10-5-2011

Autonomous Link Spam Detection in Purely Collaborative Environments

Andrew G. West
University of Pennsylvania, westand@cis.upenn.edu

Avantika Agrawal
University of Pennsylvania, aagrawal@seas.upenn.edu

Phillip Baker
University of Pennsylvania, phills@seas.upenn.edu

Brittney Exline
University of Pennsylvania, kexline@seas.upenn.edu

Insup Lee
University of Pennsylvania, lee@cis.upenn.edu

Recommended Citation

Andrew G. West, Avantika Agrawal, Phillip Baker, Brittney Exline, and Insup Lee, "Autonomous Link Spam Detection in Purely Collaborative Environments", *7th International Symposium on Wikis and Open Collaboration (WikiSym '11)* , 91-100. October 2011. http://dx.doi.org/10.1145/2038558.2038574

Seventh International Symposium on Wikis and Open Collaboration, Mountain View, California, USA, October 2011.

Autonomous Link Spam Detection in Purely Collaborative Environments

Abstract

Collaborative models (e.g., wikis) are an increasingly prevalent Web technology. However, the open-access that defines such systems can also be utilized for nefarious purposes. In particular, this paper examines the use of collaborative functionality to add inappropriate hyperlinks to destinations outside the host environment (i.e., link spam). The collaborative encyclopedia, Wikipedia, is the basis for our analysis.

Recent research has exposed vulnerabilities in Wikipedia's link spam mitigation, finding that human editors are latent and dwindling in quantity. To this end, we propose and develop an autonomous classifier for link additions. Such a system presents unique challenges. For example, low barriers-to-entry invite a diversity of spam types, not just those with economic motivations. Moreover, issues can arise with how a link is presented (regardless of the destination).

In this work, a spam corpus is extracted from over 235,000 link additions to English Wikipedia. From this, 40+ features are codified and analyzed. These indicators are computed using "wiki" metadata, landing site analysis, and external data sources. The resulting classifier attains 64% recall at 0.5% false-positives (ROC-AUC=0.97). Such performance could enable egregious link additions to be blocked automatically with low false-positive rates, while prioritizing the remainder for human inspection. Finally, a live Wikipedia implementation of the technique has been developed.

Keywords

Wikipedia, collaboration, collaborative security, information security, link spam, spam mitigation, reputation, spatio-temporal features, machine-learning, intelligent routing

Disciplines

Databases and Information Systems | Numerical Analysis and Scientific Computing | Other Computer Sciences | Statistical Models

Comments

Seventh International Symposium on Wikis and Open Collaboration, Mountain View, California, USA, October 2011.

Autonomous Link Spam Detection in Purely Collaborative Environments

Andrew G. West, Avantika Agrawal, Phillip Baker, Brittney Exline, and Insup Lee
Dept. of Computer and Information Science - University of Pennsylvania - Philadelphia, PA
{westand, aagrawal, phills, kexline, lee}@seas.upenn.edu

ABSTRACT

Collaborative models (*e.g.,* wikis) are an increasingly prevalent Web technology. However, the open-access that defines such systems can also be utilized for nefarious purposes. In particular, this paper examines the use of collaborative functionality to add inappropriate hyperlinks to destinations outside the host environment (*i.e.,* link spam). The collaborative encyclopedia, Wikipedia, is the basis for our analysis.

Recent research has exposed vulnerabilities in Wikipedia's link spam mitigation, finding that human editors are latent and dwindling in quantity. To this end, we propose and develop an autonomous classifier for link additions. Such a system presents unique challenges. For example, low barriers-to-entry invite a diversity of spam types, not just those with economic motivations. Moreover, issues can arise with how a link is presented (regardless of the destination).

In this work, a spam corpus is extracted from over 235,000 link additions to English Wikipedia. From this, 40+ features are codified and analyzed. These indicators are computed using *wiki* metadata, landing site analysis, and external data sources. The resulting classifier attains 64% recall at 0.5% false-positives (ROC-AUC= 0.97). Such performance could enable egregious link additions to be blocked automatically with low false-positive rates, while prioritizing the remainder for human inspection. Finally, a live Wikipedia implementation of the technique has been developed.

Categories and Subject Descriptors

H.5.3 [**Group and Organization Interfaces**]: *collaborative computing, computer-supported cooperative work*;
K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

Keywords

Wikipedia, collaboration, collaborative security, information security, link spam, spam mitigation, reputation, spatio-temporal features, machine-learning, intelligent routing.

1. INTRODUCTION

As of this writing, six of the Internet's ten most-trafficked sites depend heavily on collaborative or user-generated content [2]. For example, the collaborative encyclopedia, Wikipedia [9], received over 88 *billion* page views last year to its English edition [15]. Such cooperative environments are unique in that end-users can add to, and sometimes modify, others' content. Additionally, such systems often encourage participation via intentionally minimal barriers-to-entry.

Unsurprisingly, malicious users see these characteristics as an asset: open-access permissions allow attacks to be carried out at low marginal cost, and those attacks have the potential to reach a large number of viewers. The malicious use on which this paper is focused is *link spamming*, the insertion of inappropriate hyperlinks (often to selfish ends). The pervasiveness and detection of such behavior has been the subject of recent research as it pertains to social networks [22] and forum/blog comments [16, 25, 30, 37].

However, little link spam research has been done in *purely collaborative* settings, such as Wikipedia (the focus of this writing). Most applications are only *partially collaborative* because of constraints on the editing model. For example, social networks often provide only local (peer-group) access. Similarly, blogs/forums are generally append-only in nature. Without these constraints, *wiki* environments could be among the most attractive to link spam attackers.

Despite having extensive anti-spam infrastructure [41], recent research confirms Wikipedia's vulnerability. The encyclopedia's primary weakness appears to be its reliance on human-driven link spam mitigation. Our prior work [41] suggests attack vectors to exploit human detection latency, showing it feasible to conduct profitable link spam campaigns. Similarly, Goldman [23] observes that a shrinking editor/administrative population and growing readership may make Wikipedia a more viable target. Finally, [37] indicates human protections may be insufficient against mechanized and increasingly intelligent blackhat software.

Addressing these shortcomings, this work proposes an autonomous classifier for link additions in purely collaborative environments. To create the classifier, a corpus was parsed from over 235,000 link additions to English Wikipedia. For each link added, we collected: (1) Wikipedia metadata (article, editor, *etc.*), (2) source code of the document being linked, and (3) third-party data about the URL (malware status, web statistics). These links were then labeled (spam or ham) using the implicit actions of Wikipedia experts.

This corpus was then used to identify features indicative of spam behavior, emphasizing those aspects unique to purely collaborative settings. Here, we identify, describe, and provide an intuitive basis for over 40 such features. Then, a classifier built from these is evaluated using cross-validation.

We find that the model is capable of detecting 64% of spam links with 0.5% false-positives. Such performance lags considerably behind that seen in other anti-spam domains (*i.e.,* email) and speaks to the difficulty of the task. However, some quantity of link spam can certainly be blocked autonomously. Beyond that, classification scores can help prioritize the manual efforts of anti-spam defenders; a vast improvement over the brute-force strategies currently employed. The proposed system has the potential to be an asset not just for the health and survival of Wikipedia, but to the collaborative paradigm as a whole.

The novel contributions of this work are four-fold:

1. Identification of link spam detection features unique to purely collaborative environments (*i.e.,wikis*).
2. Evaluation of link spam features found useful in partially collaborative settings (*e.g.,* blogs, UGC sites), in a purely collaborative environment.
3. Establishment of a corpus and performance baseline for link spam detection on which future work can build.
4. Implementation of our technique in a live setting, to the benefit of a user community.

