ANALYSIS AND DETECTION OF LOW QUALITY INFORMATION IN SOCIAL NETWORKS

A Thesis
Presented to
The Academic Faculty

by

De Wang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computer Science
at the College of Computing

Georgia Institute of Technology
August 2014

ANALYSIS AND DETECTION OF LOW QUALITY INFORMATION IN SOCIAL NETWORKS

Approved by:

Professor Dr. Calton Pu, Advisor
School of Computer Science
at the College of Computing
Georgia Institute of Technology

Professor Dr. Edward R. Omiecinski
School of Computer Science
at the College of Computing
Georgia Institute of Technology

Professor Dr. Ling Liu
School of Computer Science
at the College of Computing
Georgia Institute of Technology

Professor Dr. Kang Li
Department of Computer Science
University of Georgia

Professor Dr. Shamkant B. Navathe
School of Computer Science
at the College of Computing
Georgia Institute of Technology

Date Approved: April, 21 2014

To my family and all those who have supported me.

ACKNOWLEDGEMENTS

When I was a little boy, my parents always told me to study hard and learn more in school. But I never thought that I would study abroad and earn a Ph.D. degree at United States at that time. The journey to obtain a Ph.D. is quite long and full of peaks and valleys. Fortunately, I have received great helps and guidances from many people through the journey, which is also the source of motivation for me to move forward.

First and foremost I should give thanks to my advisor, Prof. Calton Pu. He is not only a great research mentor guiding and inspiring me to diving deeply in the area of cyber security, but also an awesome friend who is willing to share his rich experience in life and study. He encouraged and trained me to be a great researcher by pushing me to exceed my own boundaries and excel in all aspects of what I did during the Ph.D. I really appreciate his patience on my research progress, as well as his amazing comments along my exploration within the area of cyber security. His underlying gentleness and dedication to research will continue to have a significant impact on my future.

Thanks to members of my thesis committee: Prof. Ling Liu, Prof. Shamkant B. Navathe, Prof. Edward R. Omiecinski, and Prof. Kang Li for reading and commenting on my dissertation. I really appreciate all the questions and challenges that led to an improved dissertation.

To Dr. Danesh Irani, your drive and dedication inspire me, and I will not forget your guidance and patience with a junior Ph.D. student entering the area of cyber security. I thank you for the early direction in research and constant guidances. To Dr. Qinyi Wu, you are a great friend, and I will not forget your helps on my life and

study.

The friendship, companionship, and support of my colleagues at Georgia Tech would be hard to replace. Special thanks to Rocky Dunlap, Qingyang Wang, Aibek Musaev, Yi Yang, Binh Han, Yuzhe Tang, Kisung Lee, Jack Li, Junhee Park, and Kunal Malhotra for being around in the lab. Rocky is a great lab manager. Qingyang is such a good friend to hang around and discuss about research. Aibek and Yi are valuable colleagues to work with. Binh, Yuzhe, and Kisung are my study buddies. Jack, Junhee and Kunal are good labmates.

More importantly, I would like to acknowledge the unconditional love and support from my family along the way. To my father, Chuanli Wang, I want to be as good a father as you. To my mom, Xiaoling Wu, thanks for giving me birth. To my wife, Jing Chen, I cannot go this far without you. To my little son, Eric J. Wang, I hope you will obtain a Ph.D. as well. I love you all dearly and this is for you!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Low quality information such as spam and rumors is a nuisance to people and hinders them from consuming information that is pertinent to them or that they are looking for. As social networks like Facebook, Twitter and Google+ have become important communication platforms in people's daily lives, malicious users make them as major targets to pollute with low quality information, which we also call as Denial of Information (DoI) attacks. How to analyze and detect low quality information in social networks for preventing DoI attacks is the major research problem I will address in this dissertation.

Although individual social networks are capable of filtering a significant amount of low quality information they receive, they usually require large amounts of resources (e.g, personnel) and incur a delay before detecting new types of low quality information. Also the evolution of various low quality information posts lots of challenges to defensive techniques. My work contains three major parts: 1). analytics and detection framework of low quality information, 2). evolutionary study of low quality information, and 3). detection approaches of low quality information. In part I, I proposed social spam analytics and detection framework SPADE across multiple social networks showing the efficiency and flexibility of cross-domain classification and associative classification. In part II, I performed a large-scale evolutionary study on web page spam and email spam over a long period of time. In part III, I designed three detection approaches used in detecting low quality information in social networks: click traffic analysis of short URL spam, behavior analysis of URL spam and information diffusion analysis of rumors in social networks. Our study shows promising results in analyzing and detecting low quality information in social networks.

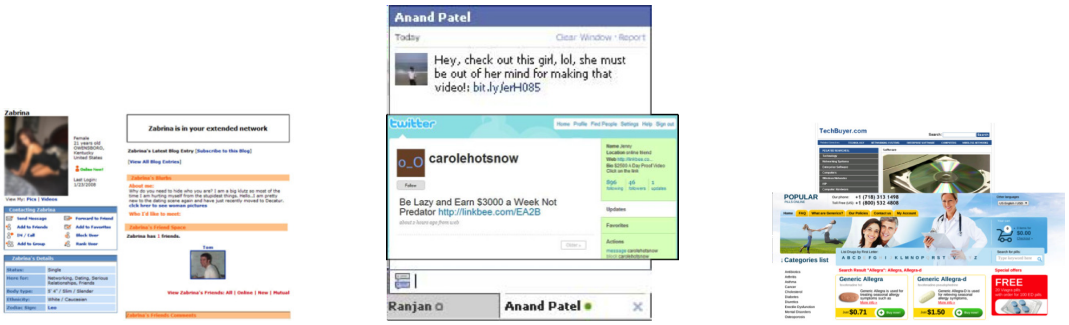
CHAPTER I

INTRODUCTION

With networks like Facebook, Twitter and Google+ attracting audiences of millions of users a month, they have been important communication platforms in everyday life of people. This in turn attracts malicious users to the social networks as well, causing an increase in the incidence of low quality information.

Such information, is a nuisance to people and hinders them from consuming information that is pertinent to them or that they are looking for. It has diverse formats in social networks such as user profile spam, message spam and web page spam. User profile spam employs text such as attractive words and multi-media such as pictures or videos to capture other users' attention or clicks in the purpose of redirecting users to phishing or spam web pages or increasing account reputations by more clicks or followings. Message spam appears in the instant message communications among users. Web page spam mainly contains false information such as phishing web page or low quality information such as irrelevant advertisings. Figure 1 shows the examples of low quality information in social networks. Individual social networks are capable of filtering a significant amount of low quality information they receive, although they usually require large amounts of resources (e.g, personnel) and incur a delay before detecting new types of low quality information.

Due to the ease of sharing information, social networks provide more efficient and numerous channels for the growth of low quality information. For example, web spam links in friend requests, in-box messages, and news feeds, are redirecting users to advertisement web sites or other types of malicious web sites. Further, social media sites have redefined the way links are shared with a tendency to share links using URL



(a) Example of user profile spam

(b) Example of message spam

(c) Example of web page spam

Figure 1: Examples of low quality information in social networks

shortener [15].

Apart from evolution of web and applications on the web being one of the reasons driving changes in low quality information, there is a constant evolution of low quality information as a reaction to defensive techniques introduced by researchers [46, 193]. Researchers have spent lots of efforts in studying the evolution of low quality information as well. Fetterly et al. [66] presented their work on a large-scale study of the evolution of web pages through measuring the rate and degree of web page changes over a significant period of time. They focused on statistical analysis on the degree of change of different classes of pages. Youngjoo Chung [46] studied the evolution and emergence of web spam in three-yearly large-scale of Japanese Web archives which contains 83 million links. His work focus on the evolution of web spam based on sizes, topics and host-names of link farms, including hijacked sites which are continuously attacked by spammers and spam link generators which will generate link to spam pages in the future. Irani et al. [98] studied the evolution of phishing email messages and they classified phishing messages into flash attacks and non-flash attacks and analyzed transitory features and pervasive features.

Owing to the evolution of low quality information and advancing techniques used in spreading low quality information, we need more novel detection approaches to address the low quality information filtering problem. More researchers have done

a lot of work on low quality information detection in social networks. Benevenuto et al. [23] presented an approach to detect spammers on Twitter. Lee et al. [121] had a long-term study on content polluters on Twitter. Zhang [207] detected spam through analyzing automated activity on Twitter. Thomas et al. [173] presented a real-time system Monarch that detects URLs. Ghosh et al. [71] investigated link farming in the Twitter network and then explored mechanisms to discourage the activity. Benevenuto et al. [164] approached the problem of detecting trending-topic spammers who include unrelated URLs with trending words in tweets. Since short URL is used often on social media recently, spammers also adopted it for camouflaging the spam URLs [15]. Klien et al. [108] proposed geographical analysis of spam in short URL spam detection. Chhabra et al. [43] presented the method to detect phishing through short URLs. Maggi et al. [130] presented security threats and countermeasures of short URLs on Twitter.

In this thesis, we mainly focus on the analysis and detection of low quality information in social networks. Specifically, we introduce our social spam detection framework SPADE across multiple social networks showing the efficiency and flexibility of cross-domain classification and associative classification. For evolutionary study of low quality information, we present the results on large-scale study on web page spam and email spam over a long period of time. The reasons why we choose web page spam and email spam are as follows: 1) web page spam is the most popular and common spam on Internet; 2) email system is the fundamental communication system supporting social networks and the trend of email spam shows the direction of low quality information to some extent. Furthermore, we provide one novel detection approach used in filtering out URL spam in social networks: click traffic analysis and behavior analysis. It involves monitoring and analysis of collective activities of legitimate users and spammers in social networks using machine learning and intrusion detection techniques. For proposed work, we will spend efforts on behavior analysis

of URL spam and information diffusion analysis of rumor in social networks.

1.1 Dissertation Statement and Dissertation Contributions

Before proceeding to the concrete contributions of my thesis, my thesis statement can be formulated as follows:

Thesis Statement: *Analysis and detection of evolving social spam and rumors in social networks requires evolution-resistant approaches that can be effectively accomplished through collaborative social spam detection framework, and activity-based analysis on click traffic, user behavior, and information diffusion.*

To support my thesis statement, we make the following three contributions:

- Various types of low quality information results in high cost of analysis and detection when we deal with them in an isolated manner. To reduce the cost and improve the performance, we propose a collaborative social spam analytics and detection framework for multiple social networks – SPADE (Section 2). It converts diverse types of low quality information into uniform models and also provides platform for cross-domain classification and associative classification to enhance detection performance. Through intensive experimental study on real dataset, we demonstrate the feasibility of SPADE and measured the effectiveness of cross-domain classification and associative classification.

- The second contribution is the evolutionary study of low quality information on large scale datasets over a long period of time, which shows interesting trends. The evolution trends of web spam are: 1) spammers manipulate social media in spreading spam; 2) HTTP headers and content of web spam also change over time; 3) spammers have evolved and adopted new techniques to avoid the detection based on HTTP header information. The evolution trends of

email spam are: 1) more email spam have main content type in text instead of multi-part and more URLs are embedded in email spam instead of HTML and image; 2) topic of email spam drifted a lot from "Business News" to "Account Information" and more email spam contain social engineering attacks; 3) sender-receiver routing network of email spam becomes more complex over time showing spammers are using complex routing to avoid detection.

- The third contribution is that we propose activity-based detection approaches to defend against malicious users. Malicious users camouflage low quality information by obfuscation techniques such as good word stuffing and short URLs. Besides, malicious users create or compromise a collection of user accounts in social networks to propagate low quality information. Through collective activities, they maximize the usage of those user accounts without being detected. The first approach is click traffic analysis of short URL spam filtering on Twitter. In this work, we generate a large-scale click traffic dataset for short URL spam. Also, we find that the majority of the clicks are from direct sources (email, Instant message, Apps) and that the spammers utilize popular websites to attract more attention by cross-posting the links. Through experimental study, we demonstrate the feasibility of detecting short URL spam by classification based on the click traffic features.

1.2 Organization of the Dissertation

We split each contribution into a separate part, which contains chapters that illustrate the details of research work, to emphasize the steps for analysis and detection of low quality information in social networks. We attempt to keep each chapter an independent unit with its own evaluation and related work.

Part I: Analytics and Detection Framework of Low Quality Information

- **Chapter II: SPADE: A Social-spam Analytics and Detection Framework**

Part II: Evolutionary Study of Low Quality Information

- **Chapter III: Evolutionary Study of Web Spam** - With over 2.5 hours a day spent browsing websites online [168] and with over a billion pages [101], identifying and detecting web spam is an important problem. Although large corpora of legitimate web pages are available to researchers, the same cannot be said about web spam or spam web pages.

  We introduce the Webb Spam Corpus 2011 — a corpus of approximately 330,000 spam web pages — which we make available to researchers in the fight against spam. By having a standard corpus available, researchers can collaborate better on developing and reporting results of spam filtering techniques. The corpus contains web pages crawled from links found in over 6.3 million spam emails. We analyze multiple aspects of this corpus including redirection, HTTP headers and web page content.

  We also provide insights into changes in web spam since the last Webb Spam Corpus was released in 2006. These insights include: 1) spammers manipulate social media in spreading spam; 2) HTTP headers also change over time (e.g. hosting IP addresses of web spam appear in more IP ranges); 3) Web spam content has evolved but the majority of content is still scam.

