A Proposal of the AdaBoost-Based Detection of Phishing Sites

Daisuke Miyamoto, Hiroaki Hazeyama, and Youki Kadobayashi

Internet Engineering Laboratory
Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, JAPAN
{daisu-mi, hiroa-ha, youki-k}@is.naist.jp

Abstract. In this paper, we propose an approach which improves the accuracy of detecting phishing sites by employing the AdaBoost algorithm. Although there are heuristics to detect phishing sites, existing anti-phishing tools still do not achieve high accuracy in detection. We hypothesize that the inaccuracy is caused by anti-phishing tools that can not use these heuristics appropriately. Our attempt is to improve accuracy by applying the AdaBoost algorithm, the most typical of the machine learning algorithms. We also evaluate the AdaBoost-based combination method by comparing with CANTINA [1], and almost of all the results show that our proposed method provides higher accuracy to detect phishing sites.

1 Introduction

Phishing is a scam in order to deceive end users in various ways [2, 3] into disclosing their personal information. Recently, the number of phishing attacks has grown rapidly. According to trend reports published by the Anti-Phishing Working Group [4], the number of reported phishing attacks was 37,439 in November 2006, for surpassing the 6,957 in October 2004. Moreover, Gartner's survey reported [5] that 120 million consumers lost $929 million due to phishing. Generally, a phishing attack is composed of two phases: attraction and acquisition. For example, *Email spoofing* [6] attracts users by a "spoofed" email, made to appear as if sent by a legitimate corporation. To acquire users' personal information, the spoofed email leads users to visit a "spoofed" web site, a so-called "phishing site."

To deal with phishing attacks, developing sophisticated algorithms for detecting phishing sites is necessary. Essentially, the detection algorithms are categorized into two distinct methods. On hand is the URL filtering method. It detects phishing sites by comparing a URL with a URL blacklist, which is composed of the URLs of phishing sites. However, the effectiveness of URL filtering is limited. Registering every phishing site to a list is difficult because phishing sites are rapidly created. Instead of using a URL blacklist, a URL whitelist-based detection method was proposed [7]. When the URL of the site is not registered on the URL whitelist, the site will be marked as a phishing site. A URL whitelist is composed of URLs of legitimate sites and is able to detect phishing sites because URLs of phishing sites cannot be registered on the whitelist. However, it is extremely difficult to register large numerous numbers of legitimate sites.

On the other hand, it is feasible to calculate the likelihood of a site being a phishing site. For example, SpoofGuard [8] checks whether the age of a site is short or not. If short, SpoofGuard deems it a phishing site, since the age of these sites tends to be short. However, the accuracy of detection by anti-phishing tools is unsatisfactory, even there are various research contributions to characterize phishing sites [2, 3, 9–12].

We hypothesize that this inaccuracy is caused by anti-phishing tools that can not use these heuristics appropriately, so we explore a suitable way to combine them. To the best of our knowledge, the most successful tool which combines heuristics is CANTINA, which has achieved high accuracy of detection without using a URL filtering method. According to [1], the true positive rate of CANTINA is 89% and the false positive rate is only 1%.

In CANTINA, the likelihood of a phishing site is calculated from weighted majority by using eight heuristics that are described in Section 3: *Age of Domain*, *Known Images*, *Suspicious URL*, *Suspicious Links*, *IP Address*, *Dots in URL*, *Forms*, and *TF-IDF-Final* heuristic.

We attempt to improve this accuracy of detection by employing a boosting algorithm for weight assignment. Boosting has its roots in a theoretical framework called the Probably Approximately Correct(PAC) learning model [13], used for studying machine learning. The key feature of boosting is that a “weak” algorithm, which performs just slightly better than random guessing, can be boosted into an accurate “strong” algorithm. AdaBoost, which is proposed by Freunde and Schapire [14], is the most typical boosting algorithm. AdaBoost solves many of the practical difficulties of the earlier boosting algorithms, and its ensembles perform better than the generic ensemble methods.

We evaluate the accuracy of the AdaBoost-based combination method in comparison to CANTINA's combination method. We also prepare two URL datasets, one is used for training, and the other is for testing. Next, we calculate the assigned weight of each heuristic in both cases with a training dataset, and investigate the accuracy by using a test dataset. The result shows that true positive rate of detection increased from 92.0 %, in the case of CANTINA's combination method, to 94.0 % in the case of the AdaBoost-based combination method, and false negative rate decreased from 4.0 % to 0.0%. We also evaluate the accuracy of detection by changing two datasets and set of heuristics, and we find that almost of all evaluation results showed that AdaBoost is able to improve the accuracy to distinguish between phishing sites and legitimate sites.

