Detecting Spam URLs in Social Media via Behavioral Analysis

Cheng Cao and James Caverlee

Department of Computer Science and Engineering, Texas A&M University
College Station, Texas, USA
{chengcao,caverlee}@cse.tamu.edu

Abstract. This paper addresses the challenge of detecting spam URLs in social media, which is an important task for shielding users from links associated with phishing, malware, and other low-quality, suspicious content. Rather than rely on traditional blacklist-based filters or content analysis of the landing page for Web URLs, we examine the behavioral factors of both who is posting the URL and who is clicking on the URL. The core intuition is that these behavioral signals may be more difficult to manipulate than traditional signals. Concretely, we propose and evaluate fifteen click and posting-based features. Through extensive experimental evaluation, we find that this purely behavioral approach can achieve high precision (0.86), recall (0.86), and area-under-the-curve (0.92), suggesting the potential for robust behavior-based spam detection.

1 Introduction

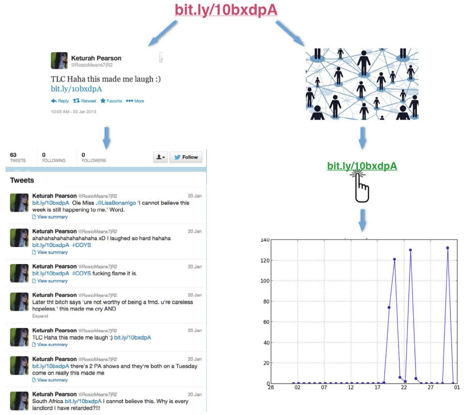
URL sharing is a core attraction of existing social media systems like Twitter and Facebook. Recent studies find that around 25% of all status messages in these systems contain URLs [7,17], amounting to millions of URLs shared per day. With this opportunity comes challenges, however, from malicious users who seek to promote phishing, malware, and other low-quality content. Indeed, several recent efforts have identified the problem of spam URLs in social media [1,5,9,16], ultimately degrading the quality of information available in these systems.

Our goal in this paper is to investigate the potential of *behavioral analysis* for uncovering which URLs are spam and which are not. By behavioral signals, we are interested both in the aggregate behavior of *who is posting* these URLs in social systems and *who is clicking* on these URLs once they have been posted. These behavioral signals offer the potential of rich contextual evidence about each URL that goes beyond traditional spam detection methods that rely on blacklists, the content of the URL, its in-links, or other link-related metadata. Unfortunately, it has historically been difficult to investigate behavioral patterns of posts and clicks. First, many social systems provide restricted (or even no) access to posts, like Facebook. Second, even for those systems that do provide research access to a sample of its posts (like Twitter), it has been difficult to assess how these links are actually received by the users of the system via clicks.

A. Hanbury et al. (Eds.): ECIR 2015, LNCS 9022, pp. 703–714, 2015.

As a result, much insight into behavioral patterns of URL sharing has been limited to proprietary and non-repeatable studies.

Hence, in this paper, we begin a behavioral examination of spam URL detection through two distinct perspectives (see Figure 1): (i) the first is via a study of how these links are posted through publicly-accessible Twitter data; (ii) the second is via a study of how these links are received by measuring their click patterns through the publicly-accessible Bitly click API. Concretely, we propose and evaluate fifteen click and posting-based behavioral features, including: for postings – how often the link is posted, the frequency dispersion of when the link is posted (e.g., is it posted only on a single day in a burst? or is it diffusely posted over a long period?), and the social network of the posters themselves; and for clicks – we model the click dynamics of each URL (e.g., does it rapidly rise in popularity?) and consider several click-related statistics about each URL, including the total number of clicks accumulated and the average clicks per day that a URL was actually clicked. Through extensive experimental study over a dataset of 7 million Bitly-shortened URLs posted to Twitter, we find that these behavioral signals provide overlapping but fundamentally different perspectives on URLs. Through this purely behavioral approach for spam URL detection, we can achieve high precision (0.86), recall (0.86), and area-under-the-curve (0.92). Compared to many existing methods that focus on either the content of social media posts or the destination page – which may be easily manipulated by spammers to evade detection – this behavior-based approach suggests the potential of leveraging these newly-available behavioral cues for robust, on-going spam detection.



