Phishing Detection with Popular Search Engines: Simple and Effective

Jun Ho Huh[1] and Hyoungshick Kim[2]

[1] Information Trust Institute, University of Illinois at Urbana-Champaign, US
jhhuh@illinois.edu
[2] Computer Laboratory, University of Cambridge, UK
hk331@cl.cam.ac.uk

Abstract. We propose a new phishing detection heuristic based on the search results returned from popular web search engines such as Google, Bing and Yahoo. The full URL of a website a user intends to access is used as the search string, and the *number of results* returned and *ranking* of the website are used for classification. Most of the time, legitimate websites get back large number of results and are ranked first, whereas phishing websites get back no result and/or are not ranked at all.

To demonstrate the effectiveness of our approach, we experimented with four well-known classification algorithms – Linear Discriminant Analysis, Naïve Bayesian, K-Nearest Neighbour, and Support Vector Machine – and observed their performance. The K-Nearest Neighbour algorithm performed best, achieving true positive rate of 98% and false positive and false negative rates of 2%. We used new legitimate websites and phishing websites as our dataset to show that our approach works well even on newly launched websites/webpages – such websites are often misclassified in existing blacklisting and whitelisting approaches.

Keywords: Phishing detection, URL Reputation, Classification.

1 Introduction

Phishing attacks are pre-dominant in today's web. The Anti-Phishing Working Group (APWG) reported that there were at least 126,697 phishing attacks in the second half of 2009, almost doubling the number of attacks counted in the first half [1]. A lot of these attacks entice people into visiting fraudulent websites that impersonate trusted entities, and persuade them to disclose their private information such as passwords and bank account details. Figure 1 shows an example of a phishing website posing as Facebook; notice how the URL in the red box (`http://h1.ripway.com/riki123/index.html`) is completely different to the original URL.

Some of the widely available and used phishing detection techniques include whitelisting [2], blacklisting [19], and heuristics [3,16,7,24,22]. Although blacklisting is effective in detecting known-bad phishing websites, it can be weak against detecting new ones. It is also difficult to efficiently update and verify the website entries of central databases.

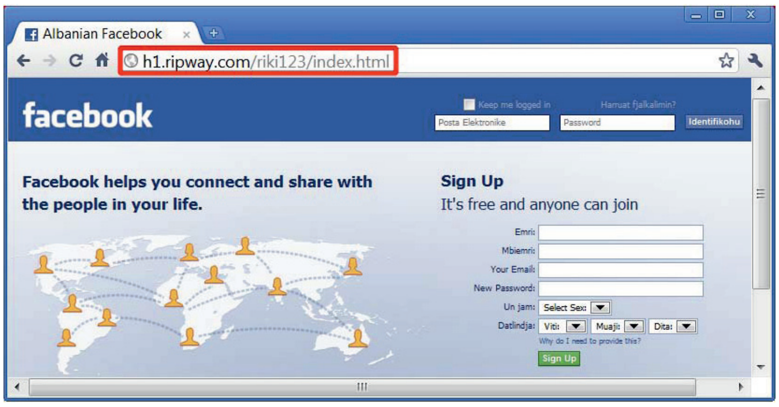


Fig. 1. Example of a phishing website

Whitelisting, on the other hand, is effective against new phishing websites since only those considered 'trusted' are accessed by users. However, it is somewhat difficult to know exactly which new legitimate websites a user will visit next and have these entries added to the whitelist prior to their visit. If the coverage of the whitelist is insufficient, it can incur a significant usability penalty. It is this usability concern that currently discourages many users from using the whitelists.

Heuristic-based techniques look for common characteristics of phishing websites. Since these techniques do not rely on central databases being up-to-date, they can be more effective against detecting new phishing websites. Nevertheless, heuristics will inherently incur false positives and false negatives, and there is always a room for phishers to design their websites to bypass the heuristics that are being checked.

This paper proposes a novel phishing detection method that uses the *reputation* of a website for classification. Our intuition is based on the observation that (1) the reputation of a legitimate website will grow over time unlike a phishing website, and (2) a webpage's reputation can be measured approximately by querying popular search engines with its full URL and analysing the returned results. In particular, the *number of results* returned and *ranking* of the website give good indication as to what the nature of the website is.

Our approach has two key advantages: (1) it can be very effective against new websites since the web search engines crawl and cache hundreds and thousands of new and old webpages per second; (2) it is *simple* to implement and deploy. To demonstrate feasibility, we evaluated performance of four well-known classification algorithms, Linear Discriminant Analysis, Naïve Bayesian, K-Nearest Neighbour, and Support Vector Machine, on three different search engines, Google[1], Bing[2] and Yahoo[3].

