SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2015; **8**:176–188
Published online 10 March 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.970

RESEARCH ARTICLE

A link graph-based approach to identify forum spam

Youngsang Shin[1]*, Steven Myers[1], Minaxi Gupta[1] and Predrag Radivojac[1]

[1] School of Informatics and Computing, Indiana University, 150 S. Woodlawn Avenue, Bloomington, IN, 47405, U.S.A.

ABSTRACT

Web spammers have taken note of the popularity of public forums such as blogs, wikis, webboards, and guestbooks. They are now exploiting them with the purpose of driving traffic to their malicious or fraudulent websites, such as those used for phishing, distributing malware, or selling counterfeit pharmaceuticals. A popular technique they use is to spam these forums with URLs to their spam websites. We consider the problem of classifying URLs posted to forums as spam or legitimate by considering the link structure of the graph rooted at the posted URL. We investigate various graph metrics and associated metadata to analyze link structures. To lessen noisy structural characteristics of the link graphs for spam classification, we also examine two techniques: differing depths and aggregating sub-graphs of the link graphs. Our results show that a support vector machine classifier based on combinations of graph metrics and metadata of link graphs can achieve a pragmatically high performance in forum spam detection. Copyright © 2014 John Wiley & Sons, Ltd.

KEYWORDS
forum spam; web spam; link graph; security

***Correspondence**
Youngsang Shin, Korea Internet and Security Agency (KISA), IT Venture Tower West Wing 3rd floor, 135 Jungdae-ro, Songpa-gu, Seoul,138-950, South Korea.
E-mail: ysshin@gmail.com

[†] Shin participated in this work while he was a PhD student at Indiana University. He is currently with Korea Internet and Security Agency, Seoul, South Korea.

1. INTRODUCTION

The Web is large, consisting of 632 million websites as of March 2013 [1]. Given its size, it is no surprise that attracting traffic to websites is challenging. Consequently, website operators are always on the lookout for new mechanisms to make their websites discoverable. This is particularly true for operators of fraudulent and malicious websites, because they do not serve any useful content that would naturally attract visitors. While email spam with links continues to be a popular technique for driving traffic to such sites, increasingly alternate means are being deployed. One such popular technique is *forum spamming*, where spammers post links to their websites at forums frequented by Internet visitors. These forums include webboards, blogs, wikis, guestbooks, and other sites where visitors can post content that can be viewed by other visitors to the site.

Forum spam offers the obvious benefit of providing direct traffic to malicious sites. Additionally, it can increase the number of links from potentially legitimate forums that point to a targeted malicious site. This in turn can boost the search engine rankings of malicious websites, as search engines often use the number of incoming links from other legitimate sites as a component in their ranking algorithms such as in *PageRank* [2]. Thus, forum spamming can be viewed as a specific type of *web spamming*, where unsavory techniques are used to obtain undeserved high rankings in search engines with respect to other websites. These techniques include *keyword stuffing* (placing popular but irrelevant keywords on a web page to improve its relevance to topics related to those keywords), *cloaking* (fooling search engines' crawlers by showing them different content than what page visitors see), and *link farming* (forming groups of heavily hyper-linked websites to exploit ranking algorithms used by search engines).

With forum spamming becoming popular, countermeasures for this attack warrant development. However, common measures forum operators normally take, including registration and CAPTCHA [3], are not practically effective enough because forum spam automators such as `XRumer` [4] can effectually defeat them [5]. Furthermore, the countermeasuring task becomes more challenging because many forums are legitimate and serve useful content. Consequently, they cannot simply be taken down by ISPs or blacklisted by search engines as a defense measure. Given its connection with web spamming, an obvious approach to defend against forum spam would be to apply the same mitigation techniques as are used to counter web spam. However, this approach is unlikely to be successful in identifying all types of forum spam, as the goals of forum spammers differ from those of web spammers. While web spammers exploit techniques like keyword stuffing, cloaking, and link farming to improve the likelihood that their sites show up higher in searches, forum spammers are less likely to be concerned about exploiting ranking algorithms used by search engines, because they can potentially attract forum visitors directly. In fact, Wang *et al.* [6] found direct evidence that forum spam pages were not part of link farms. Instead, they observed that forum spam links redirected users to web pages containing pay-per-click programs or to pages that acted as advertising portals.

There are two natural and complementary approaches for defending against forum spam. First, forum operators can take measures to protect their forums against it by using information they have access to, such as spammer behavior. Indeed, that is the approach we took in our previous work [7]. The defense mechanism developed in this paper is suited for organizations that have efficient access to web graphs, such as search engines. Specifically, we explore classifying forum spam based on the link structure of URLs found in forum spam. We investigate if the link structure of forum spam URLs possesses distinguishing characteristics that can be used to develop a classifier to defend against forum spam. Toward this goal, we crawl the Web using known forum spam URLs as seeds and build URL-based *link graphs*. In such a graph, each node is denoted by a URL, and each edge a link between URLs. The resulting graph may include both spammer-controlled and even legitimate websites because many malicious sites tend to contain links to good websites.

We analyze these link graphs in order to identify distinctive features that can be employed in a classifier. To explore the structural properties of the resulting link graphs, we examine five popular graph metrics: betweenness, degree, in-degree, out-degree, and clustering coefficient. Additionally, we examine metadata of the link graphs, including the number of nodes, edges, domains, and hosts; the Hypertext Transfer Protocol (HTTP) status codes for each URL; and type of edges. We motivate these choices and give specific definitions of the features in Section 3.

When forming the link graphs, we often encounter a large number of URLs (nodes) belonging to legitimate web hosts and links between such URLs (edges). These nodes and edges act as 'noise' that can overwhelm any malicious 'signal' that is contained in the link structure of malicious URLs embedded within. To lessen such effects, we also study the effect of collapsing sub-graphs belonging to legitimate web hosts. The results are collapsed graphs containing two types of nodes: URLs and hosts. We refer to these graphs as *hybrid link graphs*.

For the classification, we use support vector machine (SVM) classifier with linear kernels. We train them on features based on graph and metadata metrics. We find that using hybrid link graphs improves classification. Among the graph metrics, degree, in-degree, and out-degree features show the most distinctive characteristics in classifying forum spam. This is important from a practical perspective, as these can be efficiently computed, as opposed to less parallelizable metrics, such as betweenness. Furthermore, metadata metrics serve as a valuable complement to the graph metrics. We conclude that an SVM classifier based on the graph metrics and metadata can achieve high performance with a 98.87% precision and 92.78% recall. Further, once trained, such a classifier could efficiently run on massively parallel computing infrastructure, given efficient access to a web graph. This is exactly the type of infrastructure that search engine organizations possess.