Keturah Pearson @RosioMeans7jR2

TLC Haha this made me laugh :) bit.ly/10bxdpA

10:00 AM - 20 Jan 2013

Keturah Pearson @RosioMeans7jR2 20 Jan
bit.ly/10bxdpA Ole Miss .@LisaBonarrigo 'I cannot believe this week is still happening to me.' Word.

Keturah Pearson @RosioMeans7jR2 20 Jan
ahahahahahahahahahaha xD I laughed so hard hahaha bit.ly/10bxdpA #COYS

Keturah Pearson @RosioMeans7jR2 20 Jan
bit.ly/10bxdpA #COYS fucking flame it is.

Keturah Pearson @RosioMeans7jR2 20 Jan
Later tht bitch says 'ure not worthy of being a frnd. u're careless hopeless.' this made me cry AND

Keturah Pearson @RosioMeans7jR2 20 Jan
TLC Haha this made me laugh :) bit.ly/10bxdpA

Keturah Pearson @RosioMeans7jR2 20 Jan
bit.ly/10bxdpA there's 2 PA shows and they're both on a Tuesday come on really this made me

Keturah Pearson @RosioMeans7jR2 20 Jan
South Africa bit.ly/10bxdpA I cannot believe this. Why is every landlord I have retarded?!!!

Fig. 1. Studying spam URL detection in social media from two perspectives: (i) Posting behavior (left); (ii) Click behavior (right)

2 Related Work

URLs (and in particular, shortened URLs) have been widely shared on social media systems in recent years. Antoniades et al. [1] conducted the first comprehensive analysis of short URLs in which they investigated usage-related properties such as life span. With the rising concern of short URLs as a way to conceal untrustworthy web destinations, there have been a series of studies focused on security concerns of these URLs, including: a study of phishing attacks through short URLs [5], geographical analysis of spam short URLs via usage logs [9], an examination of security and privacy risks introduced in shortening services [16], and a long-term observation of shortening services on security threats [14].

Separately, Twitter spam detection has been widely studied in recent years. In general, three types of approaches have been proposed: user profile based, content based, and network relation based. User profile based methods [10,21,19] build classifiers using features extracted from account profiles, e.g., profile longevity. Content-based features [8,19] focus on the posting text. Network-based features [4,18,27] are those extracted from the social graph such as clustering coefficient. Many detection systems of suspicious Web URLs have been developed. Some of these [11,12,13,15] directly use URL lexical features, URL redirecting patterns, and URL metadata such as IP and DNS information. Some [3,6] consider features extracted from the HTML content of the landing page. Additionally, several dynamic spam URL filtering systems have also been developed [20,24,26].

Several recent works have used clicks extracted from the Bitly API, typically to study the properties of known spam links. For example, Grier et al. [8] recovered clicking statistics of blacklisted Bitly links, with the aim of measuring the success of those spam links on Twitter. Maggi et al. [14] submitted malicious long URLs to the Bitly API in order to examine the performance in terms of spam pre-filtering. Chhabra et al. [5] shortened a set of known phishing long URLs and analyzed factors like the referrer and location. There recently has been some research on using proprietary server-side click log data to defend against some types of spam (e.g., [23,25]). In contrast, our aim is to investigate how large-scale publicly-available click-based information may be used as behavioral signals in the context of spam URL detection on social media.

3 Behavior-Based Spam URL Detection

In this section, we investigate a series of behavioral-based features for determining whether a URL shared in social media is spam or not. Hence, for both the posting-based and click-based perspectives, we are interested to explore questions like: What meaningful patterns can we extract from these publicly-available resources? Are posting or click-based features more helpful for spam URL detection? And which specific features are most informative?

