An Evaluation of Machine Learning-based Methods for Detection of Phishing Sites

Daisuke Miyamoto, Hiroaki Hazeyama, and Youki Kadobayashi

Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, Japan
{daisu-mi, hiroa-ha, youki-k}@is.naist.jp

Abstract. In this paper, we evaluate the performance of machine learning-based methods for detection of phishing sites. In our previous work [1], we attempted to employ a machine learning technique to improve the detection accuracy. Our preliminary evaluation showed the AdaBoost-based detection method can achieve higher detection accuracy than the traditional detection method. Here, we evaluate the performance of 9 machine learning techniques including AdaBoost, Bagging, Support Vector Machines, Classification and Regression Trees, Logistic Regression, Random Forests, Neural Networks, Naive Bayes, and Bayesian Additive Regression Trees. We let these machine learning techniques combine heuristics, and also let machine learning-based detection methods distinguish phishing sites from others. We analyze our dataset, which is composed of 1,500 phishing sites and 1,500 legitimate sites, classify them using the machine learning-based detection methods, and measure the performance. In our evaluation, we used f_1 measure, error rate, and Area Under the ROC Curve (AUC) as performance metrics along with our requirements for detection methods. The highest f_1 measure is 0.8581, the lowest error rate is 14.15%, and the highest AUC is 0.9342, all of which are observed in the case of AdaBoost. We also observe that 7 out of 9 machine learning-based detection methods outperform the traditional detection method.

1 Introduction

Phishing is a form of identity theft whose targets are users rather than computer systems. A phishing attacker attracts victims to a spoofed web site, a so-called "phishing site", and attempts to persuade them to send their personal information.

To prevent a user from browsing phishing sites, there are two distinct approaches. One is URL filtering. It detects phishing sites by comparing the URL of a site a user visits with a URL blacklist composed of the URLs of phishing sites. However, it is difficult to build a perfect blacklist due to the rapid increase of phishing sites. According to trend reports published by the Anti-Phishing Working Group [2], the number of reported phishing sites was 23,917 in July 2007, far surpassing the 14,315 in July 2005.

The other approach is a heuristic-based solution. A heuristic is an algorithm to distinguish phishing sites from others based on users' experience, that is, a

heuristic checks if a site seems to be a phishing site. A heuristic-based solution employs several heuristics and converts results from each heuristic into a vector. Based on the vector, the heuristic-based solution calculates the likelihood of a site being a phishing site and compares the likelihood with the defined discrimination threshold. Different from URL filtering, a heuristic-based solution has a possibility to identify new phishing sites.

Unfortunately, the detection accuracy of existing heuristic-based solutions is far from suitable for practical use [3], even if various studies [4–6] discovered heuristics. To improve the detection accuracy, both discovering innovative heuristics and refining the calculation algorithm of the likelihood are important.

In our previous work [1], we attempted to employ a machine learning technique to improve the detection accuracy. We employed AdaBoost, a machine learning technique, as a calculation method of the likelihood. Our preliminary evaluation showed the AdaBoost-based detection method can achieve higher detection accuracy than the traditional detection method.

Here, we present a performance evaluation of 9 Machine Learning-based Detection Methods (MLBDMs) including AdaBoost, Bagging, Support Vector Machines (SVM), Classification and Regression Trees (CART), Logistic Regression (LR), Random Forests (RF), Neural Networks (NN), Naive Bayes (NB) and Bayesian Additive Regression Trees (BART). In the evaluation, we used f_1 measure, error rate, Area Under the ROC Curve (AUC) as performance metrics along with our requirements for detection methods. Our requirements are (i) they must achieve high detection accuracy, (ii) they must adjust their detection strategies for web users.

We let all MLBDMs classify whether a site is a phishing site or not by using a dataset of 3,000 URLs, composed of 1,500 phishing sites and the same number of legitimate sites. We employ 8 heuristics presented in CANTINA [7] and measure their performance in a less biased way. The results show that the highest f_1 measure is 0.8581, the lowest error rate is 14.15%, and the highest AUC is 0.9342, all of which are observed in the case of AdaBoost.