- **Chapter IV: A Study on Evolution of Email Spam Over Fifteen Years** - Email spam is a persistent problem, especially today, with the increasing dedication and sophistication of spammers. Even popular social media sites such

as Facebook, Twitter, and Google Plus are not exempt from email spam as they all interface with email systems. With an "arms-race" between spammers and spam filter developers, spam has been continually changing over the years. In this chapter, we analyze email spam trends on a dataset collected by the Spam Archive, which contains 5.1 million spam emails spread over 15 years (1998-2013). We use statistical analysis techniques on different headers in email messages (e.g. content type and length) and embedded items in message body (e.g. URL links and HTML attachments). Also, we investigate topic drift by applying topic modeling on the content of email spam. Moreover, we extract sender-to-receiver IP routing networks from email spam and perform network analysis on it. Our results show the dynamic nature of email spam over one and a half decades and demonstrate that the email spam business is not dying but changing to be more capricious.

Part III: Detection Approaches of Low Quality Information

- **Chapter V: Click Traffic Analysis of Short URL Spam on Twitter** - With an average of 80% length reduction, the URL shorteners have become the norm for sharing URLs on Twitter, mainly due to the 140-character limit per message. Unfortunately, spammers have also adopted the URL shorteners to camouflage and improve the user click-through of their spam URLs. In this chapter, we measure the misuse of the short URLs and analyze the characteristics of the spam and non-spam short URLs. We utilize these measurements to enable the detection of spam short URLs. To achieve this, we collected short URLs from Twitter and retrieved their click traffic data from Bitly, a popular URL shortening system. We first investigate the creators of over 600,000 Bitly short URLs to characterize short URL spammers. We then analyze the click

traffic generated from various countries and referrers, and determine the top click sources for spam and non-spam short URLs. Our results show that the majority of the clicks are from direct sources and that the spammers utilize popular websites to attract more attention by cross-posting the links. We then use the click traffic data to classify the short URLs into spam vs. non-spam and compare the performance of the selected classifiers on the dataset. We determine that the Random Tree algorithm achieves the best performance with an accuracy of 90.81% and an F-measure value of 0.913.

- **Chapter VI: BEAN: a BEhavior ANalysis approach of URL spam filtering in Twitter** - Social websites, like Twitter and Facebook, strive to detect and remove URL spam in order to keep their users happy and coming back. Although researchers have already proposed many filtering approaches such as SpamRank and TrustRank, most of which detect URL spam using content analysis on the Web pages behind or link analysis on Web graph, it is challenging to automatically detect URL spam in social media as spammers keep evolving and advancing their techniques, such as cloaking based on the IP addresses, using multiple user accounts and redirectors. In this chapter, we introduce BEAN, a behavior analysis technique, which detects URL spam by capturing the anomalous message sending behaviors of spammers. Twitter is an ideal place for our analysis due to its popularity and real-time properties. We collect over 2.4 million tweets from around a million users based on Twitter trending topics for 4 months. We apply our behavior analysis approach derived from a Markov Chain model to the Twitter dataset, and achieve a precision of 0.91 and recall of 0.88. In doing so we detected a lot of URL spam that cannot be filtered out by conventional approaches such as SVM and TrustRank, indicating that our approach is a good complement to existing URL spam detection

techniques. Also, we further investigate anomalous behavior patterns of spammers in spreading URL spam to confirm our assumption.

- **Chapter VII: Information Diffusion Analysis of Rumor Dynamics over a Social-interaction based Model** - Rumors may potentially cause undesirable effect such as the widespread panic in the general public. Especially, with the unprecedented growth of different types of social and enterprise networks, rumors could reach a larger audience than before. Many researchers have proposed different approaches to analyze and detect rumors in social networks. However, most of them either study on theoretical models without real data experiments or use content-based analysis and limited information diffusion analysis without fully considering social interactions. In this chapter, we propose a social interaction based model FAST by taking four major properties of social interactions into account including familiarity, activeness, similarity, and trustworthiness. Also, we evaluate our model on real data from Sina Weibo (Twitter-like social network in China), which contains around 200 million tweets and 14 million Weibo users. Based on our model, we create a new metrics Fractional Directed Power Community Index(FD-PCI) derived from μ-PCI to identify the influential spreaders in social networks. FD-PCI shows better performance than conventional metrics such as K-core index and PageRank. Moreover, we find influential features to detect rumors by comparison between rumor and real news dynamics.

- **Chapter VIII: Conclusion** - In this chapter, we wrap up the dissertation.

CHAPTER II

SPADE: A SOCIAL-SPAM ANALYTICS AND DETECTION FRAMEWORK

In this chapter, we will introduce a social spam analytics and detection framework (SPADE) which can be used by any social network to detect spam. Generally, social networking sites have their own spam detection mechanism and hire anti-spam personnel to develop and maintain the system. SPADE will help each social network to avoid the cost and cooperate together in the battle of spam detection. Meanwhile, SPADE will be able to identify a new type of spam automatically on all participating networks once the spam is detected on one network. Additionally, by using the framework, new social networks can quickly protect their users from social spam.

Figure 2 shows an overview of the system. SPADE can be split into three main components and we provide a brief explanation for each part here:

- Mapping and Assembly: we use mapping techniques to convert a social network specific object into a framework-defined standard model for the object (e.g., profile model, message model, or web page model). If associated objects can be fetched based on this object, it is assembled here;

- Pre-filtering: we use fast-path techniques (e.g., blacklists, hashing, and similarity matching) to check incoming objects against known spam objects;

- Classification: we use supervised machine learning techniques to classify the incoming object and associated objects. In the final labeling decision (spam or non-spam), we could use different combination strategies such as AND, OR, majority voting or Bayesian model to aggregate the classification results according

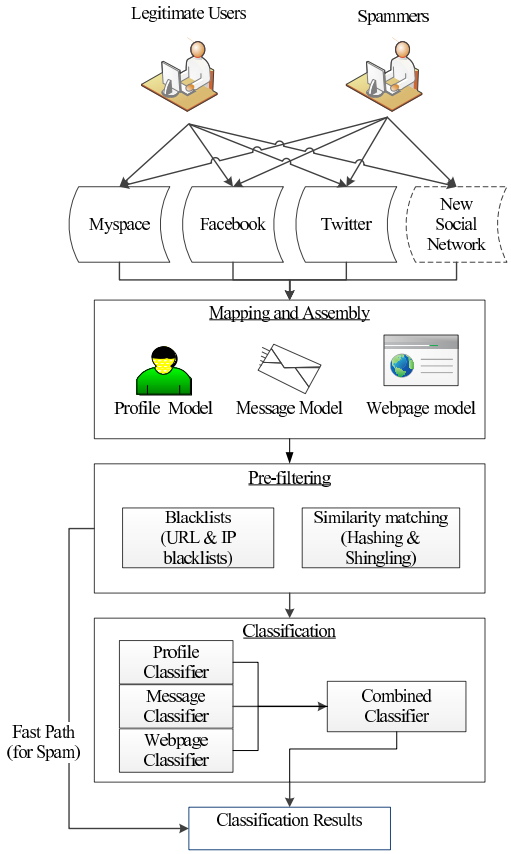


Figure 2: Overview of the spam detection framework

to certain system requirements.

More concretely, we make the following contributions in details:

- Build a social-spam detection framework to filter spam on multiple social networks. We illustrate the three main components of the system and experimentally demonstrate the use of the system on data from Twitter, MySpace, and the Webb Spam Corpus.

- Demonstrate cross social-corpora classification and measure the effectiveness. Namely, we can build a classifier for a particular model on one social network and apply it to another social network. After that, we approximate the accuracy of this technique by using existing datasets.

- Demonstrate associative classification in which the results depend not only on

the object being classified, but also on objects associated with it. For instance, classification of a message object takes into account classification results of the associated web page objects that may be linked inside the message. The feasibility of this technique is measured as well.

The remainder of the chapter is organized as follows. We introduce related work in Section 2.1 and motivate the problem further in Section 2.2. Section 2.3 provides the structure and implementation of the framework. Section 2.4 describes the experimental setup used to evaluate our framework and Section 2.5 presents our experimental results. We conclude the chapter in Section 2.6.

2.1 Related Work

Most previous work on social spam has focused on spam prevention on a single social network (e.g., Facebook [76, 163], MySpace [99], Twitter [23]). A number of techniques are employed in these papers including classification, user-report-based filtering, behavioral analysis, and in some cases friend-graph analysis. But, more researchers proposed social spam detection framework for social network sites using various techniques [32,63,87,104]. Our work differs from them on efficient information integration solution and cross-domain classification using the relationship between objects on social media. Although aspects of previous research have been incorporated into our framework to improve results. For example the study by Webb et al. [193] on automatically detect web spam using email spam on detecting Twitter spam using web pages inspired us to classify incoming social spam by taking into account classification of associated content. And we also incorporated Web spam classification methods used by Webb et al. [194] and social profile spam detection (as demonstrated on MySpace) methods used by Irani et al. [99] in our framework [184, 185, 188]. A large number of classifiers have been used in spam detection but choosing the right classifier and the most efficient combination of them is still a problem. Previous work

by Byungki et al. [35] proposes a Bayesian framework, which is theoretical efficient and practically reasonable method of combination, when investigating the integration of text and image classifiers.

Web spam detection has been over-studied by lots of researchers(see a survey in [161, 186, 189]). All existing algorithms can be categorized into three categories based on the type of information they use: content-based methods [64–66, 144], link-based methods [20], and methods based on non-traditional data such as user behavior [127], clicks [153], HTTP sessions [194, 196]. Also, some URL spam filtering technique for social media has been proposed by Kurt Thomas et al. [173] to better address different web services such as social networks. They presented a real-time URL spam filtering system named Monarch and demonstrated a modest deployment of this system on cloud infrastructure and its scalability.

Text classification (also known as text categorization, or topic spotting) is to automatically sorting a set of documents into classes (or categories, or topics) from predefined set [160]. It has attracted a booming interest from researchers in information retrieval and machine learning areas in decades. Recently, several novel classification approaches are proposed and implemented in cross-domain text classification. Pu Wang et al. [191] presented semantics-based algorithm for cross-domain text classification using Wikipedia based on co-clustering classification algorithm. Elisabeth Lex et al. [123] described a novel and efficient centroid-based algorithm Class-Feature-Centroid Classifier(CFC) for cross-domain classification of web-logs, also they have discussed the trade-off between complexity and accuracy. Pan et al. [145] proposed a spectral feature alignment (SFA) algorithm to align domain-specific words from different domains into unified clusters, with the help of domain independent words as a bridge. Zhen et al. [208] propose a two-stage algorithm which is based on semi-supervised classification to address the different distribution problem in cross-domain classification. Also, another perspective is relational classification [100, 138, 212] used

in text classification, which could boost the overall performance greatly.

2.2 Motivation

As social networks rise as an important communication platform [198], spammers have increasingly targeted social networks with spam. Social spam is spam directed at users of Internet social networking services such as MySpace, FaceBook or Twitter. One major form of social spam is profile spam which contains sexual words or attractive images in their profiles. Meanwhile, updates among friends in social networks, such as comments,feeds,messages, and friend requests,all may be used to spread spam. Also, users of social networking services can send notes, that may include embedded links to other social network locations or even outside sites, to one another [99].

Facebook, Twitter, MySpace, and other major social networks have their own anti-spam team to fight spam on their network [118]. Most of them are employing techniques such as user-report-based filtering (where users report objects that are spammy) and behavioral analysis (where logs of interactions are used to detect spamming patterns). Such dynamic methods may be eventually able to detect social spam with requiring a non-trivial amount of lag time to accumulate sufficient evidence.

In addition, social networks will also employ classification based techniques which use labeled training data to find similar occurrences of spam on the social network. Due to the evolving nature of spam and changing on-line environment [98, 149, 156], these classification based techniques need to be retrained and adapted to newer spam [150].

Although techniques to propagate spam may vary from one social network to another, anecdotal evidence suggests that spam generally fall into the category of pharmaceutical, pornographic, phishing, stocks, and business promotion campaigns due to specificities of each social network. Bot-nets have already been shown to use templates to send varying spam campaigns messages (similar to the campaigns

previously mentioned) [110] to different targets. It is only a matter of time before bot-nets and spammers also employ such techniques with approaches to distribute a legitimate message across multiple social networks already implemented in [94, 174],

To address heterogeneity issue of spam, we propose a social spam detection framework – SPADE that uses general models for profile, message, and web page objects to perform classification across social networks. Also we use misclassification or feedback via other methods to update the classification models which apply across social networks. This will allow new types of spam detected on one social network to be detected across social networks, and also reduce the burden of the social network spam teams for keeping the classifiers up to date. Furthermore, new social networks which do not have any spam detection solutions can use the SPADE to protect their users from spammers immediately .

2.3 Social-Spam Analytics and Detection Framework

In this section, we present the SPADE and the three main parts in the following subsections. An overview of the framework is shown in Figure 2.

2.3.1 Mapping and Assembly

For the objects within the social network, we have to create a standard model to build a framework that is social network agnostic. A model of an object as a schema includes the most common attributes of the object across social networks. Once a model is defined, we need to map incoming objects from the social network into objects of the model. We discuss both these steps in more detail below.

2.3.1.1 Models

In the SPADE, we define three basic models: profile model, message model, and web page model, which represent the most important objects in social networks. Other models are omitted as they are not required to demonstrate the feasibility of the

framework.

The profile model has 74 attributes, which are derived from the Google Open Social Person API [74]. Those attributes cover attributes most commonly used in user profiles across websites like Facebook, MySpace, Twitter, and Flickr.

The message model has 15 attributes based on common attributes used in messages – such as "To", "From", "Time-stamp", "Subject", and "Content". We also add a few attributes which would be common across social-network messages and e-mail messages, such as, "Sender-IP", and other header attributes.