3.1 Problem Statement and Setup

Given a URL v that has been shared on a social media platform, the *behavior-based spam URL detection problem* is to predict whether v is a spam URL through a classifier $c : v \rightarrow \{\text{spam, benign}\}$, based only on behavioral features. In this paper, we consider two types of behavioral features associated with each URL – a set of posting-related behavioral features F_p and a set of click-based behavioral features F_c. Such a behavior-based approach requires both a collection of URLs that have been shared, as well as the clicks associated with each URL. Since many social media platforms (like Facebook) place fairly stringent limits on crawling, we targeted Bitly-shortened URLs.

URL Postings. Concretely, we first used the Twitter public streaming API to sample tweets during January 2013. We collected only tweets containing at least one Bitly URL (that is, a URL that had been shortened using the Bitly link shortening service). In total, we collected 13.7 million tweets containing 7.29 million unique Bitly-shortened URLs. We observed the typical “long tail” distribution: only a few URLs have been posted upwards of 100,000 times, whereas most have been posted once or twice.

URL Clicks. We accessed the Bitly API to gather fine-grained click data about each of the 7.29 million URLs. For example, we can extract the number of clicks per time unit (e.g., minute, hour, day, month) and by country of origin. In total, we find that nearly all – 7.27 million out of 7.29 million – of the URLs have valid click information, and that 3.6 million (49.5%) of the URLs were clicked at least once during our study focus (January 2013). As in the case of postings, we find a “long tail” distribution in clicks.

3.2 Posting-Based Features

In the first perspective, we aim to study the URLs through the posting behaviors associated with them. For example, some URLs are posted by a single account and at a single time. Others may be posted frequently by a single account, or by many accounts. Similarly, URLs may be temporally bursty in their posting times are spread more evenly across time. Our goal in this section is to highlight several features that may describe each URL based on its posting behavior.

Posting Count. The first feature of posting behavior is the total number of times a URL has been posted on Twitter during our study window. Our intuition is that this count can provide an implicit signal of the topic of the link destination as well as the intent of the sharer: e.g., URLs that are posted only a few times may indicate more personal, or localized interest. We formulate this feature as *posting count*, denoted as $PostCount(u)$ given a short URL u.

Posting Standard Deviation. A Weather Channel URL and a CNN breaking news URL may have a similar *posting count* on Twitter. However, the Weather Channel URL may be posted every day of the month (linking to a routine daily forecast), whereas a breaking news URL may be posted in a burst of activity in a single day. To capture this posting concentration, we consider the standard deviation of the days in which a URL is posted. Concretely, for each URL u we have a list of days when u was posted. We refer to this list as u's *posting days*, denoted by $PostDays(u)$. We define the *posting standard deviation* of a URL u as the standard deviation of all elements in $PostDays(u)$, denoted as $std(u)$. For example, if a URL u was posted 10 times on January 22nd and not tweeted on any other day, we have $std(u) = 0$. On the contrary, a URL u shared once per day will have a much larger $std(u)$.

Posting Intensity. The posting standard deviation gives insight into how concentrated a URL has been posted, but it does not capture the total intensity of the posting. For example, two URLs both of which have only one single posting day will have the same posting standard deviation, even if one was posted thousands of times while the other appeared only once. To capture this difference, we introduce *posting intensity* to capture how intense the posting behaviors of a URL are. Given a URL u, we calculate u's "intensity score" via the following:

$$intensity(u) = \frac{PostCount(u)}{(std(u) * |set(PostDays(u))|) + 1}$$

where $|set(PostDays(u))|$ is the number of distinct posting days of u. For those URLs whose scores are the highest, they have high posting frequency, but also a low intensity of posting days. To illustrate, we find in our dataset that the URL with the highest intensity score was posted nearly 30,000 times on a single day.

Posting User Network. The sharer's personal network and reputation have certain connection with what and why she posts. A typical example is the comparison between celebrities and spammers. Spammers whose networks commonly are sparse tend to post spam links to advertise, whereas a celebrity may not share such low-quality links. Thus, for each URL, we consider features capturing the poster's personal network. We use the counts of followers and friends as simple proxies for user popularity, and take the *median* among all posters.

