Spam Behavior Analysis and Detection in User Generated Content on Social Networks

Enhua Tan[1], Lei Guo[1], Songqing Chen[2], Xiaodong Zhang[1], and Yihong (Eric) Zhao[3]
[1]Ohio State University
{etan,lguo,zhang}@cse.ohio-state.edu
[2]George Mason University
sqchen@cs.gmu.edu
[3]Yahoo! Inc.
yzhao@yahoo-inc.com

***Abstract*—Spam content is surging with an explosive increase of user generated content (UGC) on the Internet. Spammers often insert popular keywords or simply copy and paste recent articles from the Web with spam links inserted, attempting to disable content-based detection. In order to effectively detect spam in user generated content, we first conduct a comprehensive analysis of spamming activities on a large commercial UGC site in 325 days covering over 6 million posts and nearly 400 thousand users. Our analysis shows that UGC spammers exhibit unique non-textual patterns, such as posting activities, advertised spam link metrics, and spam hosting behaviors. Based on these non-textual features, we show via several classification methods that a high detection rate could be achieved offline. These results further motivate us to develop a runtime scheme, BARS, to detect spam posts based on these spamming patterns. The experimental results demonstrate the effectiveness and robustness of BARS.**

I. Introduction

With the widespread usage of user generated content (UGC) in social media, spam in these sites is explosively increasing and has become an effective vehicle for malware and illegal advertisement distributions. In order to increase the click-through rate, spammers have utilized a number of methods to attract users. By posting spam articles repeatedly to a UGC site, spam content can be shown in the striking positions of the front page on the UGC site, such as the top article list and the most recent article list. By inserting popular terms to the title or the content, spammers can make their posts highly ranked when a user searches these keywords in a UGC system. Spam content not only pollutes the content contributed by normal users, resulting in bad user experiences, but also can mislead or even trap users. Furthermore, spam in UGC sites causes a lot of Internet resources and users' time being wasted. For example, it has been estimated that 75% posts shown in the top-50 search results for commercial queries at Blogspot.com are actually spam [29]. Another study [25] shows that more than 8% of Blogspot pages are spam (by random sampling), while two other smaller blog sites have more than 50% spam.

Different from email spamming, spamming in UGC sites is easier to conduct but harder to be detected. First, it is much easier to collect spamming targets for UGC spammers than collecting email addresses on the Web, since UGC sites are easier to be identified with search engines and the number of UGC sites is much smaller than that of email accounts. Second, it is also easier to post a spam article than to send a spam email. Although CAPTCHA [2] is often used for account registration in many UGC sites, posting articles in UGC sites often does not require any CAPTCHA verification. Third, a large number of small UGC sites such as blogs and forums may not have technical teams for anti-spamming. These sites are often the target of spamming attacks. Although most of these sites are not so popular and do not have large user populations, the total number of users and the corresponding audiences of these sites are huge. This is also one reason why UGC spam has increased rapidly in recent years.

Although a content-based spam detection can be effective to some extent, in practice, it has some limitations when being applied to UGC sites. First, a content-based classification needs new training data constantly due to the constant change of spam contents. This can be addressed for emails since email recipients often label unrecognized new spam. However, labeling UGC spam by readers is not so effective and accurate due to the open nature of UGC. The large volume of UGC spam makes the human aided labeling very costly. Second, as shown by recent measurements [28], a number of spam blogs are now created by professional spammers who often copy content from recent Web articles or Web sources with specific keywords that can help boost spam blog ranking. Thus, it is more difficult for a content-based spam classification method to distinguish spam posts from normal posts as they contain very similar content. Therefore, understanding the inherent patterns of UGC spamming behavior may shed light on spam detection in UGC sites.

In this work, first we analyze the trace of a UGC site of a large commercial search engine, which has over 6 million posts involving nearly 400 thousand users in 325 days. Our trace analysis shows that UGC spammers often exhibit uniquely different behavior patterns from those of normal users, including posting patterns, advertised spam links patterns, and link host related patterns. Furthermore, based on our study of spamming behavior patterns, we show that we can achieve low false positive and high true positive rates with several spammer classification methods offline.

Motivated by these results, we design a runtime spam detection scheme, BARS (Blacklist-assisted Runtime Spam Detection), by leveraging these behavior patterns and a spam URL blacklist. In BARS, a spam classification model is trained with an initial set of labeled spammers and spam URLs. A blacklist of spammers and spam URLs is also initialized with the training set. By feeding only high confident spam URLs from classification to the blacklist when enough posting history information is collected, BARS ensures the high quality

of the auto-expanding blacklist. The high quality blacklist is essential to a low false positive rate in runtime detection, while its auto-expanding feature helps improve the true positive rate. Meanwhile, any mis-classified users and URLs can be reversed with the help of a high-priority whitelist, which further improves the detection performance. The evaluation results show the promising runtime performance of our scheme.

The paper is organized as follows. In Section II, we analyze the UGC dataset. We propose the spammer classification features we use and evaluate the dataset with multiple classifiers in Section III. In Section IV, BARS scheme is designed and evaluated. Section VI discusses related work and Section VII concludes this paper.

II. Measurement-based Analysis

A. Dataset Overview

TABLE I
DATASET SUMMARY

Type	Number	Ratio
posts	6,595,917	100%
posts w/ links	1,832,112	27.8%
XYZ spam posts	1,130,718	17.1%
user IDs	382,090	100%
user IDs w/ links	157,305	41.2%
XYZ spammer IDs	12,116	3.2%

TABLE II
TOP-20 OUTGOING LINK DOMAINS

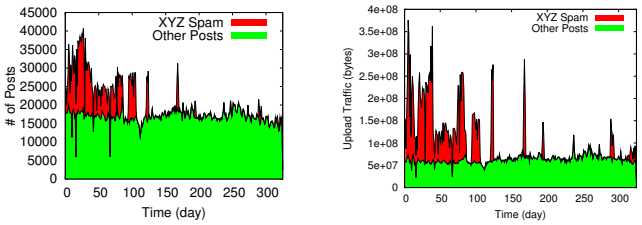
Domains	#Links	IP address	Description
xyz566.net	20,837,684	111.92.237.40	pirated software
xyz66.com	5,554,866	74.86.178.68	pirated software
tv1ccc.com	634,074	218.32.213.235	adult chatroom
xyz889.com	618,503	74.86.178.68	pirated software
258ww.com	228,985	220.229.238.55	adult chatroom
h2.idv.tw	216,058	220.228.6.5	adult chatroom
h4.idv.tw	213,333	220.228.6.5	adult chatroom
h5.idv.tw	213,194	220.228.6.5	adult chatroom
h3.idv.tw	200,001	220.228.6.5	adult chatroom
ut678.com	160,607	domain expired	adult chatroom
a5463.com	141,594	220.228.6.140	adult chatroom
xyz2007.com	140,279	74.86.178.68	pirated software
s5463.com	127,541	218.32.213.235	adult chatroom
kk0401.com	123,720	220.228.6.140	adult chatroom
kk1976.com	114,744	220.228.6.140	adult chatroom
t1.idv.tw	112,795	220.229.238.3	adult chatroom
ut789.com	108,787	domain expired	adult chatroom
youtube.com	105,181	74.125.115.113	video sharing
photobucket.com	98,156	209.17.65.42	image hosting
you.cc	93,700	208.109.181.70	domain services