The rest of this paper is organized as follows: In Section 2, we present our related work. In Section 3, we describe our evaluation conditions, and we show our experimental results in Section 4. We show our future work in Section 5 and summarize our contributions in Section 6.

2 Related Work

For mitigating phishing attacks, machine learning, which facilitates the development of algorithms or techniques by enabling computer systems to learn, has begun to garner attention. PFILTER, which was proposed by Fette et al. [8], employed SVM to distinguish phishing emails from other emails. According to [9], Abu-Nimeh et al. compared the predictive accuracy of several machine learning methods including LR, CART, RF, NB, SVM, and BART. They analyzed 1,117 phishing emails and 1,718 legitimate emails with 43 features for distinguishing phishing emails. Their research showed that the lowest error rate was 7.72% in

the case of Random Forests. In [10], Ram Basnet et al. performed an evaluation of six different machine learning-based detection methods. They analyzed 973 phishing emails and 3,027 legitimate emails with 12 features, and showed that the lowest error rate was 2.01%. The experimental conditions were different between [9] and [10], however, the machine learning provided high accuracy for the detection of phishing emails.

Aside from phishing emails, a machine learning method was also used to detect phishing sites. According to [11], Pan et al. presented an SVM-based page classifier for detection of phishing sites. They analyzed 279 phishing sites and 100 legitimate sites with 8 features, and the results showed that the average error rate was 16%.

Our previous work [1] employed AdaBoost for the detection of phishing sites. We checked 100 phishing sites and the same number of legitimate sites with 7 heuristics. Our performance evaluation showed that the average error rate was 14.7%.

We find that there are two problems in earlier research. One is that the number of features for detecting phishing sites is lesser than that for detecting phishing emails. It indicates that the detection of phishing sites is much difficult than that of phishing emails. The other is that no research contribution confirmed whether any kind of MLBDMs were available to distinguish phishing sites from legitimate sites. To the best of our knowledge, earlier research tested only one machine learning technique. In this paper, we evaluate 9 MLBDMs and show their performance by measuring the performance.

3 Evaluation Approach

In this paper, we evaluate the performance of MLBDMs. We let each machine learning method combine heuristics, perform supervised learning from the dataset, and distinguish phishing sites from other sites.

In this section, we define metrics of the performance evaluation along with our requirements for MLBDMs. We then decide the heuristics that we used in our evaluation, and describe how we construct a dataset for both training and testing. Finally, we explain the preliminary set-up of our experiments.

3.1 Evaluation Metrics

First, we defined metrics for evaluating performance along with requirements for detection methods. Our requirements were as follows.

1. *Accuracy*
   An MLBDM must achieve high detection accuracy. User safety would obviously be compromised if phishing prevention systems labeled phishing sites as legitimate. Users would also complain if prevention systems labeled legitimate sites as phishing sites because of the interruption in browsing caused by prevention systems.

2. *Adjustment Capability*
   An MLBDM must adjust its strategy for detecting phishing sites for web users. If a user is a novice, who is easily taken in by phishing attacks, phishing prevention systems should decrease false negative errors instead of increasing false positive errors. Conversely, if a user is a security expert, the system should focus on decreasing false positive errors.

For Requirement 1, we used the f_1 measure (higher is better) and the error rate (lower is better) as metrics to evaluate the detection accuracy. Statistically, f_1 measure has been used as an index of a test's accuracy. This measure can be calculated by $2 \cdot p \cdot r/(p + r)$, where p is the precision and r is the recall of the test. The average error rate has been also a reasonable metric to indicate the detection accuracy. It is calculated by dividing the number of incorrectly identified sites by the number of all sites in the dataset.