[1] `http://www.google.co.uk/`
[2] `http://www.bing.com/`
[3] `http://uk.yahoo.com/`

The next section gives an overview of how our approach works. In Section 3, we introduce the four classification algorithms. Then, in Section 4, we discuss the feasibility of our approach based on the experiment results, and recommend how it should be used in practice. Some related work is discussed in Section 5. Finally, we conclude in Section 6.

2 Phishing Detection with Popular Search Engines

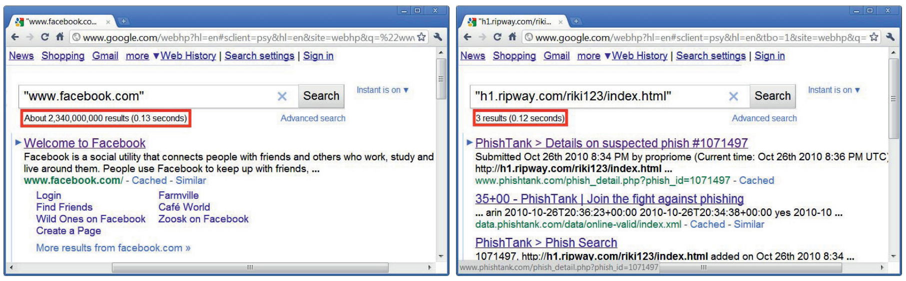
The key idea is simple: a user's trusted application queries a popular web search engine using the full URL of a website she intends to access (e.g. "`http://www.example.com`" would be your search string), and analyses the 'number of results' returned and 'ranking' of the website to make a decision.

As with any other heuristic-based methods, a system designed to use our approach can be fully automated. When the user tries to visit a website, her trusted application – this could be a web browser plug-in or a built-in feature – will first submit a search query to a web search engine using the full URL string without parameters; for example, if the user wishes to access "`http://www.facebook.com/profile.php?id=12345`", then the concatenated string "`http://www.facebook.com/profile.php`" will be submitted. From the returned search results, the application will fetch two attributes: the 'number of results' and 'ranking' of the queried URL; and use this information to determine whether the website is legitimate. The ranking is measured by looking at where the queried URL appears on the returned results: ranking n implies that the URL appears as the nth result.

Most of the time, legitimate websites that are safe for users to browse and match are ranked first; phishing websites, on the other hand, have lower ranking or are not ranked at all. Legitimate websites also have larger number of search results compared to phishing websites. This can be explained by the long-lived nature of legitimate websites and the growing number of in-links as oppose to the short-lived nature of phishing websites and the small number of in-links (usually none) they have. Figure 2 shows the differences between the search results returned from Google when queried with (a) the original URL of Facebook and (b) URL of its phishing website.

When a search engine is queried with a URL, it ranks a cached/indexed website very high, often first, if this website's URL exactly matches the queried URL. Our approach fully exploits this aspect of search engines to accurately identify even the recently published legitimate websites that do not have many in-links, or long-lived legitimate websites that are unpopular and have small number of in-links.

Moreover, new legitimate websites tend to build their reputation more quickly. Table 1 compares how the reputation of new legitimate websites and phishing websites change on Google over a period of 5 days – day 1 being the first day that a website/webpage was created. New webpages published through well-known websites like BBC and Amazon were used to observe the trends for the legitimate websites. As for the phishing websites, 5 most recently listed on an online


"www.facebook.co...
www.google.com/webhp?hl=en#sclient=psy&hl=en&site=webhp&q=%22ww
News Shopping Gmail more ▾Web History | Search settings | Sign in
Search
Instant is on ▾
Advanced search

"www.facebook.com"

About 2,340,000,000 results (0.13 seconds)

Welcome to Facebook
Facebook is a social utility that connects people with friends and others who work, study and live around them. People use Facebook to keep up with friends, ...
www.facebook.com/ - Cached - Similar

Login
Find Friends
Wild Ones on Facebook
Create a Page
Farmville
Café World
Zoosk on Facebook

More results from facebook.com »

"h1.ripway.com/riki...
www.google.com/webhp?hl=en#sclient=psy&hl=en&tbo=1&site=webhp&q=
News Shopping Gmail more ▾Web History | Search settings | Sign in
Search
Instant is on ▾
Advanced search

"h1.ripway.com/riki123/index.html"

3 results (0.12 seconds)

PhishTank > Details on suspected phish #1071497
Submitted Oct 26th 2010 8:34 PM by propriome (Current time: Oct 26th 2010 8:36 PM UTC)
http://**h1.ripway.com/riki123/index.html** ...
www.phishtank.com/phish_detail.php?phish_id=1071497 - Cached

