unlisted phishing sites as soon as they are launched and without any need to wait for the update of the phishing database, it may be bypassed by the attackers through a specially designed webpage and may also introduce a higher false alarm.

We have already designed an effective approach against phishing scams for detecting the target website completely based on the URL itself and without checking the associated web content and currently use this method as a part of our phishing detection system. However, after a period of operation, we found the characteristics of the phishing URLs have distinct feature distributions and the phishing scenarios are totally different between the regions like English and non-English countries since the processing data of our system are worldwide, APWG also pointed out these problems [7].

In this paper, we employ a prior-based transfer learning method for our phishing detection engines to solve this mismatched problem. That is, after utilizing a statistical machine learning algorithm to train the classification model from an existing data sources, we adjust the trained classifier and deploy the adaptive models to their corresponding regions according to the transfer learning. Thereafter, we construct a series of comprehensive experiments to test our proposed method. The results demonstrate that the transfer method we proposed outperforms the traditional machine learning algorithm in our anti-phishing scenario.

This paper makes three significant contributions:

- We present an analysis of the feature distributions between different regions and first propose this mismatched phenomenon in the detection scenario.
- We introduce our multi-scenario classification task and the detection strategy that fully considers these differences.
- We conduct a series of experiments to demonstrate that, according to an existing model in a familiar domain and limited data samples, our transfer learning algorithm can be utilized to generate a new model for a different phishing detection scenario.

The rest of this paper is structured as follows: Section II describes the relate works. Section III presents the background of our research. In the following section, Section IV, the methodology and the algorithms of our

doi:10.4304/jnw.7.8.1201-1207

classifier are proposed. Section V presents the experimental work and results. And then we present our conclusions in Section VI.

II. Related Work

Generally speaking, the most popular anti-phishing technique deployed by security vendors is the filtering method based on a blacklist. This mechanism is used from web browser toolbars to integrated browser phishing protection. In the past few years, the third party browser toolbars effectively help users avoid phishing attacks, like the Netcraft Toolbar [8] and the Cloudmark DesktopOne [9]. Besides, the famous antivirus software companies, like Symantec [10] and McAfee [11], proposed their anti-phishing solutions as a part of their products or deployed as a browser plug-in to protect their users. Currently, the popular browsers, like Firefox and Chrome [12], Internet Explorer [13] or Safari [14], have developed similar functions. They may use the public phishing blacklist sources like PhishTank [2], Google Safe Browsing API[15] or their proprietary repository. Although it may maintain a higher level of accuracy, the scalability and timeliness of these blacklist-based methods are not satisfactory.

To address these problems, the heuristic methods are necessary. Moreover, since the blacklist is compiled by a pack of known phishing URLs, the vendors have to utilize the heuristic mechanism to detect and mark the new phishing URLs for the compilation. Researchers extracted the DOM sketch, links attribution and irregular HTML codes as the pages' characteristics to detect the phishing webpages. Chou et al. proposed a well-known academic solution named SpoofGuard [3]. As a browser plug-in, SpoofGuard extracts domains, URLs and images hashes to evaluate similarity between the target page and the cached pages or history records. Also, many other systems continue to extract and use the similar features which demonstrate the SpoofGuard has a deep influence on the design of the anti-phishing system. CANTINA, proposed by Zhang et al., is another solution in literature. Based on eight features extracted from the pages' content, it utilizes a linear classifier to predict the suspicious websites. Some of these features, like age of domain or suspicious links, are the common characteristics against spoofing which proposed by SpoofGuard. In addition, CANTINA introduced a TF-IDF heuristic and followed the robust hyperlinks' idea as their innovations. Their system achieves an average TP of 89% and FP of 1% on 100 phishing URLs and 100 legitimate ones [16]. Xiang et al. introduced an approach to augmented CANTINA's keywords-retrieval method. They employed an identity-based detection component before the CANTINA's keywords-retrieval and evaluated in a larger test corpus that contains 11449 pages [17]. A noteworthy detection mechanism in active use is the system proposed by Google. Whittaker et al. built the Google's large-scale phishing classifier to maintain the Google's blacklist automatically. Through a logistic regression algorithm, they trained their classifier on a noisy dataset and based on 10 features including URL, hosting and page contents. After the offline training process using features collected from 103,684 phishing URLs during 3 months, their classifier maintains a false positive rate below 0.03% and the detected phishing URLs are widely used in many fields [4].