This paper proceeds as follows: First, Wikipedia fundamentals are covered (Sec. 2) prior to discussing related work (Sec. 3). Then, a corpus is created (Sec. 4), features extracted (Sec. 5), and the resulting model evaluated (Sec. 6). Next, our practical live implementation is discussed (Sec. 7), before discussing evasion/gamesmanship strategies (Sec. 8). Finally, concluding remarks are made (Sec. 9).

2. BACKGROUND

In this section, preliminaries for the remainder of this work are established. First, general terminology is discussed (Sec. 2.1), before covering those aspects specific to link spam (Sec. 2.2). Then, the *status quo* defenses Wikipedia employs against link spammers are examined (Sec. 2.3).

2.1 Terminology

Wikipedia [9] is a collaborative encyclopedia consisting of many *articles*[1]. Each article consists of a version history, $H = \{r_0, r_1, r_2, \ldots\}$, where r_0 is an empty version. One creates a new version by performing an *edit* or *revision*, and these are most often visualized by computing the `diff` between r_{n-1} and r_n. When a new version, r_n, duplicates the content of a previous one, $r_{i,i<n}$, it is termed a *revert* or *undo.* Reverts are of interest because they are often used to correct damaging contributions.

Those who make edits are termed *editors* or *contributors.* Taken as a whole, the user-base is often called a *community.* Contributors can edit *anonymously* with no barrier-to-entry, or become *registered* and have persistent credentials.

Articles on Wikipedia are inter-connected using hyperlinks. When an article references another (internal) article, this is termed a *wikilink.* When the reference points outside the encyclopedia, it is an *external link.* A syntax defines how external links are created [11], and this writing concerns itself only with well-formed links of this kind.

2.2 Defining Link Spam

Put simply, any external link that violates Wikipedia policy [11] is considered to be *link spam*[2,3]. A link can be inappropriate due to its: (1) destination or (2) presentation.

A link's *destination* is the web property to which the link points, usually an HTML page (also called the *landing site*). Links to commercial sites are generally prohibited, as are those unfit for encyclopedic use (*e.g.,* most blogs, personal sites, *etc.*). *Presentation* concerns itself with on-wiki placement. For example, links must be context appropriate and the appearance (font, size, *etc.*) should be according to convention. Those who conduct link spam are termed *spammers.* However, it should be emphasized that not all unconstructive additions are made with malicious intent.

Making a spam/ham distinction is no trivial task, especially given Wikipedia's subjective policies. Fortunately, when a corpus is assembled in Sec. 4, Wikipedia experts are relied upon to perform labeling on a case-by-case basis.

2.3 Wikipedia Anti-Spam

A thorough description of Wikipedia's anti-spam functionality can be found in our prior work [41], which is briefly summarized here. First, HTML `nofollow` is applied to all outgoing links. Thus, Wikipedia cannot be used to attain backlinks for search-engine optimization (SEO) purposes. Instead, spammers must solicit direct traffic to obtain utility from the links they place (*i.e.,* via *click-throughs*).

Evidence suggests that the majority of spam links are discovered using brute-force *patrolling* strategies, where human users manually inspect link additions. Simple systems assist *patrollers* in this task. For example, an IRC channel reports link additions and another tool provides aggregate link information (*e.g.,* all the articles in which some URL appears) [14]. Additional functionality targets systematic abuse (a URL blacklist [13], anti-bot extensions [8], *etc.*)

Assuming a URL does not have an abusive history (triggering systematic protections), all anti-spam efforts are *mitigative* – being applied *after* the link has gone live. Thus, there is an inherent latency between the insertion of a spam link and its removal. Since practically all of these efforts are human-driven, such latencies can be lengthy, and spammers can harness these windows of opportunity [41].

The proposal of this paper is a *preventative* system which is brought to bear *immediately* on link additions. Egregious contributions can be undone[4] without human intervention (as false-positive tolerances permit). Beyond that, classification scores can be used to prioritize the efforts of patrollers, minimizing their latency (relative to random search).

3. RELATED WORK

Here, related work is surveyed – both Wikipedia specific (Sec. 3.1) and in alternative collaborative settings (Sec. 3.2).

[1] While discussion is Wikipedia specific, these concepts apply broadly to all collaborative applications.

[2] Other forms of Wikipedia spam exist. For example, entire articles could be created to advertise some product/service. These alternative forms are not considered in this writing.

[3] This definition of "link spam" is broader than that in other domains (*e.g.,* blog comments). However, all inappropriate links are "undesirable traffic" from Wikipedia's perspective, so we believe the "spam" terminology is appropriate.

[4] Since our tool will not be integrated into the *wiki* software directly, it cannot *block* link spam. Instead, it will actually revert live links. However, the tool's speed will make this distinction irrelevant, and therefore it is a *de facto* preventative system.

3.1 Wikipedia Specific

On Wikipedia, link spam is a subset of *vandalism*, a term describing all unconstructive edits. Much research has examined vandalism and its detection [17, 34, 35]. This includes *bots* operating autonomously [18] and user-driven intelligent routing tools that assist patrollers [6, 40].

However, most vandalism is not spam [35], and thus the aforementioned tools are not specifically designed to detect it. Quite the opposite, most vandalism is offensive or nonsensical [35], leading to the heavy use of natural-language processing (NLP) in the development of detection schemes.

While it seems unlikely that a spammer trying to garner traffic would engage in such language patterns, this writing still builds on anti-vandalism work. For example, metadata and reputation features [17, 42] can be used agnostic of the content type. This work examines how such features perform when used exclusively for link spam detection (Sec. 5.1). Furthermore, an anti-vandalism GUI tool [40] is repurposed for use by link spam patrollers (Sec. 7.3).

While link spam may not be the most common form of vandalism currently, recent work [23, 41] has suggested vulnerabilities. Spammers may be likely to exploit these weaknesses given their well-incentivized nature. We presume link spammers aim to profit from their actions (be it financial or simply narcissistic). Therefore, they should be motivated to avoid detection and actively evade protections (Sec. 8).

Our prior work [41] was the first to examine Wikipedia link spam in-depth. After showing that *status quo* spam behaviors on Wikipedia were inefficient, we proposed a novel and aggressive attack model which estimation showed could be carried out *at profit*. The viability of such an attack was a primary motivator of this work. The efforts herein construct a similar corpus (Sec. 4), but use it to identify features indicative of spamming behaviors (Sec. 5). Our approach intends to detect both *status quo* spam strategies (as of this writing, an annoying, but non-pervasive issue) and those more aggressive proposals of our earlier work[5] (which could have more devastating affects if unchecked).

3.2 Alternative Domains

The bulk of research on link spam detection has been performed in domains besides *wikis*, such as blogs/forums [16, 30] and social networks [22]. Strategies for preventing and mitigating such spam were broadly surveyed in [27]. This work examines if these techniques are applicable within the unique confines of a *wiki*/Wikipedia environment.

For example, analysis of destination content (*i.e.,* HTML) is one such strategy, attempting to quantify "commercial intention" and SEO strategies [19, 31]. Sec. 5.2 examines how these techniques fare on Wikipedia, where `nofollow` is used and most link spam is not commercial in nature [41].

Another oft-proposed technique is semantic NLP analysis, *i.e.,* measuring how well an addition fits into the context of existing content [29, 31]. However, such measures may prove less beneficial on Wikipedia where prior research indicates a lack of "blanket spamming" [41] and one can easily find an article relevant to any link destination.

The above evidence suggests there may be difficulty in applying blog/forum detection techniques[6] to Wikipedia. Yet, the *purely collaborative* nature of *wikis* permits novel features not possible in such *partially collaborative* settings. For instance, blog comments are typically append-only in nature, while content can be inserted at arbitrary positions in *wikis*. Such Wikipedia-driven features are identified, discussed, and leveraged in Sec. 5.1.