35+00 - PhishTank | Join the fight against phishing
... arin 2010-10-26T20:36:23+00:00 2010-10-26T20:34:38+00:00 yes 2010-10 ...
data.phishtank.com/data/online-valid/index.xml - Cached - Similar

PhishTank > Phish Search
1071497, http://**h1.ripway.com/riki123/index.html** added on Oct 26th 2010 8:34 ...

www.phishtank.com/phish_detail.php?phish_id=1071497

(a) Facebook (b) A phishing website

Fig. 2. Differences in the search results returned from Google when queried with the original URL of Facebook and the URL of its phishing website. With the original URL, the 'number of results' returned is 2,340,000,000 and the website is ranked first, while it is only 3 for the phishing website which is not ranked at all.

Table 1. How the reputation of new Legitimate Websites (LWs) and Phishing Websites (PWs) change on Google over 5 days

		D-1	D-2	D-3	D-4	D-5
LW_1	No. of results	7	41	48	79	105
	Ranking	1	1	1	1	1
LW_2	No. of results	2	8	9	9	9
	Ranking	2	1	1	1	1
LW_3	No. of results	3	35	39	90	83
	Ranking	1	1	1	1	1
LW_4	No. of results	6	268	158	289	308
	Ranking	1	1	1	1	1
LW_5	No. of results	1	4	1	1	2
	Ranking	1	1	1	1	1

	D-1	D-2	D-3	D-4	D-5
PW_1	1	1	0	0	0
	0	0	0	0	0
PW_2	2	5	3	1	1
	0	0	0	0	0
PW_3	3	3	2	1	1
	0	0	0	0	0
PW_4	2	4	3	1	1
	0	0	0	0	0
PW_5	2	4	3	1	1
	0	0	0	0	0

phishing URL database called 'PhishTank'[4] were used. Even during this short period of time, we observed that the number of results for legitimate websites increased more quickly and they were all ranked first from the second day onward. In contrast, the phishing websites had fairly static, small number of results and were never ranked on Google. Our approach uses such distinct characteristics between the two groups to measure their reputation and classify them.

After collecting the two information, the user's trusted application processes them using the pre-generated classifier to determine the nature of the website. If the website is likely to be a phishing website, the user is given a warning message to deal with; otherwise, the user is free to access the website.

In the next two sections, we describe four different classification methods that may be used to train a classifier, and demonstrate how each method performs on the search results returned from three popular search engines, Google, Bing and Yahoo.

[4] `http://www.phishtank.com`

3 Applying Classification Methods

Given a set of search results for the website URLs, our problem can be reduced to a *classification* problem: the input will be a set of training samples in the form of $\langle n_u, r_u, c_u \rangle$ for a URL, u, where n_u, r_u, and c_u are the 'number of results' returned, 'ranking' of the website, and 'indicator' that shows whether u belongs to a phishing website, respectively. c_u will be `TRUE` if u belongs to a phishing website; otherwise, it will be `FALSE`.

A classification method should be carefully selected based on constraints such as the desired level of accuracy, time available for development and training, and nature of classification problems. We carried out a number of experiments to evaluate four different classification methods and to find the best performing (most suitable) one. This section introduces these classification methods and the experiment results are discussed in Section 4.

3.1 Linear Discriminant Analysis

Linear Discriminant Analysis (`LDA`) is a powerful tool for dimensionality reduction and classification [8]. Given a set of search results, `LDA` finds a linear transformation of the training samples with n_u and r_u that best discriminates between legitimate and phishing websites. `LDA` performs well if the features of the dataset are linearly independent. So we expect `LDA` to be sufficient for detecting phishing websites since n_u and r_u between legitimate and phishing websites are likely to be discriminated linearly.

3.2 Naïve Bayesian Classification

Naïve Bayesian (`NB`) classification algorithm [6,21] is one of the most successful learning algorithms for text categorisation. Based on the Bayes rule, which assumes conditional independence between classes, this algorithm attempts to estimate the conditional probabilities of classes given an observation. The joint probabilities of sample observations and classes are simplified because of the conditional independence assumption. While this Bayes rule assumption is often violated in practice, `NB` is known to perform well in many existing applications [17].

3.3 K-Nearest Neighbour Classification

K-Nearest Neighbour (`KNN`) classification is a non-parametric classification algorithm. `KNN` has been applied successfully to various information retrieval problems. `KNN` uses an integer parameter K. Given a search result for a URL, u, the algorithm finds K, the closest training data that points to this result, and uses the majority vote to determine whether the website with u is a phishing website.