The rest of this paper is organized as follows: In Section 2, we describe research related to phishing attacks; in Section 3, we explain CANTINA and its heuristics. In Section 4, we introduce the AdaBoost algorithm, and we evaluate the accuracy of detecting phishing sites in Section 5. We also discuss the limitations of our proposed method in Section 6, and we conclude our contributions in Section 7.

2 Related Work

There are many research contributions to model phishing attacks [2, 3, 9–11]. Generally, a phishing attack is separated into two distinct phases, attraction and acquisition. In many cases, phishers attract users by *email spoofing* [6], which leads users to access a phishing site. An acquisition trick which employs a phishing site is called *web spoofing* [15].

While technical means for Web spoofing are available, so far most web spoofing is caused simply by misbehavior and carelessness of users. In the case of web spoofing, a spoofed email convinces users to access a URL leading to a phishing site. The phishing site is designed as a look-alike of the targeted legitimate site. Users are likely to disclose personal information to the phishing site without verifying the URL or an SSL certification.

Regarding technical issues, although several countermeasures against phishing attacks have been studied and carried out for both users and businesses, sufficient countermeasures have not been proposed yet [2, 3].

3 CANTINA

In this section, we explain CANTINA, its heuristics, and the combination method of heuristics. We also explain our implementation of CANTINA.

To the best of our knowledge, CANTINA is the most successful tool which combines heuristics. CANTINA has achieved high accuracy of detecting phishing sites without using the URL blacklist and/or whitelist. According to [1], the true positive rate of CANTINA is 89% and false positive rate is only 1%.

In CANTINA, the likelihood of a phishing site is calculated from eight heuristics as follows.

- Age of Domain
  Checking whether or not the domain was registered more than 12 months ago. If the site has been registered more than 12 months, the heuristic will return +1, deeming it as a legitimate site, and otherwise it returns -1, deeming it as a phishing site.
- Known Images
  Checking whether or not a page contains inconsistent well-known logos such as eBay, PayPal, Citibank, Bank of America, Fifth Third Bank, Barclays Bank, ANZ Bank, Chase Bank, and Wells Fargo Bank. For example, if a site contains the eBay logos but is not on an eBay domain, CANTINA deems this site as a phishing site.
- Suspicious URL
  Checking whether or not a URL of the site contains an “at” symbol (@) or a “dash” (-) in the domain name.

- Suspicious Links
  Similar to the Suspicious URL heuristic, checking whether or not a link on the page contains an “at” symbol or a dash.
- IP Address
  Checking whether or not the domain name of the site is an IP Address.
- Dots in URL
  Checking whether or not the URL of the site contains five or more dots.
- Forms
  Checking whether or not the page contains any web input forms.
- TF-IDF-Final
  Checking whether or not the site is phishing by employing TF-IDF-Final which is the extension of Robust Hyperlinks algorithm [16].

Each heuristic returns boolean value; if a heuristic deems a page as a phishing site, it will return -1; and if a heuristic deems a page as a legitimate site, it will return +1.

Based on the result of each detection, CANTINA calculates the score (S) by weighted majority as shown in Formula 1.

$$S = \sum W_i * h_i \tag{1}$$

A positive value for the score means that it is labeled as legitimate, while a negative value or zero means that it is labeled as a phishing site.

In CANTINA, the weight assignment is important to achieve high accuracy of detection. Yue et al. considered that a heuristic should have high accuracy in detecting phishing sites while also having a low false positive rate. So, they attempted to assign weight by calculating true positives minus false positives. Given the effect e_i of each heuristic, they calculated each weight proportionally, that is:

$$W_i = \frac{e_i}{\sum e_i} \tag{2}$$

Because CANTINA is not available to download yet, we implemented five of eight heuristics that are Suspicious URL, Suspicious Links, IP Address, Dots in URL, and Forms heuristic all of which only analyze the downloaded content or the URL of the site.