Being considered as a cost-effective solution, many researchers focus on the abnormal structure of the URLs that appeared in various phishing attacks. To the best of our knowledge, Garera et al. first introduced the idea of URL-based method in phishing detection. As an early prototype of the Google's automatic classifier, their method modeled a logistic regression classifier by 18 features which include obfuscation of the URL, keywords in the URL, Google's page rank, and Quality Scores based on the Google quality guidelines. Evaluated by 2,500 URLs, their classifier achieves an accuracy of 97.3% demonstrating that the URL analysis alone can be used to solve the phishing problem and achieve a high accuracy [18]. Ma et al. proposed a series of studies based on the analysis of URLs. Similar to the Garera's work, their earlier approach uses the same classification method and part of the same features. By expanding the analysis target, they examined the malicious websites, not just in the limit of in the phishing problem [19]. Thereafter, they introduced an online algorithm, confidence-weighted learning algorithm, for the URL classification which has the ability of training a larger sample than the batch methods with a high accuracy and efficiency [20, 21].

III. Background

This section describes the background of our detection application. Specifically, we first introduce the problems and the known limitations of current anti-phishing techniques proposed by the former researchers. Next, we provide an overview of the URL resolution and a brief introduction of the phishing URLs.

A. Problems

After analyzing these methods, we can find that some of the former researchers' methods are merely a prototype or an experimental system, some of them are academic studies that have not yet transformed to any systems in active use and some of them cannot be applied in a large-scale scenario. We then designed our detection system fully considered Garera [18] and Ma's [19] system since the performance of their methodology is very satisfactory in terms of time complexity and system accuracy. However, after a period of operation, we notice that some of these features have obvious geographical features and yet are not suitable for the design goals of our system. That means, according to our phishing repository, we found the distribution of features is quite different between regions. The phishing issues are very serious both in China and America but the URLs' types are distinct. For instance, the Chinese phishers' first choice is to register a new domain while the Americans' prefer to deploy the phishing website in a hacked website. The lasted APWG's report also pointed out this difference [7]. We hence need to take these diversities into consideration when we design a detection model.

The research [21] introduced the Confidence-Weighted learning algorithm to solve the large datasets training problem and how to adapt more quickly to new features. The research [22] proposed a novel application of a clustering technique to use robust and stable consensus functions in profiling of phishing websites. A rank correlation is used to select the features for dimensionality reduction. And they trained fast supervised classification algorithms on the resulting consensus clustering to enable them to process the large data set as well as new data. However, their methods do not focus on the differences between the regions and cannot solve the problem caused by the sparse samples. Moreover, it is often accompanied by a rapidly increasing feature vectors.

B. Overview of URLs

The Uniform Resource Locator (URL) is a global address of a website or a series of resources on the Internet. As it is the basic expression way of the names and addresses refer to the objects on the World Wide Web, phish sites also need a URL to locate themselves. No matter what types the attackers prefer, they all require the recipients to click the links that contain the URLs of the phishing sites. Hence, before stating the details of our proposed algorithm and system, we first list some URLs to demonstrate the real phishing scenario.

TABLE I. EXAMPLES OF PHISHING URLS

Target	URL
Taobao	http://item.taobao.com-eis.tk /member/login.jhtml.asp? [PARAMETERS]
Paypal	http://account.member.paypal.adabim.net/paypal /webscr/loading.php? [PARAMETERS]
Chase	http://hitechsense.com/images/www.chase.com /update.php? [PARAMETERS]

As shown in the Table I, we select three typical phishing URLs from our repository. Following the standard URL syntax, we can find that the adversary wants to obfuscate the first phishing URL by the *item.taobao.com* but actually the primary domain of this URL is *com-eis*. And this obfuscated method is the most common way the attacker employs. The same situation occurs in the second phishing URL which the *account.member.paypal* is the perplexed one. As mentioned before, the adversary who attacks American targets prefers to use a hacked web server which displayed in the third URL. Generally, the *images* folder of a website contains the picture sources of the entire site. However, this URL contains an abnormal website directory since there is a subfolder named *www.chase.com* and includes some php files in the *images* folder. In contrast, the Chinese phishers are more likely to use new register domain names for their attacks like the first URL. Although the cost of this method is a little higher than the hacked way, it is more confusing since the URLs' structure, take the first case as an example, is closer to the real one which usually displayed as *http://item.taobao.com/[path]*.