We collected user posts for 325 days in a large commercial blog site till August 2009. Table I shows a summary of the dataset. The total number of posts is more than 6 millions, of which more than 27% posts include outgoing hyperlinks (or links). The number of user IDs is more than 382 thousands (XYZ is the largest spam campaign).

Table II lists the top-20 domains of the outgoing links in blog posts, ranked by the number of links. According to the rank of link domains, we find the largest spam campaign, `XYZ`, in which all posts have links to domains in the form of `xyz*.*`, such as `xyz566.net`, advertising pirated software.

```
Credit card reform bill passed by Congress
Blog Category: Uncategorized |
Blog: 2009-05-21 13:08

U.S. House of Representatives by 361 votes to 20, the result
of 64 votes in the final version of the card through the
reform bill, bill to prohibit all kinds of hidden charges
terms of consumer protection  [xyz]  interests.

Xinhua News Agency reported, a day after the Senate by 90
votes to 5, the voting resulted in the adoption of this
version of bill. As a final procedure, the Bills sent to
U.S. President Barack Obama has been signed.

Under the Act, only in the consumer credit business of at
least 60 days overdue payments can only raise interest
rates, but when consumers to maintain good faith consecutive
months  [Reductil]  with the records, credit card companies
must restore the previous low repayment rates.

In addition, credit card companies can not suddenly raise
interest rates credit card payments, must give 45 days to
inform the user of the decision to raise interest rates;
credit card bill must be sent 25 days before payment, to
avoid the  [lv bags]  are due to late receipt of the user...
```

Fig. 1. A spam blog example



(a) Daily # of posts (b) Daily upload traffic

Fig. 2. Blog post stats over time (stack graph)

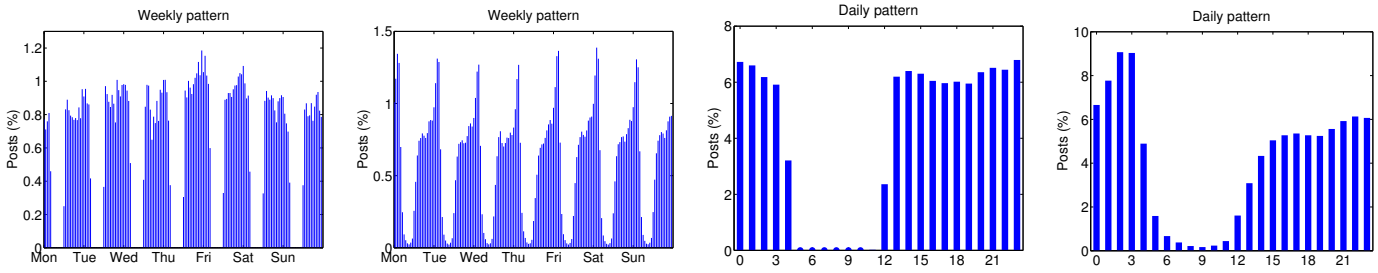
XYZ spam accounts for 17.1% of total posts in the blog site, but only 3.2% of total user IDs. The domains listed in Table II are mostly spam domains (note `youtube.com` is ranked 18th as the most popular non-spam domain). This indicates that most active links posted are dominated by spam sites. Spam in this dataset is mostly involved with pirated software, adult chatroom, etc.

Figure 1 shows an example of a crafty spam blog. This spam blog is trying to embed spam links in the post copied from a news Web site. The spam links have anchors with text keywords like `xyz`, `Reductil`, and `lv bags`, which aim to promote the spam sites selling pirated software, counterfeit medicines, and fake luxury products. When the major content of the spam blog is copied from other places, it is difficult to detect spam blogs with only traditional textual features.

Figure 2 shows the daily number of new posts and upload traffic volume of XYZ spam and other content in a stack graph. As we can observe in Figure 2(a), the daily number of XYZ spam posts is non-trivial in the system. Figure 2(b) shows that up to *83.7%* of the total posting upload traffic is from XYZ spam. The upload traffic of XYZ spam and other posts is calculated based on the content length of posts. These results show the significant resource consumption by spam content in UGC systems.

B. Weekly and Daily Spamming Patterns

We first study the weekly and daily patterns of XYZ spam and compare them with other posts. Figure 3 shows the

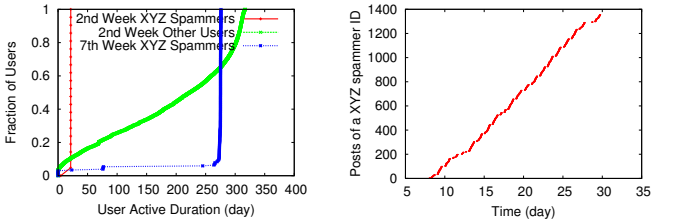


(a) XYZ spam (b) Other posts (c) XYZ spam (d) Other posts

Fig. 3. Weekly and daily patterns of XYZ spam and other blog posts

(normalized) number of posts in the time unit of week/day by binning all posts according to their posting hour in a week/day. All the timestamps are extracted from the blog post in the local time zone. Figure 3(a) and 3(b) show that XYZ spam is posted everyday regardless of whether it is weekend or not, which is not different from other posts. However, Figure 3(c) and 3(d) show XYZ spam is posted constantly every hour except for the morning (5am to 11am), while others have a daily peak. According to [20], spammers do not have peak hour posting patterns. Our analysis confirms this finding and further reveals that some spammers have an *off-hour* pattern. We conjecture that these are professional spammers who are paid for posting, and they have their own work pattern.