We also implemented Age of Domain and TF-IDF-Final heuristic, however, some of their functions were not implemented. In the case of Age of Domain heuristic, the format of the WHOIS server differs in different countries, so our implementation derives the domain name from a URL and shows us the search result of WHOIS. In the case of TF-IDF-Final heuristic, simply analyzing the downloaded content would not always work because some phishing sites use JavaScript to generate phishing sites dynamically. Hence, we browsed the target URL with Firefox 2.0, inputted all text content to our implementation by copy-and-paste from the browser screen.

In the case of Known Images heuristic, it is difficult to judge whether or not the well-known logo is used without browser’s rendering support. Thus, we manually performed the checking of Known Images heuristic, and labeled as phishing if the site used well-known logos.

4 AdaBoost

The standard AdaBoost algorithm learns a “strong” algorithm by combining a set of ”weak” algorithms h_t and a set of weight α_t:

$$H_{Ada} = \sum \alpha_t * h_t \tag{3}$$

It is similar to Formula 1, but, the algorithm for weight assignment is different. The weight are learned through supervised training off-line [17]. Formally, AdaBoost uses a set of input data $\{x_i, y_i : i = 1, \ldots, m\}$ where x_i is the input. And, y_i is the classification where $y_i = -1$ indicates a phishing site and $y_i = 1$ indicates a legitimate site. Each weak algorithm is only required to make the correct detections slightly over half the time. The AdaBoost algorithm iterates the calculation of a set of weight $D_t(i)$ on the samples. At $t = 1$, the samples are equally weighted so $D_1(i) = 1/m$. The update rule consists of three stages. Firstly, AdaBoost chooses the weight as shown in Formula 4.

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right) \tag{4}$$

Where $\epsilon_t = Pr_{i \sim D_t} [h_t(x_i) \neq y_i]$. Second, AdaBoost updates the weight by Formula 5.

$$D_{t+1} = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & if \quad h_t(x_i) = y_i \\ e^{\alpha_t} & if \quad h_t(x_i) \neq y_i \end{cases} \tag{5}$$

Where Z_t is a normalization factor; $\sum_{i=i}^{m} D_{t+1}(i) = 1$. Finally, it outputs the final hypothesis as shown in Formula 3. In this paper, a positive value for H_{ADA} means that it is labeled as legitimate, while a negative value or zero means that it is labeled as a phishing site.

5 Evaluation

In this section, we investigate whether or not the AdaBoost-based combination method can improve the accuracy to detect phishing sites. We used three metrics to evaluate the accuracy; true positive rate, false positive rate, and total accuracy rate which is calculated by dividing the number of correctly identified sites by the number of all sites in the dataset. Next, we evaluate the validity of our two datasets, which contains 50% phishing sites and 50% legitimate sites. Finally, we also discuss the effect of Known Images heuristic, which has the highest accuracy among eight heuristics but was checked manually.

Table 1. Assigned Weight by CANTINA

CANTINA	True Positive Rate	False Positive Rate	Effect	Weight
Age of Domain	62.0%	8.0%	54	0.19
Known Images	88.0%	0.0%	88	0.31
Suspicious URL	8.0%	6.0%	2	0.01
Suspicious Link	6.0%	6.0%	0	0.00
IP Address	14.0%	0.0%	14	0.05
Dots in URL	10.0%	0.0%	10	0.03
Forms	88.0%	22.0%	66	0.23
TF-IDF-Final	98.0%	44.0%	54	0.19

Table 2. Assigned Weight by AdaBoost

AdaBoost	Total Accuracy Rate	α	Weight
Age of Domain	54.0%	0.03	0.01
Known Images	88.0%	1.38	0.42
Suspicious URL	2.0%	0.00	0.00
Suspicious Link	0.0%	0.00	0.00
IP Address	14.0%	0.23	0.07
Dots in URL	10.0%	0.14	0.04
Forms	66.0%	0.56	0.17
TF-IDF-Final	54.0%	0.95	0.29

Table 3. Accuracy of normal CANTINA and CANTINA with AdaBoost

Algorithm	True Positive Rate	False Positive Rate	Total Accuracy Rate
CANTINA	92.0%	4.0%	94.0 %
AdaBoost	94.0%	0.0%	97.0 %

5.1 Evacuation of Accuracy

First, we built a training dataset. We have chosen 50 phishing URLs from PhishTank.com in May, 2007 according to the following requirements: a phishing site which (i) can still be browsed (is not expired), (ii) looks like a well known legitimate site, and (iii) can be labeled as a phishing site by a URL of the site. Next, we have also selected 50 legitimate URLs, top 25 URLs of Alexa [18], 15 URLs listed as Good URLs in 3Sharp [19], 10 URLs chosen randomly from http://random.yahoo.com/fast/ryl, and all 100 URLs (50 phishing and 50 legitimate) were English language sites. Moreover, we have also built a test dataset which was composed of 50 phishing URLs from PhishTank.com and 50 legitimate URLs from 3Sharp. We note that the URLs of the test dataset were different from those of the training dataset.