IV. CLASSIFICATION MODEL AND TRANSFER LEARNING ALGORITHM

In this section, we articulate our design goals and solutions to the problems that mentioned above. First, the theories of the classifying and the corresponding training process are proposed. Then, we detail the steps of how to transfer an existing model to the target scenario.

A. Design Goals

Considering our global detection tasks, we should propose a novel method to modify our existing anti-phishing engine. This approach must take into account the differences of the phishing scenario among the regions. These diversities not only reflect the characteristics of attack styles among these regions, but also greatly influence the accuracy of the classifier during detecting the suspicious websites. Therefore, we should take full advantage of these differences during the detection.

B. Classification Model

Our existent engine choose the logistic regression classification technique [23] to make the final decision on whether the site in question is a phishing. That is, according to the analysis result of the phishing URLs' features distribution and in order to keep the low latency during processing the URLs, we utilized our proprietary extraction method to inhibit the number of features. From its original utilization in epidemiologic study, logistic regression has now been widely used in many fields. As an efficient statistical model for binary data prediction, it is suitable for the phishing detection task, classified the target URLs as benign (represented by 0 or negative) or malicious ones (represented by 1 or positive) by a set of independent variables.

The input of the logistic regression is any value from negative infinity to positive infinity, whereas the output, expressed as a probability, is scaled to values from 0 to 1. As a generalized linear classification method, logistic regression hence needs to transform the score in logs of the odds, the *logit*, to the final output. That is,

$$odds = \frac{P}{1-P}$$

$$\log it = \log(odds) = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_n x_n$$

where ω_0 is called the "intercept" and $\omega_1, \omega_2, \ldots, \omega_n$ are the real-valued "regression coefficients" of the independent variables $x_1, x_2, \ldots, x_n$ respectively. Here, these independent variables are the features that we mentioned previously. The final score of the binomial probabilities is described as:

$$score = P(y=1|x) = \frac{e^{\log it}}{1+e^{\log it}} = \frac{1}{1+\exp\left\{-(\omega_0 + \sum_{i=1}^{n} \omega_i x_i)\right\}}$$

where $P(y=-1|x) = 1 - P(y=1|x)$. In our scenario, this score is the probability of whether the URL in question is a phishing URL. After setting a threshold, if the score is

greater than the value, the classifier labels the URL with a phishing mark and sends the result to the follow-up system.

C. Training and Knowledge Transfer

In the training phase, the regression coefficients ω can be obtained in terms of resolving the following unconstrained optimization problem for the L1-regularized logistic regression:

$$\min_{\omega} f(\omega) \equiv \sum_{i=1}^{n} |\omega_i| + C \sum_{j=1}^{n} \log\left[1 + \exp\left(-y_j \omega^T x_j\right)\right]$$

The parameter $C > 0$ is an adjustment number that specified by users for balancing the regularization term and the sum of losses. Besides, we assumed that the coefficients are fitted by the independent Gaussian prior and we modeled each weight, ω , with a Gaussian distribution: $\omega_i \sim N(\mu_i, \sigma_i^2)$. Generally, the prior mean is 0 and the variance is 1.

For the task of model transferring, we should treat the analysis and the processing among regions differently. As described before, we found the distributions of phishing URLs' characteristics are totally distinct among regions. And this problem should be considered during training the classification model.

Take the Chinese phishing scenario as an example. According to the Huawei Symantec anti-phishing lab's report [24], the attacks that target the Chinese websites have become increasingly serious. Since the most studies focus on the English regions and our existing system that in active use as well, we have to deploy our anti-phishing detection engine in China. However, simply utilizing the prior model that is trained by the datasets from other regions is inappropriate. Besides, the training data from the new region are not enough for us to rebuild a complete classification model.