2. DATA COLLECTION AND OVERVIEW

To collect forum spam URLs, we retrieved blog comments at an active blog maintained by the security research group at the Computer Laboratory at University of Cambridge.[†] The blog is built on the WordPress [8] software platform. It uses two mechanisms for filtering spam: (i) posters must provide a properly parsable email address prior to commenting, and (ii) it runs the Akismet [9] plugin for forum spam filtering. The details of Akismet's classification algorithm are proprietary, and its accuracy has not been formally established. However, any false positives or false negatives in our data set have already been accounted for by the blog's administrator.

Even though the comments in our data set do not contain any false positives, spam comments may still include non-spam URLs in addition to spam URLs. This is because forum spammers often intentionally insert non-spam URLs in their comments in order to make their comment appear legitimate. To remove such non-spam URLs, we manually investigate each URL to ensure it indeed is spam. To be safe, we exclude any URLs inaccessible during our sanitizing step from further consideration. While automating the sanitization process is conceivable, we kept it manual, mainly because many inaccessible URLs resulted in a redirected web page containing non-standard error messages to denote that the page was not found instead of returning the standard HTTP 404 error code as one would expect.

We begin by extracting URLs from both spam and legitimate comments and labeling them appropriately. We use each collected URL as a seed and crawl the Web to build link graphs. Toward this goal, we use a custom crawler written in Python. For each such URL, our crawler follows all links, including redirections, at the retrieved HTML page in a breadth first manner, up to a specified depth. The structural properties of link graphs are affected by how deep we crawl from a given seed URL. Intuitively, forum spam sites should not make use of a deep structure, as it is unlikely that users would "click" deeply on to another site,

[†] http://www.lightbluetouchpaper.org.

following the proverbial rabbit down its hole. Additionally, deep structures have little influence on web rankings, providing another disincentive for forum spammers to follow this strategy.

The resulting link graph consists of the following:

(1) A set N of nodes that are URLs followed during crawling, and
(2) A binary relation E on N. E denotes a hyperlink between nodes.

In this paper, we define the 'depth' of link graphs as the count of links (including automatic redirections), which our crawler follows from a root node. We count multiple successive redirections as one. By default, we crawl to a depth of four. However, we build link graphs for crawls of depths of two and three for comparison purposes. For each URL encountered, our crawler logs its URL, parents, and HTTP status code to infer availability and redirections. To minimize the effects of cloaking, our crawler mimics HTTP headers used by a popular web browser, `Firefox`.

We collected 2656 spam and 89 legitimate URLs during two periods totaling nine weeks. The data collection ended on 27 February 2010. Of these, we successfully crawled 2245 spam and 85 non-spam URLs. This also means that we built 2245 and 85 link graphs with spam URLs and non-spam URLs as seed URLs, respectively.

As previously alluded to, the resulting graphs may contain large numbers of URLs and interconnected links belonging to legitimate web hosts. If the resulting legitimate structure dominates the graphs, it can make the classification of spam versus legitimate URLs more difficult. In order to prevent "legitimate" sub-graphs from suppressing distinguishing characteristics of forum spam, we collapse sub-graphs belonging to a whitelist of known legitimate hosts into single nodes. The whitelist was derived from the top one million hosts in the Alexa list [10] of popular URLs. Note that all the links to/from any spam URL belonging to legitimate web hosts are still retained even after the collapsing. These graphs are referred to as *hybrid link graphs*. Hybrid link graphs thus contain two types of nodes: hosts for sub-graphs of URLs belonging to whitelisted hosts and URLs for all others. Figure 1 shows an example of URL-based link graph and its collapsed hybrid URL-based link graph. We studied classification results for both regular URL-based link graphs and the resulting hybrid link graphs.

In Tables I(a) and I(b), we show the average number of nodes and edges of URL-based and hybrid link graphs, respectively. A few things are noteworthy in these tables.

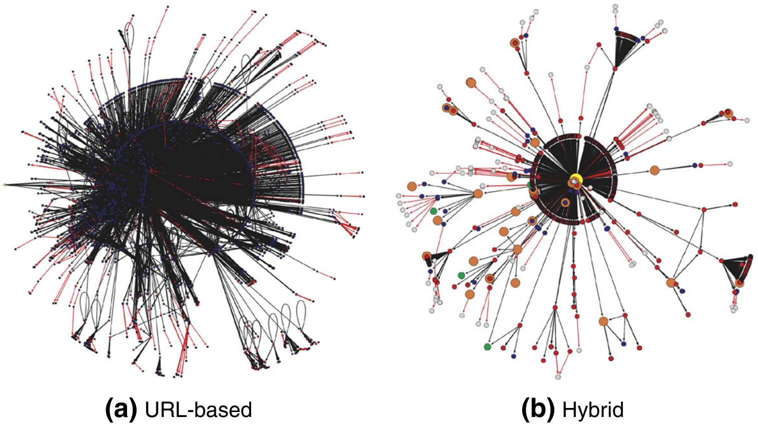


Figure 1. Example of (a) URL-based link graph and (b) its collapsed/hybrid graph.

Table I. Overview of data: There were a total of 2245 spam and 85 legitimate URLs.

	Avg. # of nodes		Avg. # of edges	
Depth	Spam	Non-spam	Spam	Non-spam
(a) URL-based link graphs				
2	330	1,457	937	4,343
3	1,216	6,863	10,805	19,033
4	7,237	93,621	32,491	316,154
(b) Hybrid link graphs				
2	161	397	517	724
3	468	1,541	8,907	2,523
4	1,803	14,581	13,673	39,172

DOI: 10.1002/sec

First, each increase of the depth of crawls expands graph sizes by an order of magnitude. Second, for a given depth, the average number of nodes and edges in non-spam graphs are an order of magnitude more than their spam counterparts. Finally, as expected, the number of nodes and edges in the collapsed hybrid link graphs is smaller than their uncollapsed counterparts for both spam and non-spam seed URLs.

3. LINK GRAPH METRICS

Here, we present the two categories of metrics we use in classifying forum spam. The first category contains metrics based on graph properties of the link graphs. The second is based on metadata about nodes and edges in link graphs.

3.1. Graph metrics

Web pages linked through forum spam tend to contain deliberately crafted links to other web pages just as in the case of link spam [11]. These links might be directed to other spam web pages owned by the spammer or even legitimate web pages. In the first case, spammers may wish to keep visitors stay within their own website(s). In the latter, they may try to make visitors believe that their web pages are legitimate. Thus, spam web pages may have palpable characteristics such as a relatively large number of links or a tight connection with their neighbors. To understand such characteristics, we pick standard graph metrics that measure the degree of such clustering. Broadly, the graph metrics we use fall into three categories. The first category consists of three metrics based on the degree of each node in the graph. These metrics quantify how connected in a graph a node is. The first metric in this category, *degree*, is the number of edges incident to a node. We pick this metric because our data showed that link graphs rooted at forum spam URLs have higher degrees, perhaps because of their designers' desire to keep victims within spam pages. Because links are directed, we also separately measure *in-degree* and *out-degree*, the number of incoming and outgoing edges to a node. The second category of graph metrics consists of *betweenness*, which is a measure of centrality of a node within a graph or how many shortest paths in the graph traverse a given node. A node with high betweenness denotes a hub or central node in the graph. Our data showed that forum spam URLs have higher betweenness than non-spam URLs. We suspect that this is due to spammers trying to direct victims toward specific web pages, such as those containing a specific malware payload. The third and final category of metrics contains the *clustering coefficient*. It measures the degree to which all neighbors of a given node in a graph tend to interconnect. This metric can show the degree of local interconnectivity among URLs. We chose this metric based on our observation that forum spam pages have high interconnectivity among their pages, perhaps to boost search-engine rankings.