Without any prior knowledge, the `KNN` classifier usually applies Euclidean distances as the distance metric. This simple method can often yield competitive results even compared to other sophisticated machine learning methods. The performance of KNN is primarily determined by the choice of K as well as the distance metric applied [13]. However, choosing a suitable K value is not easy when the points are not uniformly distributed [5].

3.4 Support Vector Machines

Support Vector Machine (SVM) [20,4] is known as one of the best supervised learning techniques for solving classification problems with high dimensional feature space and small training set size. The idea is to find the optimal separating hyperplanes that classify data by maximising the geometric margin space between the classes' closest points. SVM is receiving great attention due to some excellent performance it has achieved on real-world applications [9].

4 Experiment Results

The aim of our experiment was to demonstrate feasibility and effectiveness of our approach, and determine the best performing classification method. We tested with real legitimate websites and phishing websites, querying three popular search engines, Google, Bing and Yahoo. The dataset (search results) collected was then trained and classified using the four classification algorithms described above (LDA, NB, KNN, and SVM).

4.1 Description of the Dataset

We quickly noticed that the search results returned for highly targeted websites (e.g. banks, online payment services, and social networks) were strongly distinct from those returned for phishing websites. So rather than investigating the well-established ones (which are relatively easy to classify), we focused more on analysing the performance of classifying new websites/webpages: the legitimate websites we used were only a few days or weeks old, and the phishing websites were a few hours old.

Note, based on our original intuition about the growing reputation of legitimate websites (see Section 1), this relatively new dataset represents the *worst case scenarios* in the experiment; we expect more established websites such as banks to have more distinct characteristics from the phishing websites. We tested our approach under several conditions by changing the search engines and query string formats. The dataset is summarised as follows:

- legitimate websites – 100 URLs of recently launched websites were collected from 'New Websites'[5], which is a showcase for the latest websites;
- phishing websites – 100 URLs of the latest phishing websites were collected from two well-known online phishing URL databases (50 from each), 'Artists Against 419'[6] which contains phishing URLs as well as other forms of scams, and 'PhishTank'[7] which only contains phishing URLs.

[5] http://www.newlywebsite.com
[6] http://www.aa419.org
[7] http://www.phishtank.com

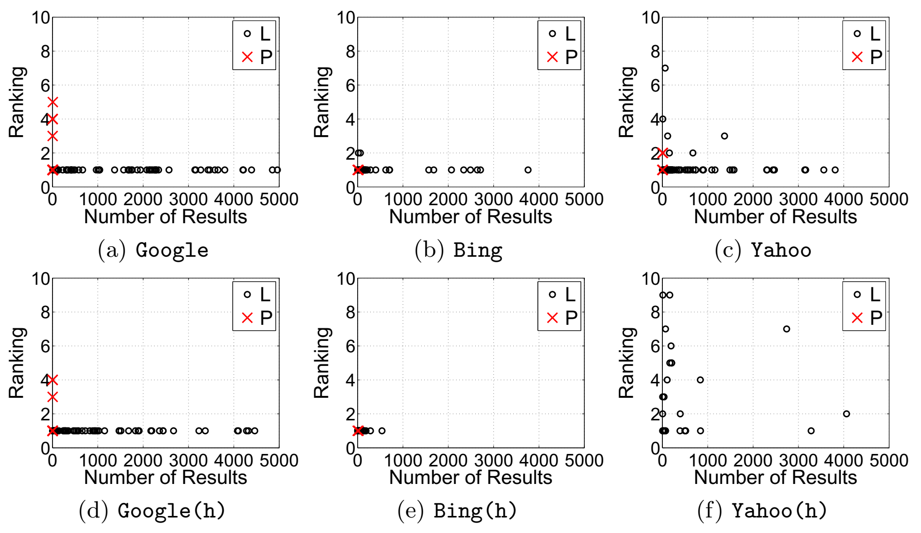


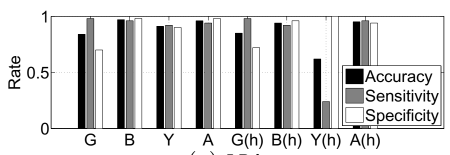
Fig. 3. Scatter plot graphs showing the collected search results. X-axis and Y-axis represent the 'number of results' returned and 'ranking' of the website, respectively.

We tested the three search engines individually (`Google, Bing, Yahoo`) and also observed the performance of combining all of the results together (`All`). With `All`, a total of six attributes, two from each search engine, were used for classification. Having realised that the query string format, whether it includes the string '`http://`' in front or not, could affect the returned results, we used two string formats: the first is 'URL only', e.g. "`www.facebook.com`"; and the second is 'URL with protocol', e.g. "`http://www.facebook.com`".