In recent years, the idea of transfer learning has been widely used in many knowledge engineering areas like data mining and machine learning. Since the training and predicting datasets often do not have the same distribution in many real-world applications, the transfer learning has been utilized to solve these mismatched problems. For us, if our integrated detection engine adds a new data source node, we must adjust our classification model to fit the new distributions of these features since these URLs from different domains have distinct characteristics. However, in general, the labeled URLs in the target domain are rough and limited; it is impossible and expensive to recollect enough corresponding URLs to rebuild the models for the new scenarios. Hence, we should transfer the knowledge from our prior domain to the new task domain. That is we employ the transfer learning or knowledge transfer algorithm to improve the performance of the classifier by resolving the issues of expensive data labeling and model rebuilding.

As we mentioned before, the values of μ_i and σ_i^2 are set to 0 and 1 respectively since their each weight ω follows the Gaussian distribution $\omega \sim N(0,1)$. When we transfer the classification models, we change the value of μ and σ^2 to adjust the existing model. That is

$$\sum_{i=1}^{n} \left| \frac{\omega_{ti} - \mu_i}{\sigma_i} \right| + \sum_{j=1}^{n} \log P\left(y_j \mid x_j\right)$$

Specifically, we first fetch some sample datasets of the target domains. And according to some specific rules, like the date of the data or the origin vendors, we cluster them into a number of *K* different groups. After that, we generate the *K* independent logistic regression models by these grouped data and obtain the target domain's weights $\{\omega_{ti}^{k}\}$, $k = 1,...,K$ respectively (where the subscript *i* stands for the number of the features and the superscript *k* indicates the different groups). From these, we calculate the mean value and the standard deviation of *k* groups to estimate the tendency of the weights that belong to the target domain. After that, we perform this tendency to influence our original domains' model by the following:

$$\omega_{ti} = \omega_{oi} \bullet \sigma + \mu$$

where the ω_{oi} means the original model's weight. Then, we get the transferred model and adjust these parameters in the classifying application by new labeled datasets.

V. Evaluation

In this section, we conduct the experiments on two datasets that come from different regions to demonstrate the feasibility of the transfer learning in anti-phishing applications. Since there are often some irregular HTTP streams and the extracted URLs may be improperly constructed, hence, we need to omit these worthless URLs from our dataset. After purifying the evaluation corpora, we build these data on two ratios of negative-to-positive to simulate the real phishing scene and then perform our test.

Evaluation of Datasets

In our evaluation, we perform the experiments on two positive (phishing) corpora. The first one, which we call A-corpus, is a copy of Huawei Symantec's phishing repository containing data collected from a detection node in North America between May 1, 2011 and May 7, 2011. We use their data to train the classifier to simulate our familiar anti-phishing scenario. The second corpus which we intend to illustrate the transfer performance of our proposed method contains data collected from one of Huawei Symantec's Chinese anti-phishing node between May 18, 2011 and May 31, 2011. We call this C-corpus. The negative dataset, which we call N-corpus, is from less popular legitimate websites. We deploy an independent spider to collect part of the data and draw others from Yahoo's directory listing by visiting the link http://random.yahoo.com/bin/ryl. Table II details the statistics of these datasets.

TABLE II. DATASET STATISTICS

Type	Total	Valid
Positive URLs	37116	36560
A-corpus[a]	14920	14793
C-corpus[b]	22196	21767
Negative URLs	191949	140857
N-corpus	191949	140857
Total URLs	247462	193175

a. From Huawei Symantec in North America between May 1, 2011 and May 7, 2011
b. From Huawei Symantec in China between May 18 and May 31, 2011

The ratio of negative-to-positive

The ratio of legitimate-to-phishing training data will influence the outcomes greatly [25]. Different studies construct their experiments on different proportions of positive samples (phishing) in their training or evaluation datasets [4, 21]. Experts from the Messaging Anti-Abuse Working Group (MAAWG) estimate the ratio of authentic site to the phishing ones as anywhere from 100:1 to 1000:1. In our detection applications, this ratio is more than 5000:1. That is because the research of MAAWG focuses on the spam and our data are mainly collected from the network traffic. However, if the ratio of benign-to-phishing URLs is 1000:1, it is clear that even without a machine learning algorithm, we can reach an accuracy of 99.9% if we directly predict that all samples were negative. Therefore, in order to evaluate the performance of our classifier, we not only construct the experiments by varying the ratio of negative-to-positive samples used to train the classifier, also gain more insight by examining the True Positive Rate and True Negative Rate.