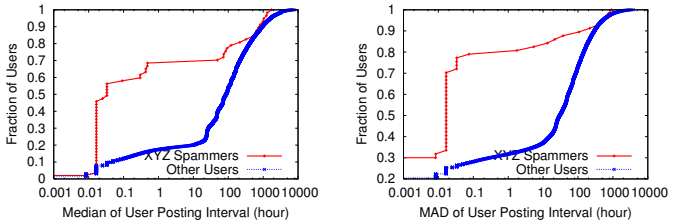
C. Spammer Posting Patterns



(a) User active duration (b) Posting time of a XYZ spammer ID

Fig. 4. User activities: for users joined in the same week

Because ID revoking is a straightforward method to thwart spam, spammers commonly have many different IDs. In order to examine the impact of spammer IDs, we study the active duration (the duration between the user's first and last posts) of user IDs in the dataset. We check the set of users who started posting in the same week since the active durations are smaller for newly joined users. Figure 4(a) shows the active duration of users joined in the second week of our trace collection. In this case, XYZ spammer IDs have shorter active durations than those of other users. Figure 4(b) shows the posting time of a XYZ spammer ID, which has a clear posting pattern with off-hours and ends after a few weeks. We have also studied spamming behavior for spammer IDs born in a different week. Figure 4(a) shows that for XYZ spammer IDs joined in the 7th week, the life span can be as large as our trace duration. After checking these IDs, we find spammers *reuse* their IDs after a long inactive duration if these IDs are not disabled in time.



(a) Median (b) MAD (median absolute deviation)

Fig. 5. User posting interval

Since spammers may advertise spam sites in an aggressive or machine-like manner [23], we thus investigate the posting intervals of XYZ spammers and other users. Figure 5(a) shows that a number of XYZ spammers post more frequently than other users, based on the median of user posting intervals. The vertical pattern around 1 minute in Figure 5(a) indicates that some spammers post with an almost constant frequency of one post every minute. These spammers exhibit bot-like behaviors, as most email spammers. On the other hand, these bot-like spammers still have off-hours as shown in Figure 4(b). Figure 5(b) further shows the MAD (median absolute deviation) distribution of posting intervals. This figure shows that spammers' posting intervals have smaller variances than those of normal users. In contrast to previous findings [23], there exist non-negligible spammer IDs posting with intervals *indistinguishable* from those of normal IDs.

D. Distribution of Posting Contributions

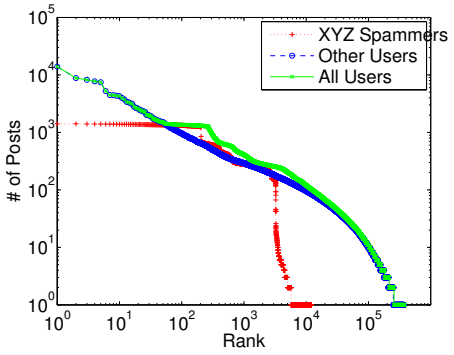


Fig. 6. Distribution of posting contributions

In UGC networks, we define the contribution of a user as the number of posts posted by the user. With spam, this

distribution could be vastly distorted as shown in Figure 6. `All Users` in Figure 6 shows the log-log scale rank distribution of all users' contributions, and it has an abnormal flat step around the first hundreds of users. `Other Users` in Figure 6 shows that after removing XYZ spammers, the rank distribution of users' post numbers is much smoother than before. `XYZ Spammers` in Figure 6 shows that the top-200 XYZ spammer IDs post a substantial number of posts, causing the abnormal flatting in the curve for all users.

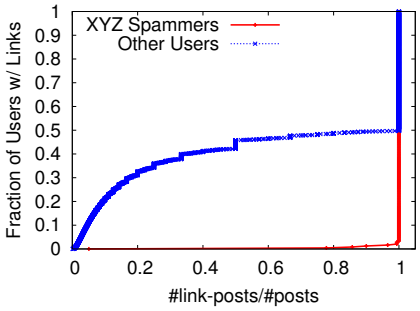


Fig. 7. Ratio of #link-posts per user

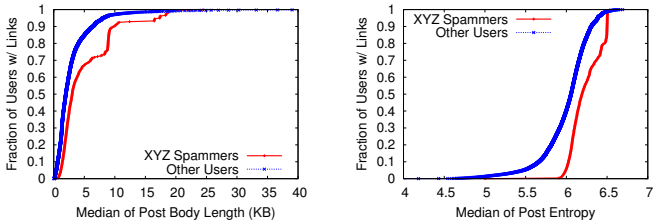
A spam article may have some outgoing links embedded in the content, in order to attract readers to click. For each user with links (a user who has at least one outgoing link in his/her posts), we calculate the number of posts having link(s) inserted (`#link-posts`) and the total number of posts, then plot the ratio of these two numbers in Figure 7. The figure shows that almost all XYZ spammer IDs *constantly* post articles with links inserted. UGC spammers are not willing to post text-only posts that cannot directly get any clicks to their customers' Web sites, since a user may not want to copy and paste a text-only URL in a Web browser to access the URL, and it is hard for the customer of spammers to evaluate the effectiveness of spamming without any link.

E. Link Patterns

As links in spam post are the entrance to the sites advertised by spammers, the presentation of links is usually optimized for their advertising purposes. Figure 8(a) shows that XYZ spam posts typically have either 1 or 2 links, which advertise a specific site, or more than 70 links, which advertise a number of items such as different video discs.

Looking into the links, we can see in Figure 8(b) that the median URL lengths of XYZ spam links are much *shorter*. According to our observation, a spam link usually points to a spam site without any path, or to a html file hosted in the root directory, with the intent of redirecting users to click as much content as possible. On the other hand, the link in normal posts is often composed by a query with multiple parameters, or has a long page depth, i.e., to a specific resource on the Web. Figure 8(c) shows that the anchor part of a XYZ spam link (the displayed text of a link) also has a shorter length. The analysis shows that normal users often post a link with the same anchor as the URL, while XYZ spammers tend to use shorter keywords to attract user's attention. Due to the intrinsic link advertising purposes of spammers, there exist substantial differences between spammers and non-spammers on these link metrics, which could be used for spam detection.

F. Content Characterizations



(a) Median of post content length per user (b) Median of post entropy per user

Fig. 9. Cumulative distribution of content metadata

Figure 9(a) shows the median of post length distribution of users who posted at least one link. XYZ spammers have a slightly larger post length, as some spam posts are copied from Web articles with links or spam content inserted. We also calculate the entropy of the post content by treating it as an ordinary file. Figure 9(b) shows that the difference between XYZ spammers and other users is small on this entropy metric.