The web page model has attributes based on common HTTP session header information (see work done by Steve et al. [196]) and content. For instance, "Connection", "Server" and "Status" et al. are common features in HTTP session header. For the content of web pages, we extract visible text (after striping out HTML tags) that will be used on text classification.

A model is akin to a class, and an object is an instance of the model. For the purpose of extensibility and scalability, all models are stored in XML.

Data Types: An attribute can be one of four types: Numerical, String, Categorical (Nominal), and Date namely. These types are standard attribute types found in many machine learning implementations. A snippet of the person model definition is shown as follows.

```
<model type="person">
<attribute>
<attributeName>AboutMe</attributeName>
<attributeType>String</attributeType>
</attribute>
<attribute>
<attributeName>Age</attributeName>
<attributeType>Numerical</attributeType>
```

```
 </attribute>
 ...
</model>
```

2.3.1.2 Mapping

After the definition of models, we use mapping process to transform incoming social network objects into the respective object model in the framework. In this mapping process, it automatically provides to the SPADE a list of incoming attributes and their attributes in the respective model. Due to easy updatability and simplicity, the mapping relationships are specified in an XML file.

It is easy to handle the name mappings and also straight-forward for type mappings except in illegal cases that we disallow such as "Date" to "Categorical". An example of name mappings is shown in Table 1 (shown in table format for simplicity). For some data types such as categorical type, we need to specify the mapping for each value in the domain of the data type. While for semantic mapping, we handle it using manual code written within a special XML tag to perform the necessary conversion.

Table 1: The name mapping between Twitter profiles and our profile model. Empty mappings are omitted

Twitter Profile	**Profile Model**
Id	Id
Name	NickName
Location	CurrentLocation
Description	AboutMe
Profile_Image	ProfileImage
Url	ProfileUrl

2.3.1.3 Assembly

As we obtained uniform models from the mapping process, assembly is the next process which probes each model object for associated objects and then subsequently

fetches those model objects. For instance, when we are dealing with a message object, if the content contains URLs, we fetch the web pages associated with those URLs and create web page objects which are then assembled together with the message object. As it can provide a rich source of information for the further stages, this additional information is often critical for spam detection .

2.3.2 Pre-filtering

To reduce classification cost, we adopt fast-path techniques to quickly filter out previous classified or similar spam in incoming social network objects. Some of these techniques are listed as follows:

- Blacklists: lists of entries, such as URL, DNS, and IP address, which are to be immediately rejected due to prior spamming or bad behavior.
- Similarity matching: Hashing and shingling can be used to quickly calculate similarity against previous spammy entries. The number of previous spammy entries can be limited to avoid high lookup costs.

These techniques may have shortcomings for their lag-time in detecting new spam, but they significantly reduce classification cost.

2.3.3 Features

Based on the attributes of our models, we chose the common attributes as features in classification. But the datasets from different domains may share different common attributes. To preserve all the common information, we keep all attributes in the models as features. If the dataset from certain social media doesn't have a subset of features, we will only use the common features to do the classification and predication. For example, Twitter profile doesn't have the attribute "age" and "sex" but it has attribute "bio" then we will use the feature "AboutMe" which is shared by Twitter and MySpace datasets in classification.

Meanwhile, we adopt bag of words model to generate word features for textual attributes of each model. Bag of words model takes each word as features and use the appearance of the words in document as value of corresponding features. For instance, we apply bag of words model to the content of web page which may generate hundreds of word features. Considering performance and accuracy of classification, we reduce the size of feature set by adding constrains such as frequency of word features and remove the stop words such "the", "a", and "an".

2.3.4 Classification

For each model, we build one classifier using over 40 different types of supervised machine learning algorithms implemented in the Weka [1] software package [84], including standard algorithms such as Naïve Bayes [157], Support Vector Machine (SVM) [60] and LogitBoost [36, 68].

We retrieve associated objects for incoming objects to be classified in the Assembly stage (see Section 2.3.1.3). Thus, it will involve the classification of different objects. This process is illustrated in Figure 3. For instance, a profile is passed to the profile classifier for classification followed by associated messages being passed to message classifier to do the classification. If the message object contains a URL (such as a Tweet), then the associated object will be a web page object which will be passed to the web page classifier. We apply combination strategies to the results of the classifiers after obtaining all results.

After the classifier for each model involved returns a decision, it is passed on to the combiner. We have four different combination strategies available for us to adopt in our framework: AND strategy, OR strategy, Majority voting strategy, and Bayesian model strategy.

[1] Weka is an open source collection of machine learning algorithms that has become a standard tool in the machine learning community.

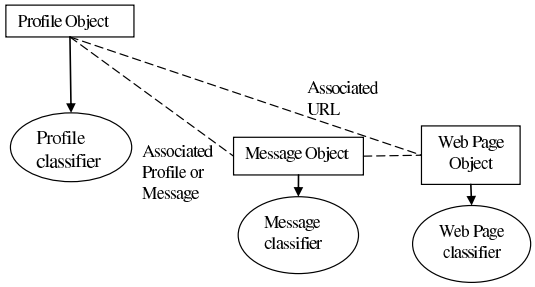


Figure 3: Using associated objects to assist in classification.

- AND strategy classifies an object as spam if all classifier, for each model, classifies it as spam.
- OR strategy classifies an object as spam if any classifier, for each model, classifies it as spam.
- Majority voting strategy classifies the object as spam only when majority of classifier, for each model, classifies it as spam.
- Bayesian model strategy is a slightly modified version of a strategy from previous research on creating an anti-spam filter combination framework for text-and-image emails [35].

Bayesian model strategy may need more details to explain. We use a subscript i to distinguish different models and t to denote incremental learning cycles at time t. Suppose we receive a object x and ω is the class associated with x, either being spam or legitimate. Then, assuming a hidden variable Z for an event to select one model, a probability for a class ω given $x, P(\omega|x)$, can be expressed as a marginal probability of a joint probability of Z and ω.

$$P(\omega|x) = \sum_i P(\omega, Z_i|x) = \sum_i P(\omega|Z_i, x) P(Z_i|x). \quad (1)$$

Here, we use external knowledge $P(Z_i|x)$ to express each classifier's confidence

given x. For instance, if a certain classifier model becomes unavailable, the corresponding $P(Z_i|x)$ will be zero. Also one could assign a large probability for the corresponding $P(Z_i|x)$ if one classifier dominates over other classifiers. Most data types are supported directly by the classifiers we use, except for the String data type. Using bag of words model, we convert the string into a list of boolean attributes (where each boolean attribute represents the presence or absence of a word) through stemming and removing stop words. Before classification, we represent each object (or model instance) as a attribute vector f of n attributes: $\langle f_1, f_2, ..., f_n \rangle$. All of attributes are boolean; hence, if $f_i = 1$, the attribute is present in a given object; otherwise, the attribute is absent in a given object.

2.4 Experimental Plan

In this section, we first discuss the datasets used to demonstrate our framework, followed by the implementation of our framework, and finally detail our evaluation plan.

2.4.1 Datasets

The datasets contain raw profiles or messages from the social networks. And later we parse them into their respective XML objects which are submitted to the SPADE. We use these datasets as sample input to perform experiments and measure the performance of the SPADE.

MySpace Profile Dataset: It is a previously collected sample of over 1.8 million MySpace profiles [40] from June to September 2006. Besides, approximately 1,500 spam MySpace profiles were also gathered from a previous study [195] on honey-pots in MySpace collected in late 2007. We summarize the strategies used to collect the MySpace profiles as follows:

- Legitimate Top 8 Crawl: the top 8 most popular friends were crawled in a breath

first search manner starting with a seed list of random (legitimate) profiles, which resulted in a collection of over 890,000 connected profiles.

- Random Crawl: profiles were crawled by generating random UserId's and retrieving the profile represented by that user, which resulted in a collection of over 960,000 profiles.

- Honey-pot Spam: Social honey-pot accounts were configured across the U.S. and were used to crawl profiles that initiated contact with them and were identified as spam accounts.

Twitter Profile, Message, and Web Page Datasets: Irani et al. [99] collected over 900,000 Twitter users, over 2.4 million Tweets and fetched any links in the Tweets. Twitter users are represented in the profile model, Tweets can be represented in the message model, and web pages associated with the links in the web page model. These tweets were gathered by querying the top trending topics every minute and represent a over 600 topics over a span of November 2009 to February 2010. Twitter users and Tweets (messages) marked as suspended or removed due to Twitter terms of service violations were marked as spam and there were over 26,000 Twitter such users and 138,000 such Tweets.

TREC Email Datasets: the TREC 2007 [181] corpus contains over 50,000 spam emails and over 25,000 legitimate emails. Spam messages in this dataset was collected from various email honey-pots, with legitimate messages being donated by various users. We include the use of emails in our social-spam detection study to have additional data for the message model.

Webb Spam corpus and WebBase dataset: The Webb Spam corpus [193] is a collection of nearly 350,000 spam web pages, crawled from spam links found in email messages between November 2002 to January 2006. As there were no legitimate pages in the Webb Spam Corpus, we augmented it with a dataset from the WebBase

Web Page Repository [93]. We downloaded and used December 2005 crawl of the WebBase corpus and used a stratified random sample of over 392,000 legitimate web pages.

2.4.2 Experiment Implementation

2.4.2.1 Mapping and Assembly

In mapping process, we create a mapping file from attributes in the dataset to attributes in the model for each of the datasets earlier presented. The assembly process takes care of retrieving associated object models if required.

Specifically, mapping/assembling MySpace and Twitter (users) to the profile model demonstrates the mapping and assembly in relation to the profile model. Mapping/assembling TREC and Twitter (tweets) to the message model demonstrates the mapping and assembly in relation to the message model, whereas mapping/assembling Twitter (web-pages linked in tweets) and Webb Spam Corpus/WebBase to the web page model demonstrates the mapping and assembly in relation to the web page model.

2.4.2.2 Pre-filtering

Due to the lack of historic blacklists to apply to our data and the likelihood that using new blacklists on older data would skew the results, we do not use blacklists in our experiments.

2.4.2.3 Features

We chose the most common features across different social media for each model as examples. The common attributes are listed in Table 2. We could expand the feature set to include social network related features such as number of friends/followers. The framework is capable to embrace more features. For simplicity, we illustrate the classification in our framework only based on textual attributes.

Table 2: The most common attributes for different models

Model	Common Attribute
Profile Model	AboutMe
Message Model	Body
Web page Model	Content

From Table 2, the most common attributes are all textual attributes. Thus, we need to convert them into word features using bag of words model. After that, we select discriminative word features for classification.

Features Selection: The purpose of features selection is to remove noisy features and select powerful features in terms of information gain. The measures we have adopted in the process can be separated into two steps:

- The first step is to set up the frequency of token threshold for all features. For example, if the times of one token appearing in all documents are fewer than 10 times, we will not consider it as a word feature in the classification.
- The second step is to filter out stop words using stop word list. For those popular and semantically no-meaning words, we will not take them into account when doing classification. For example, "a", "and", and "the".

χ^2 **test:** It is a statistical test which used to compare real data with data we would expect to obtain according to a specific hypothesis. In a classification problem, we generate predicted labels and compare them to real labeled data to obtain the deviation between them. The value of χ^2 test will tell us the significance of difference between real labels and predicted labels. The formula for calculating chi-square(χ^2) is:

$$\chi^2 = \sum (r - p)^2 / p \tag{2}$$

That is, chi-square is the sum of the squared difference between real(r) and the predicted(p) data(or the deviation, d), divided by the predicted data in all possible categories.

In our experiments, we use χ^2 test to rank out the usefulness of each feature. Top features ranked by χ^2 test could tell us the trends/strongest features of spam objects in different datasets.

2.4.2.4 Classification

Cross social-corpora classification and associative classification are the two core parts of our detection framework SPADE. To evaluate the effectiveness of cross social-corpora classification, we build a web page model classifier using the Webb Spam Corpus and WebBase dataset, followed by classifying the web page part of Twitter dataset. Next, for incoming objects, we retrieve their associated objects and use cross social-corpora classification for each object model. For the profile classifier, we use the MySpace profile dataset for training and evaluating the cross classifier on the Twitter profile dataset. For the message classifier, the TREC 2007 dataset is used for training and evaluating the cross classifier on the Twitter message dataset. We then combine the results from web page model and message model classifications to obtain the final predicted labels of message objects. Finally, the incoming profiles classification labels are obtained by the combination of the labels of message model and profile model classification.

We use classifiers implemented in the Weka software package [84]. Most data types used in our models can be directly used by classifiers in Weka except for the String data type. As previously mentioned, we use bag of words after using stemming and removing stop words. The StringToWordVector filter performs this transformation for us and includes applying the Snowball Stemming algorithm (variation of the Porter stemmer) and removing a built-in list of stop words.

Once the object is in an amenable form, steps to build a classifier model include re-sampling and classifier selection. When re-sampling, we use stratified random sampling to balance the spam and legitimate classes in order to avoid any biases that

may arise from imbalanced classes.

2.4.3 Evaluation

We use several criteria evaluate the classifiers' performance, namely F1-Measure and Accuracy. In our framework SPADE, F1-measure (also F1-score) is calculated based on precision and recall. Before introducing the details of precision and recall, we review the relationship between true positive, true negative, false positive, and false positive which is shown in Table 3.