To make the experiments more comprehensive, we introduce the different ratios of negative-to-positive to test our classifier. In particular, we experiment on the classifier by a 10:1 and 100:1 ratio of legitimate-to-phishing URLs since these two ratios are very close to our anti-phishing scenario. Our integrated detection system needs to process massive URLs with a low latency and the most of the URLs in the network traffic are legitimate ones, we, hence, must deploy a whitelist to limit the proportion of the benign URLs in our trained and predicted data. Generally, we limit the range of the ratio from 10:1 to 100:1. In this experiment, we compile two evaluation datasets with a 10:1 and 100:1 ratio of negative-to-positive respectively, and name them as Dataset I and Dataset II. In order to simulate our real anti-phishing scene, we randomly sample the required URLs from the A-corpus and the N-corpus to compile the Dataset I with a 10:1 ratio and the Dataset II with a 100:1 ratio.

ClassifierFeatures

Adversaries usually build their phishing URLs in a common way to masquerade as a trustworthy entity, hence, it is feasible to determine a web site is a phishing or not simply by these common features. We extract millions of string tokens from our labeled URLs and analyze them to discovery their inherent relations among these URLs. Some previous studies, like [26], have already proposed their findings, and we fully considered these suggestions during our work. Additionally, since our data feeds are global and not limited in the spam collections, we also find some characteristics that can be used to distinguish the malicious websites from the authorized ones. According to the types of these features that we extracted from our URL repository, we group and categorize them as follows:

TABLE III. DATASET STATISTICS

	Name	Types
General Features	Length of URLs	float
	Length of Hostnames	float
	Segments of URLs	float
	Segments of Hostnames	float
	Hexadecimal Number in URLs	boolean
	IP Address in URLs	boolean
	Unusual Top-Level Domains	boolean
	Dots in Path	boolean
	Numerical Primary Domain	boolean
Hosted Features	Virtual Web Hosting	boolean
	Signs of Being Hacked	boolean
	URL-Aliasing Services	boolean
Lexical Features	Phishing Sensitive Words	float
	Brand Words	float
	Long Words	boolean

Transfer Performance

For the transfer performance experiment, two corpora, Dataset I and Dataset II, are utilized as the datasets of the familiar domain to simulate the real anti-phishing scenario. Since our original models are trained on the data with a legitimate-to-phishing ratio between 10:1 and 100:1, we compare the transferred model with these two previous models respectively. Besides, we compile the evaluation data by C-corpus and N-corpus as the target domain dataset. We use random sampling method to select the records from the target dataset and create 10 subsampled datasets as the prior knowledge of the target domain. After the training process on these 10 subsampled datasets is completed, we transfer our original models by adjusting the regression coefficients according to these 10 trained models. First, we calculate the values of the mean and the standard deviation of each regression coefficients trained on these 10 subsampled datasets. Next, we transfer the weights by the algorithm mentioned before and generate the new models. Here the threshold score is set to 0.5.

TABLE IV. TRANSFERRED MODEL FROM THE EVOLUTION DATA OF 10:1 RATIO OF N-TO-P

Classifier	Recall	Specificity	Accuracy	F-measure
Model A	76.6619%	99.2347%	96.2134%	84.4227%
Model B	86.8838%	99.5918%	97.8908%	91.6857%