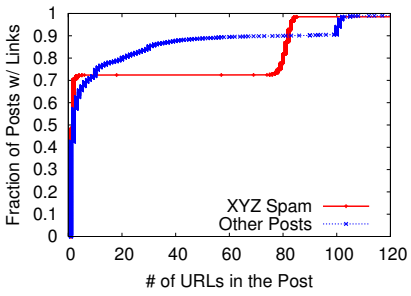
We further study the content of spam posts by comparing the word frequencies of the most active spam posts. The words are extracted from the sample of manually labeled spam posts (see Section III-B for details). As shown in Table III, the top-20 most frequently used words of spam type 1 are related to pirated software or movies, while those of spam type 2 are related to adult chatroom or pornography. There is almost no overlapping between them except `film` and `movie`. This indicates that if we only rely on content textual features to detect spam, we have to discover new textual features for every new type of spam, which requires repetitive labeling and training effort.

TABLE III
CONTENT COMPARISON OF DIFFERENT SPAM

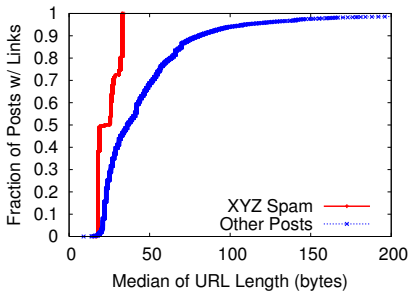
Type	Top-20 Most Frequently Used Words (in English)
spam type 1	official, english, software, DVD, chinese, traditional, disc, film, ULE, xyz, tools, subtitles, price, compilations, sound, DOD, movie, XYZ, CD, TND.
spam type 2	chatroom, video, adult, dating, beauty, erotic, av, chat, spicegirls, film, picture, movie, free, porn, japanese, game, photo, passion, girls, lover.

G. Hosting Behaviors

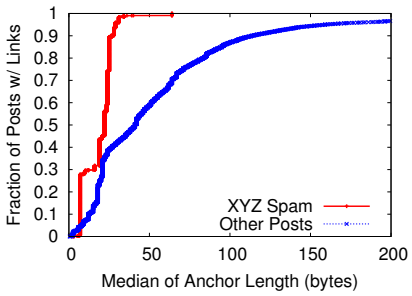
A spam host refers to the Web host in the spam link. In order to defeat blacklist-based spam detection, the host owners often register many host names or even different domain names [9]. Moreover, due to the extra cost to obtain individual IP addresses in Web hosting, lots of spam hosts share IP addresses (Table II also shows this phenomenon). Thus we expect that for spam hosts, the ratio of unique IP addresses to unique hosts should be small. Figure 10(a) shows that more than half of XYZ spammers' posts point to hosts with *less IPs*.



(a) # of Links

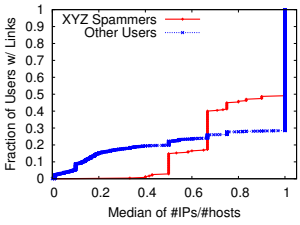


(b) Median of URL length

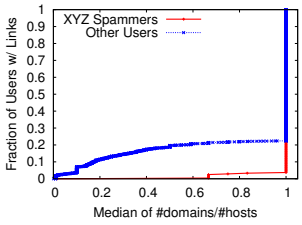


(c) Median of anchor length

Fig. 8. Link patterns



(a) Median of #IPs/#hosts



(b) Median of #domains/#hosts

Fig. 10. Hosting Behaviors

Figure 10(b) plots the ratio of the number of link domains to the number of link hosts. In this case, most XYZ domains have only one host name. This indicates that the cost-effectiveness consideration of hosting services could in fact serve as an important metric to detect spam advertised hosts.

H. Link Spam in UGC Sites

Because spam is prevalent on the Web among blogs, forums, or other UGC sites, we quantify the extent of this problem by querying a sample of manually confirmed spam blog links in Yahoo Site Explorer [4], which can return the inlinks of a URL, i.e., the list of Web pages having that URL. Figure 11 shows that the queried spam links have a median of 6 to 7 inlink domains (some links have no query results). This implies that a spam link may often be posted to several UGC sites to promote the same spam site. This also suggests that an anti-spam *collaboration* network among UGC sites can more effectively prevent new spam from being posted in multiple UGC sites.

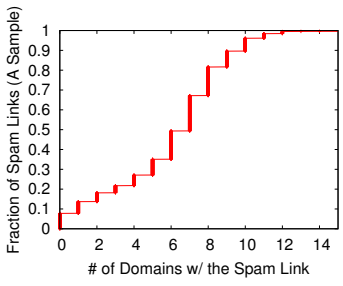


Fig. 11. Inlink #domains of sample spam links

III. Offline Spammer Detection

With the spamming characteristics identified in the previous section, we aim to evaluate their effectiveness with offline classification in this section.

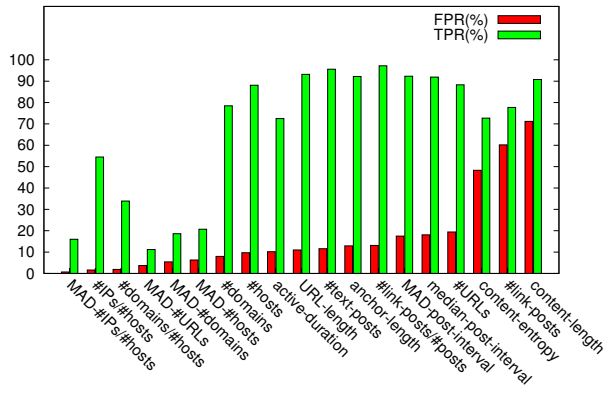
A. Features and Classifiers

To study the effectiveness of UGC spammer classification by using non-textual features of spamming behavior, we only use the features listed in Table IV in our evaluations. The features we choose are unique in that we collect all posts of a user and use the median or deviation to capture the user's patterns. Five sets of non-textual features are shown in Table IV, including user activities, post contributions, link patterns, hosting behaviors, and content metadata. All these features are selected based on the measurement analysis shown in Section II.

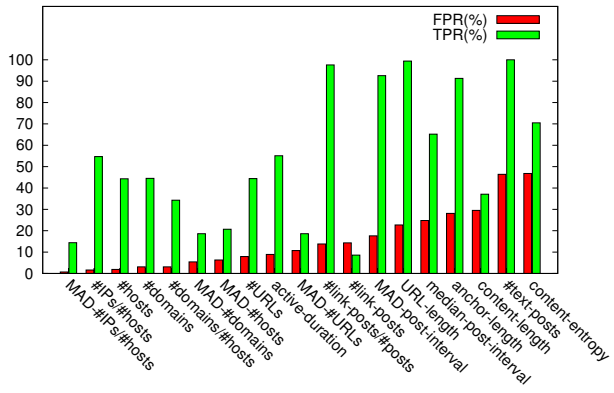
TABLE IV
COMPLETE LIST OF FEATURES USED IN EVALUATIONS. `link-posts` REPRESENTS THOSE POSTS WITH LINK(S) INSERTED.