Table II. Graph metrics.

Metric	Definition
Degree	# of links incident upon a node
In-degree	# of links incoming to a node
Out-degree	# of links outgoing from a node
Betweenness	$\sum_{s \neq n \neq t \in N} \frac{\sigma_{st}(n)}{\sigma_{st}}$
Clustering coefficient	$\frac{\lvert\{e_{jk}\}\rvert}{k_i(k_i-1)} : v_j, v_k \in N_i, e_{jk} \in E$

Table II lists the metrics used, along with their formal definitions. The notation used in Table II is as follows: $G = (V, E)$ is a graph with nodes V and edges E. Denoted by e_{ij} is edge $(i,j) \in E$. The neighborhood N for a node $i \in V$ is denoted as $N_i = \{v_j : e_{ij} \in E \vee e_{ji} \in E\}$. Let $k_i = |N_i|$. Let σ_{st} be the number of shortest paths from s to t in G, and $\sigma_{st}(n)$ be the number of shortest paths from s to t that pass through n. Note that each metric applies to each node in a link graph. Thus, for each metric, we have a set of corresponding values whose cardinality is the same as the number of nodes in the graph.

3.2. Metadata metrics

The graph metrics explained in Section 3.1 capture only connectivity information among nodes of a link graph. To complement this information, we characterize nodes and edges of each link graph through the metadata metrics (Table III).

The first metadata-based metric, M_1, is a pair consisting of the total number of nodes and edges in the link graph. The motivation for this metric comes from Table I, which shows that the average number of nodes and edges in link graphs rooted at forum spam URLs tend to be smaller than those of their non-spam counterparts. Metric M_3 extends this notion to the number of unique hosts and domains contained in the link graph.

Metric M_2 contains information about the HTTP status code [12] returned by the web server for the URLs of the link graph. A web server returns an HTTP status code when a web client requests a specific URL to the server. For example, a status code in the 200 range indicates that the request has succeeded. Similarly, a status code of 301 denotes that the requested URL has been permanently moved and assigned a new permanent URL. Any future reference to the page should use one of the returned URLs. HTTP status codes may indicate aspects of how a web server is configured. The justification for this metric comes from the observation that more forum spam URLs are redirected than legitimate URLs. For each link graph, a 28-dimensional vector is returned, one value for each status code. Each element of the tuple refers to the fraction of nodes in the graph that return that corresponding status code. For hybrid link graphs, we do not include the HTTP status codes of the collapsed nodes in the tuple.

The rest of the six metadata metrics relate to edges in link graphs. They are motivated by the desire to capture

Table III. Metadata metrics.

Set ID	Description
M_1	# of nodes and edges
M_2	Proportion of nodes returning specific HTTP status codes
M_3	# of domains and hosts
M_4	Ratio of out-links to node's domain versus to all domains
M_5	Ratio of in-links to node's domain versus to all domains
M_6	Ratio of links to node's domain versus to all domains
M_7	Ratio of out-links to Alexa domains versus to all domains
M_8	Ratio of in-links to Alexa domains versus to all domains
M_9	Ratio of links to Alexa domains versus to all domains
M_{10}	Ratio of forms to same domain versus to all domains

HTTP, Hypertext Transfer Protocol.

linking strategies used by forum spammers. We focus on edges connecting different domains because they are likely to be the ones showing the strategies used by operators of malicious domains. In M_4, M_5, and M_6, we capture the ratio of links to different domains of nodes of the graph. Specifically, for each node v, M_6 measures the ratio of the number of links to the same domain as v, to the number of all domains. M_4 and M_5 capture the same ratios restricted to looking at only outgoing and incoming links, respectively. Metrics M_7, M_8, and M_9 are similar in spirit, with the only difference being that the numerator in the ratio is for links going to Alexa domains and the denominator is the sum of links to Alexa and non-Alexa domains. Note that the cardinality of each of these metrics is the number of nodes in a link graph or less, the latter being in the case where adjacent nodes belong to the same host; in which case, we collapse them from the perspective of computing these metrics.

The last metadata metric, M_{10}, captures metrics similar to M_4–M_9 but only in the context of HTML forms. Specifically, for each node, if the page contains HTML forms with an action, this metric contains the ratio of forms to different domains versus all forms. If a node has no forms with action, this metric has the default value of zero. The motivation for this metric comes from the observation that many spam websites tend to provide secure HTTP Secure (HTTPS) connections. However, they tend to direct all visitors to one back end server. This is presumably in an attempt to manage the costs of procuring HTTPS certificates.

4. CLASSIFICATION METHODOLOGY

We model forum spam classification as a binary classification problem, classifying each link graph as spam or legitimate using features based on metrics defined in Section 3. We use an SVM as our classifier. Specifically, we run SVM^{light} [13] with a linear kernel function and default capacity parameter. In the later text, we describe how we derive features out of the metrics and how we judge the performance of our classifier.

4.1. Feature vector representation of metrics

Our metrics are represented as sets with different cardinalities. Some metrics, for example, M_1, have a cardinality of two while several others, for example, the *degree* metric, have a cardinality of the number of nodes in a link graph. The cardinality of the latter varies with the number of nodes in a link graph. Because a classifier must have the same number of features corresponding to each data point, the obvious strategy of taking a metric and making a feature out of each element of its set is infeasible. Thus, we adopt two strategies to create feature sets from metrics described in Section 3. For metadata metrics with a variable number of features, we take a mean value per host in the link graph and then use the average of those mean values as the feature. Metadata metrics, M_1, M_2, and M_3, have the same cardinality for each link graph, so we simply take each of their two elements for M_1 and M_3 and 28 elements (the number of all of the observed HTTP status codes) for M_2 and compute features from them.

The strategy we use for metadata metrics is less than ideal for graph metrics, because it would cause much information on the distribution of the metric values over nodes to be lost. Hence, for graph metrics, we denote the distribution of values for each metric as a cumulative distribution function (CDF). Each value in a CDF representation corresponds to the fractions of nodes (or edges) in a link graph with values lower than particular threshold. Then we take *quantiles* [14] from the CDFs. This means that the CDFs are then sampled in regular intervals between 0 and 1. The discrete values of sampled CDF values, *quantiles*, are used as a feature vector.