Table 3: The relationship between true-positive, true-negative, false-positive and false-positive

Actual Label	Predicted Label	
	Positive	*Negative*
Positive	True-Positive (TP)	False-Negative (FN)
Negative	False-Positive (FP)	True-Negative (TN)

The definitions of precision(P), recall(R), F-measure(FM), and accuracy(A) for classification are based on above terms, and are given by the following formulas.

$$P = \frac{TP}{(TP + FP)}; \quad R = \frac{TP}{(TP + FN)} \tag{3}$$

$$FM = 2 \cdot \frac{P \cdot R}{P + R} \; ; \; A = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{4}$$

Precision is obtained by the number of true positives divided by the sum of true positives and false positives. Recall is obtained by the number of true positives divided by the sum of true positives and false negatives. F1-measure is the Harmonic mean of precision and recall (traditionally F-measure is represented in terms of precision and recall). Accuracy represents the number of instances correctly classified and is equals to the sum of true positives and true negatives divided by the total number of instances.

2.5 Experimental Results

In this section, we present the results of the two core parts of our detection framework, namely the cross social-copora classification and associative classification.

2.5.1 Baseline

Before the details of two main classifications, we first introduce baseline for comparison purpose. The baseline here is single domain classification. In the classification, the feature sets are from our uniform models. Meanwhile, we applied 10-fold cross-validation for all the classifications. For simplicity, we only use Naïve Bayes algorithm in classification. For Twitter dataset, we listed the classification result in Table 4.

Table 4: The results of single domain classification using Naive Bayes

Classification Category	TP	FP	F-Measure	Acc.
Twitter Web Page	61.5%	24.56%	0.6611	68.47%
Twitter Message	60.83%	23.01%	0.6618	68.91%
Twitter Profile	89.54%	57.90%	0.7237	65.16%

Table 4 shows the results of classification on dataset from Twitter domain. We also show the precision and recall for each model in Figure 4. Twitter profile classification shows high recall but low precision.Precision and recall of Twitter web page and message classification are very close but the overall performance is lower than we expected.

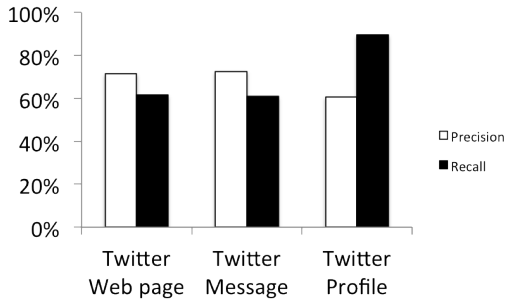


Figure 4: Precision and Recall in Twitter single domain classification

The results are not promising due to a number of factors. In the following sections,

we will use cross social-corpora classification and associative classification to perform classification and compare them with the baseline. Also, we will discuss the reasons for errors in classification.

2.5.2 Cross Social-Corpora Classification

During practical usage of our framework, we expect our classifier models to be built using a number of social networks. Incoming objects from the same social network or other social networks can then be classified using these models. We evaluate an extreme case of this classification, where we build a classifier using one social-network dataset and test the results using another dataset.

2.5.2.1 Classification Process

The purpose of cross-domain classification is to prove that spam information across different domains is from common spam sources. We have three basic models and datasets from several different social media so the process of classification is illustrated in detail in the following. Meanwhile, for easy illustration, we have listed the process of our classification experiments in Figure 5.

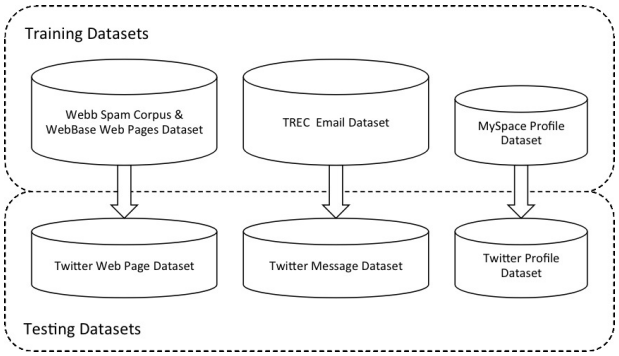


Figure 5: The process of cross social-corpora classification

For the web page model, we use the Webb Spam Corpus dataset combined with the WebBase web pages dataset as training dataset. The Twitter web page dataset is used as testing dataset.

For the message model, we use TREC 2007 as training dataset and take Twitter message dataset as testing dataset.

For the profile model, we use MySpace Profile dataset which contains both spam profiles and legitimate profiles as training dataset and take Twitter profile dataset as testing dataset.

In the process of classification, we compare different algorithms in Weka package to see which algorithm can achieve best accuracy and performance. We also test out whether the classifier model built using a dataset from one domain can test the dataset from another domain. The performance of classification indicates the effect of common spam sources. If the classification performs well, it implies that the datasets from two different domains share common spam sources. The results of the experiments are listed in the following sections.

2.5.2.2 Classification Experiments

We first show cross social-corpora classification based on web page model as the web page model can be used in conjunction (via associated objects) with both profile and message models in our framework. We will use the web page model classifier to improve the accuracy of the other models.

To build (train) the web page model classifier, we use the Webb Spam Corpus (spam web pages) and WebBase (legitimate web pages). We apply the classifier to labeled web pages associated with Tweets from the Twitter dataset. These datasets previously described in Section 2.4.1 consist of HTTP session headers and content web pages.

Previous work [194] used HTTP session headers to detect spam for web pages, but based on our datasets we found that HTTP session headers are not robust to temporal differences in the cross-corpora classification. This is likely due to HTTP session

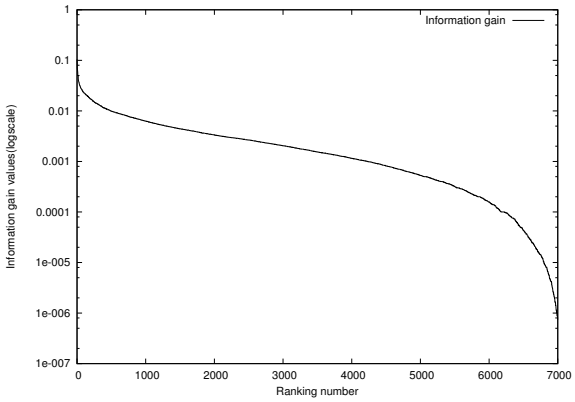


Figure 6: Information gain values of 7,000 features for Webb Spam Corpus and WebBase web pages

headers containing transitory features that become exiting due to the arms-race between spammers and spam-researchers [98, 149]. We therefore perform classification on content of web pages for cross-corpora and cross-temporal datasets.

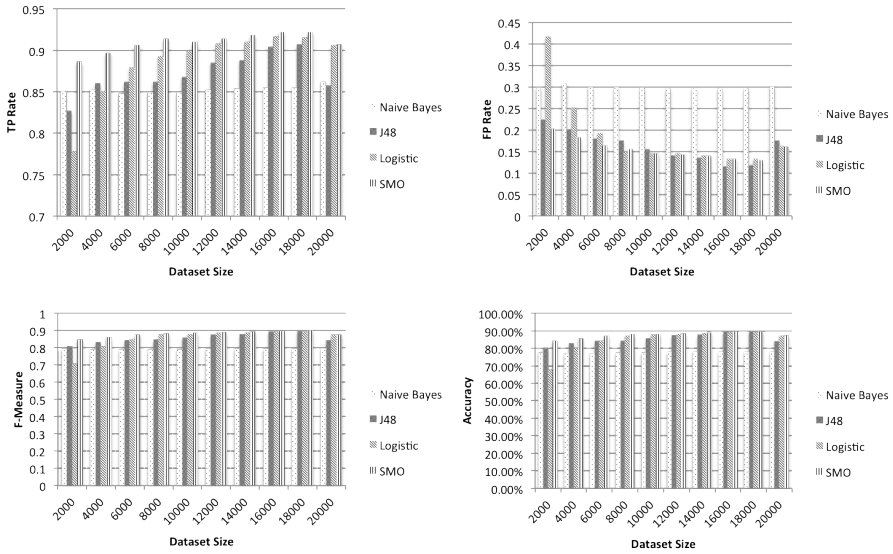


Figure 7: The performance metrics results for Webb Spam Corpus and WebBase web pages classification

Using a bag of words approach on the content of web pages results in over 50,000 words (after stripping HTML tags, removing stop words, and removing words which occur less than 10 times). As this attribute set is too large to use practically, we explore the impact of feature set size and corpus sample size on the effectiveness of

our classifiers. We vary the dataset size between 2,000 and 20,000, based on the features with most information gain, and varied the corpus sample size similarly in the range between 1000 and 10,000, with a unified random sample of spam and legitimate instances to generate an equal class distribution (thereby minimizing the class-specific learning biases). The size of total features in datasets influences the size of feature set we choose. After performing this evaluation, we found the majority of our classifiers consistently exhibited their best performance with 7,000 retained features (information gain values shown in Figure 6) and a corpus sample size of 16,000 instances. The performance metrics of the most four effective classifier are shown in Figure 7. We use these settings for the rest of our experiments involving the web page model.

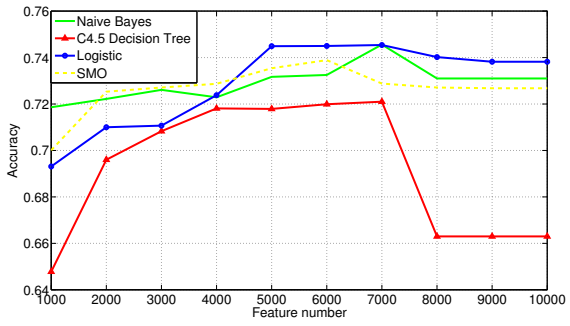


Figure 8: Classifier performance results for Cross-corpus learning on web page model

Using the sample size and feature size above we evaluated the performance of 40 classifiers in Weka. The performance metrics for a sample of the four most effective classifiers from our evaluation are shown in Figure 8. We can see that all of our classifiers preformed adequately, with Naïve Bayes performing the best overall in terms of average accuracy (Table 5 shows the confusion matrix that resulted from the Naïve Bayes classification choosing 7000 features).

We also ranked the Top 10 positive and negative attributes by χ^2 Test shown in Figure 9 — positive attributes are those words which appear more in legitimate web pages than spam web pages. Negative attributes are those words which appear more

Table 5: The confusion matrix in results of Naïve Bayes classifier

	Predicted Legitimate	Predicted Spam
True Legitimate	3286	1714
True Spam	830	4170

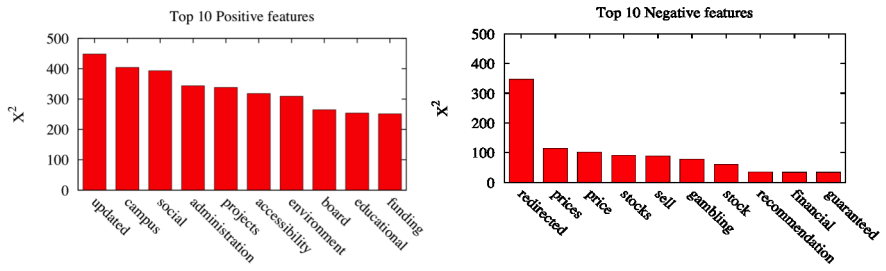


Figure 9: Top 10 positive and negative attributes for Webb Spam Corpus and WebBase.

in web spam than legitimate web pages.

2.5.2.3 Result Analysis

To investigate the misclassification, we manually reviewed the 1,714 misclassified legitimate web pages from Twitter and put them into several categories based on their content (shown in Table 6). Non-text category mainly were web pages primarily consisting of pictures, flash, video, audio, radio, TV, and other types, which usually contain very little visible text content. Foreign language category is misclassified due to unrecognized words, such as web pages in Japanese, Germany, and Chinese. Short text category mostly is from the messages or comments in social network (e.g. Twitter messages the length of which is under 140 characters). Comments list category is represented by web page which contains a list of comments following an object in social network like an article or blog. Download links category includes web pages which link to file downloads. Search engine category contain index pages for search engines like Google or Bing. Shopping/advertisement sites category contains pages which are filled with many descriptions and pictures of products.

The classification of these types of web pages can be fixed by white-listing the some legitimate sites. The white-list is from WebBase project and contains about

Table 6: The Categories of Misclassified Legitimate Web Pages

Categories	Amount	Ratio
Non-text	543	32.7%
Foreign language	366	21.4%
Short text	188	11%
Comments list	51	3%
Download links	137	8%
Search engine	57	3.3%
Shopping/advertisement sites	17	1%
Unidentified	355	20.7%

6,300 domain names. After white-listing these sites, we obtained better results using Naïve Bayes classifier. The F-Measure and accuracy achieved 0.9244 and 91.27% respectively.

Table 7: The confusion matrix in results of Naïve Bayes Classifier after white-listing legitimate sites

	Predicted Legitimate	Predicted Spam
True Legitimate	2890	121
True Spam	561	4242

Table 8: The performance metrics in results of Naïve Bayes Classifier after white-listing legitimate sites

Algorithm Name	TP	FP	F-Measure	Acc.
Naïve Bayes	88.32%	4.02%	0.9244	91.27%

We performed the same cross social-corpora experiments above on message model. We used the TREC email corpus to build the message model classifier and the Twitter message dataset for testing. Through similar setup as web page cross-domain classification, we obtain the results (shown in Table 9) using Naïve Bayes algorithm. The results achieved were not as good as those achieved with the web page model. The accuracy and F-measure are both showing that the classification contains lots of errors. After further investigation, we found the reasons as follows. Firstly, Twitter messages have a limit of 140 characters and are very short. In addition, a large

number of messages contain short-hand, abbreviated words, or contain URLs, which reduce the amount of content available for classification. For TREC data, the length of most email messages is longer than 140 characters and the lexicon is vastly different from Twitter's. To address the short-form and abbreviation issues in Twitter message dataset, we use lexicon normalization technique [86] to process the dataset firstly. The basic idea is to replace those popular short-form and abbreviation words with normal words in our dictionary. Then, we compare top 20 word features of two datasets based on information gain.