Recent work [37] also examines blackhat/SEO spamming software that specifically targets collaborative functionality (including *wikis*). That writing proposes that such software can be detected by packet-level analysis. Not having access to Wikipedia's network traffic flows, we are unable to quantify the performance of such techniques.

Finally, it deserves mention that Wikipedia anti-spam is an entirely volunteer effort. In contrast, profit-oriented sites often employ dedicated individuals to perform content inspection [38]. Thus, it seems especially pertinent to optimize what shrinking human resources are available [23], as our live implementation attempts to do (Sec. 7).

4. SPAM CORPUS

Next, the production of a link spam corpus is discussed. This required collecting external links added to Wikipedia, as well as associated data (Sec. 4.1). Then, spam/ham labels were applied to a subset of these additions (Sec. 4.2).

4.1 Link Collection

Wikipedia link additions were collected in real-time using an extension to the STiki framework [40]. By examining every `diff` to Wikipedia's article namespace (ignoring user profiles, discussion pages, *etc.*), external link additions were parsed per their syntax [11]. For each link added, data fields were retrieved and stored at three granularity:

1. Wikipedia: Using [7], all *wiki* metadata for the link is stored (editor, article, timestamp, edit summary). The URL and its hypertext description are also recorded, along with the full text of the article of appearance.
2. Landing site: Visiting the URL of the external link, the source code of the destination is obtained.
3. Third-party: Data was obtained about the URL via the Google Safe-Browsing Project [5, 36] (malware and phishing lists) and the Alexa Web Information Service [3] (`whois` fields, backlink quantity, *etc.*)

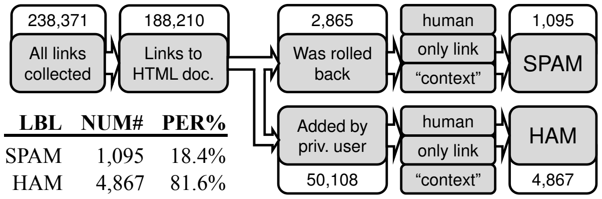
Data was collected during portions of Feb. and Mar. 2011, retrieving over 235,000 rows, consuming roughly 20GB of storage. It should be noted that the fetching/archival of arbitrary Internet documents does raise certain legal and ethical issues (*e.g.,* child pornography [43]). On the advice and approval of our institution's General Counsel, steps were taken to avoid acquiring/rendering image content.

4.2 Link Labeling

Having acquired link data, we produce a corpus of labeled spam/ham entries. The corpus composition is summarized by Fig. 1. Then, the accuracy and consequences of this labeling approach are discussed. As a preliminary, only rows where the destination is an HTML document are considered (all file types can be handled in practice, see Sec. 7.1). This filter removed some 50,000 (21%) of rows from eligibility.

[5] Note that because [41] proposes novel attack vectors, such behaviors are not captured in our corpus. Thus, static rules must be written to prevent spam campaigns of that kind (see Sec. 7.2).

[6] Two systems believed to be employing such techniques are Akismet [1] and Defensio [4]. These services are closed-source commercial offerings filtering blog comments and other postings. An exacting comparison is not possible due to their proprietary nature. Our system's implementation (Sec. 7) is free/open-source.



LBL	NUM#	PER%
SPAM	1,095	18.4%
HAM	4,867	81.6%

Figure 1: Summarizing corpus creation

Labeling Spam Revisions: To find spam edits, we rely upon a technique developed in prior literature [41, 42]: the *rollback* action is an expedited form of revert/undo available to privileged users. Rollback is only appropriate when undoing "blatantly unproductive" contributions, and thus any edit undone via rollback can be considered vandalism [42].

However, extracting the link spam subset requires additional work. First, we consider only rolled-back edits where *exactly one* link was added. Then, manually inspecting those `diffs`, we discard revisions where modifications were made apart from the link and its immediate supporting context. The remaining revisions are *link spam.* Intuitively, we know these to be link spam edits because the link is the *only* change made, and therefore the decision to rollback the edit speaks *directly* to the inappropriateness of that one link [41].

Of the 188,210 collected links with HTML destinations, 2,865 (1.5%) were undone via the rollback action. Of these, 1,510 were the only link added in their edit. Finally, 1,095 passed the manual inspection and "no-bot" criteria[7], forming the spam portion of the corpus.

Labeling Ham Revisions: Having identified link spam edits, the complementary ham labels are produced. Desiring accurate tags and low noise, it is insufficient to treat all non-spam links as appropriate ones (as done in prior work [41]). To arrive at ham labels, we consider those links added by privileged users. Given that we trust such individuals to label poor additions, by extension, they can be trusted to apply the same wisdom when *adding* links. For consistency, these links are also subjected to the "one link added", "manual inspection", and "no bot" filters.

Of the 188,210 links with HTML destinations, 50,108 (26.6%) were added by a user with rollback privileges. Of these, 4,867 met all criteria for inclusion.

Discussion: Combining the labeled sets, we arrive at a corpus[8] with 5,962 entries: 81.6% ham and 18.4% spam (see Fig. 1). Given the labeling technique, this does *not* speak to the actual prevalence of inappropriate links.

Though just 2.5% of all links collected are in the final corpus, our labeling strategy allows us to arrive at tags with high confidence. This strategy is an advantageous one because it: (1) autonomously operates based on implicit actions, (2) allows a case-by-case interpretation of link spam, and (3) leverages the experience and knowledge of Wikipedia experts. Trusting these experts is justified: just one spam edit (0.09% of spam) was committed by a privileged user.

However, as a consequence of the labeling technique *some features cannot be utilized.* All ham edits are made by privileged users, making it biasing to encode how they attained or wield that status. For example, "user registration status" and "account age" are two prohibited features that would otherwise be of interest. Similarly, quantifying `diff` magnitude is inappropriate given the "context criteria." Such biases are carefully avoided when developing features in Sec. 5.

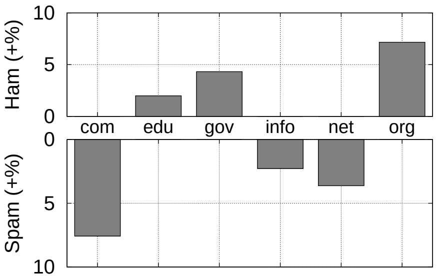


Figure 2: Top-level domain (`URL_TLD`)

Other corpus constraints are less severe, and Sec. 7.1 describes generalizations so our classifier can score all edits.

5. FEATURE SELECTION

The corpus is used to determine features indicative of link spam behavior. Space considerations[9] prevent a comprehensive discussion of all features, which Tab. 1 lists. Such a diversity of features is needed because of varying spam strategies [41]. For example, one could use subtle strategies, in the hopes of having a link become *embedded* with a long survival time. Alternatively, an attacker could be aggressive, attempting to maximize utility until detection.

This write-up concentrates on novel features and those weighted heavily in the classifier (see Tab. 1). Discussion closely follows the presentation order of that table. Features are organized by data source: Wikipedia (Sec. 5.1), the landing site (Sec. 5.2), and third-party services (Sec. 5.3). All features operate only on information available at the time an edit was committed (*i.e.,* zero-delay detection [17]).

5.1 Wikipedia-driven

Wikipedia-specific features are our starting point. While some "metadata" and "article" based signals are inspired by anti-vandalism work [17, 34], we also develop novel features capturing properties of link presentation and history.

URL Properties: The URL itself is first scrutinized[10]. At median, spam URLs are 1.7× shorter than ham ones (`URL_LEN`, Tab. 2), likely because spam links point to domains 30% more often than a specific file (`URL_IS_DOMAIN`, Tab. 2). This is intuitive: "main pages" are unlikely to contain encyclopedic information, but can be promotional.