There are two issues that dictate *quantile* sampling. First, the effectiveness of features so created depends on the number of *quantiles*. We found that creating 10 or 100 *quantiles* was ineffective, while 10000 *quantiles* could lead to over-fitting because the total number of the input link graphs in our data is only 2330. As a result, we chose to sample each CDF 1000 times, which created feature vectors with 1000 elements each corresponding to one of the five graph metrics. The second issue with sampling relates

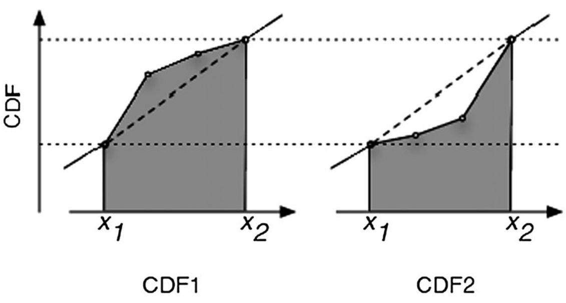


Figure 2. Example of sampling for two CDFs.

to the shape of the CDF itself. Figure 2 illustrates the issue. If only two points, x_1 and x_2, are sampled from each of the CDFs, they will fail to capture the fact that the shape of these CDFs is different because the y-axis values for both points x_1 and x_2 are the same. To avoid this aspect from affecting our features, we use the area between consecutive sample points as features.

4.2. Classifier performance

Generally, an SVM classifier maps the labeled training data to a high-dimensional feature space and separates the two classes of data with a maximum margin hyperplane. Its output corresponds to a signed distance between a feature vector and the learned hyperplane, with the magnitude indicating the strength of the prediction. In order to decide what values of the SVM output correspond to spam (class label +1) or legitimate URLs (class label –1), one needs to apply a *decision threshold*, which is then used to control the fraction of false positive versus false negative predictions. Specifically, a high decision threshold can minimize false positives at the cost of increasing false negatives.

To assess the accuracy of classifier performance, we estimate the fractions of correct and incorrect predictions for each class. We then use the area under the receiver operating characteristic (ROC) curve to compare accuracies across classifiers. The ROC curve shows true positive rate (also referred to as recall) as a function of the false positive rate and is usually plotted for a number of different decision thresholds (see Figure 3 for definitions). An area under the curve (AUC) of 1 corresponds to a perfect classifiers, while an AUC of 0.5 corresponds to a random classifier. While an AUC has a useful probabilistic interpretation, because its value corresponds to the probability that a randomly selected positive example will be ranked higher than a randomly selected negative example, it aggregates the performance over all decision thresholds. Thus, it is also useful to emphasize parts of the ROC curve with low false positive rate (thus, high precision) where the confidence of a positive prediction is high. Such performance can be expressed as recall for a pre-specified value of precision. High values of precision minimize the number of good URLs classified as spam.

		Classified class	
		Non-spam	Spam
Correct class	Non-spam	*True Negative*	*False Positive*
	Spam	*False Negative*	*True Positive*

$$True\ positive\ rate = \frac{True\ Positives}{(True\ Positives + False\ Negatives)}$$
$$False\ positive\ rate = \frac{False\ Positives}{(False\ Positives + True\ Negatives)}$$
$$Precision = \frac{True\ Positives}{(True\ Positives + False\ Positives)}$$
$$Recall = \frac{True\ Positives}{(True\ Positives + False\ Negatives)}$$

Figure 3. Confusion matrix and definitions used in the context of SVM classification.

Finally, the performance of the classifier was estimated using a stratified 10-fold cross-validation [15]. Briefly, the data are randomly split into 10 non-overlapping partitions (each containing one tenth of the positives and one tenth of the negatives). In each of the 10 steps, the i-th partition ($i \in 1, 2, \ldots, 10$) is used for testing while the remaining partitions were used for training. The model is then trained, and the prediction scores are calculated on the test data points. After 10 steps, each data point will contain one prediction value and its true class. These values are then used to compute the ROC curve. Note that we also applied a z-score data normalization [16], where the mean and standard deviation for each feature were calculated on training data only and then applied to the test data.

5. CLASSIFICATION RESULTS

Here we discuss the results of our forum spam filtering algorithms and the impact of various parameters on classification, including depth of crawl, collapsing of link graphs, and feature sets.

5.1. Performance of individual graph metrics

We start by testing the impact of each graph metric on the performance of the classifier. Toward this goal, we train an SVM on the features of each metric. We do so for URL-based as well as hybrid link graphs of depths two, three, and four. Figure 4 shows the performance of the classifier for each graph metric. First, we note that the three degree-based metrics perform the best at all crawl depths. In fact, the AUC for the classifier trained on the clustering coefficient metric is close to 0.5, indicating that its performance is close to a classifier that guesses randomly. Second, all but the clustering coefficient metric perform best on graphs of depth three. The fact that the link graphs of depth four from the root node did not result in improved accuracy suggests that *it was not necessary to explore deeper graph structures for this problem and the data under study*. Owing to this observation, we use graphs with a crawl depth of three for subsequent analysis.

Next, we plot the ROC curves for each of the graph metrics. Note that our application calls for high true positive rates but not at the cost of high false positive rates, for we would not want to penalize good URLs on forums for

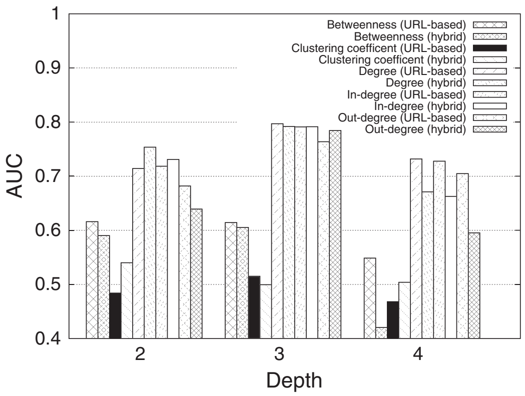


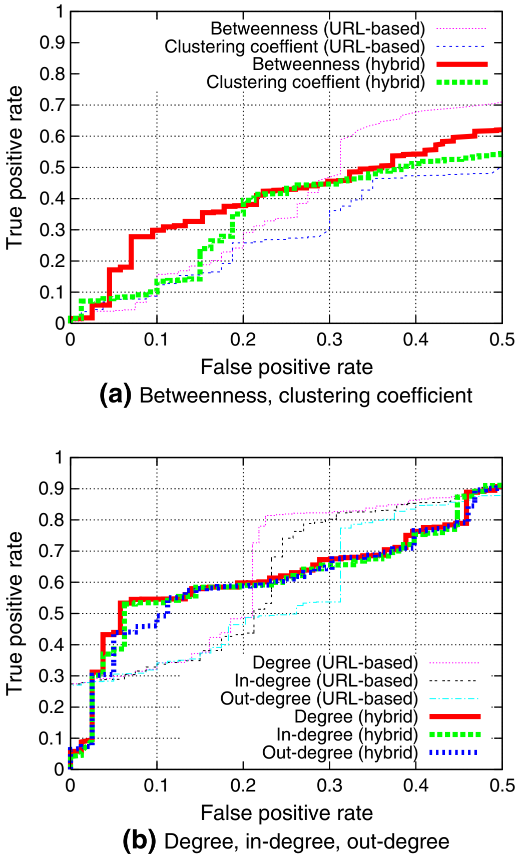
Figure 4. Impact of depth of crawl on classifier performance using graph metrics.