3.3 Click-Based Features

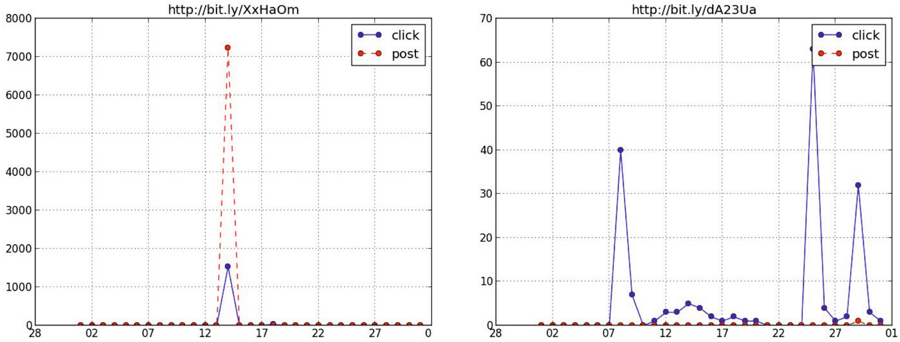
Now we turn our attention to how URLs are received in social media by considering the clicks that are associated with each URL in our dataset. We consider two kinds of clicking patterns: *clicking timeline* features that consider the temporal series of daily received clicks, and *clicking statistics* features that capture overall statistics of the clicks. For the first kind of clicking pattern, we have every short URL's fine-grained daily clicking data – which we can plot as its *clicking timeline.* We adopt three features extracted from this clicking timeline curve:

Rises + Falls. The first question we are interested is: how to capture the overall shape of a URL's clicks – do some go up continuously? Or do some periodically go up and down? To measure these changes, let n_i denote the number of clicks on the ith day. We define a *rise* if there exists an i such that $n_{i+1} - n_i > \alpha * n_i$ where α is a threshold and we set it to be 0.1, ensuring the change is nontrivial. Based on this criteria, we observe eight rises in Figure 2b (some are quite small). Similarly, let n_i denote the number of clicks on the ith day. We define a *fall* if there exists an i such that $n_i - n_{i-1} > \beta * n_{i-1}$ where β is a threshold value (set to 0.1 in our experiments). We observe eleven falls in Figure 2b.

Spikes + Troughs. In Figure 2b, we observe that while there are 8 rises, there are only 5 spikes of interest. So rather than capturing consecutive monotonic changes (as in the rises and falls), we additionally measure the degree of fluctuation of a URL through its *spikes* and *troughs.* That is, if there is an i such that $n_{i-1} < n_i > n_{i+1}$ we call it a *spike.* If there exists an i satisfying $n_{i-1} > n_i < n_{i+1}$, then it is a *trough.* Figure 2b has 5 spikes and 3 troughs.

Peak Difference. Naturally, there is a relationship between how and when a URL is posted and the clicks the URL receives. For example, Figure 2a illustrates a close relationship between posting and clicking for a URL. In contrast, Figure 2b demonstrates a much looser connection, indicating some external interest in the

URL beyond just its Twitter postings (in this case, the URL refers to a university website which attracts attention from many sources beyond Bitly-shortened links on Twitter). To capture the extent to which posting behaviors influence clicks, we define the *peak difference*. For each URL, we identify its *clicking peak* as the day it received the most clicks. Similarly, we identify its *posting peak* as the day it was posted the most. Note that a URL may have more than one posting peak and clicking peak. Here we define the *peak difference* as the minimum difference between two peaks among all pairs. The range of peak difference is from 0 to 30. In this way, peak difference can represent the level of tightness between clicking and posting.



(a) **(b)**

Fig. 2. The click and post timelines for two URLs. In (a), post and click behaviors are tightly coupled. In (b), the relationship is more relaxed.

We augment these timeline-based features with several click statistics:

Total Clicks. The first statistic is the *total clicks* a URL received in the period of study, which is a clear indicator of the popularity of a URL.