Feature sets	Features
User activities	median (MAD) of posting interval active duration
Post contributions	#link-posts #text-posts #link-posts/#posts
Link patterns	median (MAD) of #URLs median of URL length median of anchor length
Hosting behaviors	median (MAD) of #hosts median (MAD) of #IPs/#hosts median (MAD) of #domains median (MAD) of #domains/#hosts
Content metadata	median of content length median of content entropy

Our classifiers are built based on Orange [3], a python-based data mining library. We have conducted experiments of blog spammer classification with several machine learning classifiers: Naive Bayes (NB), Logistic Regression (LR), and Decision Tree (DT). The Naive Bayes method is conducted as the baseline for performance comparison because of its simplicity. Both Naive Bayes and Logistic Regression return the probability of a user being a spammer. The Decision Tree learning method is also conducted in evaluation, with the standard C4.5 algorithm.



(a) Decision Tree classification



(b) Logistic Regression classification

Fig. 12. Classification results using each feature only

We use false positive rate (FPR) and true positive rate (TPR) to evaluate the classification performance. False positive rate is defined as the ratio of false positive items to the sum of true negative items and false positive items (ratio of non-spam classified as spam), and true positive rate is defined as the ratio of true positive items to the sum of true positive items and false negative ones (ratio of spam detected). Another metric false negative rate is defined as $1 - TPR$ (ratio of spam not detected).

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{TN + FP}$$

$$True\ Positive\ Rate\ (TPR) = \frac{TP}{TP + FN}$$

B. Spammer Classification

TABLE V
CLASSIFICATION RESULTS OF DIFFERENT METHODS

Methods	FPR	TPR
Naive Bayes (NB)	6.2%	99.4%
Logistic Regression (LR)	5.0%	99.5%
Decision Tree (DT)	1.6%	98.6%

We randomly sampled 1.37% (2,167) users from the total 157,305 users who posted at least one article containing link(s). Then we labeled each sample user as spammer or non-spammer based on the links from their posts. The labeling work was done by two persons without any knowledge of the features used in blog spammer detection. Among these 2,167 users, 1,087 are spammers, which accounts for 50.2% of the sample users. The labeled 2,167 users created a total number of 65,456 posts.

In the first set of experiments, we classify the labeled dataset with different classifiers based on all features shown in Table IV. All experiments are performed by using 10-fold cross-validation [18] to avoid biased selections of training and testing sets. Table V shows the classification results. As shown in the table, Naive Bayes, Logistic Regression, and Decision Tree have comparable performance. Decision Tree has the lowest false positive rate of 1.6% (with 98.6% true positive rate), while Logistic Regression has the highest true positive rate of 99.5% (with 5% false positive rate).

TABLE VI
CLASSIFICATION RESULTS ON ENTIRE DATASET

Methods	spammer		non-spammer	
	XYZ	non-XYZ	XYZ	non-XYZ
NB	12,087	70,108	29	75,081
LR	12,100	69,855	16	75,334
DT	12,072	67,228	44	77,961

Figure 12(a) shows the classification results when each feature is evaluated separately with Decision Tree. Figure 12(a) shows that a few features from different feature sets have very good true positive rate with low false positive rate, including the number of domains or hosts (hosting behavior), active duration (user activity), URL length (link pattern), and the number of text-only posts (post contribution). On the other hand, some features only provide limited performance, such as the content length/entropy, as their values are largely overlapping between spammers and non-spammers. Figure 12(b) shows the results of Logistic Regression (the results of Naive Bayes are similar and are omitted due to the page limit). Because these features are mostly presented as continuous values, Decision Tree is better at splitting the value space and thus gets better results utilizing each feature.

C. Spammer Detection on Entire Dataset

Because the entire dataset we collected is too large (over 1.8 million posts with links), it is infeasible to label every users in the dataset. We then use the labeled dataset as the training set, and run the three classifiers on the entire dataset. Because the classifiers do not include the feature of whether the user is a XYZ spammer or not, they could possibly classify XYZ spammers as non-spammers when being applied to the entire dataset. As shown in Table VI, all three classifiers detect almost all XYZ spammers: only up to 44 out of 12,116 XYZ spammers are incorrectly classified as non-spammers. There are about 50% of IDs classified as spammers among

the user IDs with link-post(s), and XYZ spammer IDs only account for 15% of them. We randomly sample 1,000 users out of the classified spammers (by Decision Tree classifier), and find only 7 of them are non-spammers, which indicates an estimated 0.7% false positive rate. We also randomly sample 1,000 users out of the classified non-spammers and find 9 of them are spammers, indicating an estimated 0.9% false negative rate. The performance results are consistent with the results on the labeled dataset, showing the effectiveness of spamming behavior features based classifications.

IV. BARS: Blacklist-assisted Runtime Spam Detection

The result in the previous section shows that spammers can be well detected offline based on non-textual features of all posts of a user. However, for a runtime system, a new spam post needs to be detected right away to minimize its adverse impact. This implies that a runtime spam detection scheme is demanded to classify a new post as soon as it is posted, based on the features of that post and past posts by the same user. In this case, it is challenging to classify the first post of a new user as no history information is available.

In this section, we propose a runtime spam detection scheme BARS (blacklist-assisted runtime spam detection) utilizing non-textual features, with the help of an auto-expanding spam blacklist, and a high priority non-spam whitelist. Our proposed scheme BARS is built on user behavior machine learning (ML) model. The non-textual behavior features are generated at runtime based on the new post and past posts of the same user. A spam URL blacklist is also maintained to help identify new spam posts. For a new post, if it has a URL in the blacklist, it will be classified as spam. With a highly accurate URL blacklist, spam posts containing these links can be promptly detected without the user's posting history. By providing high confident spam URLs from the ML model classified results to the blacklist, we can detect more spam than before. Meanwhile, the mis-classified users and URLs in the blacklist are reversed with the help of a high priority whitelist, which is essential for maintaining low false positives. Algorithm 1 shows how BARS works.