Table 9: The results of message and profile cross-domain classification using Naive Bayes

Classification Category	TP	FP	F-Measure	Acc.
Message	39.11%	3.7%	0.5164	66.18%
Profile	56.76%	22.46%	0.6334	67.15%

Table 10 shows the top 20 word features of two datasets based on information gain. It shows that two datasets have different top 20 word features sets. Therefore, if we use TREC data to test Twitter message data based on text content, a lot of Twitter messages may be misclassified.

Table 10: Top 20 word features for Twitter message data and TREC data

Rank	TREC	Twitter	Rank	TREC	Twitter
1	org	episode	11	statistics	ties
2	list	season	12	math	history
3	mail	watch	13	minimum	justin
4	wrote	pilot	14	reproduct	bieber
5	listinfo	fool	15	project	children
6	code	tube	16	ethanks	saves
7	mailman	rt	17	unsubscrib	jumped
8	comment	enemy	18	check	voices
9	post	turning	19	guide	lost
10	read	point	20	provide	funny

Another cross-corpora experiment we performed was using the user profile model. We use the MySpace corpus for building the user profile model classifier and the

Twitter user dataset for testing. Still we used Naïve Bayes algorithm and the same setup in profile cross-domain classification. We find that the results that shown in Table 9 are not good as previous achieved for web page classification.

After investigating the difference between MySpace and Twitter profile. we found that each Twitter profile only has five attributes: "Picture", "Name", "Location", "Web", and "Bio", whereas a MySpace profile has 11 attributes. We compared the spam and legitimate profiles on Twitter, expecting that spam profiles would use "Web" URLs and "Bio"s to propagate spam URLs (based on an observation made in a previous study [40]), but we find the percentage of empty values in "Web" URL and "Bio" attributes for spam profile is higher than the percentage of empty values in legitimate profiles. Meanwhile, the length of "Bio" attribute on Twitter also contains a small amount content due to a limit of 160 characters (enforced by Twitter). These reasons all result in performance not being as good as when training using MySpace profiles to test Twitter profiles.

Table 11: Statistics on "Web" and "Bio" fields in Twitter profile

	Web and Bio (not empty)	Total	Percentage
Total Profiles	536291	938190	57.16%
Spam Profiles	4675	26818	17.43%

Cross-domain classification proves that web page classification has good results in our case. Our error analysis shows that the major reason of poor results for message and profile datasets is different lexicons and special format. Thus, cross-domain classification is limited by common parts among multiple social networks. Some of the challenges faced with cross-temporal and cross-dataset issues can be helped by using the associative classification component of our framework to deal with the issues. In the following section, we introduce the associative classification.

2.5.3 Associative classification

Associative classification takes advantage of models that may be associated with a model being classified. Previous studies have shown that the integration of diverse spam filtering techniques can greatly improve the accuracy of classification [35]. Spammers are typically interested in advertising, which is usually done via links to websites hosting the spam. Thus, spammers usually post URLs whereas non-spammers post messages or status updates without URLs. To take advantage of the web page model, we use the associated URL information from our message model and profile model.

2.5.3.1 Classification Process

For easy illustration, we listed the process of our classification experiments in Figure 10. To simplify the experiments, we use OR combination strategy in "Result combination I" and "Result combination II" which means the entry will be labeled as spam whenever one classification shows positive result.

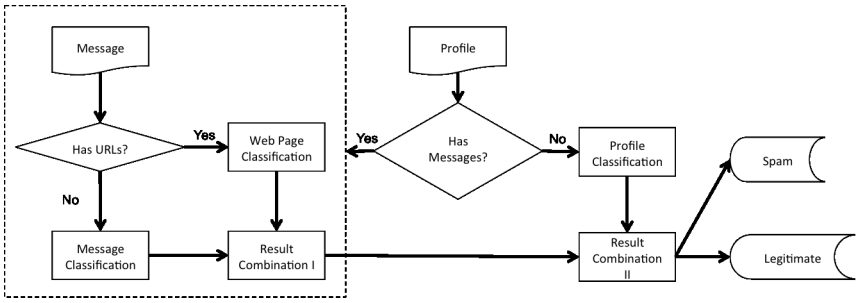


Figure 10: The process of associative classification

Taking the message model, as an example, the classification process works as follows. If a new message arrives with a URL in it, we extract the URL and fetch the associated web page content. Using the web page model classifier, we can then classify the web page and use the result to assist in making a determination of the classification of the message — combination strategies for different classifier results have been described in Section 2.3.4.

For messages which do not contain URLs, we classify them using the message model classifier. Previous research has achieved accurate classification results using content attributes and user behavior attributes [39, 82]. Under our data constraints, we only perform classification based on the content of messages.

For profile classification, we also combine the results from message classification and non-message profile text classification. The combination strategy is straightforward: The results from message classification are processed to use in threshold system for identifying profile spam. After finishing all the classification, we sum the results and obtain the final decisions whether the user profile is spam or not.

2.5.3.2 Classification Experiments

We demonstrate associative classification using the profile model as an example. We randomly choose 3,000 spam and legitimate Twitter profiles, also retrieving the associated messages. For messages, if they have an associated URL, we retrieve it using our crawler. Finally, we obtain all objects needed to help us in the profile classification. This amounts to 6,000 profiles, 28,841 messages, and 43,905 URLs (of which 17,211 are non-redirection). Using the WebbSpam corpus and WebBase corpus, the F-measure and accuracy of our associative classifier is 0.90 and 91% respectively.

Table 12: The results of web page model using Naïve Bayes Classifier

Algorithm Name	TP	FP	F-Measure	Acc.
Naïve Bayes	94.43%	15.42%	0.9000	90.98%

The result is shown in Table 12. We once again manually checked the misclassification and find they fall into similar categories as previously described. Some web pages contain sparse text or mainly non-text content and some are in foreign languages. The results of Twitter messages classification also have two parts: one is for messages which contain URLs, in which we use the content of the messages as well as web page model classifier to help us predict the final classification label

for the message. The other is for messages which do not contain any URLs, we use the content only to do the classification. We use the TREC 2007 corpus to train the message model (the training of the web page model has been previously described). Combining results from both classifiers, we obtain the final results of Twitter message classification for which F-measure value is 0.8955 and the accuracy achieved is 89.31% shown in Table 13.

Table 13: The results of message classification

Classification Category	TP	FP	F-Measure	Acc.
Non-URL message	69.96%	9.59%	0.7792	86.84%
Combined result	90.68%	11.84%	0.8955	89.31%

Using the above to obtain the results of messages associated with profiles, we perform threshold-based classification of Twitter profiles (akin to reputation systems). We apply threshold-based classification and vary the threshold between 1 and 10, depending on how many messages have been sent by users. We choose the threshold 1 to obtain the largest spam profiles for our dataset, although this also leads to the largest number of false-positives. The results of which are shown in Figure 11. One of the possible reasons for errors is that some spam profiles may have not been

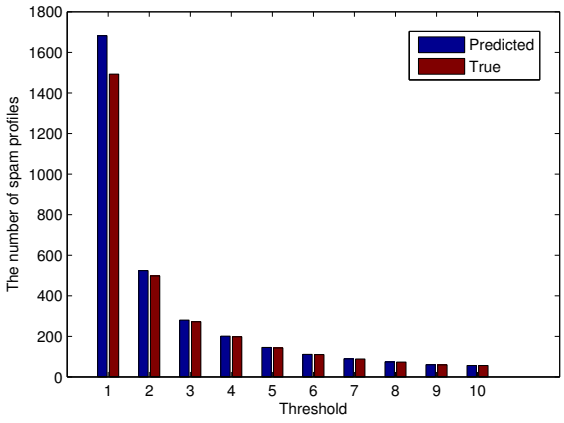


Figure 11: Using messages to identify spam profiles

suspended at the time we collected data (thus leading to the labels on the profiles

being legitimate instead of spam). Another reason is that for the Twitter dataset, there are some profiles which do not have any messages associated with them.

2.5.3.3 Result Analysis

Finally, we combined the results from text classification for profiles and message classification. From Table 14, we see the F-measure value is 0.8725 and the accuracy achieved by the profile classifier is 86.42%.

Table 14: Results of profile classification

Classification Category	TP	FP	F-Measure	Acc.
Message associated profile	86.82%	0.80%	0.9255	93.16%
Non-message associated profile	98.53%	39.53%	0.8278	79.94%
Combined result	92.93%	20.10%	0.8725	86.42%

The FP Rate in the final result of profile classification is higher than 20%. The majority of error classification is from the error classification of text classification. If we look at the associative learning using message to obtain spam profiles, the FP rate will be less than 0.9%. While the underlying reason for high FP rate for text classification is that user profiles don't contain much textual content, it would not be able to avoid the noises in the textual features of dataset for our classifiers. However, our associative classification shows its capability of improving the accuracy by 7% and FP rate by about 20%.

2.5.4 System Performance Evaluation

In this section, we use two metrics to measure the system performance of our framework: one is latency and the other is throughput. Here, latency means the time interval to process all the requests. Throughput means the number of items being processed in a period of time. The hardware setup is shown in Table 15.

Social networks generate huge amount of new information(e.g. Twitter generates about 5000 messages per second) over time. They have back-end data centers and high-speed network infrastructure to support. But we have limited resources so that

Table 15: Hardware setup of system performance experiments

Model	Dell PowerEdge R610
CPU	Six-core Intel Xeon processors 5500 and 5600 series
Memory	96 GB
Storage	1TB

we only can simulate the real-time scenarios proportionally. In the experiments, we use web page model as example, in which the dataset contains around 1 million web pages. By adjusting the workload (number of web pages), we have obtained the result of system performance evaluation shown in Figure 12.

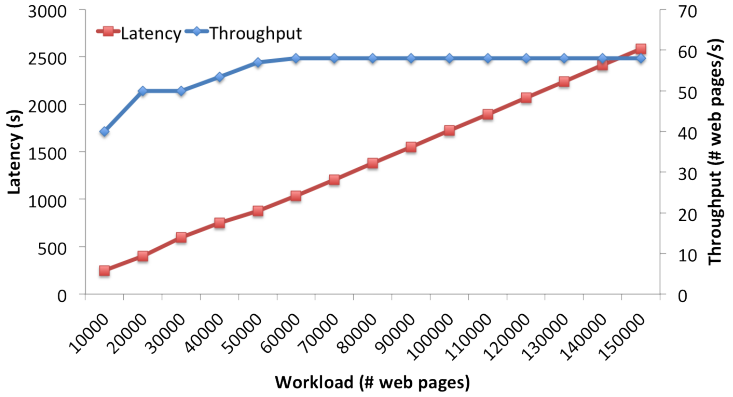


Figure 12: Result of system performance evaluation

Figure 12 shows that system throughput increases as the increasing of workload until it achieves the bottleneck. The bottleneck of throughput is around 60 web pages per second. The latency also increases as the increasing of workload, which shows linear increasing trend. Since our system performance heavily relies on the hardware setup of the experiments, we will need more computation resources to overcome the bottleneck.

With enough computation resources, we will adopt existing approaches of parallel classification such as parallel implementation of classification algorithms based on MapReduce [91] and parallel SVM algorithm based on MapReduce [128]. Those approaches need the supports of MapReduce framework and system resources. Also,

one active open source project named Apache Mahout, which is to build a scalable machine learning library, is a good source of ideas for our framework and algorithm design. The implementation of the parallel classification system will be considered as future work.

2.5.5 Discussion

We measured the efficiency and feasibility of cross-domain classification and associative classification in our experiments. It shows that combination of cross-domain classification and associative classification could improve the overall performance of the common objects in social networks. Also, we observed that cross-domain classification is not suitable for the objects which have few common parts while the associative classification could boost the performance of other objects' classification by using the good result from web page cross-domain classification. Even that we used very simple combination strategies such as majority voting and OR, the result is still promising. For more details of comparisons among different combination strategies, it is out of the scope of our work. Here, we will discuss more about data sharing among multiple social networks, real-time deployment and adjustment to new spam in our framework.

2.5.5.1 Data Sharing among Multiple Social Networks

First challenge for our framework is data sharing among multiple social network. One possible concern is user privacy. Our framework only uses public information such as messages sent out to the trending topics or public user profiles on social networks. In addition, we will adopt data anonymization technique to preserve users' privacy by anonymizing their personal information such as name and user id. In this way, privacy should not become a concern for our framework.

Also, another concern is the willingness of sharing data for social networks service

providers. The anti-spam collaboration among multiple social networks will be beneficial to all parties including the users on social networks. Under the condition that preserving users' privacy, they are likely to cooperate with each other. In addition, we could propose a volunteer program that allows users to voluntarily share data to our framework. But we have to admit that it will take time for social networks realize the benefits of collaborative framework.

2.5.5.2 Real-time Deployment

The real-time deployment of our framework acquires cooperations from social networks. Besides this, we need three main steps to achieve it. The first step is to launch our framework as a web service that provides data sharing APIs and spam notification mechanism to social network providers. The second step is to create a data storage center (e.g. data server) to hold all the incoming data from social networks. The last step is to design efficient machine learning classification algorithms on big data (e.g. using Mahout [2] from Hadoop framework). Meanwhile, we still need to do more work to evaluate the system performance of our framework.

2.5.5.3 Adjustment to New Spam

New spam detected by our framework will be added into pre-filtering module for filtering out similar spam in the future. Meanwhile, the framework will send the feedback to participating social networks making them aware the new type of spam and block it. Also, participating social networks may report some kind of spam from users to our framework. We also will add them into the pre-filtering module so that any similar or same type of spam will be filtered out quickly in our framework. In the end, the pre-filtering module will build up a common spam pool for all existing spam, which will greatly reduce the overall cost of spam detection.