Second, we applied each heuristic to the training dataset, and assigned the weight to each heuristic. The result of CANTINA's weight assignment is shown in Table 1. In this paper, true positive denotes labeling a phishing site as phishing, and false positive denotes labeling a legitimate site as phishing.

Third, we also assigned the weight to each heuristic in the case of employing an AdaBoost algorithm, and the result of weight assignment is shown in Table 2. In order to facilitate comparing the weight assignment of AdaBoost with that of CANTINA, the weight α_i was normalized by Formula 6, which has same meaning as Formula 2.

$$W_i = \frac{\alpha_i}{\sum \alpha_i} \tag{6}$$

In AdaBoost, heuristics should be applied into a training dataset in order of higher accuracy, so we measured the total accuracy rate on each heuristic. When there are same total accuracy rate among several heuristics, the heuristic with lower false positive rate was applied. In the case of our training dataset, heuristics were applied in order of Known Images, Forms, Age of Domain, TF-IDF-Final, IP Address, Dots in URL, Suspicious URL and Suspicious Links.

By comparing the weight assignment of AdaBoost with that of CANTINA, we found that Known Images heuristic was assigned the highest weight, and Suspicious Links heuristic was assigned zero weight in both cases. However, the assigned weight of Age of Domain heuristic was lower than that of TF-IDF-Final heuristic, otherwise normal CANTINA assigned higher weight to Age of Domain heuristic than TF-IDF-Final heuristic. This reversion was caused by that almost of all the sites which Age of Domain heuristic correctly labeled had been already identified correctly by Known Images heuristic. Conversely, TF-IDF-Final heuristic often labeled correctly where Known Images heuristic had mislabeled. Thus, AdaBoost assigned higher weight to TF-IDF-Final heuristic and lower weight to Age of Domain heuristic.

Finally, we applied both algorithms to our test dataset and the results(Table 3) showed that false positive rate deceased from 4.0% to 0.0%. This means that the AdaBoost-based combination method never labeled legitimate sites as phishing in this case. True positive rate increased from 92.0% to 94.0%, and the total accuracy rate increased from 94.0% to 97.0%. According to this result, it can be assumed that the AdaBoost-based weight assignment algorithm has more effective than CANTINA's algorithm to detect phishing sites.

5.2 Percentage of phishing sites in dataset

Essentially, the number of legitimate sites is much larger than that of phishing sites, whereas our dataset mentioned in Section 5 contained 50% phishing sites and 50% legitimate sites.

Here, we presented the verification result with five pairs of a training dataset and a test dataset. They were composed of (i) 50 phishing sites and 50 legitimate sites, (ii) 40 phishing sites and 50 legitimate sites, (iii) 30 phishing sites and 50 legitimate sites, (iv) 20 phishing sites and 50 legitimate sites, and (v) 10 phishing sites and 50 legitimate sites. In the case of (i), both the training dataset and the test dataset are the same as those we used in Section 5. In the cases of (ii), (iii), (iv) and (v), we have chosen the phishing sites by random sampling manner.

The result of weight assignment is shown in Table 4 and that of AdaBoost is shown in Table 5. Based on assigned weight, we measured the accuracy of

Table 4. Weight Assignment by CANTINA

	(i)	(ii)	(iii)	(iv)	(v)
Domain Age	0.19	0.17	0.19	0.17	0.18
Known Images	0.31	0.30	0.29	0.31	0.26
Suspicious URL	0.01	0.01	0.01	0.00	0.04
Suspicious Links	0.00	0.01	0.00	0.00	0.04
IP Address	0.05	0.05	0.05	0.05	0.06
Dots in URL	0.03	0.04	0.05	0.02	0.03
Forms	0.23	0.23	0.23	0.25	0.23
TF-IDF-Final	0.19	0.19	0.18	0.20	0.16