Similarly, Fig. 2 visualizes data about the top-level domain utilized (`URL_TLD`). It is unsurprising to see that `*.gov` and `*.edu` TLDs are well-behaved, given their greater administrative governance. Appropriately, `*.info` domains are penalized, as these are some of the *cheapest* domains to register and could therefore be used in Sybil attacks [20].

[7] To maintain a human validated set, edits that were undone (or made) by autonomous bots were removed from analysis.

[8] Efforts to open-source the corpus are encumbered due to it containing: (1) potentially copyrighted/illegal content, and (2) non-free data points [3]. Interested parties should contact the authors.

[9] Our open-source implementation (Sec. 7) allows feature calculation to be examined at greater depth.

[10] We consider these to be "wiki" features because they are a matter of presentation; a single landing site could have multiple URLs.

FEATURE	SRC	TYP	RNK	DESCRIPTION
URL_TLD	W	enum	●●●	Top-level domain of the URL (*e.g.,* *.com or *.edu)
URL_LEN	W	num	●●	Length (in characters) of the URL being added
URL_IS_DOMAIN	W	bool	●●	Whether the URL points to a broad domain/folder or specific file
URL_SUBDOMAINS	W	num	●●	Quantity of subdomains in the URL (*i.e.,* sub.example.com = 3)
LINK_IS_CITE	W	bool	●●●●●	Whether the link was added per a special reference/citation format
LINK_PLACEMENT	W	num	●●●	Where in the article the link was added (as function of article length)
LINK_TEXT_LEN	W	num	●●	Length (in characters) of the hypertext description of added link
LINK_DISCUSSED	W	bool	●●	Whether the link/URL is found on the article's discussion page
ART_TS_CREATION	W	num	●●●●●	Age of the article to which link was added (*i.e.,* time-since creation)
ART_REPUTATION	W,A	num	●●●●	Historical, time-decayed measure of vandalism/controversy on article, per [40]
ART_REFERENCES	W	num	●●	Quantity of citations/references in the article of link addition
ART_LENGTH	W	num	●	Length of the Wikipedia article to which the link was added
ART_POPULARITY_*	W	num	-	Article visitors in last $t \in \{hour, day, week, month, 6\text{-}months\}$, per [10]
ART_EDITS_TIME_*	W,A	num	-	Article edits committed in last $t \in \{hour, day, week, month, 6\text{-}months\}$
URL_ADDS_TIME_*	W,A	num	-	Links to URL added in last $t \in \{hour, day, week, month, 6\text{-}months\}$
DOM_ADDS_TIME_*	W,A	num	-	Links to domain added in last $t \in \{hour, day, week, month, 6\text{-}months\}$
URL_REPUTATION	W,A	num	●●●●	Historical, time-decayed measure of spam-iness for added URL
URL_DIVERSITY	W,A	num	●●●●	Of all the times the URL has been linked, the % added by the current editor
DOM_REPUTATION	W,A	num	●●●	Historical, time-decayed measure of spam-iness for added domain
DOM_DIVERSITY	W,A	num	●●●	Of all the times the domain has been linked, the % added by the current editor
META_COMM_LEN	W	num	●●●●●	Length (in characters) of the revision summary
META_TIME_DAY	W	num	●●●●	Time-of-day when the link was added (UTC locale)
META_DAY_WEEK	W	enum	●	Day-of-week when the link was added (UTC locale)
SITE_PROFANE	L	num	●●●	Measure of the prevalence of profane language on the landing site
SITE_NUM_IMGS	L	num	●●●	Quantity of images displayed on the landing site
SITE_SIZE	L	num	●●●	Size (in bytes) of the textual content on the landing site
SITE_COMPRESS	L	num	●●●	Ratio of raw content-size to compressed size; speaks to repetitiveness
SITE_TITLE_LEN	L	num	●●	Length of the HTML title, in characters (*i.e.,* <title>. . . </title>)
SITE_NUM_META	L	num	●●	Quantity of HTML <meta keywords="$w_1, w_2, \ldots, w_n$"> on site
SITE_VOCAB_LEN	L	num	●●	The average word length of visible textual content on the landing site
SITE_COMMERCIAL	L	num	●	Measure of the commercial intent of the landing site
SITE_RELEVANT	L,W	bool	●	Whether the landing site is topic-similar to Wikipedia article of addition
ALEXA_BACKLINKS	T	num	●●●●●	Quantity of incoming links to landing site, per the crawling by [3]
ALEXA_DELTAS	T,A	num	●●●●●	Meta-feature speaking to site's historical traffic patterns, per [3]
ALEXA_ADULT	T	bool	●●●●	Whether or not the URL contains adult content, per [3]
ALEXA_SPEED	T	num	●●●	Load time of landing site, as a percentile of all sites, per [3]
ALEXA_AGE	T	num	●●●	Time that the landing site has been online, per the crawling by [3]
ALEXA_CONTINENT	T	enum	●●	Continent to which the whois registration of site maps, per [3]
GOOG_MALWARE	T	bool	●	Whether URL is active on the Safe-Browsing "malware" list, per [5]
GOOG_PHISHING	T	bool	●	Whether URL is active on the Safe-Browsing "phishing" list, per [5]

Table 1: Comprehensive listing of features used, organized by data source. Sources are: (*W*)ikipedia, (*L*)anding site, and (*T*)hird-party. (*A*)ggregate features are also indicated. Feature rank/importance was calculated by performing a greedy step-wise comparison over feature subsets [24, 28]. More bullets indicate greater weight in the final classifier. For brevity, rank is omitted for features having multiple variations.

Link Properties: Link presentation is also of interest. One heavily weighted feature is if the link is part of a "citation" environment (LINK_IS_CITE). As Tab. 2 shows, citations are 6.5× less likely to be spam. This feature also correlates well with *where* in the article the link is placed (LINK_PLACEMENT). By convention, straightforward hyperlinks (*i.e.,* not citations) are confined to an "External Links" section at the bottom of an article. Even spam links adhere to this rule – being placed about $^3/_4$ of the way through the article (Tab. 2). Clearly, spammers are not using prominence to solicit reader (or administrative) attention [41].

Although uncommon, when a link appears on the article's "discussion" page before it is posted to the article, it tends to be constructive (LINK_DISCUSSED, Tab. 2). This is likely an attempt to reach consensus on if the link should be added.

Article Properties: Focus now shifts to the Wikipedia articles to which links are added. We find that spam tends to target *more popular* (ART_POPULARITY_*, Fig. 3c) and *older* (ART_TS_CREATION, Tab. 2) articles than ham links. This may be an attempt to maximize link exposures, but could also invite administrative scrutiny.

Similarly, the scatter-plot of Fig. 4 shows that the previous section's LINK_PLACEMENT feature correlates strongly with ART_TS_CREATION. Links that are added: (1) far down the article, on an (2) old article, have a high spam probability. This is logical: old articles are likely to have mature/stable content (links included). While citations (likely to be in the article body) may be required to update an article, it is far less likely that general "external links" will be ham.

Moreover, articles which have been problematic in the past tend to continue that trend. This makes an article reputation metric (ART_REPUTATION, per [42] and its API [40]) particularly relevant. In 43% of spam cases, the article had a recent history of spam and/or vandalism (vs. 24% for ham).

URL/Domain Aggregates: The Wikipedia history of a web property (*i.e.,* URL or domain) is one of the best indicators of its quality, capturing intuitions such as:

- Web properties with a spam history are suspicious.
- Unusually rapid linking to a web property is suspicious.
- Little editor diversity for a web property is suspicious.