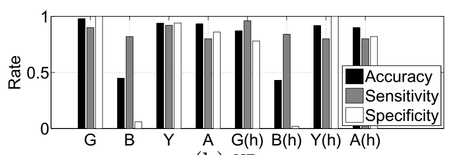
Classifiers generated by querying 'URL with protocol' to a search engine `S` are denoted as '`S(h)`', while classifiers generated by querying 'URL only' is denoted as '`S`' alone. `Google`, for instance, represents search results collected from Google through querying 'URL only', and `Google(h)` represents results collected through querying 'URL with protocol'.

To visually observe the differences between the search results collected for legitimate websites and phishing websites, we use scatter plot graphs with the 'number of results' on the X-axis and the 'ranking' of the website on the Y-axis. These graphs are shown in Figure 3; for better visualisation of the dataset, the X-axis only shows up to 5,000 and the Y-axis up to 10.

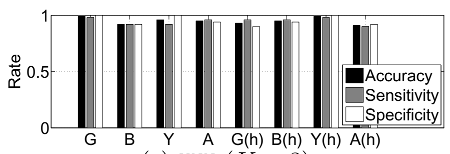
A number of distinct characteristics can be observed from these graphs. First, all of the phishing websites, plotted with a red `X`, have a very small number of returned results (quite a few have none), and most of them are not ranked or ranked below 10 – this explains why many phishing websites are missing from the graphs. Second, the majority of the legitimate websites are ranked first and have large number of returned results; quite a number of legitimate websites that have more than 5,000 results are not shown in the graphs. Interestingly,



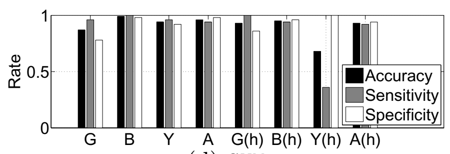
(a) LDA



(b) NB



(c) KNN ($K = 3$)



(d) SVM

Fig. 4. Performance of the four classifiers with respect to *Accuracy*, *Specificity*, and *Sensitivity*

`Yahoo(h)` shows no phishing websites, indicating that most of them are either ranked below 10 or not ranked at all; also, there are many legitimate websites that are not ranked first.

4.2 Classification Results

For classification, we used the first half of the collected search results for training and the rest for testing. The classifiers, as discussed in Section 3, were generated with the first half of the data.

We then assigned the legitimate websites with 'Positive' answers (P) and the phishing websites with 'Negative' answers (N). True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) can be summarised as below:

- TP – legitimate websites correctly classified as legitimate websites;
- FP – legitimate websites incorrectly classified as phishing websites;
- TN – phishing websites correctly classified as phishing websites;
- FN – phishing websites incorrectly classified as legitimate websites.

The graphs in Figure 4 show the performance of the four classifiers using the following three measurements:

- *Accuracy* – the rate of the websites correctly classified ($\frac{TP+TN}{P+N}$);
- *Specificity* – the rate of true negative ($\frac{TN}{TN+FP}$);
- *Sensitivity* – the rate of true positive ($\frac{TP}{TP+FN}$).

We also measured the running time of the classifiers to show the relative efficiency of the classification algorithms; the results are shown in Table 2. These classifiers were implemented using the built-in MATLAB library functions. The PC we used for the experiment was equipped with an Intel quad-core 2.4GHz CPU and 64-bit Windows operating system. The results show that all of the algorithms except

Table 2. Average running time of the classifiers measured in *seconds*

		LDA		NB		KNN		SVM	
		Train	Test	Train	Test	Train	Test	Train	Test
Google	–	0.000s	0.009s	0.110s	0.027s	0.000s	0.019s	0.392s	0.005s
	(h)	0.000s	0.001s	0.020s	0.007s	0.000s	0.009s	0.040s	0.004s
Bing	–	0.000s	0.005s	0.021s	0.007s	0.000s	0.014s	0.057s	0.005s
	(h)	0.000s	0.001s	0.021s	0.008s	0.000s	0.009s	0.007s	0.004s
Yahoo	–	0.000s	0.001s	0.021s	0.008s	0.000s	0.009s	0.089s	0.005s
	(h)	0.000s	0.001s	0.021s	0.007s	0.000s	0.009s	0.057s	0.004s
All	–	0.000s	0.002s	0.058s	0.019s	0.000s	0.009s	0.207s	0.001s
	(h)	0.000s	0.001s	0.057s	0.019s	0.000s	0.010s	0.091s	0.004s
Average	–	**0.000s**	**0.003s**	**0.041s**	**0.013s**	**0.000s**	**0.011s**	**0.118s**	**0.004s**

SVM incur computational overhead that is less than 0.05 seconds for training and less than 0.02 seconds for classifying. We argue that this is a reasonable overhead.