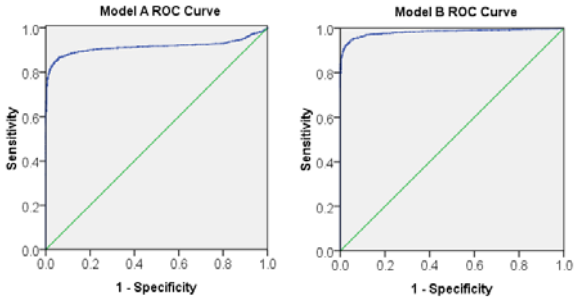

Figure 1. The ROC Curve of Model A and Model B

Table IV and Table V describe the results of the transfer performance. In Table IV, Model A is the original model that trained on the Dataset I and the Model B is the corresponding transferred model. In order to undertake a comprehensive experiment, we first directly predict the target domain dataset by Model A then following by Model B. We then use the same steps in the test of the Table V. That is, Model C for Dataset II and Model D is the adjusted model from Model C. We also add the corresponding ROC curves to demonstrate their performance that showed in Fig. 1 and Fig.2. According to these results, we can find that the transferred model made the performance of the classifier improve greatly.

TABLE V. TRANSFERRED MODEL FROM THE EVOLUTION DATA OF 100:1 RATIO OF N-TO-P

Corpus	Recall	Specificity	Accuracy	F-measure
Model C	57.3207%	99.8715%	94.1761%	72.4880%
Model D	85.5745%	99.7182%	97.825%	91.3290%

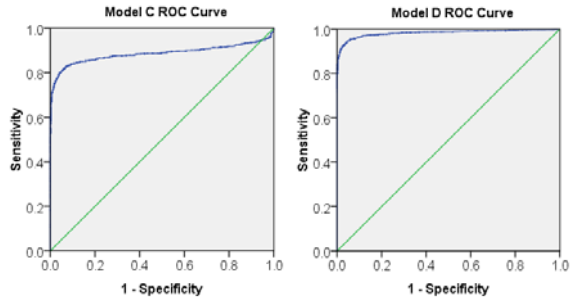

Figure 2. The ROC Curve of Model C and Model D

Discussion of the Results

Through these experiments, the results demonstrate that our proposed method has already achieved the goals of the phishing detection task. According to Table IV and V, the classifier that is based on the L1-logistic regression and transfer learning algorithm maintains a high accuracy with different evaluation datasets and different ratios of negative-to-positive. And the classifier achieves a high accuracy which more than 97%. Since this detection strategy is a part of our integrated detection engine and there is a secondary verification mechanism in the follow-up system to recheck the outcomes and mark the URL with a brand, the total accuracy is much higher than the previous works like [4, 21]. Besides, the ROC curves of the classifier on different corpus and different ratios (Fig. 1 and 2) suggest that our proposed model transfer method allows us to generate an effective model for the unfamiliar scenario by using a limited dataset. In our experiments, the datasets for the evaluations are subsampled randomly to simulate the different groups that are mentioned before. Since these datasets are all collected from the real phishing data and these groups in practice are different from each other, so the experiment results can reflect actual anti-phishing scenario.

VI. CONCLUSION

With the rapid growth of phishing attacks, there is a need for a real-time, non-language related phishing detection mechanism to complete the multinational anti-phishing task. Our URL-based method in phishing detection is an appropriate solution. To fulfill the detection requirements of features' mismatched, we proposed a transfer learning method to generate an adaptive model for the new detection scenario since it is impossible to train a new model with a limited training sample. The adaptive model also performs well in phishing detection abilities in the target domain with limited sample datasets.

ACKNOWLEDGMENT

We are grateful to the Huawei Symantec Technologies for providing us with kind help. We cooperated with Huawei Symantec Technologies Co.,Ltd. Anti-phishing Lab. This work is supported by National Natural Science Foundation of China (No. 61121061), National S&T Major Program (2011ZX03002-005-01). We thank the anonymous reviewers for their comments on this work.

REFERENCES

[1] Wikipedia Phishing. http://en.wikipedia.org/wiki/Phishing, June 4, 2011
[2] OpenDNS Phishtank. http://www.phishtank.com/, 2011
[3] Chou, N., et al. "Client-side defense against web-based identity theft", in *Proceedings of the 11th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, Citeseer, 2004
[4] Whittaker, C., B. Ryner, and M. Nazif. "Large-scale automatic classification of phishing pages", in *Proceedings of the 17th Annual Network and Distributed Security Symposium (NDSS)*, Citeseer, 2010
[5] Cranor, L., et al. "Phinding phish: An evaluation of anti-phishing toolbars", in *Proceedings of the 14th Annual Network and Distributed System Security Symposium (NDSS)*, Citeseer, 2007
[6] Sheng, S., et al. "An Empirical Analysis of Phishing Blacklists", in *Proceedings of the 6th Conference on*