the sake of finding forum spam. In particular, we stipulate that a good classifier would keep false positive rates low while maximizing the true positive rate. Figure 5 shows the ROC curves corresponding to each of the graph metrics for both URL-based and hybrid link graphs. The key observation from this figure is that *hybrid link graphs offer higher true positive rates for relatively lower false positive rates*. Because of this reason, we primarily examine the hybrid link graphs subsequently in this paper.

5.2. Performance of individual metadata metrics

Here, we examine the performance of individual metadata metrics. Figure 6 shows the performance of SVM classifiers trained on individual metadata metrics. We note that metric M_2, whose feature vector contains the proportion of graph nodes containing specific HTTP status codes, shows the best performance, closely followed by M_3, which contains the number of unique hosts and domains in a graph. M_3's equivalent, M_1, which contains the number of nodes and edges in hybrid link graphs, also has good performance. Beyond these three, only M_4 and M_7 perform better than random. These are both vectors containing the ratio of out-links to the same domain as the node / Alexa domains versus all out-links for each node in a graph, indicating that *out-links are more important to study in the context of forum spam URLs*. This is intuitive, because outgoing links are better controlled by spammers than incoming links. The remaining metrics have AUCs close to 0.5 and sometimes worse. Therefore, we use only the top five metadata-based feature sets, $M_1 \sim M_4$ and M_7 for the remaining experiments.

A noteworthy observation is that although the AUC of the classifier with M_7 is smaller than that with M_4 by 0.07, the ROC curve for the classifier with features from M_7 shows higher true positives at lower false positive rates, as shown in Figure 7. This emphasizes the importance of ROC curves in judging performance and that combining feature sets may help to maximize gains.



(a) Betweenness, clustering coefficient

(b) Degree, in-degree, out-degree

Figure 5. ROC curves for classifiers trained on graph metric features for URL-based versus hybrid link graphs at depth three. (a) Betweenness, clustering coefficient and (b) Degree, in-degree, out-degree.

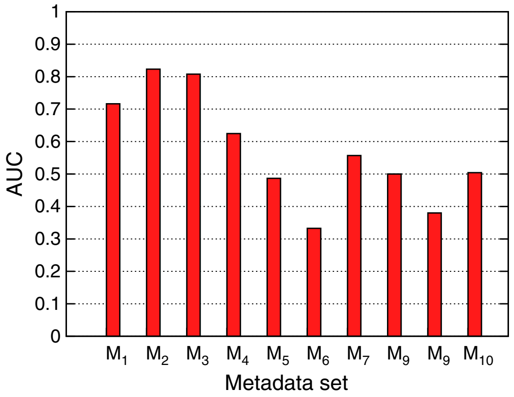


Figure 6. Performance of individual metadata metrics for hybrid link graphs at depth three.

5.3. Performance of metric combinations

While it is safe to ignore metrics whose SVMs fared no better than a random guesser, picking only the best performing

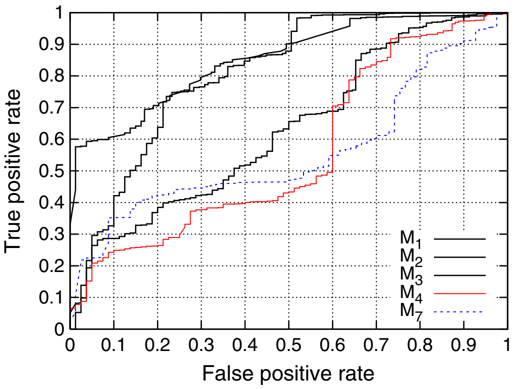


Figure 7. ROC curves for SVM classifiers for best performing metadata metrics (for hybrid link graphs at depth three).

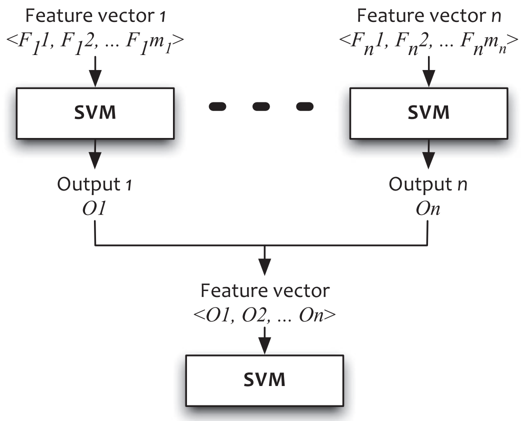


Figure 8. Overview of combination of feature sets.

metrics is not a good approach. This is because different metrics may complement each other to improve the performance of classifiers trained on their combinations. We now examine metric combinations, first within graph metrics, then with metadata metrics, and subsequently when both types of metrics are combined.

5.3.1. Technique to combine metrics.

Each of our graph metrics translate into feature vectors with cardinality 1000, as described in Section 4.1. Collectively, the five graph metrics alone give 5000 features, which exceed the number of graphs in our data set, which are 2330. Even though SVM classifiers are known to be highly immune to such problems, this could reduce the predictive power of the classifier. To avoid a situation with a large number of features, the classifiers based on different metrics were combined by building a second stage classifier that uses the outputs of the first-level classifiers as features. This situation is presented in Figure 8.

As an example of combination of graph metrics explained in Section 5.3.2, an SVM classifier is executed for each graph metric in the first layer. Thus, each SVM classifier in the first layer uses a feature vector with cardinality 1000 in our experiment. In the second layer, the decision values, O_n, produced from the SVM classifiers in the first layer are used as a feature vector. Therefore, even for the combination of all the five graph metrics, the cardinality of the feature vector is only five in the second layer.

Table IV. Combinations of graph metrics.