Overall, the KNN ($K = 3$) classification algorithm produced the best results – when 'URL with protocol' was queried on Yahoo (Yahoo(h)), KNN achieved 98% in all accuracy, specificity and sensitivity. Such a high performance implies that each data point, legitimate or phishing, can find two or three neighbours in the same class and that are very closely related. The False Negative rate can also be calculated using $1 - Sensitivity$, which gives a False Negative rate of 2% for Yahoo(h).

The results were somewhat inconsistent for other three classification algorithms. For example, both SVM and LDA performed bad with Yahoo(h) – unlike our expectations, SVM did not outperform other algorithms. Since SVM is one of the most generalised form of linear classifiers, it is likely that Yahoo(h) has a nonlinear property. Although SVM performed well with Bing, when we consider how expensive the tuning phase of SVM is (see Table 2), we would not recommend using SVM. In contrast, NB performed bad with Bing: we speculate that this is because NB relies on each attribute being independent, but the two attributes were not truly independent in Bing. Hence, we would not recommend using Bing or Bing(h) with NB; more obvious recommendations would be to use Google or Yahoo. If one wishes to use these as the classification algorithms, the search engine and/or the query string format should be selected carefully.

In addition, we observed that Yahoo queried with 'URL only' and All search engines combined together can both achieve reasonable and stable performance with any classification algorithm. The linear property of Yahoo allows all of the classification algorithms to perform reasonably well. If one wishes to use All, the response time of querying all three search engines should be carefully evaluated first. In our experiment, the average response time for querying Google with both the phishing and legitimate websites was 0.161 seconds. If we assume that the response time for querying the other two search engines is the same, then the total response time for using All would be 0.483 seconds. On average, it would take about 0.483 seconds (plus the running time for classification as shown above) more to access a webpage. Hence, for any given classification algorithm, the use of `All` should be considered only if it outperforms the best performing single search engine.

5 Related Work

Over the years, a wide range of phishing detection techniques have been proposed and deployed. One of the most used techniques seems to be blacklisting. Most of the anti-phishing applications available, including those built into mainstream web browsers, use blacklists for detecting phishing sites. Typically, a URL database of known-bad websites is managed and used for detecting phishing websites.

These blacklisting approaches [19], however, only provide a partial solution with partial list of global phishing websites, and are not completely effective against new phishing websites. To make the matters worse, the majority of the phishing websites are short-lived (e.g. lasting hours) and hundreds of new ones appear everyday [15], making it difficult to update and check against the central databases.

In contrast, whitelists manage a list of known-good websites. Whitelists are generally divided into global lists updated by central servers and personalised lists managed by the end users as needed. Due to its inherent usability issues, whitelists are currently used only in the preprocessing step, i.e. before the heuristics are checked, to reduce false positives. Kirda and Krugel [12] have developed a browser extension called AntiPhish to maintain trusted websites' domain names and credentials. Cao et al. [2] have proposed a method for constructing the user's personalised whitelists using the Naïve Bayesian classification algorithm.

One of the main limitations with using blacklists and whitelists is that they can only classify previously-known phishing or legitimate websites. Inevitably, these lists are not very effective when it comes to identifying a newly formed website. To overcome this limitation, many heuristics-based techniques have been proposed that analyse the HTML structure, URL, domain name, and webpage contents [3,16,7,24,22]. These methods are capable of achieving true positive rates between 85% and 95%, and false positive rates between 0.43% and 12%. Here, the most challenging task is to extract the right features that correctly identify phishing websites. When this task is not managed properly, they will incur false positives and false negatives.

To minimise false positives and false negatives, Ronda et al. [18] have proposed a user-assisted system that uses visual previews of the original web form and the form the user intends to access — the user is expected to visually compare the two previews and make a decision. Considering that many phishing websites are now highly effective in replicating the original websites and forms, how much this type of approach will add to the overall usability is questionable.

The majority of researchers have used the webpage contents to measure the likelihood of phishing. For example, CANTINA [24] determines whether a webpage is dangerous by scrutinising webpage features like 'consistency of well-known logos' and 'use of HTML form tags'. Again, this type of approach has limitations: if the heuristic looks for common properties of phishing websites, the phishers can simply avoid these properties when designing new ones. In practice, most of these ideas are implemented as toolbars (built-in web browsers), showing different types of security messages to help users mitigate phishing attacks [3,14]. Some of the browser anti-phishing mechanisms were compared and evaluated in [23]. More recently, Kim and Huh [11] have proposed a method for identifying phishing websites using the network performance characteristics such as the round-trip time. This method is capable of achieving true positive rate of 99.4% and false positive rate of 0.7%.