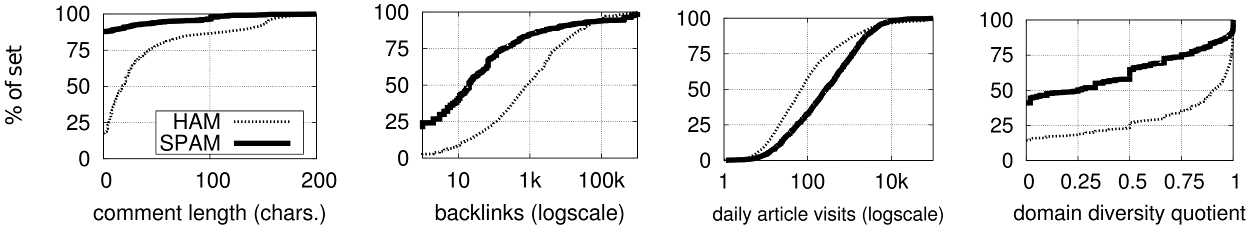


Figure 3: CDFs for (a) `META_COMM_LENGTH`, (b) `ALEXA_BACKLINKS`, (c) `ART_POPULARITY_DAY`, and (d) `DOM_DIVERSITY`

FEATURE	UNIT	HAM	SPAM
`URL_LEN`	chars.	64	38
`LINK_PLACEMENT`	% of article	41	73
`LINK_TEXT_LEN`	chars.	26	24
`ART_TS_CREATION`	months	146	192
`URL_IS_DOMAIN`	boolean	6.3%	37.5%
`LINK_IS_CITE`	boolean	53.9%	8.3%
`LINK_DISCUSSED`	boolean	4.5%	2.4%

Table 2: Wikipedia feature comparison. Non-boolean features presented at median.

These notions are represented by: (1) raw counts, (2) time-decayed reputations built atop the rollback action [42], and (3) user-link diversity quotients. Each of these signals is calculated over varying time windows to capture historical trends (see Tab 1). Additionally, each feature is quantified at both URL and domain granularity[11] (again, Tab. 1).

Diversity quotients lend themselves to human interpretation. For example, Fig. 3d indicates that 40% of spam links are added by an editor who is responsible for *all* recent links to that domain, versus 15% for ham (`DOM_DIVERSITY`). No matter the contributor, long-term prevalence is indicative of link quality: ham domains have 5× the 6-month quantity of spam ones (`DOM_ADDS_TIME_6MOS`).

Reputation and raw counts are also strong benchmarks, but trend discovery requires multi-dimension analysis (easy for a classifier, but non-trivial to present). Normalization is an important component of such reasoning: consider that YouTube averages nearly 2,000 link additions monthly.

Metadata: Entire anti-vandalism systems have been built atop revision metadata [42], and such features are now evaluated in an anti-spam setting. For example, the length of the revision summary/comment (`META_COMM_LEN`) is the second-most heavily weighted feature in the classifier. Some 88% of spam leaves this field blank (versus 17% of ham, see Fig. 3a). Omitting a summary hints that one is not familiar with Wikipedia conventions and therefore may be unaware of the linking rules under which the encyclopedia operates.

Prior work [42] also showed that most vandalism happens on weekdays (`META_DAY_WEEK`) during normal "business hours" (`META_TIME_DAY`). However, the inability to localize UTC timestamps using IP-geolocation (registered users' IPs are hidden) hampers comparison with that prior result. Regardless, there exists strong temporal patterns separating spam and ham edits, as Fig. 5 demonstrates.

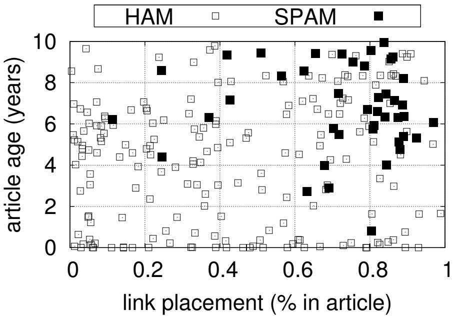


Figure 4: `ART_TS_CREATION` vs. `LINK_PLACEMENT`

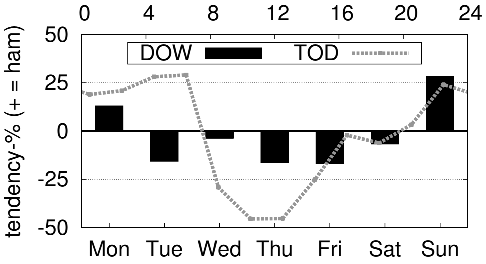


Figure 5: Day-of-week (DOW, `META_DAY_WEEK`) and time-of-day (TOD, `META_TIME_DAY`)

5.2 Landing Site Processing

Inspired by spam webpage detection [19, 31] (regardless of the delivery mechanism), our classifier implements a landing site processing component. However, we find the contribution of these features to be incremental, as the diversity of inappropriate links added to Wikipedia exceeds that of "stereotypical" spam. These findings are now summarized.

Language Properties: Foremost, one might expect spam links to be overwhelming commercial. To quantify this, destination content was run through extensive (1000+ element) regular expressions capturing marketing terminology. Spam sites were found to be only marginally more commercial than ham ones (`SITE_COMMERCIAL`, Tab. 3), supporting prior Wikipedia research [41]. In a similar manner, vulgarity was quantified to mitigate "shock sites" and inappropriateness, with slightly better results (`SITE_PROFANE`, Tab. 3).

Literature [19, 31] also describes statistical language properties typical of spam webpages. To this end, we implemented features for the size (`SITE_SIZE`), compressibility (`SITE_COMPRESS`), and a measure of the vocabulary complexity (`SITE_VOCAB_LEN`) at the destination. While these features figure moderately in the classifier (see Tab. 1), the results sometimes disagree with those in prior research. For

[11] Domains capture broad trends which may be able to evade URL-specific analysis. However, some domains may be too broad (*e.g.,* social-networking sites). Future work intends to draw distinctions at all points along a URL's domain/folder hierarchy.

FEATURE	UNIT	HAM	SPAM
SITE_COMMERCIAL	ratio	1.00	1.03
SITE_PROFANE	ratio	1.00	1.08
SITE_SIZE	kilobytes	32.99	27.05
SITE_COMPRESS	ratio	3.82	3.97
SITE_VOCAB_LEN	chars.	4.21	3.98
SITE_RELEVANT	boolean	37.3%	33.8%

Table 3: Landing site feature comparison. Non-boolean features presented at median.

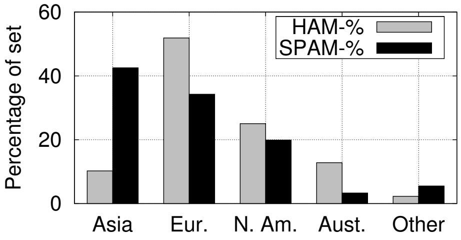


Figure 6: Host continent (ALEXA_CONTINENT)

example, spam content was found to have slightly shorter average word-lengths than ham (Tab. 3), contrasting with [31].

Other research [29] relies on "language model disagreement", the notion that spam contributions do not fit the "context" of the surrounding content. To measure this, a naïve measure of relevance was constructed: whether the Wikipedia article *title* appears *verbatim* on the landing site (SITE_RELEVANT). With 37% of ham and 33% meeting this criteria (Tab. 3), the feature's weight is nominal. Future work intends to leverage more rigorous Bayesian and n-gram probabilities over the entire Wikipedia article. However, such techniques may scale poorly in a live implementation.

SEO Tactics: Given that a spammer has taken to Wikipedia to publicize a site, one might expect that he/she would attempt to maximize traffic via other tactics (*i.e.,* search-engine optimization). Lengthy <meta keywords="..."> and <title> blocks are two simple and common SEO tactics. Surprisingly, we observe that spam edits have slightly shorter titles (SITE_TITLE_LEN, 6.8 vs. 7.5 words) and fewer meta keywords (SITE_NUM_META, 5.3 vs. 6.6 words). This may suggest there is fallacy in assuming the contributor of a spam link is actually the landing site operator: one could simply be *lobbyist* for a particular person or agenda.