Table 5. Weight Assignment by AdaBoost

	(i)	(ii)	(iii)	(iv)	(v)
Domain Age	0.01	0.01	0.00	0.06	0.00
Known Images	0.42	0.41	0.41	0.51	0.34
Suspicious URL	0.00	0.00	0.03	0.00	0.10
Suspicious Links	0.00	0.00	0.00	0.00	0.00
IP Address	0.07	0.08	0.08	0.00	0.22
Dots in URL	0.04	0.04	0.18	0.00	0.00
Forms	0.17	0.17	0.14	0.16	0.25
TF-IDF-Final	0.29	0.28	0.25	0.28	0.10

CANTINA as shown in Table 6 and that of CANTINA with AdaBoost in Table 7. We found the accuracy of CANTINA with AdaBoost is as well or better than that of CANTINA in every case.

We also found that the weight assigned by AdaBoost was concentrated on particular heuristics, unlike that of normal CANTINA. Figure 1 showed that the variance of each heuristic in the case of AdaBoost is higher than the case of normal CANTINA.

Notably, Known Images heuristic of AdaBoost is always assigned higher weight than that of normal CANTINA. Within our dataset, we observed that Known Images heuristic showed the best accuracy among eight heuristics; The total accuracy rate of Known Images is 88.0% in the case of (i), and 96.7% in the case of (v). Thus, We assumed that the concentration of the weight on Known Image heuristic has affected the accuracy of distinguishing between legitimate sites and phishing sites.

5.3 Effect of Known Images heuristic

According to [1], the true positive rate of Known Images was 37.0%, while that of our experiments was 88.0%. We assumed the diffrence was caused by our manually checking whether or not the site contains well-known logos which are

Table 6. Accuracy of normal CANTINA

	(i)	(ii)	(iii)	(iv)	(v)
True Positive Rate	92.0%	90.0%	92.5%	95.0%	80.0%
False Positive Rate	4.0%	4.0%	4.0%	4.0%	4.0%
Total Accuracy Rate	94.0%	93.3%	94.4%	95.7%	93.3%

Table 7. Accuracy of CANTINA with AdaBoost

	(i)	(ii)	(iii)	(iv)	(v)
True Positive Rate	94.0%	90.0%	92.5%	95.0%	90.0%
False Positive Rate	0.0%	0.0%	0.0%	0.0%	0.0%
Total Accuracy Rate	97.0%	95.6%	96.7%	98.6%	98.3%

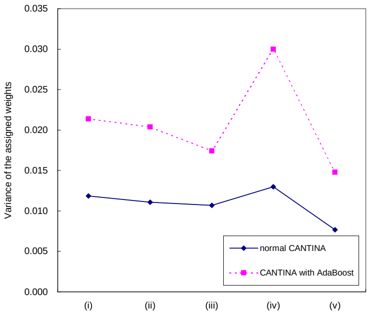

Fig. 1. Variance of Assigned Weight

listed in Section 3. If we implement Known Image heuristic, it is necessary to have the function of pattern matching in a digitized image, although this might result in many misjudgments.

In our previous experiments, AdaBoost assigned the highest weight to Known Images heuristic, so it can be assumed that the accuracy of the AdaBoost-based combination method depends heavily on the accuracy of Known Images heuristic.

In order to verify whether or not AdaBoost can build a “strong” learning algorithm even if the accuracy of Known Images decreases, we tested the accuracy of detecting phishing sites by combining heuristics which were removing Known logo heuristic and using only the other seven heuristics. We calculated the weight with the training dataset of (i), (ii), (iii), (iv) and (v), and measured the accuracy. The result is shown in Table 8 and 9.

In the case of (iii), we observed that the false positive rate of AdaBoost was rapidly decreased and the total accuracy rate of AdaBoost was lower than that of CANTINA. This was caused by overfitting, which we will discuss in Section 6.1.