Combination ID	Description
$Combination_g 1$	Degree, betweenness
$Combination_g 2$	Degree, clustering coefficient
$Combination_g 3$	Degree, betweenness, clustering coefficient

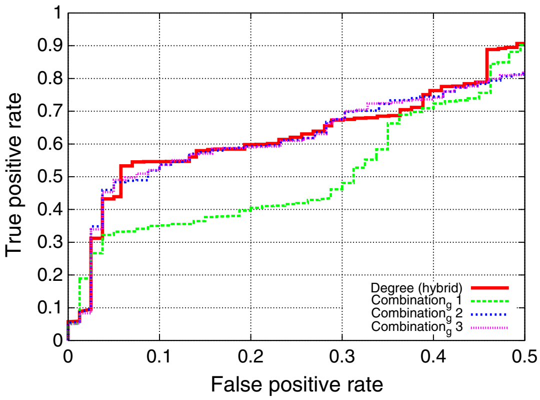


Figure 9. Performance of SVM classifiers based on different combinations of graph metrics (for hybrid link graphs at depth three).

5.3.2. Combinations of graph metrics.

Here we explore the performance of a classifier trained on graph metrics. Because classifiers based on the features of individual metrics performed at least as well as a random guesser according to Figure 4, we try all metric combinations. Here, we show only selected results in the interest of brevity. In particular, we exclude combinations that include two of the three degree-based metrics because these metrics expectedly perform similar to each other and their combinations yield no better results than any of them individually, as noted in Figure 4. Table IV shows the metric combinations depicted in Figure 9.

Figure 9 shows that *graph metric combinations 2 and 3 show the best performance*, but their performance is similar to the SVM classifier we had trained with just features based on three individual degree-based metrics. We show only the degree metric for comparison purposes. Also, the recall of the best graph metric combination at low false positive rates is about 60%, suggesting the need for additional complementary metrics.

Table V. Combinations of metadata metrics.

Combination ID	Description
$Combination_m1$	M_2, M_3
$Combination_m2$	M_1, M_2, M_3
$Combination_m3$	M_2, M_3, M_4
$Combination_m4$	M_1, M_2, M_3, M_4
$Combination_m5$	M_1, M_2, M_3, M_7
$Combination_m6$	M_1, M_2, M_3, M_4, M_7

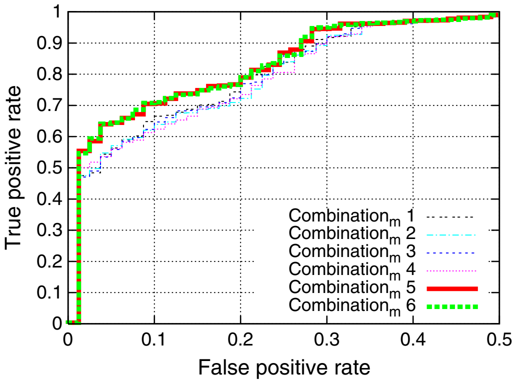


Figure 10. Performance of SVM classifiers based on different combinations of metadata metrics (for hybrid link graphs at depth three).

5.3.3. Combinations of metadata metrics.

We now examine the performance of combinations of metadata metrics. We focus only on the five metadata metrics whose classifiers performed better than a random guesser, namely, $M_1 \sim M_4$ and M_7. In the interest of brevity, we only show the results for the top six best performing combinations. The combinations are shown in Table V.

The performance results for the best metadata metric combinations are shown in Figure 10. The results of metadata metric combination 1, 2, 3, and 4 are similar. The top three metadata metrics, M_1, M_2, and M_3, do not actually provide notably better performance results in the combinations. The performance results are similar to the best one among those of individual metadata metrics in Figure 7. *The best performing classifiers are a result of metadata metric combinations 5 and 6*. The latter is simply a combination of each of the five metadata metrics considered. Also, given that the performance of metric combination 5 comes close, we conclude that metric, M_4, which is a feature vector containing the ratio of out-links to a node's domain versus all out-links, is dispensable. We also emphasize the positive effect of M_7 improving the performance result in the metric combination 5. As we discussed in Section 5.2, the ROC curve for the classifier with features from M_7 provides higher true positives at lower false positive rates in Figure 7. Nevertheless, the best metric combination itself still offers a recall of just under 80% at low false positive rate, which is less than satisfactory.

Table VI. Combinations of graph and metadata metrics.

Combination ID	Description
$Combination1$	Degree, $Combination_m5$
$Combination2$	$Combination_g2$, $Combination_m5$
$Combination3$	$Combination_g3$, $Combination_m5$

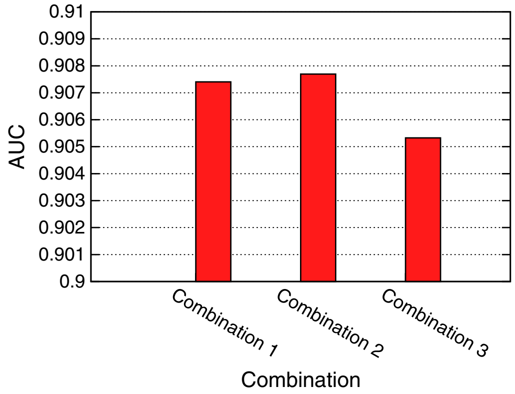


Figure 11. AUC of SVM results with combinations of graph metrics and metadata (for hybrid link graphs at depth three).

5.4. Overall classifier performance

Finally, we combine the combination of features explored in Sections 5.3.2 and 5.3.3 to see the performance of the combined classifier. We train SVMs for all possible combinations but show the results only for the three best performing combinations shown in Table VI. In all cases, hybrid graphs of crawl depth three were used.

Figure 11 shows the performance of various combinations. We note that all combinations perform well, as the AUC for each combination is above 0.9. Also, because all AUCs are below 0.91, it shows that the differences across combinations are rather small. This is to be expected because combinations 2 and 3 include the metrics, clustering coefficient and betweenness, which did not improve the performance of the classifier trained on the combinations of graph metrics. However, because combination 2 has the highest AUC, we use that classifier for final results.

Finally, with knowledge about the best feature combinations to use, we vary decision thresholds and observe the precision and recall of our classifier. *We find that our classifier has a precision of 98.87% with a recall of 92.78%.*

6. RELATED WORK

Many works investigate identifying malicious links contained in emails or on web pages [17–29]. Ma *et al.* [21,22] propose an online learning system to detect malicious URLs by just using lexical and host-based features of the URLs. In contrast, some other works follow or crawl URLs for links with the goal of detecting web spam

[17–20,23–26,28,29]. These approaches try to mitigate web spam by detecting various Search Engine Optimization (SEO)-based techniques including cloaking, keyword stuffing, and link farms. Niu *et al.* [23] identify cloaking by setting `User-Agent` and `Referer` HTTP headers with appropriate values to make the HTTP request come from a web browser and through web search result links. Wu and Davison [26] detect cloaking by probing a web page from multiple unaffiliated IP addresses. Whittaker *et al.* [25] crawl URLs and extract content-based features for each. They then use content-based features along with lexical and host-based features of URLs. Drost and Scheffer [19] and Ntoulas *et al.* [24] used content-based analysis for detecting spam URLs, by studying relevant properties such as page size or distribution of keywords in the pages crawled from the URLs. Castillo *et al.* [18] and Gan and Suel [20] build host-level link graphs to detect link farms and combine the link graph information with content-based features. Then they decide if a host is a spamming machine or not. The works in [17,28] propose methods for graph regularization to classify remaining nodes of a partially spam classified collection of sites represented as directed graphs. Saito *et al.* [29] extract link farms from web crawls by applying various graph algorithms.