5.3 External Data

Next, third-party services are used to discern spam landing sites, namely Alexa Web Information [3] and Google's Safe-Browsing project [5, 36]. These well-regarded providers can perform analysis at a depth and breadth that would otherwise be outside the scope of this work.

Alexa [2, 3]: Alexa is a company producing web-statistics via Internet usage monitoring and a web crawler. For each link added, we query their subscription service which provides data about backlinks, traffic patterns, and site hosting.

The quantity of backlinks (ALEXA_BACKLINKS) a landing site has, as visualized in Fig. 3b, is the feature weighted *most heavily* in our classifier. In the median case, a ham site has ≈850 backlinks, compared to just 20 in the spam case (a 40× difference). This is unsurprising given that backlinks are recognized as a good measure of site reputation and the basis for well-known search-engine rank algorithms [32].

Site popularity and traffic trends can also capture reputation. One would expect that sites with a consistently high number of visitors might be appropriate destinations. Such notions are captured by ALEXA_DELTAS, a meta-feature (*i.e.,* lower-order classifier) built from ≈50 data points. Its final rank (Tab. 1) speaks to its predictive nature. Similarly, reputable sites are likely to be quick loading (ALEXA_SPEED) and maintain their Internet presence (ALEXA_AGE). At median, spam sites are two years *younger* than ham ones.

Distinct from reputation, one might consider the genre of the site content. Alexa indicates adult hosts (ALEXA_ADULT), and spam sites are 8× more likely to be adult in nature (0.8% of ham and 6.5% of spam links have this property[12]). One can also examine *where* the site is hosted (ALEXA_CONTINENT, Fig. 6). Similar to email spam [39], Asia and Europe are common spam sources. In fact, Asia hosts four times as many Wikipedia spam destinations (relatively) as ham ones.

Finally, in some cases, Alexa is *missing data* about a URL, likely because their crawler has yet to encounter it. Missing data might suggest a site is new or poorly connected, both indicative of low quality. Empirical data shows that 26% of spam links have missing crawler data, compared to 5% for ham links. However, it is somewhat dubious to leverage the "shortcomings" of another service. Therefore, our current classifier treats such features as "missing", incurring no penalty. No feature is codified to formalize this notion.

Google Safe-Browsing [5, 36]: By overlaying machine-learning and virtual-machine sandboxing atop its Internet crawler, Google produces lists of suspected malware and phishing sites (GOOG_MALWARE, GOOG_PHISHING). Ostensibly, utilizing these lists could prevent Wikipedia from becoming a vehicle for malware delivery and scamming behaviors.

The *entire* data collection (of 235,000 links) produced just 31 hits on these lists. None of these links were assigned the "spam" label for a variety of reasons, and thus the features are a non-factor in the classifier. Nonetheless, this data point is still described and collected so we can write static rules (Sec. 7.2) capable of mitigating future malware attacks, regardless of their *status quo* prevalence.

6. CLASSIFER PERFORMANCE

Having identified individual features, their performance is now analyzed in combination. To build the classifier, the Weka [24] implementation of the alternating decision tree (ADTree) algorithm [21] is used. ADTree is chosen because of its: (1) performance, (2) support for enumerated and missing features (as sometimes occur with third-party data), and (3) output of a human-readable model. All results were obtained via 10-fold cross-validation over the corpus.

Simple Performance: Results are summarized by Fig. 7 and Tab. 4. Examining precision-recall (Fig. 7a), it is clear our method significantly outperforms a control classifier. While pure chance operates at ≈18% precision (the percent of the corpus which is spam), our system has precision greater than 90% for 80% of the recall spectrum.

[12] The fact that not *all* adult content is spam underscores why Wikipedia link spam detection is difficult. An adult film star's article can legitimately link to his/her "official site", but in many other contexts the same link would be grossly inappropriate.

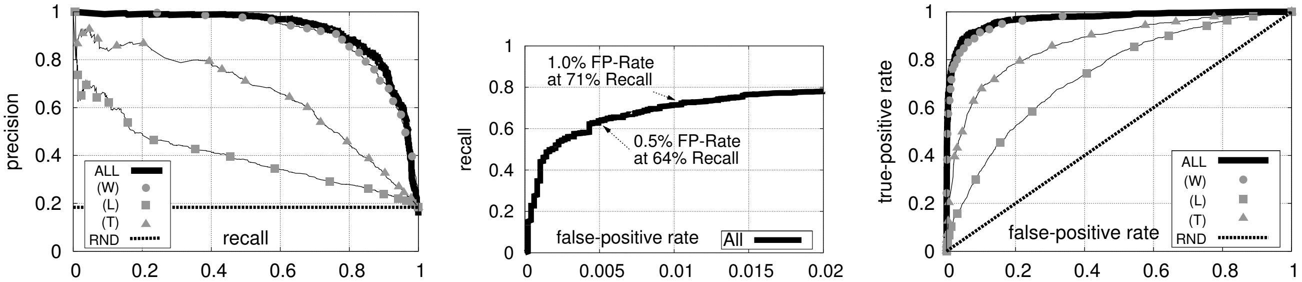


Figure 7: Result curves for (a) precision-recall, (b) recall vs. false-positive rate, and (c) ROC. Feature subsets include: (W)ikipedia, (L)anding site, and (T)hird-party (see Tab. 1).

Also of interest is Fig. 7b, which plots recall as a function of false-positive (FP) rate. In order for the classifier to autonomously block spam contributions, it must adhere to the FP-tolerances of the community (see Sec. 7.3). Fig. 7b shows that 64% of spam could be detected at a 0.5% FP-rate, suggesting that a considerable portion of the spam mitigation burden could be lifted from human patrollers.

Feature Subsets: Fig. 7a and Fig. 7c also visualize the performance of feature subsets. Most striking is that the "Wikipedia" class alone is nearly equivalent in performance to the complete classifier. While it is encouraging to see the strong contributions of these novel features, it also forces one to consider the necessity of the other subsets. However, by the very nature of *wikis* and collaborative environments, it is "Wikipedia" features which are most easily manipulated (see Sec. 8). Therefore, the robustness of the other sets could prove critical in capturing evasive tactics and play a greater role when the live implementation (Sec. 7) is retrained.

Given the quantity and strength of "Wikipedia" features, it is interesting to examine what "sub-subset" is most heavily weighted. Such groupings are delineated per the organization of Sec. 5.1. We find that "metadata" (PR-AUC=0.59) and "URL/domain aggregates" (PR-AUC=0.66) are most significant, but neither approaches the composite performance of all Wikipedia signals (PR-AUC=0.91, Tab. 4).

This work also implemented features motivated by the anti-spam efforts of blogs and webpage processing [19, 29, 31]. Of particular interest was how these features, captured by the "landing site" subset, would fare in a purely collaborative environment. As Fig. 7 shows, this is the worst performing subset by a substantial margin. While discouraging, it also confirms some of our initial intuition that Wikipedia spam behaviors are a unique phenomena that require distinct detection machinery from "typical" spam links.

Finally, Alexa features (the force of the "third-party" subset) perform surprisingly well in isolation, especially considering that the service is designed as a marketing data service, not an anti-spam tool.

Performance Discussion: Unfortunately, our technique performs far less accurately than state-of-the-art *email spam* mitigation schemes. However, the system performs comparably to Wikipedia anti-vandalism classifiers [34], a domain that has received considerable research attention.