Table 8. Accuracy of CANTINA's weight assignment

	(i)	(ii)	(iii)	(iv)	(v)
True Positive Rate	98.0%	95.0%	97.5%	95.0%	90.0%
False Positive Rate	28.0%	28.0%	28.0%	28.0%	20.0%
Total Accuracy Rate	85.0%	82.2%	83.3%	78.6%	81.7%

Table 9. Accuracy of the AdaBoost-based weight assignment

	(i)	(ii)	(iii)	(iv)	(v)
True Positive Rate	98.0%	95.0%	70.0%	65.0%	60.0%
False Positive Rate	20.0%	20.0%	10.0%	10.0%	10.0%
Total Accuracy Rate	89.0%	86.7%	81.1%	82.9%	85.3%

However, the rest of the result showed that false positive rate was decreased by applying AdaBoost, and we find that AdaBoost increased the total accuracy rate in almost of all cases.

6 Discussion

In this section, we discuss the limitations of our AdaBoost-based detection of phishing sites. At first, we analyze the result to find the reason why the true positive rate of the AdaBoost-based combination method was lower than that of normal CANTINA in Section 5.3. Next, we discuss the way of building both a training dataset and a test dataset.

6.1 Overfitting problem

In the training dataset of (iii) in Section 5.3, we found that one legitiamte site was labeled as phishing by Forms, Age of Domain and TF-IDF-Final heuristic all of which were the top three heuristics in accuracy in this case. The AdaBoost-based combination method, firstly, has checked this site by Forms heuristic. Form heuristic showed the highest accuracy among seven heuristics, but could not identify this site correctly . In this case, an AdaBoost algorithm assigns higher weight to the heuristic which is able to label the site correctly, and assigns lower weight to heuristic which labeled incorrectly. Age of Domain and TF-IDF-Final heuristic, as well, could not identify the site correctly, so the weight of these two heuristics have been reduced.

We analyzed that it was an overfitting problem which caused this reduction. The overfitting problem is known as the one of the weak points of AdaBoost, it decreases the accuracy otherwise AdaBoost attempted to fit the weight for identifying the site. In the case of (i) and (ii), there are much phishing sites that both Age of Domain and TF-IDF heuristic labeled correctly. Therefore, these

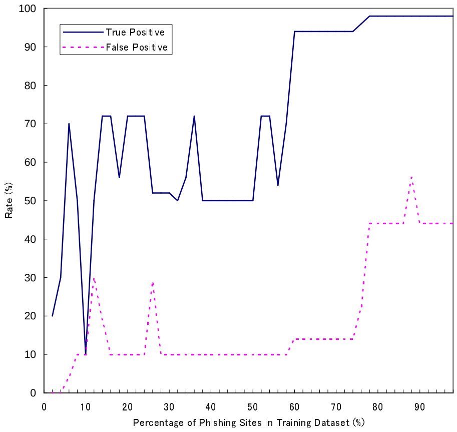

Fig. 2. Effect of the percentage of phishing sites in training dataset

heuristics could be assigned high weight otherwise they lost weight by the issued site. However, in the case of (iii), (iv) and (v), there are fewer phishing sites, and these two heuristics could not be assigned high weight. Hence, the true positive rate of AdaBoost was falling. If the sites was not contained in training dataset, the true positive rate of (iii) was increased to 97.5%.

Increasing samples in dataset is the remedy for reducing the effect of overfitting. We assumed the automated dataset collection is desirable for updating the assigned weight suitably. However, AdaBoost is designed as a supervised learning algorithm. Before applying AdaBoost to the training dataset, all of the sites might be identified as legitimate or phishing. Hence, we manually checked and labeled each site.

In order to facilitate to collect samples, we assumed that it is the ideal environment in which numerous sites have been manually labeled. For example, the user of PhishTank.com can vote whether a site which was reported as phishing is really a phishing site or not. We considered the dataset collection is facilitated with this voting information.

6.2 Effect of percentage of phishing sites in training dataset

On constructing a training dataset, it is important that not only the size of the dataset, but also the percentage of phishing sites in the dataset. If we have chosen our training dataset from all the sites in a random manner, almost of all the sites in training dataset would be legitimate sites. Within such a training dataset, AdaBoost would focus on identifying the legitimate sites as legitimate, but this would decrease the accuracy of detecting phishing sites.

Figure 2 shows how a training dataset affects the true positive rate and false positive rate. The X-axis denotes the percentage of phishing sites in training dataset, and the Y-axis denotes the true positive rate and the false positive rate in each training dataset. We first have build a training dataset where all 50 sites were legitimate. Next, we have randomly removed one legitimate site from training dataset and have added one phishing site into the training dataset. We also calculated the weight by using seven heuristics, and measured the true positive rate and the false positive rate in our test dataset which were composed of 50 phishing sites and 50 legitimate sites. In this way, we changed the percentage of phishing sites in order to the effectiveness of test dataset.