Zhang and Gu [27] focus on an observation that spammers are likely to reuse a limited set of forums to place their spam links because it is not easy to harvest target forums. Thus, their proposal builds a graph whose node is a forum, which is linked with another node when there is a commonly posted spam URL between them. After building the graph, the proposal continuously monitors a group of forums that seem to be used by the same spammer. Once a new URL is posted on one of forums in the group, it sees if the URL is also found in other forums in the group. If so, it decides that the URL is a spam link. However, their approach is not tolerant enough to defeat spammer's evasion once spammers know its main idea. For example, the edges in their graph are characterized by a spam URL. Spammers can easily put polymorphic URLs into different forums instead of placing the same URL. Even though a domain can be used instead of a full URL, spammers can easily exploit a reputable domain to generate a non-spam looking link that is eventually connected to a spam link in a multiple level [7].

Because of forum spam's aspects related to web spam, one might be tempted to directly apply the same mitigation techniques that are used for web spam. However, two issues make it difficult to directly apply the link structure analysis-based techniques to detect link farms. First, most web spam techniques identify each *host* as spam or non-spam, while for forum spam, one needs to classify individual URLs. This is because forum spam URLs often belong to legitimate hosts, whereas this is not the case in link farming. As an example, a spammer can create an account on `amazon.com` and place a malicious URL on his or her profile page, with appropriate semantic data to attract traffic to the malicious site. Next, the spammer can insert the URL of the `amazon.com` profile page on forums. Using link farm identification techniques could classify `amazon.com` as a spam host, which would clearly be wrong. In our approach, we build a link graph for an individual URL. Thus, we can identify each URL as spam or non-spam. Second, link structures around forum spam URLs often differ from those of web spam, because there is little motivation for forum spam URLs to form link farms. In fact, Wang *et al.* [6] observed that redirection spam links redirected users to web pages containing pay-per-click programs, or pages that acted as advertising portals. They also found these redirection spam links to various spammer-controlled domains that did not form link farms. While mitigation techniques for web spam highly depend on identifying the characteristics of link farms, our approach rather focuses on classifying link graphs with similar link structures.

Even though we analyze the link structure as well, we build and investigate URL-based link graphs rooted to URLs found in forum spam to detect spam URLs. Thus, our main intuition is to analyze the link structures of URLs that web users confront by following the URLs found in forum spam. Even when forum spamming is used as web spamming, the URLs found in forum spam could be discovered by web users in web search results. By examining the link structure of URLs followed by web users, we can obtain the structural characteristics of spammers' infrastructure, that is, webhosts containing the URLs. However, the link structure analysis by web spamming studies does not crawl and build a web graph from a URL that needs to be classified as spam or not. Instead, it investigates the link structure of individual URLs within a large web graph that is already crawled and built with some chosen seed URLs. Therefore, the relationship between individual URL and other URLs directly linked to or by the URL is more important. Then it aggregates the extracted information by host level and classifies each host as spam or not.

In our previous work [7], we identified light-weight features based on forum spammers' IP addresses, commenting activity, and the anatomy of their posts. Then we used these features to train a classifier that can identify forum spam. While this approach focused on mitigation from the perspective of forum operators, the method in this paper is more suited for organizations that have efficient access to web graphs, such as search engines. These two techniques are complementary and should ideally co-exist; the work in this paper would minimize the possibility that users reach spammed forum pages through search engines, and the previous work would ensure that any forums visited through search engines or otherwise have defenses in place to prevent their visitors from falling prey to forum spam.

7. CONCLUDING REMARKS

In this paper, we inspected the link structure in the neighborhood of forum spam URLs in order to identify and characterize features one could use for detecting forum spam. We found that collapsing URL nodes belonging to legitimate web hosts is helpful in identifying distinctive structural characteristics in the neighborhood of spam pages. We also discovered that following three links from a URL is enough to build link graphs with distinctive features in our experimental data. This is a significant savings over crawling to a depth of four, because the latter gives rise to graphs whose nodes and edges significantly outnumber those of graphs at crawl depths of three. However, it does not necessarily denote that a link graph with depth three is always best for every case. Instead, it shows that a link graph with a deeper depth does not always come with better information and strongly implies possible optimization for building a link graph for discovering some common characteristics.

We explored five standard graph metrics and 10 metadata metrics in our classifier. Degree was the most effective graph metric to distinguish spam link graphs from non-spam graphs. In-degree and out-degree were similar enough to each other to the degree that either could be used. The other two metrics are relatively compute intensive and offered benefits small enough that we would not advocate in their favor. We also found that metadata metrics on nodes and edges were useful in that they complemented the graph metrics. In particular, the most effective metadata metrics, $M_1 \sim M_3$, were also the simplest to compute, for they were derived out of simple statistics, the number of nodes, edges, hosts, and domains in link graphs, and the proportion of nodes that returned specific HTTP status codes. While many nodes returned 200 status codes implying that the retrieval was successful, the spam graphs had a disproportionately high number of nodes with status codes 404 (not found) and 301 (moved permanently). Finally, even though the best-performing classifier was based on the combination of both metadata and graph metrics, the performance of a classifier based only on metadata metrics came close.

While the performance of our classifier is good, there is room for improvement, particularly in the context of recall, which currently stands at 93%. Thus, to extend this work, a few experimental metrics such as graphlets [30] may be helpful. Another direction is to extend the analysis of link graphs by investigating weighted hybrid link graphs. Currently, in our hybrid link graph model, the degree of links between two collapsed nodes is not represented. By assigning a weight with respect to the degree of links, we can employ a measurement of the relationship between two nodes for structural analysis of link graphs. Similarly, we do not make use of in-bound data links to web pages (such as images, ads, etc.). This information may play an important role in disambiguating good and bad links. There is also a direction to extend this work for forum spam with different forms of media instead of text. For example, a forum spammer can insert an image including spamming URLs. A robust method to extract URLs from it such as Optical Character Recognition (OCR) needs to be combined with our approach.

Our approach requires crawling the Web but can be readily employed by search engines, which already crawl and cache a significant portion of the Web. Thus, their operators can build link graphs around URLs without any additional crawling. Furthermore, classifiers only have to be periodically trained, which is an offline operation. Once trained, the classification overhead of an SVM-based classifier is marginal [31]. Thus, web search engines can use our approach to identify spam URLs in forum spam instantly, in turn preventing the indexing of spam URLs to their users.