[2] https://mahout.apache.org/

2.5.6 Countermeasures for Spam Evolution

Spammers will likely not stop evolving and wait us to be caught. In addition to building up the framework for all social media, we also established an anti-evolution component in framework. In the following, we outlined the possible attacks from spammers which may be applied to avoid our classification, followed by our countermeasures dealing with attacks.

2.5.6.1 Possible Attacks from Spammers

One possible attack is to crack our cross-domain classification by propagating distinct spam in different domains. Here, we consider every independent social medium as one domain. The important assumption of our cross-domain classification is that there exists common spam across domains. So if spammers know our strategy and send different types of spam to different domains, then it will be difficult for us to do cross-domain classification.

Another possible attack targets the associative classification by cut off or change the relationship among different objects in social media. Spammers change the strategy to put the spam URLs in pictures and videos instead of putting spam URLs in message and profile. In this way, our classifiers will not be able to detect them unless we discover those changes in advance and have corresponding detection techniques.

Third one camouflages spamming behaviors by stuffing legitimate objects. For instance, spammers send normal messages using spam user account and insert legitimate URLs in messages and profiles. All those legitimate objects in this case become noise in dataset. How to ignore those noise and obtain high accuracy is the critical part for us to address.

Here, we just list a few possible attacks which may established by spammers for avoiding our detection. In fact, we cannot predict all possible attacks from spammers.

2.5.6.2 Countermeasures

To deal with attacks from spammers mentioned above, we have the following countermeasures:

For cross-domain classification, we will have a common spam pool which collects all common spam so far. If spammer propagate different types of spam to different domains, it will increase the cost of spamming and violate the purpose of spreading spam in social media. After our framework is supported by enough social media, the cost of attack one will overweight the benefit of avoiding detection since our common spam pool will cover new types of spam from participating social media.

For associative classification, we will upgrade the relationships between objects in our framework. A monitor will be setup for overseeing the performance of associative classification. If there is abnormal decrease in result, an alarm will be launched. It still require the expertise to check the new types of spam and may have some time delay in detection. After identify new types of spam or new relationship, the framework needs to add new relationship to classification. For example, if spammer put URL in pictures, we need the technique to extract the URL from pictures and again use web page model classification to identify the web spam. The new relationship should be the picture is spam picture if the URL is spam URL.

For the camouflage attack, we will adopt multiple ways to detect spammers not only by the content features of objects in social media. Temporal and spatial features also will be used in spam analysis. For instance, the potential features could be time interval of sending messages, the distance between message sender and message retweeters (the distance could be defined as the number of users on the shortest path between two users), and number of URL links or topics in messages. Those features are selected based on some assumptions of spammers' behaviors. For example, spammers tend to use bots to automatically send messages in short period of time. Also

spammers may create multiple accounts in spreading message with URL links to multiple topics. Many researchers also have proposed different approaches or systems to address spam filtering using temporal and spatial features [88]. User behavior analysis in social media is another focus of our framework. For behavior analysis, we could use some common objects (e.g. URL link in message and profile) among objects in social networks to detect collective spamming activities [119]. Those kinds of analysis techniques are listed in our plan to improve our framework's scalability. The details of the implementation of those techniques is out of scope of this chapter.

2.6 Conclusion

The growth of social networks not only enables large-scale information sharing around the world but also attracts large number of spammers spreading social spam. Traditional spam detection approaches such as user-report-based filtering and content-based filtering are no longer efficient in distributed social media environment. The goal of our framework SPADE is to detect spam on multiple social networks. In this chapter, we introduced the definition of social spam and examples of user profile spam, message spam, and web page spam accordingly. It enables people to better understand the current status of social spam and recognize the potential consequences it may incur. Furthermore, we have demonstrated the feasibility and effectiveness of SPADE. It consists of three main components: Mapping and Assembly, Pre-filtering, and Classification. Mapping and Assembly component is similar to ETL(extract, transform, and load) tool which is to convert different input formats into uniformed ones. Pre-filtering component is a dynamic technique integration part to speed up filtering. Classification component is comprised of two major types: cross-domain and associative classifications. Through the experiments, it shows that SPADE can detect social spam on our basic models using several standard classifiers such as Naïve Bayes with respect to high F-measure and accuracy. In possible attacks analysis, we

explained why SPADE can be applied to multiple social networks and be resilient to evolution due to the spam arms-race. To overcome the limitation on the dataset, the next step for us is to deploy and evaluate SPADE on live feeds from social networks. Meanwhile, we will take system performance such as throughput and latency into account in evaluation. Also, future work will include integrating more detection techniques such as behavior analysis of spammers to our framework.

CHAPTER III

EVOLUTIONARY STUDY OF WEB SPAM

Web spam is defined as web pages that are created to manipulate search engines and deceive web users [82, 83]. Email has long been the primary method to spread web spam, although spammers are evolving with the times and quickly employing new techniques to spread web spam. One clear trend is the move towards social media due to the ease of sharing information providing more efficient and numerous channels for the growth of web spam. For example, web spam links in friend requests, inbox messages, and news feeds, are redirecting users to advertisement web sites or other types of malicious web sites. Further, social media sites have redefined the way links are shared with a tendency to share links using URL shortener [15].

Apart from evolution of web and applications on the web being one of the reasons driving change in web spam, there is a constant evolution of spam as a reaction to defensive techniques introduced by researchers [46, 193]. Improvements in defensive techniques used in web spam are enabled by researchers having access to corpora of web spam and being able to collaborate on developing and reporting results on web spam filtering techniques.

Previous studies [193, 194, 196] have introduced and studied first large-scale web spam corpus – Webb Spam Corpus 2006 through content and HTTP session analysis. The Webb Spam Corpus 2006 [193] is a collection of nearly 350,000 spam web pages, crawled from spam links found in email messages between November 2002 to January 2006. For legitimate web pages, the Standford WebBase project [162] provides topic focused snapshots of Web sites, in which the resulting archives are available to the public via fast download streams. After performing classification on those datasets,

the experiments results demonstrated that good performance and high efficiency of classification using HTTP session information.

In this chapter, we introduce the Webb Spam Corpus 2011, a new corpus of approximately 330,000 spam web pages. We compare this corpus with the previous, and first of its kind, web spam corpus [193] released in 2006. More concretely, we make the following contributions:

First, we create a new large-scale Web spam corpus – Webb Spam Corpus 2011 – which is a collection of approximately 330,000 spam web pages. Web spam links are extracted from spam email messages received between May 2010 to November 2010. Additionally, we also perform data cleansing to remove legitimate pages which may have been inadvertently collected (similar to the data cleansing performed in the prior Webb Spam Corpus by Webb et al. [196]).

Second, we analyze the Webb Spam Corpus 2011 from various perspectives. For example, we evaluate the new corpus on three main aspects: redirections, HTTP session information and content. Based on these aspects, we also make insightful observations. For example, when investigating legitimate web link attack in data cleansing, we found that social networks and search engines have become major targets of attacks.

Lastly, we studied the evolution of web spam by comparing Webb Spam Corpus 2011 with Webb Spam Corpus 2006. For redirections, Webb Spam Corpus 2011 has less redirection. Specifically, it has less "302 Found" redirections and location redirection but more iFrame redirections. The host names in redirection chains have new category – social networks sites, which indicates that social media have been manipulated to spread Web spam through hosting profiles, like plug-ins, and widgets. For HTTP session information, the percentages of hosting IP addresses for web spam in the ranges of 63.* -69.* and 204.* -216.* have changed from 45.4% and 38.6% in Webb Spam Corpus to 28.1% and 21.7% respectively. Additionally, we compared

the top 10 HTTP headers in the datasets. In terms of content, there are few exact content duplications between the datasets. We also compared the contents of the datasets from other content aspects: most popular words, top words based on information gain, and n-gram(n is from 2 to 3) sequences based on frequency. To evaluate the classification performance and feature change over time, we also performed new classification experiments on new dataset.

The remainder of the chapter is organized as follows. We motivate the problem further in Section 3.1. Section 3.2 introduces corpus including the data collection and cleansing methods. Section 3.3 compares Webb Spam Corpus 2011 with Webb Spam Corpus. Section 3.4 performs classification comparison on two datasets. We discuss related work in Section 3.5 and conclude the chapter in Section 3.6.

3.1 Motivation

Web spam has received a lot of attention with search engines constantly adjusting techniques to identify web spam [53] and social networks trying to prevent web spam propagating through their networks [175]. With web links being one of the most popular and easiest ways to share information on the web, web spam will remain a problem.

One of the most common technique to fight web spam is using machine learning, more specifically supervised learning techniques, to build classifiers for web spam using headers, content, or link features. As a prerequisite to using such techniques or researching new ones, having access to a large amount of labeled web spam is important and thus we collect, cleanse, and release a corpus of web spam as an enabler for researchers to improve and develop new web spam techniques. A standard corpus released for any number of researchers to use, as is the case with our corpus, allows and encourages collaboration between researchers to share and improve on each others results.

Although the release of the previous Webb Spam Corpus achieved this a number of years ago, we found that web spam has changed significantly enough to warrant an update to the Webb Spam Corpus. Namely, as detection techniques improve, spammers evolve and introduce new techniques to avoid detection. A concrete example of this is popular tools such as URL shortener (which reduce the length of a URL by mapping an identifier on a standard web link to a long URL) were quickly picked-up by spammers as a cheap method of obfuscation and redirection. Further, looking back at the year of 2006, social networks such as Facebook do not exist or are in early stage of startup or microblog sites such as Twitter. Thus, not only do we release the Webb Spam Corpus 2011, with real-time data collection, we also provide an analysis of evolution and major changes we have observed between the 2006 and 2011 version of the Webb Spam Corpus.

3.2 Webb Spam Corpus 2011

In this section, we introduce the data collection method, as well as the data cleansing process for the Webb Spam Corpus 2011.

3.2.1 Data Collection

3.2.1.1 Collection Method

We introduce the Webb Spam Corpus 2011 which is available for download for collaborative research investigation and reporting as an .arff file (Weka file format[1]) at the Webb Spam Corpus' home page—`http://www.cc.gatech.edu/projects/doi/WebbSpamCorpus.html`. The two main parts involved in creating the Webb Spam Corpus 2011 are data collection and data cleansing. These steps are detailed below and a high-level overview of the process is provided in Figure 13.

[1]Weka: http://www.cs.waikato.ac.nz/ml/weka/

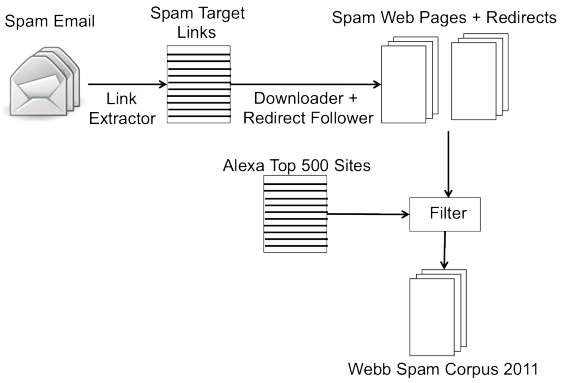


Figure 13: Illustration of data collection and data cleansing process.

3.2.1.2 Source URL and Actual URL

We distinguish URL links into two groups: source URLs and actual URLs. Here, source URLs are the original URLs extracted from email messages and are typically what the end user will see in the email message. Actual URLs are the final URLs or the URL of the web page that the user finally sees in their browsers. That is, this is the final URL after all redirects (HTTP redirects, Javascript redirects, meta-tag redirects, and more) have been followed. If a web page does not redirect a user, the actual URL could be the same as the source URL. To clarify this, the relationship between source URL and actual URL is shown in Figure 14:

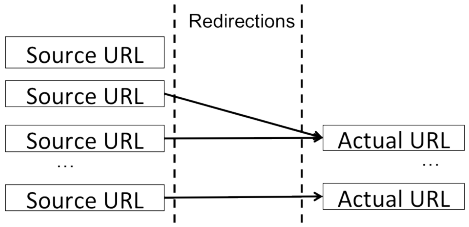


Figure 14: The relationship between source URL and actual URL.

The relationship between source URL and actual URL has the following characteristics:

a). One redirection chain leads from source URL to actual URL;

b). Many source URLs may redirect/map to a single actual URL;

c). Source URLs which were successfully accessed without resulting in a redirect is actual URL.

3.2.1.3 Source URL links

We start with a set of source URLs extracted from 6.3 million spam emails collected between May 2010 to November 2010 to a moderately sized email service provider. We only extract HTTP and HTTPS URLs (although HTTPS links make up only 0.2% of all the spam links we extracted), using Perl's URI::Find::Schemeless and Html::LinkExtor modules to extract URLs from text and HTML respectively. We end up with 30.7 million web links (15.1 million unique links). Figure 15 shows the distribution of URL links in months. We also investigate the top level domains in source URLs and list top 10 TLDs in Table 16. "RU" is top level domain for Russian Federation and "DE" is top level domain for Federal Republic of Germany. In this study, we focus on English language web pages only, which are about 1.7 million web pages (before cleansing) which were crawled in March 2011.