Algorithm 1 BARS: blacklist-assisted runtime spam detection

```
Learner ← ML model training with user behavior features
blacklist ← spam URLs in training set (or other sources)
blackusr ← spammer IDs in training set
whitelist ← non-spam URLs in training set (or other sources)
seedwhiteusr ← non-spammer IDs in training set
remove URLs in both blacklist and whitelist from the blacklist
for a new post do
    pid ← this new post; uid ← pid's user ID
    if pid has URL in blacklist then
        blSpam ← True
    else
        blSpam ← False
    end if
    mlSpam ← Learner(features generated from pid and uid's past posts)
    n ← 1 + # of uid's past posts
    if blSpam AND not mlSpam AND n > T_history then
        detectSpam ← False
    else if blSpam OR mlSpam then
        detectSpam ← True
    else
        detectSpam ← False
    end if
    if detectSpam AND (blSpam OR (mlSpam AND n > T_history)) then
        newURLs ← pid's new URLs, i.e. not in whitelist/blacklist
        if uid not in blackusr then
            blackusr.add(uid)
            newURLs.append(new URLs from uid's past posts)
        end if
        blacklist: include newURLs
    end if
    if not detectSpam AND (uid in seedwhiteusr OR (not mlSpam AND n > T_history)) then
        whitelist: include pid's new URLs
        remove URLs in both blacklist and whitelist from the blacklist
    end if
end for
```

A. ML & Blacklist Interaction

A blacklist is initialized by the training set, and can automatically expand by adding new URLs from a new spam post if the post has any URL in the blacklist. If the post has no URL in the blacklist but can be classified as spam based on spamming behaviors, new URLs from the post are spam URL candidates. In light of how the classification model works, it is intuitive that the classification is likely to be more trustworthy if the user have produced several posts in the system. Therefore, we set a threshold ($T_{history}$) and only provide new URLs of the classified spam post to the blacklist when the number of existing posts by the same user is larger than the threshold. In this way, the blacklist expands with high confidence.

Algorithm 1 details how the blacklist expands with more spam URLs. The classifier can possibly classify a spammer's first several posts as non-spam due to the lack of user history. After this boosting period, the classifier is finally able to detect this user as a spammer with enough history. Added as a new spammer to the blacklist, URLs in past posts of the spammer can now be added to the blacklist to improve the detection performance of future spam.

B. Anti-Detection Prevention

In our trace, spam posts rarely contain any non-spam URL. However, to escape from our spam detection scheme, future spammers may add non-spam URLs to spam posts, so that the blacklist expanding may incorrectly include non-spam URLs. To defeat this kind of anti-detection, a high priority steady-growing whitelist containing non-spam URLs is used in BARS. The whitelist is initialized with non-spam URLs from the training set, and is updated when a new post is confidently identified as non-spam (URLs of the user's past posts are not included to minimize false positives). The whitelist is set to have a higher priority than the blacklist. When a URL is firstly included in the blacklist, it can still be removed from the blacklist and inserted to the whitelist, if the URL repeatedly

triggers conflicts between the high confident classifier and the blacklist detection results. The whitelist is used solely to maintain the low false positive rate of the blacklist.

C. User Clustering

In addition, the accuracy of the ML model can also be improved by overcoming its limitation of not having enough history information for a new user ID. For this purpose, we can cluster user IDs based on shared URLs. As we know, spammers typically have multiple user IDs to promote the same spam site. Thus, for the first post by a new user ID, although it has no posting history, we can incorporate recent posts of other users sharing the same URLs to generate features for classification. Besides the improvement to the ML classifier, the blacklist can also get more and accurate input from the ML classification results with user clustering.

V. BARS Evaluation

We evaluate our runtime spam detection scheme on the labeled blog posts (refer to Section III-B) by splitting the dataset into training sets and a testing set according to the posting time. The training sets are selected from the latest posts of the first 160 days, with increasing durations. For all training sets, the testing set is selected from the 161st to 325th days, excluding posts by those user IDs existing in the first 160 days. As a result, the testing set is completely *independent* of the training set, which enables us to assess the efficiency of runtime spam detection.

A. Performance Evaluation

We compare three schemes in our evaluation experiments. *ML* uses machine learning classifier only. For a new post, all the features are generated based on the post and past posts of the same user. We use the spamming features listed in Table IV only. The machine learning classifier uses C4.5 Decision Tree. And we also cluster users who share URLs (clustering users who share domains or hosts has a higher false positive rate). *BARS* uses the algorithm shown in Algorithm 1 based on ML. We set the $T_{history}$ threshold to 10. *OPT* provides optimal feedback from the classifier to the blacklist (or whitelist) by filtering non-spam URLs (or spam URLs), which is a cheating algorithm as it requires the labels of the testing set as input. In our evaluation, ML is used as a baseline algorithm, and OPT shows the best result we can get by combining machine learning classifier and the blacklist.

Figure 13(a) and 13(b) show the runtime spam detection results of posts in the testing set. Figure 13(c) and 13(d) show the results of users, which are similar to the results of posts. The false positive rate of ML generally decreases with the increase of the training set duration, while the true positive rate increases accordingly. Comparing ML to BARS, the true positive rate is increased from 91% to 95% while the false positive rate is decreased from 15% to 13%, for the 1 day training set case. That is, the runtime spam detection performance is improved by BARS mainly on the increase of the true positive rate. OPT only outperforms BARS within

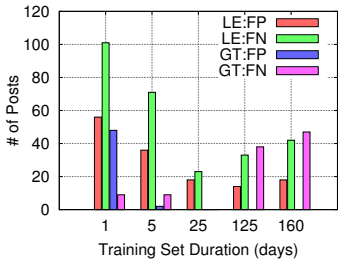
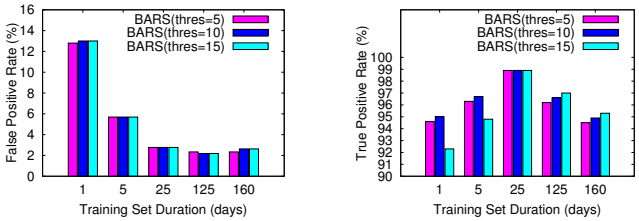


Fig. 14. Runtime spam detection FP/FN of ML model: posts with history length less or greater than $T_{history}$=10



(a) False positive rate (of posts) (b) True positive rate (of posts)

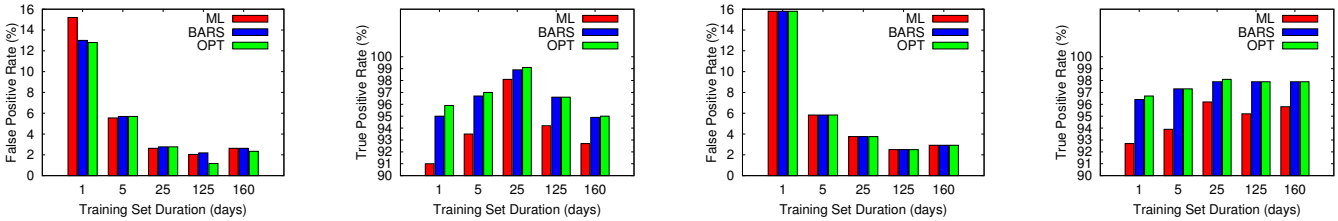
Fig. 15. BARS performance: tuning $T_{history}$

less than 1% in our evaluation, which further indicates the effectiveness of BARS in detecting spam.