Email and AntiSpam (CEAS), Mountain View, California USA, 2009
[7] AARON, G. and R. RASMUSSEN Global Phishing Survey 2H2010. http://apwg.org/reports/APWG_GlobalPhishingSurvey_2H2010.pdf, 2011
[8] Netcraft Netcraft Anti-Phishing Toolbar. http://toolbar.netcraft.com, 2010
[9] Cloudmark Cloudmark DesktopOne. http://www.cloudmarkdesktop.com, 2011
[10] Symantec Norton 360-The Best Firewall-Protect agains viruses and phishing. http://us.norton.com/360/, 2011
[11] McAfee SiteAdvisor Software. http://www.siteadvisor.com/, 2011
[12] Google Google Safe Browsing for Firefox. http://www.google.com/tools/firefox/safebrowsing/, 2011
[13] Microsoft Phishing Filter: Help protect yourself from online scams. http://www.microsoft.com/uk/athome/security/online/phishing_filter.mspx, Oct 28, 2006
[14] Apple Safari 5 Features. http://www.apple.com/safari/features.html#security, 2011
[15] Google Google Safe Browsing API Developer's Guide (v2). http://code.google.com/intl/zh-CN/apis/safebrowsing/developers_guide_v2.html, 2009
[16] Zhang, Y., J. Hong, and L. Cranor. "CANTINA: A Content-Based Approach to Detecting Phishing Web Sites", in *Proceedings of the 16th International World Wide Web Conference*, Banff, Alberta, Canada, ACM, pp.639-648, 2007
[17] Xiang, G. and J.I. Hong. "A Hybrid Phish Detection Approach by Identity Discovery and Keywords Retrieval", in *Proceedings of the 18th International World Wide Web Conference*, Madrid, Spain, ACM, pp.571-580, 2009
[18] Garera, S., et al. "A Framework for Detection and Measurement of Phishing Attacks", in *Proceedings of the 2007 ACM Workshop On Recurring Malcode*, Alexandria, Virginia, USA, ACM, pp.1-8, 2007
[19] Ma, J., et al. "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*, Paris, France, ACM, pp.1245-1254, 2009
[20] Ma, J., et al. "Identifying Suspicious URLs: An Application of Large-Scale Online Learning", in *Proceedings of the 26th Annual International Conference on Machine Learning*, Montreal, Quebec, Canada, ACM, pp.681-688, 2009
[21] Ma, J., et al., "Learning to detect malicious URLs", *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, pp. 30, 2011
[22] Beliakov, G., J. Yearwood, and A. Kelarev, "Application of Rank Correlation, Clustering and Classification in Information Security", *Journal of Networks*, vol. 7, pp. 935-945, 2012
[23] Hosmer, D.W. and S. Lemeshow, Applied Logistic Regression, Second Edition. Vol. 354. 2000: John Wiley and Sons.
[24] HuaweiSymantec Huawei Symantec Phish Report. http://www.huaweisymantec.com/cn//download.do?id=661203, May 2011
[25] Zadrozny, B., J. Langford, and N. Abe. "Cost-Sensitive Learning by Cost-Proportionate Example Weighting", in *Proceedings of the 3rd IEEE International Conference on Data Mining*, IEEE Computer Society, pp.435, 2003
[26] McGrath, D.K. and M. Gupta. "Behind Phishing: An Examination of Phisher Modi Operandi", in *Proceedings of the 1st Workshop on Large-Scale Exploits and Emergent Threats*, USENIX Association, pp.1-8, 2008

Jianyi Zhang is a Ph.D. candidate in the Information Security Center of the State Key Laboratory of Networking and Switching Technology, National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, with research projects also in the Huawei Symantec Technologies Anti-phishing research Lab. His graduated and has been a researcher in the Beijing Electronic Science and Technology Institute. His current research is in the information security especially in the protection of privacy. Email: nese@bupt.edu.cn