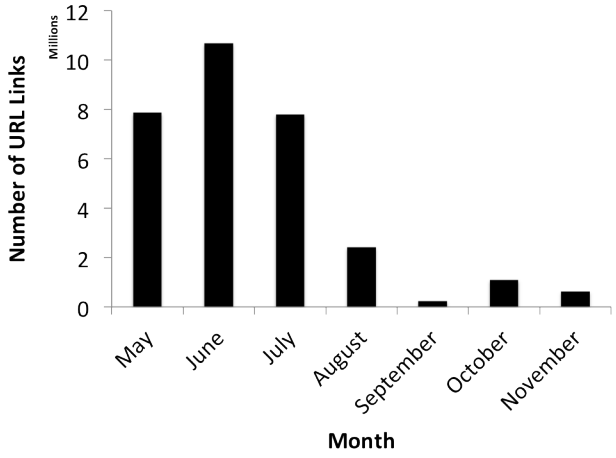


Figure 15: Distribution of source URL links in months.

Table 16: List of Top Level Domains in Source URLs.

Top Level Domain	Number of Unique Source URLs
RU	10,052,443
COM	3,063,766
NET	205,311
UK	191,583
INFO	168,192
DE	125,472
NL	117,099
PL	106,287
IP Addresses	13,263
Other	1,061,023

3.2.1.4 Web spam downloading

Once we have a set of source URLs, we proceed to download all the web pages. We use a custom crawler written using Perl's LWP::Parallel::UserAgent module to download corresponding web pages. We then follow any iFrame-redirects, http-redirects, Javascript redirects (using Mozilla's Rhino), or meta-tag redirects. More details can be found in [193] which uses similar techniques. We keep the raw headers and HTML content of the page, and do not crawl or spider links from it. We downloaded a total of 1.7 million pages (including redirections) and in-total collected over 1 GB of data.

3.2.2 Data Cleansing

Data cleansing on Webb Spam Corpus 2011 is split into two parts:

3.2.2.1 Removing False-positives

False-positives in corpus include legitimate URLs and error pages. Spammers often include legitimate URLs in spam emails to avoid spam rules or to appear legitimate [196]. Using Alexa's top 500 site list[2], we list top 10 legitimate actual URLs in Webb Spam Corpus 2011 shown in Figure 16.

Figure 16 shows that 4 social networks websites (`www.facebook.com`, `support.`

[2]http://www.alexa.com/topsites

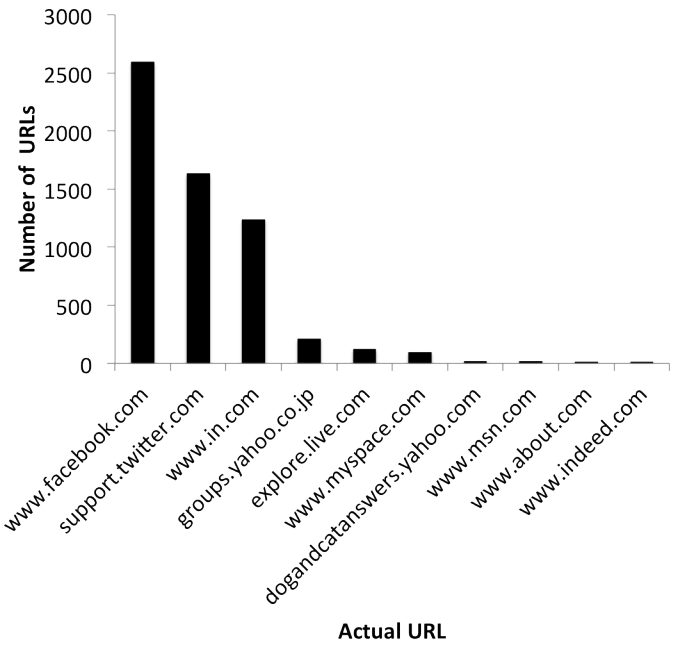


Figure 16: Top 10 legitimate actual URLs in Webb Spam Corpus 2011.

`twitter.com`, `www.in.com`, and `www.myspace.com`), 5 search engines (`groups.yahoo.co.jp`, `explore.live.com`, `dogandcatanswers.yahoo.com`, `www.about.com`, and `www.indeed.com`), and 1 information portal (`www.msn.com`) are in the top 10 list. It indicates that spammers are using popular social networks and search engines in legitimate URL attack. We removed $6,175$ legitimate actual URLs and $6,494$ legitimate source URLs in this process.

Besides legitimate URL links in spam emails, the downloaded web pages also contain other false-positives. Although these actual URLs may have been spam URLs, due to the delay in setting up our downloading and cleansing system, the spam URLs were crawled a few months after the source URLs were extracted. This resulted in a number of 404 HTTP errors or custom served "404 error web pages".

We eliminate such pages as well as previously mentioned false-positives leaving us with $673,489$ spam web pages in the corpus.

3.2.2.2 Removing Non-Textual Web Pages

Approximately 98% of web pages identify their "Content-type" as text/html. After cleansing false-positives in corpus, we discard non text/html pages. By removing non-textual web pages based on the attribute "Content-Type" in HTTP header information, we kept 673, 313 web pages including 342, 478 redirections.

3.2.3 Data Statistics

After finishing downloading all web pages, we investigate the distribution of top level domains and HTTP status codes. The purpose is to find which top level domain hosts the most web spam and the most common HTTP responses when we click through those spam URL links.

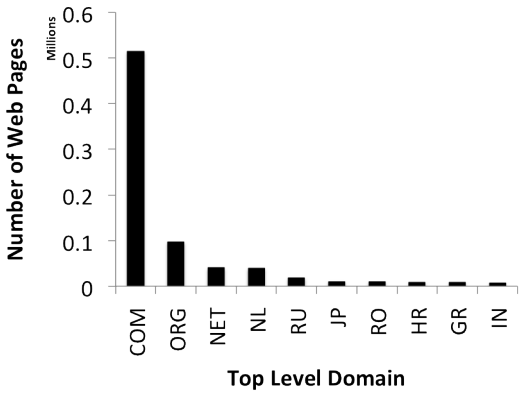


Figure 17: Top 10 top level domains.

To obtain popular top level domains, we process the dataset in the following steps. First, we collect all top level domains from IANA Data[3], which contains 313 top level domains (last updated Jun 20, 2012). By matching all the source URLs in downloaded files with the top level domains list, we aggregated the count of web pages in the same top level domain. The 10 most popular top level domains are shown in Figure 17. We see that the three most popular top level domains COM, ORG, and

[3]http://data.iana.org/TLD/tlds-alpha-by-domain.txt

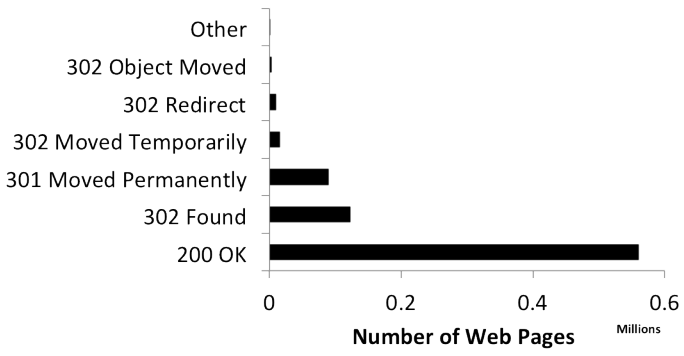


Figure 18: Distribution of HTTP status codes.

NET almost represent more than 80% of the TLDs. Especially, the percentage of web pages which are belonging to top level domain COM is over 60%.

For HTTP status codes, we aggregate all status codes based on the number of web pages and list the distribution of status codes shown in Figure 18. It shows that "200 OK" is the most common of status code in Webb Spam Corpus 2011 – over 70%. Also other status codes which are primarily used in redirection, such as "302 Found", "301 Moved Permanently", and "302 Moved Temporarily", are quite popular.

3.3 Comparison between Two Datasets

We compare the Webb Spam Corpus 2011 with Webb Spam Corpus 2006 in three dimensions: redirections, HTTP session information, and content.

3.3.1 Redirections

Redirections are normally used by spammers to camouflage the actual spam URL links and avoid being blocked by URL blacklists. We look into redirections returned by source URLs in the Webb Spam Corpus 2011 shown in Table 17.

To compare fairly with redirections in the Webb Spam Corpus 2006, we compute the percentage of source URLs versus number of redirections shown in Figure 19. It shows that Webb Spam Corpus 2011 has more source URLs returning no redirects

Table 17: Number of redirects returned by source URLs.

Number of Redirects	Number of Source URLs
0	254,315
1	15,075
2	2,880
3	387
4	1361
5	86
6	58
7	46
8	31
9	27
10	26
11	19
12	13
13	15

(more source URLs which are also the actual URLs). The possible reasons are as follows: a) spammers are using less redirections for camouflaging actual spam URLs; b) Webb Spam Corpus 2011 has more URL links than Webb Spam Corpus 2006; c) there may exist false positives in Webb Spam Corpus 2011 before data cleansing.

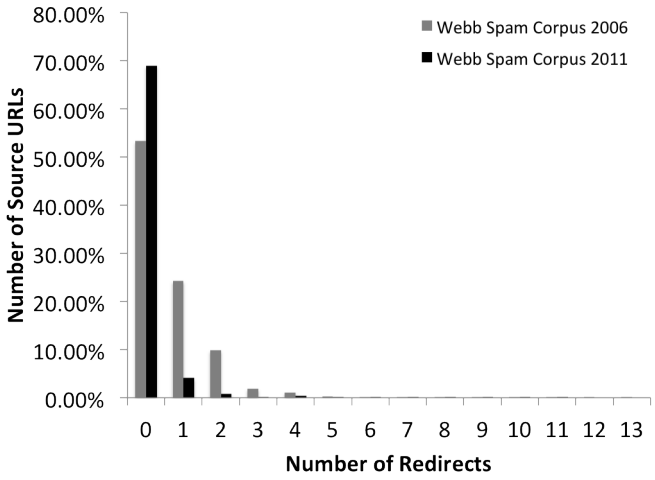


Figure 19: Comparison based on percentage of source URLs vs number of redirections.

We also aggregate source URLs based on the actual URLs they are mapping to and generate the distribution of number of source URLs that point to the same actual

URL shown in Figure 20. It shows similar trend as the distribution of the number of source URLs that point to the same actual URL in Webb Spam Corpus 2006.

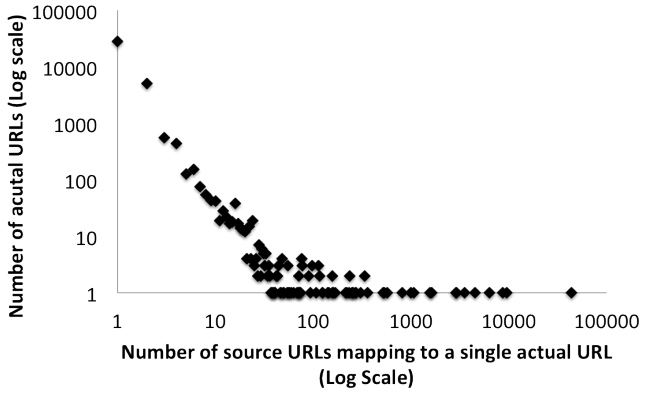


Figure 20: Distribution of the number of source URLs that point to the same actual URL.

Redirections have different categories including HTTP redirect, frame redirect, iFrame redirect, meta-refresh redirect and location redirect [196]. For HTTP redirect, it also has some subcategories based on response status such as "301 Moved" HTTP redirect and "302 Found" HTTP redirect. We compare the redirection distribution of two datasets which is shown in Figure 21.

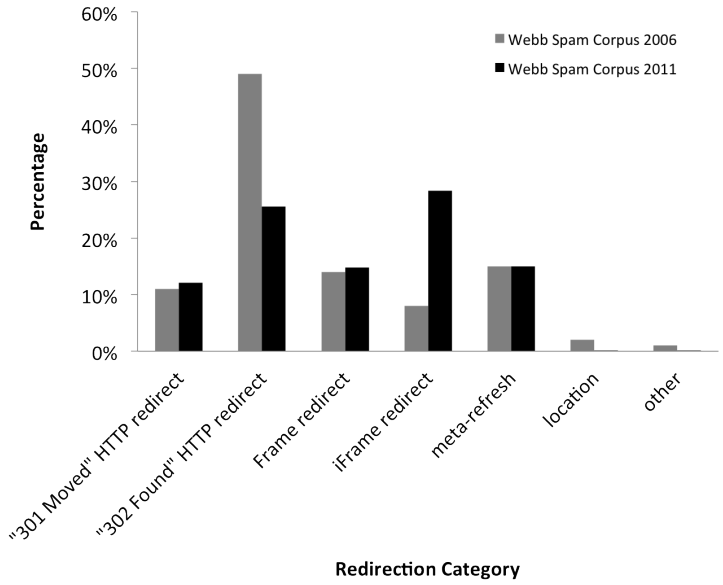


Figure 21: Comparison between redirection distributions of the two datasets.

Figure 21 shows HTTP redirect in Webb Spam Corpus 2011 still occupies the majority of redirections, accounting for 41.7% of the redirections (25.6% for "Found" redirects, 12.1% for "Moved Permanently" redirects, and 4.0% for other HTTP redirects). HTML frame and HTML iFrame redirects account for 14.8% and 28.4% respectively. Redirection using meta-refresh tags account for 15.0% and location redirect accounts for less than 1% of all redirects.

We observe that Webb Spam Corpus 2011 has fewer "302 Found" redirections and location redirection. But it has more iFrame redirections. Meanwhile, we found that Webb Spam Corpus 2011 has other HTTP redirects which occupies 4% redirections. The response status examples of other HTTP redirects includes: a) "302 Object moved"; b) "302 Moved Temporarily"; c) "302 Redirect".

Besides showing the distribution of redirections, we also look into the common host names in redirection chains which will tell us what kinds of websites have been taken advantage of by the spammers. The most common host names in redirection chains including HTTP redirection, frame redirection, iFrame redirection, and meta-refresh redirection are shown in Table 18.