Figure 14 shows the FP (false positives) and FN (false negatives) for posts with a history length greater than $T_{history}$ (GT), and those otherwise (LE) in ML. When the training set duration is larger than 1 day, the false positives for posts with a long history are significantly decreased. This indicates that we can be more confident with the classification results for posts with enough history. As a result, the feedback from the classifier to the blacklist is trustworthy, and that is the reason why the performance of BARS is close to OPT. Figure 15 shows the spam post detection results of BARS by tuning $T_{history}$. As we can see, the false positive rate is not sensitive to the change of the threshold, and the true positive rate varies within 1% when the training set duration is larger than 5 days. We have similar results of users and are omitted here.

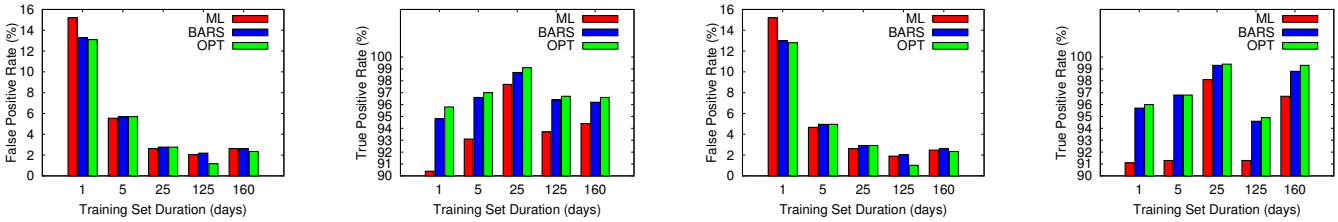
B. Anti-Detection Attacks Evaluation

In order to validate the effectiveness of whitelist, we also evaluate the performance of our schemes by artificially inserting non-spam URLs to spam posts. First, we introduce non-spam URLs (in our labeled dataset) to spam posts after their appearance, as spammers can achieve this by copying links from non-spam posts. The spam detection results of posts in Figure 16(a) and 16(b) show that BARS performs as well as before due to the steady growth of whitelist (the results of users are similar and thus omitted). We also evaluate the case of inserting popular non-spam URLs to spam posts before their first appearance in non-spam posts, considering that some popular links can be firstly copied from the Web by spammers. The results in Figure 16(c) and 16(d) indicate that BARS still performs better than ML with only minor increase of the false positive rate. The reason is that a popular non-spam URL is firstly included in the blacklist, causing some



(a) False positive rate (of posts) (b) True positive rate (of posts) (c) False positive rate (of users) (d) True positive rate (of users)

Fig. 13. Runtime spam detection performance



(a) FPR: inserting afterward (b) TPR: inserting afterward (c) FPR: inserting popular URLs beforehand (d) TPR: inserting popular URLs beforehand

Fig. 16. Runtime spam detection results: manually inserting non-spam URLs to spam posts

non-spam posts to be detected as spam. With the growth of user history length, BARS detects repetitive conflicts between classifier and blacklist detection results, and then removes the non-spam URL from the blacklist.

In summary, BARS can effectively improve the spam detection performance than using the ML model alone, especially in the small training set case. The blacklist and whitelist well maintain the low false positive rate and increase the true positive rate, even under several anti-detection attacks.

C. Discussions

BARS is sensitive to the history length of a spammer ID. If a spammer resorts to use a new ID each time when posting a new spam, it would be difficult for our scheme. However, the cost of spam posting is also increased for such spammers as the new ID registration requires extra effort. On the other hand, we can cluster user IDs based on shared URLs, which has already shown its effectiveness in our evaluation.

URL shortening has also been increasingly used recently, particularly in micro-blogs like Twitter. URL shortening, however, does not compromise the URL-based features used in classifications, as well as the blacklist and whitelist. The reason is that the original URL can be retrieved by visiting the shortened URL, and then be used for spam detection. The increased cost of retrieving the original URL is trivial.

VI. Related Work

It is estimated that 78% of emails on the Internet are spam [1]. Plenty of research has been conducted on email spam. For example, researchers have characterized spam traffic [12] and network-level spammers' behavior [26]. Many schemes, such as Naive Bayes based classifications [5], DNS Blacklists [16], and domain-based email authentication methods [8], have been studied or deployed to fight against email spam. Ramachandran et al. [27] proposed behavior-based blacklist by grouping email spammers with similar sending target domains. Xie et al. [30] proposed to automatically extract spam URL patterns from distributed and bursty botnet-based spam campaigns. Hao et al. [15] proposed to detect spammers with network-level features which are hard to change.

Web spam, especially link spam, that contains a large number of links to boost the page rank of linked sites in search engines, has drawn significant attention in recent years. Web spam hosts are discovered based on a small seed set by using the link structure of the Web to propagate trust [14]. Becchetti et al. were able to detect 80.4% Web spam based only on link properties with 1.1% false positives [6]. Castillo et al. combined link-based and content-based features for a decision tree classifier and were able to detect up to 88.4% of spam hosts with 6.3% false positives [7]. Wang et al. studied Web spam traffic and found that Blogspot.com was responsible for 25% of Web spam, which served as the doorway domains for Web spammers [29].

The volume of spam increases quickly in UGC sites. For example, Niu et al. [25] found that more than half of blog posts on two blog sites were spam, and forum spam often showed up in major search engines. Kolari et al. [20] have characterized spam blog properties such as non-power-law degree distributions and no-peak daily patterns. Goetz et al. [11] proposed a generative model to produce temporal and topological properties of blog networks, such as the inter-posting time. Sato et al. [28] found most of their studied spam blogs were created by a very small number of professional spammers. These spammers copied spam blogs from recent Web content or Web sources with specific keywords in order to avoid spam detection and promote spam links. Grier et al. [13] studied spam in Twitter and found the click-through rate of Twitter spam is much higher than email spam.