Average Clicks. Given a URL's total clicks and posting count, we can measure its *average clicks* per posting. By intuition more exposures bring more clicking traffic, but the average clicks is not necessarily large. Compared to total clicks, average clicks has a starker representation of popularity: many clicks via few postings suggest highly popular.

Clicking Days. We measure the number of *clicking days* in which a URL received clicks. This feature captures the consistency of attention on a URL.

Max Clicks. *Max clicks* is the maximum daily clicks. Unlike total clicks, this statistic can distinguish URLs that receive a burst of attention.

Effective Average Clicks. For those URLs with great total clicks, we observe some have a large number of clicking days while some have only one clicking day but thousands of clicks. Since average clicks considers only the relationship between total clicks and posting count, here we introduce *effective average clicks* defined as the following: Effective average clicks $= \frac{\text{total clicks}}{\text{clicking days}}$.

Click Standard Deviation. We already have features representing the fluctuation of timelines, now we consider a feature for the fluctuation of daily clicks given that we have specific sequence of daily clicks. We can calculate the standard deviation of daily clicks, defined as *click standard deviation*. Note that we fix a month as our time window of study. So, for each short URL we have a sequence of 31 daily clicks and we can compute the standard deviation.

Mean Median Ratio. Finally, given 31 daily clicks of a URL u, we can calculate its mean and median daily clicks, denoted as $mean(u)$ and $median(u)$ respectively. Now suppose we have a URL obtaining thousands of clicks on a day but very few on other days. It may have a considerable mean value but a low median. To build a connection between mean and median, we define *mean median ratio* of u as the following: Mean median ratio (u) $= \frac{mean(u)}{median(u)+1}$.

4 Experiments

In this section, we report a series of experiments designed to investigate the capacity of these two behavioral perspectives – posting-based and click-based – on the effectiveness of spam URL detection. Recall that our goal here is to examine the effectiveness of *behavioral signals alone* on spam detection. The core intuition is that these signals are more difficult to manipulate than signals such as the content of a social media post or the content of the underlying destination page. Of course, by integrating additional features such as those studied in previous works – e.g., lexical features of tweet texts, features of user profiles, and so forth – we could enhance the classification performance. Since these traditional features may be more easily degraded by spammers, it is important to examine the capability of a behavioral detector alone.

4.1 Experimental Setup

We consider two different sources of spam labels:

Spam Set 1: List Labeled. For the first set of spam labels, we use a community-maintained URL-category website *URLBlacklist* (`http://urlblacklist.com`) that provides a list of millions of domains and their corresponding high-level category (e.g., "News", "Sports"). Among these high-level categories are two that are clearly malicious: "Malware" and "Phishing", and so we assign all URLs in our dataset that belong to one of these two categories as *spam*. We assign all URLs that belong to the category "Whitelist" as *benign*. It is important to note that many URLs belong to potentially dangerous categories like "Adult", "Ads", "Porn", and "Hacking"; for this list-based method we make the conservative assumption that all of these URLs belong to the *unknown* class. For all remaining URLs, we assume they are *unknown*. This labeling approach results in 8,851 spam URLs, 223 benign, and 1,009,238 unknown. Of these URLs, we identify all with at least 100 total clicks, resulting in 1,049 spam, 21 benign, and 60,012 unknown. To balance the datasets, we randomly select 1,028 URLs from the unknowns (but avoid those above-mentioned dangerous categories), and consider them as *benign*, leaving us with 1,049 spam and 1,049 benign URLs.

Spam Set 2: Manually Labeled. We augment the first spam set with this second collection. We randomly pick and manually label 500 short URLs, each of which has been posted at least 30 times along with at least 5 original tweets (i.e., not a retweet, nor a reply tweet). We label a URL as "spam" if its landing page satisfies one of the following conditions: (1) The browser client (Google Chrome in our work) or Bitly warns visitors that the final page is potentially dangerous before redirecting; (2) The page is judged as a typical phishing site; (3) After several redirectings, the final page is judged to be a typical "spam page"; (4) Apparent Crowdturfing Web sites such as what were introduced in [22]. Finally, we end up with 124 manually-labeled malicious URLs: 79 spam ones, 30 irrelevant ads ones, and 15 pornographic ones. We also collect 214 benign URLs: 85 news ones, 70 blog ones, 49 video-audio ones, and 10 celebrity-related ones.