For Requirement 2, we performed Receiver Operating Characteristic (ROC) analysis. Generally, detection methods calculate the likelihood of being phishing sites L and compare the likelihood with the defined discrimination threshold th. In our experiment, MLBDMs distinguish a phishing site by checking if L is less or equal than $th(= 0)$. Imagine that th was higher than 0. In this case, MLBDMs would tend to label a site as phishing rather than as legitimate. Conversely, MLBDMs would tend to label a site as legitimate if th was lower than 0. Accordingly, we assumed that adjusting th provides different detection strategies. Based on this assumption, we employed ROC analysis because it has been widely used in data analysis to study the effect of varying the threshold on the numerical outcome of a diagnostic test. We also used the Area Under the ROC curve (AUC; higher is better) as a metric to evaluate adjustment capability.

3.2 Heuristics

In our evaluation, we employ 8 heuristics presented in CANTINA [7]. To the best of our knowledge, the most successful tool for combining heuristics is CANTINA, which has achieved high accuracy of detecting phishing sites without using the URL blacklist. In CANTINA, the likelihood of the phishing site is calculated from weighted majority by using 8 heuristics, without using machine learning techniques.

3.3 Dataset

We then built a dataset with the criteria for choosing URLs. Based on the criteria in the original CANTINA, we collected URLs with the same number of phishing sites and legitimate sites. All sites were English language sites because CANTINA does not work well if the sites are not written in English. First, we chose 1,500 phishing sites that were reported on PhishTank.com [12] from November, 2007 to February, 2008. Second, we also selected 227 URLs from 3Sharp's study of anti-phishing toolbars [13]. There were listed 500 URLs of legitimate sites in [13], however, we could not connect to many listed URLs. Third,

we attempted to collect 500 URLs from Alexa Web Search [14] and observed 477 URLs. Finally, we gathered 796 URLs from yahoo random link [15].

Each site was checked with our implementation of heuristics, and was converted into a vector $\boldsymbol{x} = (x_1, x_2 \ldots x_p)$, where $x_1 \ldots x_p$ are the values corresponding to a specific feature. The dataset consisted of 8 binary explanatory variables and 1 binary response variable.

To perform our evaluation in a less biased way, we employed 4-fold cross validation. Furthermore, our cross validation was repeated 10 times in order to average the result.

3.4 Experimental Set-up

We adjusted the parameters for MLBDMs to minimize the error rate in training. For decision tree-based machine learning techniques such as RF, we tested them using different numbers of trees, namely 100, 200, 300, 400, and 500 trees. The minimum error rate (14.27%) was observed when the number of trees was 300, followed by 200 and 400 (14.28%), 500 (14.30%), and 100 (14.37%). Thus, we set the number of trees to 300 for RF-based detection methods.

The iteration time was set to 500 in all of our experiments if the machine learning technique needed to analyze iteratively for reducing training errors. The minimum error rate (14.27%) was observed when the number of iterations was 500, followed by 300 and 400 (14.28%), 200 (14.30%), and 100 (14.31%). In addition, finding the optimal iteration number is important, however, the choice of the exact value of the optimal iteration number is not often a critical issue since the increase in test error is relatively slow.

For some types of machine learning techniques, we used threshold value to approximate the prediction output. For example, BART is designed for regression, not for classification. Therefore, BART gives quantitative value whereas we need an MLBDM to output binary value that indicates whether a site is a phishing site or not. In such cases, we employed threshold value and observed if the result of BART regression was greater than the threshold. We decided the threshold in the same fashion as the original CANTINA. In the case of CANTINA, the maximum likelihood of being a phishing site is -1 and that of being a legitimate site is 1; therefore, it employs the middle value 0 as the threshold value.

In SVM, we tested both linear and non-linear kernel functions. The average error rate in training by using *Radial Based Function* (RBF), one of the typical non-linear kernel functions, was 14.18%, less than 21.02% of linear kernel. Thus, we used RBF in our experiments.

In NN, we selected the number of units in the hidden layer, namely 1, 2, 3, 4, and 5 units, for finding the minimum average error rate. The minimum error rate (14.14%) was observed when the number of units was 5, followed by 4 (14.23%), 2 (14.46%), 3 (15.48%), and 1 (16.03%).

4 Evaluation

In this section, we evaluate the performance of all MLBDMs by measuring f_1 measure, error rate and AUC, and studying them comparatively. We also compare MLBDMs with the original CANTINA.