There is little doubt the proposed system can help Wikipedia control *status quo* spamming behaviors, which might be characterized as a "nuisance." More significant is its ability to mitigate aggressive and mechanized tactics that could lead to pervasive damage. Even in its purest form (*i.e.,* absent the static rules of Sec. 7.2), we are confident the classifier can deflect the recently proposed attacks of [41]. Static rules will add an additional level of reassurance.

Throughout this work, evaluation has been performed over tagged corpus edits, yet these edits compose just 2.5% of those collected. While definitive and noise-free labels are advantageous for training, it remains to be seen if the associated edits capture all the subtleties necessary to make accurate spam/ham predictions. Certainly, a sizable portion of unlabeled data is rife with ambiguity (regarding its quality), and it is unlikely even human editors could reach a definitive spam/ham distinction. In a live implementation, however, all links must be scored and the non-human nature of our tool might invite criticism over false-positives.

Improving Performance: While the classifier performs well, future improvements intend to build on this foundation. Data collection is ongoing to improve corpus scope. Moreover, a corpus built without labeling bias (Sec. 4.2) would enable additional features. Recently, [33] assembled a vandalism corpus using outsourced human annotators, and a similar configuration is imaginable for anti-spam purposes.

7. LIVE IMPLEMENTATION

Having demonstrated that our classifier significantly outperforms random search (the *status quo* patrol technique), it seemed prudent to encode the technique for the Wikipedia community. We undertook this task, with an implementation currently operational on English Wikipedia (open-source code available at [40]). This required practical considerations outside of those encountered with the offline corpus (Sec. 7.1) and static rules to handle special circumstances (Sec. 7.2). Further, the tool provides streamlined access to the classification scores (Sec. 7.3). Fig. 8 visualizes the system model/architecture of this implementation.

7.1 Generalizations

Our corpus was designed with the goal of having accurate labels, leading to many constraints on the complete set of edits collected. In practice, however, the classifier should be able to *score* all revisions adding an external link(s)[13]:

Multiple Links: The corpus contains edits where exactly one link was added. In order to score revisions contributing multiple links, we begin by processing each link independently. Then, the score assigned (to the edit) should be the

[13] When *scoring* a link, the classifier outputs a real-value which speaks to the probability a revision is link spam (not a binary prediction). Higher scores are more indicative of spam.

FEATURES	PR	ROC
Random	0.184	0.500
Wikipedia (W)	0.909	0.968
Landing site (L)	0.399	0.738
Third-party (T)	0.656	0.866
Combo (W+L)	0.902	0.965
Combo (W+T)	0.915	0.970
Combo (L+T)	0.667	0.872
All (W+L+T)	0.917	0.971

Table 4: Area-under-curve (AUC) for precision-recall (PR) and receiver-operating-characteristic (ROC), for various feature subsets.

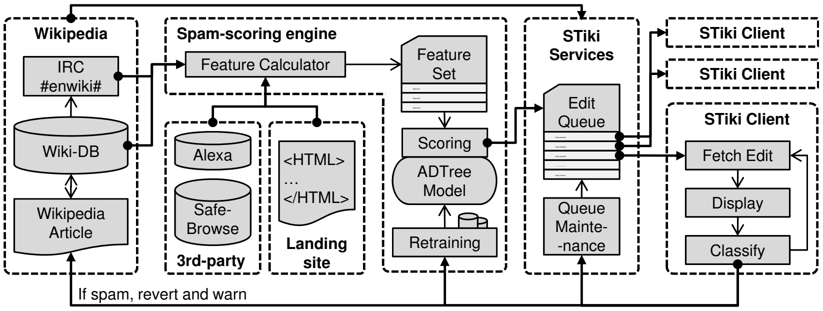


Figure 8: Architecture for spam detection implementation

maximum of those scores. In this manner, spammers cannot use constructive links to dilute inappropriate ones.

Non-HTML Destinations: A majority (79%) of destinations are HTML documents, for which there are specific features. However, other content types can be harnessed for malicious purposes. To prevent evasion, such additions are scored without using "landing site" features (at slightly decreased performance, see Tab. 4).

7.2 Static Rules

Other implementation practicalities are more acute. In these cases, static scoring rules are installed:

Acquisition Errors: Destinations returning HTTP 4xx or 5xx error codes (*e.g.,* "404 Not Found") are scored arbitrarily high. An inaccessible landing site serves no purpose, speaks to unreliability, and violates policy [11].

Novel Attack Vectors: Recent work [41] has identified novel link spam attacks, not yet in active use, and therefore not trained upon. Static detection rules are authored to prevent widespread damage via these channels. For example, the Safe-Browsing lists [5, 36] (per Sec. 5.3) are utilized in such a fashion. Similarly, if novel spam strategies do arise, the classifier can be retrained to capture them.

7.3 Utilizing Revision Scores

Having quantified a probabilistic link spam metric for revisions, these scores need to be applied and disseminated.

One goal is for the system to *autonomously* undo egregious link additions. Such operation requires Wikipedia approval, which is being sought as of this writing. Generally, a false-positive tolerance is set and score thresholds are tuned accordingly. For example, one anti-vandalism bot [18] operates at a FP-rate of 0.5%. Per Sec. 6 and Fig. 7b, our system could detect 64% of link spam at such tolerances.

Regardless of the outcome of that approval process, patrollers can use classifier scores to prioritize spam search efforts (*i.e.,* for *intelligent routing*). This has already been achieved by interfacing with STiki [40] – GUI software providing crowdsourced access to a shared priority queue of revisions in need of inspection (see Fig. 8). STiki requires only that ID/score pairs are provided, as its core engine handles all backend logic (*e.g.,* de-queuing inspected or non-current revisions). Critically, the human assessments gathered using the tool can be used to refine scoring techniques.

Finally, an API and IRC feed have been made available [40] so other developers can access the calculated features/scores.

8. EVASION & GAMESMANSHIP

Having implemented a link spam classifier/scorer for Wikipedia, we now consider how a user might evade our system. Given spammer's well-incentivized nature, such attempts at gamesmanship are a realistic concern.

First, any attacker who is aware of the model and the intuition on which the system is built has some advantage. An attacker could manipulate his/her edit or landing site so that it is scored more favorably. Admittedly, some features can be easily gamed (*e.g.,* `META_COMM_LENGTH`, the revision summary length). Fortunately, others are more robust in that they, (1) are not easily affected, or (2) increase marginal costs for attackers. For example, Sec. 5.3 described the "traffic" (`ALEXA_DELTAS`) and "backlinks" (`ALEXA_BACKLINKS`) features, which are difficult to manipulate. Similarly, using Sybil attacks [20] to side-step URL and domain reputations (`URL_REPUTATION` and `DOM_REPUTATION`, Sec. 5.1) would require multiple domain registrations. Such spatio-temporal signals have been shown difficult to circumvent [26, 42] and the classifier integrates several features of this kind.

Absent content-optimization against the model, we (non-exhaustively) consider several other attack vectors:

TOCTTOU attack: A time-of-check-to-time-of-use sattack leverages the fact that scoring is performed only at link addition. By altering destination content (or using redirection) after this time, one can link to sites that would otherwise be penalized. Such behavior has already been seen in active use [12] against human patrollers. An obvious solution is to re-scan sites on some interval and report on significant scoring changes. This, of course, would require substantial resources (English Wikipedia currently has some 36 million external links). Scalability could be increased by producing a whitelist of domains that are trusted to have stable content.

Crawler redirection: Similar to a TOCTTOU attack, an attacker could serve benign content to our crawler, but serve spam content to ordinary visitors. Detecting the IP address from which our service operates would be straightforward. One solution is to distribute the fetch operation, possibly using anonymization networks. Ultimately, such landing site manipulation is a reason we implemented orthogonal feature types (*i.e., wiki*-centric and third-party data).