For each dataset, we construct the five posting-based features and the ten click-based features for all of the URLs. Then, we adopt the Random Forest classification algorithm (which has shown strong results in a number of spam detection tasks, e.g., [2,4,19]), using 10-fold cross-validation. The output of the classifier is a label for each URL, either *spam* or *benign*. We evaluate the quality of the classifier using several standard metrics, equally-weighted for both classes.

4.2 Experimental Results

Classification on the List-labeled Dataset. For the first dataset, we report the evaluation results in Table 1. We find that using all features – both posting-based and click-based – leads to a 0.74 precision, recall, and F-Measure, and a ROC area of 0.802. These results are quite compelling, in that with no access to the content of the tweet nor the underlying web destination, spam URLs may be identified with good success using only behavioral patterns.

Next, we ask whether posting-based features or click-based features provide more power in detecting spam URLs. We first exclude the five posting-based features and report the *Click-based only* result in the table. We see even in this case we find a nearly 0.65 precision, recall, and F-Measure. When we drop click-based features in favor of a *Posting-based only*, we see a similar result. These results show that individually the two feature sets have reasonable distinguishing power, but that in combination the two reveal complementary views of URLs leading to even better classification success. We additionally consider the very restricted case of *clicking statistics only* (recall that our click-based features include both clicking statistics and clicking timeline features). Using only the seven click statistics, we observe only a slight degradation in quality relative to all click-based features.

To provide more insights into the impact of each feature, we use the Chi-square filter to evaluate the importance of features to the classification result. The top 6 features are shown in Table 2. Median friends and average clicks are the most two important features. Generally speaking, click-based features tend to play more important roles than posting-based features. Recall that our list-labeled dataset are those URLs with abundant clicks received, but it is not guaranteed that they have adequate posting counts, which may explain the ranking. For

Table 1. Evaluation results for the list-based dataset

Set of features	Precision	Recall	F-Measure	ROC area
All 15 features	0.742	0.737	0.736	0.802
Click-based only	0.647	0.647	0.647	0.705
Posting-based only	0.648	0.695	0.694	0.756
Clicking statistics only	0.622	0.622	0.622	0.679

instance, if most URLs, either malicious or benign, have only one or two posting days and posting counts is less than 5, their posting counts and posting standard deviations will tend to be similar.

Table 2. Top-6 features for list-labeled dataset (Chi-square)

Rank	Features	Score	Category
1	Median friends	277.43	Posting
2	Average clicks	199.11	Clicking
3	Median followers	159.53	Posting
4	Effective average clicks	150.72	Clicking
5	Click standard deviation	141.62	Clicking
6	Mean median ratio	141.49	Clicking

Classification on the Manually-labeled Dataset. We repeat our experimental setup over the second dataset and report the results here in Table 3. When we use the complete 15 features, the precision, recall, and F-Measure are all even higher than in the list-labeled dataset case, around 0.86, with a ROC area of around 0.92. These results are encouraging. We attribute the increase in performance relative to the first dataset to the more expansive labeling procedure for the second dataset. In the list-labeled dataset, we only considered extremely “bad” URLs since we considered only the “Malware” and “Phishing” categories. This conservative assumption may lead to many spam-like URLs lurking in the set of benign URLs. In contrast, the manually-labeled dataset considers more broadly the context of what makes a spam URL.