Bain et al. [10] have evaluated a number of online resources in assisting phishing detection. In particular, they have demonstrated that Google PageRank and Yahoo! Inlink, which rank websites according to the number of in-links, can be effective in identifying phishing websites. The domains of legitimate websites usually have in-links from credible websites whereas the phishing websites do not. However, one concern is that such ranking systems *only* use the in-link information to rank a website. Many newly formed legitimate websites, which are not so much exposed to the public, may not have any (or very small number of) in-links from credible websites. This will introduce high false positives – that is, legitimate websites appearing as phishing websites. Our approach, by submitting the website URL as the search string, also checks whether there is a cached website/webpage with the exact same URL. Most of the time, the search engines we tested with, Google, Yahoo, and Bing, ranked a website *first* in their search results if the queried URL exactly matched this website's URL, regardless of the number of in-links it may have. Hence, even the new legitimate websites that have small number of in-links are likely to be ranked very high using our approach. We have demonstrated this in Table 1: even from day 1, most of the legitimate websites are ranked first; the phishing websites are not ranked at all.

Moreover, Google PageRank is updated approximately every 3 months. This means a website may have to wait 3 months for its ranking to change: a legitimate website that is currently ranked low (or not ranked) and is misclassified as a phishing website, would continuously be misclassified until its ranking goes up after 3 months. Our approach, on the other hand, uses the latest information returned straight from the search engine, which is updated daily or even hourly. Table 1 shows how quickly such information can change, and the need to capture the latest information possible to make an accurate decision.

6 Conclusion and Future Work

Existing phishing detection techniques based on blacklists, whitelists, and various heuristics tend to achieve low performance when it comes to classifying new, recently launched websites. To address this weakness, we proposed a heuristic-based technique that uses the reputation of the URL of a website, a property that can be measured easily and quickly through any popular web search engine. Our experiment results show that, when a search engine is queried with the website's full URL as the search string, the legitimate websites tend to get back large number of results and are ranked first most of the time. The phishing websites, on the other hand, tend to get back very small number of results (one or two) and are often ranked below 10 or not ranked at all.

For performance evaluation, we used three most popular search engines, Google, Bing and Yahoo, to measure the reputation of 100 new legitimate websites and 100 new phishing websites. These recorded values were experimented with several classification methods, Linear Discriminant Analysis, Naïve Bayesian, K-Nearest Neighbour, and Support Vector Machine algorithms, demonstrating high accuracy in classifying the new websites.

Although we relied on a relatively small sample pool, our best performing classification method outperformed existing heuristics (these achieve true positive rates between 85% and 95% and false positive rates between 0.43% and 12%); and unlike previous experiments, our experiment focused more on analysing the performance of classifying relatively new websites. Hence, we expect the performance of classifying more established websites such as banks and social networks to be significantly higher.

There are a few things that a phisher might try to bypass our system: the phisher could provide a legitimate website for a while (to build a good reputation) and then abuse this reputation, or compromise a legitimate website (that already has a good reputation) and replace it with a phishing website. But these attacks would be expensive, and perhaps forcing the phishers to perform such costly attacks is a good thing. The phisher may also try 'gaming' the search engines to manipulate their ranking/reputation algorithms, and such attacks may turn out to be cheaper to perform than working around existing phishing filters. However, as the phishing attacks become more sophisticated, we imagine that a *combination* of anti-phishing mechanisms would be used in the future, rather than just relying on one particular solution. Some combination of our method and other existing heuristics may be used to provide a highly effective solution.

As part of the future work, we plan to increase the size of the dataset and investigate any change in performance. Moreover, we would like to try these classification methods on a more fresh set of phishing websites which are yet to be registered on online databases. It would also be interesting to analyse how different combinations of URL formats (e.g. leaving some parameters in the URL) affect performance.

Acknowledgements. The authors would like to thank Richard Clayton and Dongting Yu for their constructive comments and suggestions. The authors would also like to thank the anonymous reviewers for their careful attention and insightful comments.