Table 1. Precision, Recall and f_1 measure, False Positive Rate(FPR), False Negative Rate(FNR), Error Rate, and AUC

	Precision	*Recall*	f_1 *measure*	FPR	FNR	ER	AUC
AdaBoost	0.8614	0.8551	0.8581	14.49%	13.83%	14.15%	0.9342
Bagging	0.8492	0.8573	0.8527	14.27%	15.36%	14.82%	0.9231
SVM	0.8629	0.8498	0.8562	15.02%	13.57%	14.29%	0.8946
CART	0.8330	0.8542	0.8384	14.58%	18.16%	16.37%	0.9062
LR	0.8510	0.8588	0.8548	14.12%	15.10%	14.60%	0.9172
RF	0.8566	0.8546	0.8554	14.54%	14.37%	14.45%	0.9296
NN	0.8633	0.8512	0.8570	14.88%	13.54%	14.21%	0.9310
NB	0.8464	0.8636	0.8547	13.64%	15.74%	14.69%	0.9215
BART	0.8567	0.8550	0.8555	14.50%	14.39%	14.45%	0.9321
CANTINA	0.9134	0.6519	0.7606	34.81%	06.21%	20.52%	0.9162

First, we measured the accuracy of all MLBDMs. We calculated Precision, Recall and f_1 measure for each pattern of dataset respectively, and also calculated their average as shown in Table 1. The highest f_1 measure was 0.8581 in AdaBoost, followed by NN (0.8570), SVM (0.8562), BART (0.8555), RF (0.8554), LR (0.8548), NB (0.8547), Bagging (0.8527) and finally CART (0.8384).

We then calculated the false negative rate, false positive rate and error rate in Table 1. The lowest error rate was 14.15% in AdaBoost, followed by NN (14.21%), SVM (14.29%), RF and BART (14.45%), LR (14.60%), NB (14.69%), Bagging (14.82%), and finally CART (16.37%). The lowest false negative rate was 13.64% in NB, and the lowest false positive rate was 13.54% in NN.

We also calculated AUC as shown in Table 1. The highest AUC was 0.9342 in AdaBoost, followed by BART (0.9321), NN (0.9310), RF (0.9296), Bagging (0.9231), NB (0.9215), LR (0.9172), CART (0.9062), and finally SVM (0.8956).

Finally, we compared all MLBDMs with CANTINA's detection method. We evaluated the performance of CANTINA in the same way as that described in Section 3, and observed f_1 measure was 0.7607, error rate was 20.52%, and AUC was 0.9162 as shown in Table 1, respectively. According to our comparison, 7 out of 9 MLBDMs, namely AdaBoost, Bagging, LR, RF, NN, NB, and BART-based detection methods, outperformed CANTINA.

5 Future Work

In our future work, we will implement a phishing prevention system according to the detection result. Within such a system, we should adjust the discrimination

Table 2. FPR given FNR rate $< 5.00\%$, and FNR given FPR rate $< 5.00\%$

	FP(FN< 5.00%)	*FN(FP< 5.00%)*
AdaBoost	30.65%	26.15%
Bagging	34.28%	30.35%
SVM	62.10%	71.10%
CART	46.24%	36.77%
LR	33.90%	41.27%
RF	31.56%	25.51%
NN	31.56%	26.48%
NB	31.86%	35.55%
BART	31.35%	25.76%

threshold for each web user. For simple examples, we considered the cases of novices and security experts. If a user is a novice, who is easily taken in by phishing attacks, the system should decrease the false negative rate instead of increasing the false positive rate. Conversely, if a user is a security expert, the system should emphasize decreasing the false positive rate.

Table 2 shows the false positive rate when the false negative rate was less than 5.00%, and the false negative rate when the false positive rate was less than 5.00%. The lowest false positive rate was 30.65% in the case of AdaBoost, and the lowest false negative rate was 25.51% in the case of RF. This indicated that if novices could accept 30.65% of false positive errors, 95.00% of phishing sites would be blocked as phishing sites. Similarly, if security experts could accept 25.51% of false negative errors, 95.00% sites of legitimate sites would be browsed normally. It is beyond of scope of this paper, however, we need to decide the optimal threshold for each user by both measuring each user's knowledge for the detection of phishing sites and estimating the acceptable level of false positives and false negatives.