Detecting spam in UGC sites has different challenges from detecting traditional Web spam. Since spam content locates

in a single site, it is hard to use the link structure of the Web to help detect such spam. Kolari et al. [19] used SVM to evaluate spam blog detection based on local (content) features such as bag-of-words, bag-of-anchors, and bag-of-urls. In [21], global features, such as incoming or outgoing links to a node, were shown to be less effective than local features on spam blog detection. Lin et al. [23] proposed a spam blog detection method based on temporal, content, and link self-similarity properties. Their results show up to 95% accuracy by combining all the features including traditional content features. Ma et al. demonstrated the effectiveness of host-based features to classify malicious URLs, including the TLD, URL's path tokens (e.g., `ebayisapi`, `banking`), WHOIS dates, and DNS record [24]. Katayama et al. [17] evaluated the impact of sampling confidence to SVM learning for spam blog detection. A recent work [10] on Facebook proposed an unsupervised algorithm to detect the wall posts of malicious attackers by clustering posts based on shared links. Lee et al. [22] deployed honeypots on MySpace and Twitter and evaluated the effectiveness of social spam signatures generation.

Different from existing schemes, we characterize the spamming behavior patterns at the user level in this work. Because non-textual spamming features do not change as fast as the spam content, it provides a unique opportunity for us to detect UGC spammers. We can utilize these features to improve the runtime spam detection accuracy and robustness.

VII. Conclusion

The massive volume of user generated content in social media has witnessed the surge of spam in UGC sites, such as spam blogs. Due to the volatility of spam content, we seek to explore the spamming behavior patterns for such spam detection. In this work, we have conducted a thorough analysis of a large blog trace to study the user activities in about one year. Our analysis provides several new findings on the spamming behavior in blog-like UGC sites. Based on these non-textual features, we have applied several classifiers to classify UGC spammers. The experimental results not only show effectiveness of our proposed scheme, but also confirm the features we have identified through analysis. We further design and evaluate a runtime spam detection scheme, BARS, which shows promising detection performance.

VIII. Acknowledgement

We thank the constructive comments from the anonymous reviewers. This work is partially supported by the US National Science Foundation under grants CNS-0746649, CNS-0834393, CCF-0913150, and CNS-1117300.

References

[1] http://en.wikipedia.org/wiki/E-mail_spam.
[2] CAPTCHA. http://www.captcha.net/.
[3] Orange. http://www.ailab.si/orange/.
[4] Yahoo site explorer. http://siteexplorer.search.yahoo.com.
[5] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, and C. D. Spyropoulos. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *Proc. of the 23rd ACM SIGIR Annual Conference*, 2000.
[6] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates. Link-based characterization and detection of web spam. In *Proc. of AIRWeb*, 2006.
[7] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: web spam detection using the web topology. In *Proc. of SIGIR*, pages 423–430, 2007.
[8] M. Delany. Domain-based email authentication using public keys advertised in the DNS (DomainKeys). In *RFC 4870*, 2007.
[9] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proc. of WebDB*, 2004.
[10] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, , and B. Y. Zhao. Detecting and characterizing social spam campaigns. In *Proc. of IMC*, 2010.
[11] M. Goetz, J. Leskovec, M. Mcglohon, and C. Faloutsos. Modeling blog dynamics. In *Proc. of AAAI Conference on Weblogs and Social Media (ICWSM)*, 2009.
[12] L. H. Gomes, C. Cazita, J. M. Almeida, V. Almeida, and W. Meira Jr. Characterizing a spam traffic. In *Proc. of Internet Measurement Conference (IMC)*, 2004.
[13] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: The underground on 140 characters or less. In *Proc. of ACM Conference on Computer and Communications Security (CCS)*, 2010.
[14] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proc. of the thirtieth international conference on Very large data bases (VLDB)*, pages 576–587, 2004.
[15] S. Hao, N. Syed, N. Feamster, A. Gray, and S. Krasser. Detecting spammers with SNARE: Spatio-temporal network-level automatic reputation engine. In *Proc. of USENIX Security Symposium*, 2009.
[16] J. Jung and E. Sit. An empirical study of spam traffic and the use of DNS black lists. In *Proc. of IMC*, 2004.
[17] T. Katayama, T. Utsuro, Y. Sato, T. Yoshinaka, Y. Kawada, and T. Fukuhara. An empirical study on selective sampling in active learning for splog detection. In *Proc. of AIRWeb*, pages 29–36, 2009.
[18] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1137–1143. Morgan Kaufmann, 1995.
[19] P. Kolari, T. Finin, and A. Joshi. SVMs for the blogosphere: Blog identification and splog detection. In *Proc. of AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*, March 2006.
[20] P. Kolari, A. Java, and T. Finin. Characterizing the splogosphere. In *Proc. of the 3rd Annual Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 15th World Wid Web Conference*, 2006.
[21] P. Kolari, A. Java, T. Finin, T. Oates, and A. Joshi. Detecting spam blogs: A machine learning approach. In *Proc. of the 21st National Conference on Artificial Intelligence (AAAI)*, 2006.
[22] K. Lee, J. Caverlee, , and S. Webb. Uncovering social spammers: Social honeypots + machine learning. In *Proc. of SIGIR*, July 2010.
[23] Y.-R. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. L. Tseng. Splog detection using self-similarity analysis on blog temporal dynamics. In *Proc. of AIRWeb*, 2007.
[24] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Beyond blacklists: Learning to detect malicious web sites from suspicious URLs. In *Proc. of the ACM SIGKDD Conference (KDD)*, Paris, France, 2009.
[25] Y. Niu, Y.-M. Wang, H. Chen, M. Ma, and F. Hsu. A quantitative study of forum spamming using context-based analysis. In *Proc. of the 14th Annual Network and Distributed System Security Symposium (NDSS)*, 2007.
[26] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. *SIGCOMM Comput. Commun. Rev.*, 36(4):291–302, 2006.
[27] A. Ramachandran, N. Feamster, and S. Vempala. Filtering spam with behavioral blacklisting. In *Proc. of CCS*, 2007.
[28] Y. Sato, T. Utsuro, Y. Murakami, T. Fukuhara, H. Nakagawa, Y. Kawada, and N. Kando. Analysing features of Japanese splogs and characteristics of keywords. In *Proc. of AIRWeb*, pages 33–40, 2008.
[29] Y.-M. Wang, M. Ma, Y. Niu, , and H. Chen. Spam double-funnel: Connecting web spammers with advertisers. In *Proc. of WWW*, May 2007.
[30] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov. Spamming botnet: Signatures and characteristics. In *Proc. of SIGCOMM*, 2008.