We found that both the true positive rate and false positive rate tended to increase when we increased the percentage of phishing sites in training dataset. We also found both of two line graphs were complected, so it was hindered to predict whether or not the accuracy of detection is high before applying AdaBoost combination method to the test dataset.

Hence, our AdaBoost-based combination method should constantly investigate whether or not the assigned weight is suitable for detection of phishing sites. Our method should also change the training dataset and calculate new weight, if the accuracy is not so high.

7 Conclusion

We presented an approach which employs AdaBoost algorithm to combine heuristics for detection of phishing sites. Our proposed combination method calculated the likelihood of a phishing site by weighted majority, in which weight was assigned by AdaBoost. In this paper, we focused on an AdaBoost-based combination method, and applying its algorithm to eight heuristics which were introduced by CANTINA. Through the evaluation of both combination method of AdaBoost and that of CANTINA, we found that almost results showed that employing AdaBoost could be improved the accuracy. Accordingly, we assumed that AdaBoost is an effective algorithm to combine heuristics for detecting phishing sites.

We also showed the limitations of the AdaBoost-based combination method, identified overfitting as the weak point of our approach and discussed the way of increasing the sample of dataset as a countermeasure of overfitting problem. We presented the effect of the percentage of phishing sites in training dataset, and found that our method should investigate constantly whether or not the assigned weight is suitable.

References

1. Zhang, Y., Hong, J., Cranor, L.: CANTINA: A Content-Based Approach to Detect Phishing Web Sites. In: Proceesings of the 16th World Wide Web Conference (WWW'07), Banff, Canada (2007)

2. Kumar, A.: Phishing - A new age weapon. Technical report, Open Web Application Secuirtry Project (OWASP) (2005)
3. Tally, G., Thomas, R., Vleck, T.V.: Anti-Phishing: Best Practices for Institutions and Consumers. Technical report, McAfee Research (2004)
4. Anti-Phishing Working Group: Phishing Activity Trends Report - November, 2006 (2006)
5. McCall, T., Moss, R.: Gartner Survey Shows Frequent Data Security Lapses and Increased Cyber Attacks Damage Consumer Trust in Online Commerce (2005)
6. Drake, C.E., Oliver, J.J., Koontz, E.J.: Anatomy of a Phishing Email. In: Proceedings of CEAS 2004. (2004)
7. Miyamoto, D., Hazeyama, H., Kadobayashi, Y.: SPS: A Simple Filtering Algorithm Thwart Phishing Attacks. In: Proceedings of AINTEC 2005. (2005)
8. Chou, N., Ledesma, R., Teraguchi, Y., Boneh, D., Mitchell, J.C.: Client-side defense against web-based identity theft. In: Proceedings of 11th Annual Network and Distributed System Security Symposium (NDSS '04). (2004)
9. Van der Merwe, A., Loock, M., Dabrowski, M.: Characteristics and responsibilities involeved in a phishing attack. In: Proceedings of The 4th International Symposium on Information and Communication Technologies (ISICT 2005). (2005)
10. Jakobsson, M.: Modeling and Preventing Phishing Attacks. In: Proceedings of Financial Cryptography '05, Pishing Panel. (2005)
11. M. Jakobsson: Distributed Phishing Attacks. In: Proceedings of DIMACS Workshop on Theft in E-Commerce: Content, Identity, and Service. (2005)
12. Fette, I., Sadeh, N., Tomasic, A.: Learning to Detect Phishing Emails (2006)
13. Valiant, L.G.: A theory of the learnable. In: Communications of the ACM, 27(11). (1984) 1134–1142
14. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Journal of Computer and System Sciences 55(1). (1997) 119–139
15. Felten, E.W., Balfanz, D., Dean, D., Wallach, D.S.: Web Spoofing: An Internet Con Game. In: Proceedings of 20th National Information Systems Security Conference (NISSC '97). (1997)
16. Thomas A. Phelps, R. Wilensky: Robust hyperlinks and locations. (2000)
17. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: ICML. (1996) 148–156
18. Alexa Internet, Inc.: (Alexa the Web Information Company)
19. Robichaux, P., Ganger, D.L.: Gone Phishing: Evaluating Anti-Phishing Tools for Windows (2006)