Continuing our experiments, we again consider subsets of features in the classification experiment. Again, we find that using only a single feature type – either *Click patterns only* or *Posting patterns only* – leads to fairly strong classification performance. But that in combination, the two provide complementary views on URLs that can be used for more successful spam URL detection

Again, we use Chi-square filter to rank features, as shown in Table 4. Interestingly, the ranking is quite different from what we found in Table 2, though again we observe a mix of both posting and click-based features. We attribute some of this difference to the click data’s availableness in the manually-labeled dataset; most of the URLs in the manually-labeled dataset have abundant posting information and we can see that the posting behavior features play important roles in classification. On the contrary, most of the URLs in the manually-labeled

Table 3. Evaluation results for the manually-labeled dataset

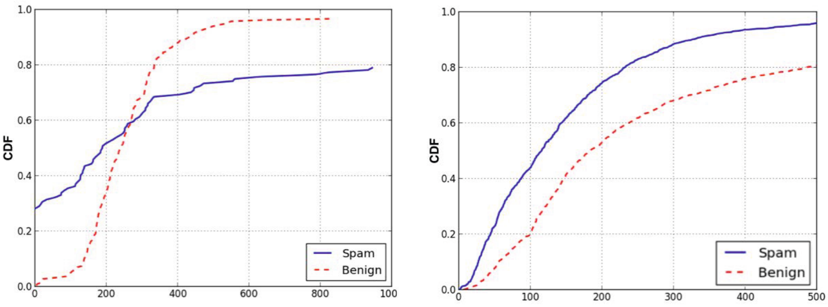
Set of features	Precision	Recall	F-Measure	ROC area
All 15 features	0.860	0.861	0.859	0.921
Click-based only	0.828	0.828	0.828	0.888
Posting-based only	0.839	0.84	0.837	0.904
Clicking statistics only	0.842	0.843	0.841	0.875

dataset do not have very large clicking traffic to support clicking-based features. However, these two results – on the two disparate ground truth datasets – demonstrate the viability of integrating click-based features into spam URL detection in social media, and the importance of integrating complementary perspectives (both posting-based and click-based) into such tasks.

Table 4. Top-6 features for manually-labeled dataset (Chi-square)

Rank	Features	Score	Category
1	Average clicks	149.41	Clicking
2	Posting count	144.23	Posting
3	Median followers	123.24	Posting
4	Median friends	118.19	Posting
5	Score function	87.00	Posting
6	Posting standard deviation	63.66	Posting

To further illustrate the significance of click and posting-based features, we consider two of the top-ranked features in both datasets (recall Table 2 and Table 4): median friends and average clicks. We compare the distributions of these two strongly correlated features for all spam URLs and benign URLs, in Figure 3. For URLs in the list-based dataset, as in Figure 3a, around 20% spam URLs are posted by users with a median friends count of 0, and yet around 20% have a median friends count that exceeds 1,000. These two types of posters could correspond to newly-registered accounts (0 friend) and "high-quality" accounts



(a) Median Friends **(b)** Average clicks

Fig. 3. Example feature comparison for spam and benign URLs

like those in a for-pay campaign. In contrast, legitimate accounts who posted benign URLs have relatively "normal" distribution of median friends, that is, most have median friends less than 300 and almost none has a zero median. For URLs in manually-labeled dataset, as in Figure 3b, we see that spam URLs tend to have a lower average clicks. A potential reason is that malicious URLs require more exposure or other "abnormal means" to support consistent clicks, while legitimate URLs can survive longer due to their appealing contents. We find similar distributions for other click-based statistics, including the click standard deviation and the effective average clicks.

5 Conclusions

In summary, this paper investigated the potential of behavioral analysis aiding in uncovering spam URLs in social media. Purely by behavioral signals, we have considered two perspectives – (i) how links are posted through publicly-accessible Twitter data; and (ii) how links are received by measuring their click patterns through the publicly-accessible Bitly click API. The core intuition is that these signals are more difficult to manipulate than signals such as the content of a social media post or the content of the underlying destination page. Through an extensive experimental study over a dataset of 7 million Bitly-shortened URLs posted to Twitter, we find accuracy of up to 86% purely based on these behavioral signals. These results demonstrate the viability of integrating these publicly-available behavioral cues into URL spam detection in social media.

Acknowledgment. This work was supported in part by AFOSR Grant FA9550-12-1-0363.