References

1. Aaron, G., Rasmussen, R.: Global phishing survey: Trends and domain name use in 2h2009 (May 2010), `http://www.antiphishing.org/reports/APWG_GlobalPhishingSurvey_2H2009.pdf`
2. Cao, Y., Han, W., Le, Y.: Anti-phishing based on automated individual white-list. In: DIM 2008: Proceedings of the 4th ACM Workshop on Digital Identity Management, pp. 51–60. ACM, New York (2008)

3. Chou, N., Ledesma, R., Teraguchi, Y., Mitchell, J.C.: Client-Side Defense Against Web-Based Identity Theft. In: NDSS 2004: Proceedings of the Network and Distributed System Security Symposium (2004)
4. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines: and other kernel-based learning methods, 1st edn. Cambridge University Press (March 2000)
5. Domeniconi, C., Peng, J., Gunopulos, D.: Locally adaptive metric nearest-neighbor classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 1281–1285 (2002)
6. Domingos, P., Pazzani, M.: On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. Machine Learning 29(2-3), 103–130 (1997)
7. Fu, A.Y., Wenyin, L., Deng, X.: Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover's Distance (EMD). IEEE Transactions on Dependable and Secure Computing 3(4), 301–311 (2006)
8. Fukunaga, K.: Introduction to statistical pattern recognition, 2nd edn. Academic Press Professional, Inc., San Diego (1990)
9. Hearst, M.A., Dumais, S.T., Osman, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intelligent Systems and their Applications 13(4), 18–28 (1998)
10. Bian, K., Park, J.-M., Hsiao, M.S., Belanger, F., Hiller, J.: Evaluation of Online Resources in Assisting Phishing Detection. In: Ninth Annual International Symposium on Applications and the Internet, SAINT 2009, pp. 30–36. IEEE Computer Society, Bellevue (2009)
11. Kim, H., Huh, J.H.: Detecting DNS-poisoning-based phishing attacks from their network performance characteristic. Electronics Letters 47(11), 656–658 (2011)
12. Kirda, E., Kruegel, C.: Protecting Users Against Phishing Attacks with AntiPhish. In: COMPSAC 2005: Proceedings of the 29th Annual International Computer Software and Applications Conference, pp. 517–524. IEEE Computer Society, Washington, DC, USA (2005)
13. Latourrette, M.: Toward an Explanatory Similarity Measure for Nearest-Neighbor Classification. In: Lopez de Mantaras, R., Plaza, E. (eds.) ECML 2000. LNCS (LNAI), vol. 1810, pp. 238–245. Springer, Heidelberg (2000)
14. Liu, W., Deng, X., Huang, G., Fu, A.Y.: An Antiphishing Strategy Based on Visual Similarity Assessment. IEEE Internet Computing 10(2), 58–65 (2006)
15. Moore, T., Clayton, R.: Examining the impact of website take-down on phishing. In: eCrime 2007: Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit, pp. 1–13. ACM, New York (2007)
16. Pan, Y., Ding, X.: Anomaly Based Web Phishing Page Detection. In: ACSAC 2006: Proceedings of the 22nd Annual Computer Security Applications Conference, pp. 381–392. IEEE Computer Society, Washington, DC, USA (2006)
17. Rish, I.: An empirical study of the naive Bayes classifier. In: Proceedings of IJCAI-2001 Workshop on Empirical Methods in Artificial Intelligence (2001)
18. Ronda, T., Saroiu, S., Wolman, A.: Itrustpage: a user-assisted anti-phishing tool. ACM SIGOPS Operating Systems Review 42(4), 261–272 (2008)
19. Sheng, S., Wardman, B., Warner, G., Cranor, L.F., Hong, J., Zhang, C.: An empirical analysis of phishing blacklists. In: CEAS 2009: Proceedings of the 6th Conference on Email and Anti-Spam (2009)
20. Vapnik, V.N.: Statistical Learning Theory. Wiley-Interscience (September 1998)
21. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.-H., Steinbach, M., Hand, D.J., Steinberg, D.: Top 10 algorithms in data mining. Knowledge and Information Systems 14(1), 1–37 (2007)

22. Xiang, G., Hong, J.I.: A hybrid phish detection approach by identity discovery and keywords retrieval. In: WWW 2009: Proceedings of the 18th international conference on World wide web, pp. 571–580. ACM, New York (2009)
23. Zhang, Y., Egelman, S., Cranor, L., Hong, J.: Phinding Phish: Evaluating Anti-Phishing Tools. In: NDSS 2007: Proceedings of the 14th Annual Network and Distributed System Security Symposium (2007)
24. Zhang, Y., Hong, J.I., Cranor, L.F.: Cantina: a content-based approach to detecting phishing web sites. In: WWW 2007: Proceedings of the 16th International Conference on World Wide Web, pp. 639–648. ACM, New York (2007)