Denial-of-service: By overwhelming the service with requests (*i.e.,* link additions to Wikipedia) an attacker could delay the processing of subsequent link spams. In addition to parallel analysis, our system uses static rules to handle unreasonably large landing sites and edits adding *many* links.

9. CONCLUSIONS

In this work, we have described the problem of purely collaborative link spam and justified the need for its autonomous detection. To this end, we proposed a mitigation strategy and evaluated it over Wikipedia revisions.

From the outset we suspected that purely collaborative environments (*e.g.,* wikis) were unique from partially collaborative ones (*e.g.,* blogs) and might require specialized anti-spam machinery. This was confirmed by implementing features inspired by past blog/forum research, finding their performance nominal. These shortcomings, however, were overcome by leveraging properties specific to *wiki* environments. When combined with third-party data, features built on these properties produce an effective and robust classifier.

It is clear this work will benefit the Wikipedia community, especially given our live implementation of the technique. Offline analysis demonstrated that two-thirds of Wikipedia link spam can be automatically mitigated (at low false positives), while prioritizing the remainder for human inspection. This is a considerable improvement over current strategies, which rely on brute-force human effort.

However, this work also intends to have broader implications. Our extensive feature set captures properties that exist in general-purpose *wikis*, not simply those specific to encyclopedic content. Moreover, a performance baseline has been established on which future work can build. Ultimately, we hope our technique is a foundation towards better securing the entire collaborative paradigm.

Acknowledgements

This research was supported in part by ONR MURI N00014-07-1-0907. Further, the authors would like to thank Andrew King (UPenn, Ph.D. student) and Oleg Sokolsky (UPenn, professor) for their support throughout this project. Also, they gratefully acknowledge the assistance of Robert Terrell (UPenn, Office of the General Counsel).

References

[1] Akismet. `http://akismet.com/`.

[2] Alexa: The web info. company. `http://www.alexa.org`.

[3] Alexa web info. service. `http://aws.amazon.com/awis/`.

[4] Defensio: Social web security. `http://www.defensio.com`.

[5] Google safe browsing API. `http://code.google.com/apis/`.

[6] Huggle. `http://en.wikipedia.org/wiki/WP:Huggle`.

[7] MediaWiki API. `http://en.wikipedia.org/w/api.php`.

[8] MediaWiki extensions. `http://www.mediawiki.org/wiki/Category:Extensions`. (extending core engine functionality).

[9] Wikipedia. `http://www.wikipedia.org`.

[10] Wikipedia article statistics. `http://dammit.lt/wikistats`.

[11] Wikipedia: External links. `http://en.wikipedia.org/wiki/Wikipedia:External_links`. (syntax and policy).

[12] Wikipedia: Long-term abuse: Universe Daily. `http://en.wikipedia.org/wiki/Wikipedia:UNID`.

[13] Wikipedia spam blacklist (English language version). `http://en.wikipedia.org/wiki/MediaWiki:Spam-blacklist`.

[14] WikiProject spam. `http://en.wikipedia.org/wiki/Wikipedia:WikiProject_Spam`.

[15] Wikistats. `http://stats.wikimedia.org/`.

[16] S. Abu-Nimeh and T. Chen. Proliferation and detection of blog spam. *IEEE Security and Privacy*, 8:42–47, 2010.

[17] B. Adler, L. de Alfaro, S. M. Mola-Velasco, P. Rosso, and A. G. West. Wikipedia vandalism detection: Combining natural language, metadata, and reputation features. In *CICLing'11: Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics, LNCS 6609*, pages 277–288, February 2011.

[18] C. Breneman and C. Carter. Cluebot NG. `http://en.wikipedia.org/wiki/User:ClueBot_NG`.

[19] H. Dai, Z. Nie, L. Wang, L. Zhao, J.-R. Wen, and Y. Li. Detecting online commercial intention (OCI). In *WWW'06: Proceedings on the 15th World Wide Web Conference*, 2006.

[20] J. R. Douceur. The Sybil attack. In *First IPTPS*, 2002.

[21] Y. Freund and L. Mason. The alternating decision tree algorithm. In *Proceedings of the 16th International Conference on Machine Learning*, pages 124–133, 1999.

[22] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In *IMC'10: Proc. of the 10th Internet Measure. Conf.*, 2010.

[23] E. Goldman. Wikipedia's labor squeeze and its consequences. *Journal of Telecomm. and High Tech. Law*, 8, 2009.

[24] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witen. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[25] S. Han, Y. yeol Ahn, S. Moon, and H. Jeong. Collaborative blog spam filtering using adaptive percolation search. In *WWW'06 Workshop on the Weblogging Ecosystem*, 2006.

[26] S. Hao, N. A. Syed, N. Feamster, A. G. Gray, and S. Krasser. Detecting spammers with SNARE: Spatio-temporal network-level automated reputation engine. In *18th USENIX Security Symposium*, August 2009.

[27] P. Heymann, G. Koutrika, and H. Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. *IEEE Internet Computing*, 11(6):36–45, 2007.

[28] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence Journal*, 97(1):273–324, 1997.

[29] G. Mishne, D. Carmel, and R. Lempel. Blocking blog spam with language model disagreement. In *AIRWeb'05: Proc. of the Wkshp. on Adversarial Info. Retrieval on the Web*, 2005.

[30] Y. Niu, Y. min Wang, H. Chen, M. Ma, and F. Hsu. A quantitative study of forum spamming using context-based analysis. In *NDSS'07: Proceedings of Network and Distributed System Security Symposium*, 2007.

[31] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *WWW'06: Proceedings of the 15th World Wide Web Conference*, 2006.

[32] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 422-1999, Stanford University, 1999.

[33] M. Potthast. Crowdsourcing a Wikipedia vandalism corpus. In *SIGIR '10: 33rd Intl. ACM SIG Information Retrieval Conference*, pages 789–790, 2010.

[34] M. Potthast, B. Stein, and T. Holfeld. Overview of the 1st intl. competition on Wikipedia vandalism detection. In *Notebook Papers of CLEF 2010 LABs and Workshops*, 2010.

[35] R. Priedhorsky, J. Chen, S. K. Lam, K. Panciera, L. Terveen, and J. Riedl. Creating, destroying, and restoring value in Wikipedia. In *GROUP '07: Proc. of the Intl. Conf. on Supporting Group Work*, pages 259–268, 2007.

[36] N. Provos, P. Mavrommatis, M. A. Rajab, and F. Monrose. All your iFrames point to us. In *SECURITY'08: The 17th USENIX Security Symposium*, pages 1–15, 2008.

[37] Y. Shin, M. Gupta, and S. Myers. The nuts and bolts of a forum spam automator. In *LEET'11: Proc. of the 4th Wkshp. on Large-Scale Exploits and Emergent Threats*, 2011.

[38] B. Stone. Policing the Web's lurid precincts. *The New York Times*, page B1, July 18, 2010.

[39] Symantec MessageLabs. 2010 Security Report. `http://www.messagelabs.com/resources/mlireports.aspx`.

[40] A. G. West. STiki: A vandalism detection tool for Wikipedia. `http://en.wikipedia.org/wiki/Wikipedia:STiki`.

[41] A. G. West, J. Chang, K. Venkatasubramanian, O. Sokolsky, and I. Lee. Link spamming Wikipedia for profit. In *CEAS '11: Proc. of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse, and Spam Conference*, September 2011.

[42] A. G. West, S. Kannan, and I. Lee. Detecting Wikipedia vandalism via spatio-temporal analysis of revision metadata. In *EUROSEC '10: Proceedings of the Third European Workshop on System Security*, pages 22–28, April 2010.

[43] J. Winter. Wikipedia distributing child porn, co-founder tells FBI. *FoxNews.com*, April 27, 2010.