We mainly employ the characteristics of web infrastructure used by forum spammers in our approach. While forum spammers can effortlessly change their spamming content to avoid any content-based spam detection, they cannot easily alter their infrastructure behind their campaign, because it is hard to build an effective infrastructure to attract visits by web users. Because of this reason, many spammers are willing to pay for third-party infrastructure instead of building their own infrastructure [6]. We exploit this aspect in our approach, because there is a high probability that forum spam by different spammers may share common characteristics in terms of their this is almost certainly not what you meant to say, but i don't know how to fix it although we focus on detecting URLs in forum spam, our approach can be easily extended to detect URLs in any other domain that are connected to a web infrastructure with a similar link structure. However, because of crawling cost, we do not assert that our approach may replace the existing spam detection methods including content-based ones. Instead, we believe that our approach can effectively complement the existing methods.

While our approach shows promise, the current study is limited in that it uses the links in forum spam from only one blog. However, there is little reason to believe that the blog we studied would differ from a random forum on the Web. In fact, we studied how forum spammers performed their spamming campaigns by using forum spam automating tools and showed that the forum spam in our data source was distributed to a large number of forums on the Web in our previous works [5,7]. Thus, we believe that the characteristics we identified of links in forum spam are likely to be valid for spam found in other forums on the Web.

ACKNOWLEDGEMENTS

We would like to thank Steven Murdoch from the University of Cambridge for providing data on forum spam. This material is based upon work supported by the National Science Foundation (NSF) under grant no. CNS-0831988 and by a grant from the Center for Applied Cybersecurity Research (CACR) at Indiana University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF or CACR.

REFERENCES

1. Netcraft. Web server survey. Available from: http://news.netcraft.com/archives/category/web-server-survey/. [Accessed on 10 April 2013].

2. Brin S, Page L. The anatomy of a large-scale hypertextual web search engine, *International World Wide Web Conference (WWW)*, Brisbane, Australia, 1998; 107–117.
3. University CM. CAPTCHA: telling human and computers apart automatically. Available from: http://www.captcha.net. [Accessed on 15 Decemeber 2013].
4. XRumer. Available from: http://www.botmasternet.com. [Accessed on 1 June 2013].
5. Shin Y, Gupta M, Myers S. The nuts and bolts of a forum spam automator, *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, Boston, U.S.A., 2011; 17–24.
6. Wang YM, Ma M, Niu Y, Chen H. Spam double-funnel: connecting web spammers with advertisers, *International World Wide Web Conference (WWW)*, Banff, Canada, 2007; 291–300.
7. Shin Y, Gupta M, Myers S. Prevalence and mitigation of forum spamming, *IEEE International Conference on Computer Communications (INFOCOM)*, Shanghai, China, 2011; 2309–2317.
8. WordPress. Available from: http://wordpress.org. [Accessed on 15 December 2013].
9. Akismet. Available from: http://akismet.com. [Accessed on 15 December 2013].
10. Alexa Internet I. Alexa top sites, 2010. Available from: http://www.alexa.com/topsites. [Accessed on 27 February 2013].
11. Davison BD. Recognizing nepotistic links on the web, *AAAI-2000 Workshop on Artificial Intelligence for Web Search (W4)*, Austin, U.S.A, 2000; 23–28.
12. Fielding R, Gettys J, Mogul J, Frystyk H, Masinter L, Leach P, Berners-Lee T. Hypertext transfer protocol – HTTP/1.1. RFC2616, 1999.
13. SVM light support vector machine. Available from: http://svmlight.joachims.org. [Accessed on 15 December 2013].
14. Serfling RJ. *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons: 605 Third Avenue, New York, NY, U.S.A, 1980.
15. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection, *International Join Conference on Artificial Intelligence (IJCAI)*, Montreal, Canada, 1995; 1137–1143.
16. Han J, Kamber M. *Data Mining: Concepts and Techniques*, Second edn. Morgan Kaufmann: 225 Wyman Street, Waltham, MA, U.S.A, 2006.
17. Abernethy J, Chapelle O, Castillo C. Web spam identification through content and hyperlinks, *WWW International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Beijing, China, 2008; 41–44.
18. Castillo C, Donato D, Gionis A, Murdock V, Silvestri F. Know your neighbors: web spam detection using the web topology, *ACM Special Interest Group on Information Retrieval (SIGIR) Conference*, Amsterdam, Netherlands, 2007; 423–430.
19. Drost I, Scheffer T. Thwarting the nigritude ultramarine: learning to identify link spam, *European Conference on Machine Learning (ECML)*, Porto, Portugal, 2005; 96–107.
20. Gan Q, Suel T. Improving web spam classifiers using link structure, *WWW International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Banff, Canada, 2007; 17–20.
21. Ma J, Saul LK, Savage S, Voelker GM. Beyond blacklists: learning to detect malicious web sites from suspicious URLs, *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, Paris, France, 2009; 1245–1254.
22. Ma J, Saul LK, Savage S, Voelker GM. Identifying suspicious URLs: an application of large-scale online learning, *International Conference on Machine Learning*, Montreal, Canada, 2009; 681–688.
23. Niu Y, Wang YM, Chen H, Ma M, Hsu F. A quantitative study of forum spamming using context-based analysis, *Internet Society Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, U.S.A., 2007.
24. Ntoulas A, Najork M, Manasse M, Fetterly D. Detecting spam web pages through content analysis, *International World Wide Web Conference (WWW)*, Edinburgh, Scotland, 2006; 83–92.
25. Whittaker C, Ryner B, Nazif M. Large-scale automatic classification of phishing pages, *Internet Society Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, U.S.A, 2010.
26. Wu B, Davison BD. Detecting semantic cloaking on the web, *International World Wide Web Conference (WWW)*, Edinburgh, Scotland, 2006; 819–828.
27. Zhang J, Gu G. NeighborWather: a content-agnostic comment spam inference system, *Internet Society Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, U.S.A., 2013.
28. Zhou D, Burges CJ, Tao T. Transductive link spam detection, *WWW International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Banff, Canada, 2007; 21–28.
29. Saito H, Toyoda M, Kitsuregawa M, Aihara K. A large-scale study of link spam detection by graph algorithms, *WWW International Workshop on Adversarial Infor-*

mation Retrieval on the Web (AIRWeb), Banff, Canada, 2007; 45–48.

30. Przulj N. Biological network comparison using graphlet degree distribution. *Bioinformatics* 2007; **23**: 177–183.
31. McGrath DK, Kalafut A, Gupta M. Phishing infrastructure fluxes all the way. *IEEE Security and Privacy Magazine's Special Issue on DNS Security* September/October 2009; **7** (5): 21–28.

DOI: 10.1002/sec