References

1. Antoniades, D., et al.: we.b: the web of short urls. In: WWW (2011)
2. Benevenuto, F., Magno, G., Rodrigues, T., Almeida, V.: Detecting spammers on twitter. In: CEAS (2010)
3. Canali, D., Cova, M., Vigna, G., Kruegel, C.: Prophiler: a fast filter for the large-scale detection of malicious web pages. In: WWW (2011)
4. Castillo, C., Donato, D., Gionis, A., Murdock, V., Silvestri, F.: Know your neighbors: web spam detection using the web topology. In: SIGIR (2007)
5. Chhabra, S., Aggarwal, A., Benevenuto, F., Kumaraguru, P.: Phi.sh/$ocial: the phishing landscape through short urls. In: CEAS (2011)
6. Cova, M., Kruegel, C., Vigna, G.: Detection and analysis of drive-by-download attacks and malicious javascript code. In: WWW (2010)
7. Cui, A., Zhang, M., Liu, Y., Ma, S.: Are the urls really popular in microblog messages? In: CCIS (2011)
8. Grier, C., Thomas, K., Paxson, V., Zhang, M.: @spam: the underground on 140 characters or less. In: CCS (2010)

9. Klien, F., Strohmaier, M.: Short links under attack: geographical analysis of spam in a url shortener network. In: HT (2012)
10. Lee, K., Caverlee, J., Webb, S.: Uncovering social spammers: social honeypots + machine learning. In: SIGIR (2010)
11. Lee, S., Kim, J.: WarningBird: Detecting suspicious URLs in Twitter stream. In: NDSS (2012)
12. Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Beyond blacklists: learning to detect malicious web sites from suspicious urls. In: KDD (2009)
13. Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Identifying suspicious urls: an application of large-scale online learning. In: ICML (2009)
14. Maggi, F., et al.: Two years of short urls internet measurement: security threats and countermeasures. In: WWW (2013)
15. McGrath, D.K., Gupta, M.: Behind phishing: an examination of phisher modi operandi. In: LEET (2008)
16. Neumann, A., Barnickel, J., Meyer, U.: Security and privacy implications of url shortening services. In: W2SP (2010)
17. Rodrigues, T., Benevenuto, F., Cha, M., Gummadi, K., Almeida, V.: On word-of-mouth based discovery of the web. In: SIGCOMM (2011)
18. Song, J., Lee, S., Kim, J.: Spam filtering in twitter using sender-receiver relationship. In: Sommer, R., Balzarotti, D., Maier, G. (eds.) RAID 2011. LNCS, vol. 6961, pp. 301–317. Springer, Heidelberg (2011)
19. Stringhini, G., Kruegel, C., Vigna, G.: Detecting spammers on social networks. In: ACSAC (2010)
20. Thomas, K., Grier, C., Ma, J., Paxson, V., Song, D.: Design and evaluation of a real-time url spam filtering service. In: SP (2011)
21. Thomas, K., Grier, C., Song, D., Paxson, V.: Suspended accounts in retrospect: an analysis of twitter spam. In: IMC (2011)
22. Wang, G., et al.: Serf and turf: crowdturfing for fun and profit. In: WWW (2012)
23. Wang, G., et al.: You are how you click: Clickstream analysis for sybil detection. In: USENIX (2013)
24. Wang, Y., et al.: Automated web patrol with strider honeymonkeys: Finding web sites that exploit browser vulnerabilities. In: NDSS (2006)
25. Wei, C., et al.: Fighting against web spam: A novel propagation method based on click-through data. In: SIGIR (2012)
26. Whittaker, C., Ryner, B., Nazif, M.: Large-Scale automatic classification of phishing pages. In: NDSS (2010)
27. Yang, C., Harkreader, R.C., Gu, G.: Die free or live hard? Empirical evaluation and new design for fighting evolving twitter spammers. In: Sommer, R., Balzarotti, D., Maier, G. (eds.) RAID 2011. LNCS, vol. 6961, pp. 318–337. Springer, Heidelberg (2011)