6 Conclusion

In this paper, we evaluated the performance of machine learning-based detection methods (MLBDMs) including AdaBoost, Bagging, Support Vector Machines (SVM), Classification and Regression Trees (CART), Logistic Regression (LR), Random Forests (RF), Neural Networks (NN), Naive Bayes (NB) and Bayesian Additive Regression Trees (BART). Because we assumed that the detection method must be accurate and must have adjustment capability, we used f_1 measure, error rate and AUC as performance metrics in the evaluation.

We employed 8 heuristics presented in [7] and analyzed 3,000 URLs, which were composed of 1,500 phishing sites and the same number of legitimate sites. We performed 4-fold cross validation 10 times and measured the average of f_1 measure, error rate, and AUC.

The result showed that the highest f_1 measure was 0.8581 in AdaBoost, followed by NN (0.8570), SVM (0.8562), BART (0.8555), RF (0.8554), LR (0.8548),

NB (0.8547), Bagging (0.8527) and finally CART (0.8384). The lowest error rate was 14.15% in AdaBoost, followed by NN (14.21%), SVM (14.29%), RF and BART (14.45%), LR (14.60%), NB (14.69%), Bagging (14.82%), and finally CART (16.37%). The highest AUC was 0.9342 in AdaBoost, followed by BART (0.9321), NN (0.9310), RF (0.9296), Bagging (0.9231), NB (0.9215), LR (0.9172), CART (0.9062), and finally SVM (0.8956). Additionally, we plotted the ROC curve and found that all MLBDMs could achieve both high true positive rates and low false positive rates.

References

1. Miyamoto, D., Hazeyama, H., Kadobayashi, Y.: A Proposal of the AdaBoost-Based Detection of Phishing Sites. In: Proceedings of the 2nd Joint Workshop on Information security. (2007)
2. Anti-Phishing Working Group: Phishing Activity Trends Report - July, 2007 (2007)
3. Zhang, Y., Egelman, S., Cranor, L., Hong, J.: Phinding Phish: Evaluating Anti-Phishing Tools. In: Proceesings of the 14th Annual Network and Distributed System Security Symposium (NDSS'07). (2007)
4. Kumar, A.: Phishing - A new age weapon. Technical report, Open Web Application Secuirtry Project (OWASP) (2005)
5. Tally, G., Thomas, R., Vleck, T.V.: Anti-Phishing: Best Practices for Institutions and Consumers. Technical report, McAfee Research (2004)
6. 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)
7. 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). (2007)
8. Fette, I., Sadeh, N.M., Tomasic, A.: Learning to detect phishing emails. In: Proceedings of the 16th International Conference on World Wide Web (WWW'07). (2007)
9. Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: A comparison of machine learning techniques for phishing detection. In: Proceedings of eCrime Researchers Summit (eCryme'07). (2007)
10. Basnet, R., Mukkamala, S., Sung, A.H.: Detection of phishing attacks: A machine learning approach. Studies in Fuzziness and Soft Computing **226** (2008) 373–383
11. Pan, Y., Ding, X.: Anomaly based web phishing page detection. In: Proceedings of the 22nd Annual Computer Security Applications Conference on Annual Computer Security Applications Conference(ACSAC'06). (2006)
12. OpenDNS: PhishTank - Join the fight against phishing. (Available at `http://www.phishtank.com`)
13. Robichaux, P., Ganger, D.L.: Gone Phishing: Evaluating Anti-Phishing Tools for Windows. (Available at `http://www.3sharp.com/projects/antiphishing/gone-phishing.pdf`)
14. Alexa Internet, Inc.: Alexa the Web Information Company. (Available at `http://www.alexa.com`)

15. Yahoo!Inc.: Random Yahoo Link. (Available at
http://random.yahoo.com/fast/ryl)