From Table 18, we investigated all the host names and found that there are three major categories: domain parking websites, social networks websites, and advertiser websites. For example, `bodisparking.com` and `sedoparking.com` are domain parking websites. `facebook.com` and `in.com` are social networks websites. `ad.doubleclick.net` is advertiser websites. The first set of counts represent the view of all of the HTTP, HTML, and JavaScript redirection techniques. This list consists of 3 domain parking services, 1 advertiser, and 1 social networks. The top 5 HTTP redirect host names consist of 1 domain parking service, 3 advertisers and 1 social networks. The top 5 frame redirect host names consist of 3 domain parking services, 2 advertisers. The top 5 iFrame redirect host names consist of 1 domain parking services, 3 advertisers, and 1 social networks. The top 5 meta refresh redirect

Table 18: Most common host names in redirection chains.

Top 5 host names in redirection chain	
Host name	**Count**
domdex.com	59,004
www.facebook.com	37,580
domains.googlesyndication.com	9,934
bodisparking.com	9,530
potentbusy.com	9,431
Top 5 host names of HTTP redirection	
Host name	**Count**
mrs45.hosteur.com	9,046
home.wanadoo.nl	8,624
arpitjain.in	6,054
sharepoint.microsoft.com	4,596
www.in.com	4,336
Top 5 host names of frame redirection	
Host name	**Count**
bodisparking.com	9,530
potentbusy.com	9,430
www.ndparking.com	7,192
www.sedoparking.com	1,306
searchportal.information.com	1,209
Top 5 host names of iFrame redirection	
Host name	**Count**
domdex.com	59,004
www.facebook.com	14,960
ad.doubleclick.net	2,649
areasnap.com	2,219
bullishcasino.com	1,672
Top 5 host names of meta refresh redirection	
Host name	**Count**
www.facebook.com	19,931
domains.googlesyndication.com	9,875
www.lawtw.com	6,736
www2.searchresultsdirect.com	1,838
www.sedoparking.com	1,472

host names consist of 3 domain parking services, 1 advertiser, and 1 social networks.

Domain parking for idle domains is used to display advertisements and earn money. It is easy to understand that spammers are using these domains for monetary benefit. Advertisers are similar to domain parking services on displaying advertisements

which may not be useful for users . For social networks websites, we studied in detail about Facebook URLs in Webb Spam Corpus 2011. We found that the majority of redirections from Facebook belongs to iFrame redirection, meta-refresh redirection and HTTP redirection. In iFrame redirection, there are three types of URL redirections based on the sub path of URL links: "connect", "plugins", and "widgets", which accounts for 72.6%, 24.4%, and 3% respectively. Also the "connect" URL link redirects users to the profiles hosted Facebook. In our dataset, 10,820 "connect" URL link redirects to "t35.com" profile hosting in Facebook. "t35.com" is a domain parking services website. For 3,655 "plugins" URL links, 3,379 of them are "like" box plug-in and 140 of them are "activity" plugin. Normally, if you click on "like" box plugin, you will become a fan of events, products, or profiles so that you will be kept updated with news feeds and status changes. For "activity" plug-in, you will join the activity if you click on it. "Widgets" URL links are similar to "plugins" URL links. 444 "widgets" URL links provide "like" button for users to click. Therefore, we can conclude that spammers are using the power of social networks to spread spam information.

3.3.2 HTTP Session Information

Webb Spam Corpus 2011 also contains the HTTP session information that was obtained from the servers that were hosting those pages. In this section, we compare two datasets focusing on the most common server IP addresses and session header values.

3.3.2.1 Hosting IP Addresses

Hosting IP address is the IP address that hosts a given web spam page. Figure 22 shows the distribution of all of the hosting IP addresses over network number in Webb Spam Corpus 2011. Here network number is the first 8 bits of IPV4 address. Previous study [196] said that the 63.* -69.* and 204.* -216.* IP address ranges account for

45.4% and 38.6% of the hosting IP addresses respectively in Webb Spam Corpus. While in Webb Spam Corpus 2011, the percentages of IP addresses in those two ranges change to 28.1% and 21.7% respectively. Another two IP address ranges 70.* -100.* and 170.* -203.* account for 21.3% and 14.0% of the hosting IP addresses respectively in Webb Spam Corpus 2011.

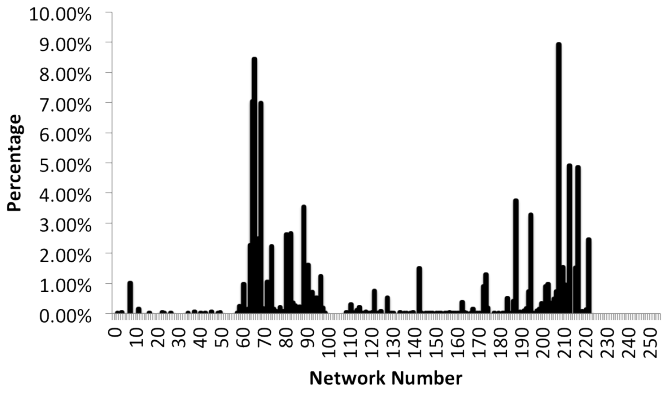


Figure 22: Distribution of hosting IP address.

It implies that spammers are comprising more various hosting IP addresses to spread web spam. The reason may be the IP blacklists used in popular anti-spam filters which force spammers to use new IP addresses for hosting web spam. To investigate most popular hosting IP addresses in Webb Spam Corpus 2011, we list top 10 hosting IP addresses based on the count of web pages. Meanwhile, through whois service, we obtain the server location and ISP (Internet service provider) for every hosting IP address.

Table 19 shows 4 IP addresses from 63.* -69.*, 2 IP addresses from 204.* -216.*, and 4 IP addresses from other ranges. Also, it shows that 4 IP addresses from US servers, 2 IP addresses from France servers, and 4 IP addresses from other countries (Australia, Korea, Netherlands, and Germany). We can see that all servers are legitimate servers which does not mean those legitimate servers are the spammers. It only means the web services provided by those servers are used by the spammers for the spamming purpose.

Table 19: Top 10 hosting IP addresses.

Hosting IP Address	Count	Server Location	ISP
208.073.210.029	23,785	Los Angeles, CA USA	Oversee.net
065.055.011.238	21,205	Redmond, WA USA	Microsoft Hosting
213.186.033.019	17,543	France	Ovh Systems
066.196.085.048	16,542	Sunnyvale, CA USA	Inktomi Corporation
069.043.160.174	13,289	Beaumaris,Victoria In Australia	Castle Access
066.045.237.214	10,834	Secaucus, NJ USA	Interserver
222.122.053.065	9,090	Seoul, Republic of Korea	Korea Telecom
217.016.006.170	9,073	France	AB Connect
195.189.117.037	8,624	Nijmegen, Gelderland in Netherlands	Bluedome Internet Application Services BV
188.040.054.131	8,538	Germany	Hetzner Online AG

3.3.2.2 HTTP Session Header

Previous study [194] has shown that HTTP session information is used for predict web spam efficiently. As the evolution of web spam, we intend to see whether HTTP session information of web spam has changed over time. To obtain most popular HTTP session information, we rank out top 10 HTTP session headers based on the count of web spam which those headers are associated with, shown in Table 20.

Table 20: Top 10 HTTP session headers.

Header	Total Count	Unique Count	Most Popular Value (Count)
CONTENT-TYPE	379,721	120	text/html(147,428)
SERVER	369,985	919	Apache(82,004)
CONNECTION	359,786	5	close(312,186)
CONTENT-LENGTH	271,654	12,004	77(22,039)
X-POWERED-BY	148,944	191	ASP.NET(70,088)
CACHE-CONTROL	141,062	585	private(70,712)
SET-COOKIE	134,063	116,522	parkinglot=1;domain=.potentbusy.com; path=/;(3931)
LINK	122,352	5,012	`http://l.yimg.com/d/lib/yg/css/dynamic_200602130000.css`;rel= “stylesheet”; type=“text/css” (15,446)
P3P	92,591	248	policyref=“`http://www.dsnextgen.com/w3c/p3p.xml`”(24,180)
EXPIRES	90,915	7,668	Mon, 26 Jul 1997 05:00:00 GMT(25,641)

Compared with top 10 HTTP session headers, Table 20 shows some changes as

follows: a). new header P3P appears in top 10 list and old header PRAGMA has been removed from the list; b) the most popular values for the header SERVER and CONTENT-LENGTH have changed from "microsoft-iis/6.0" to "Apache" and from $1,470$ to 77 respectively; c). the order of the header CONTENT-LENGTH moves before X-POWERED-BY but the others keep the same relative order. Also, we find that 79.1% of the web spam pages with a SERVER header were hosted by "Apache" (60.5%) or "Microsoft IIS" (18.6%). In Webb Spam Corpus, 94.2% of the web spam pages with a SERVER header were hosted by "Apache" (63.9%) or "Microsoft IIS" (30.3%). Most popular value for the header CONTENT-LENGTH is not able to show the trend of content length so we also obtain the distribution of content length shown in Figure 23.

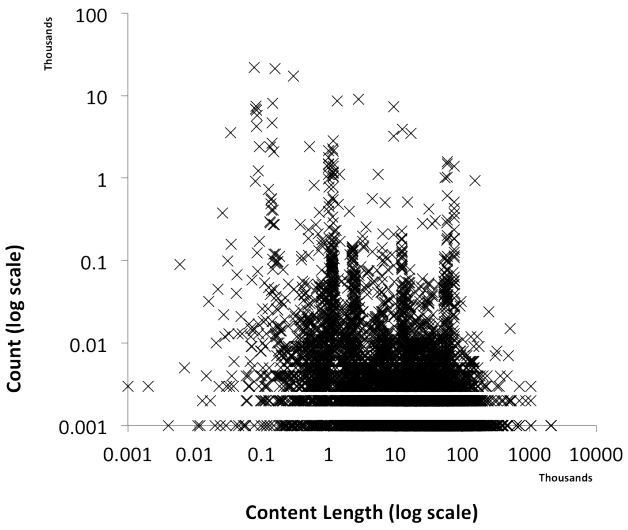


Figure 23: Distribution of content length.

Figure 23 shows the average value of content-length is between $1,000$ and $1,0000$ although the most popular value is 77 bytes. As more multimedia used in web spam, the content length of web spam text gradually becomes shorter. Another thing we also need to check is whether the content of web spam also evolve over time.

3.3.3 Content

In this section, we compare two datasets on duplications and syntax changes between them. For duplications, we try to find the overlap between them based on MD5 hash values of content of web spam. For syntax changes, we intend to obtain the evolution of web spam syntax by comparing information gain of words and n-gram phrases.

3.3.3.1 Duplications

We compute MD5 hashes on the content of HTML web pages when we crawl the URL links. After evaluating these results, we find that there are $122,618$ unique MD5 values in Webb Spam Corpus 2011. Thus, $247,367$ of the web spam pages (66.9%) have the same HTML content as one of $122,618$ unique web spam pages. The percentage of exact content duplicates is much higher than the percentage (42%) in Webb Spam Corpus 2006 [193]. One possible reason is more URL duplications in the Webb Spam Corpus 2011.

To check the duplications between the two datasets, we iteratively compared MD5 codes of every web spam in Webb Spam Corpus 2011 and Webb Spam Corpus. The result of comparison is that $7,257$ web spam in Webb Spam Corpus 2011 are overlap with $2,834$ web spam in Webb Spam Corpus 2006. The percentages of duplications between two datasets are 2.0% and 1.3% in Webb Spam Corpus 2011 and Webb Spam Corpus 2006 respectively. Therefore, there are very few exact content duplicates existing between the two datasets.

3.3.3.2 Syntax Analysis

We analyze syntax of Webb Spam Corpus 2011 by computing the information gain of words in the content of web pages. Information gain, which is also called Kullback-Leibler divergence [111] in information theory, is calculated based on entropy as follows:

$$IG(T, a) = H(T) - H(T|a) \tag{5}$$

Here, T denotes a set of training examples and a presents the ath attribute of instance. $H(T)$ is the entropy of T and $H(T|a)$ is the conditional entropy of T with knowing the value of a.

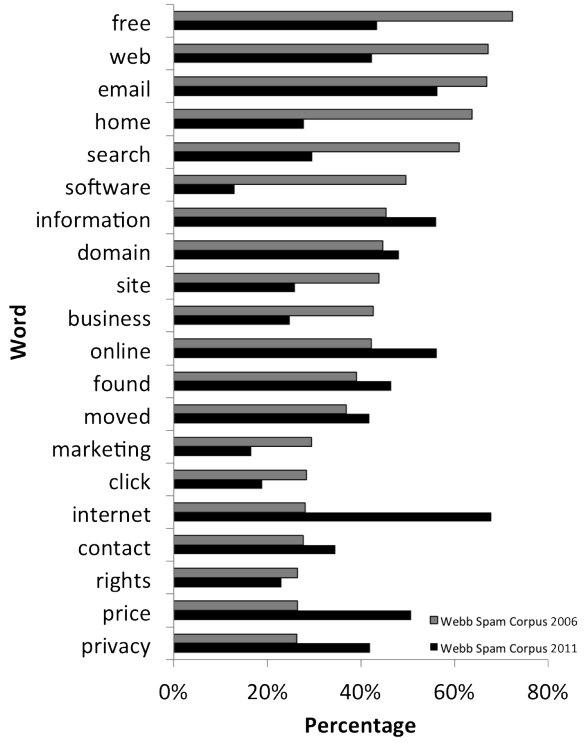


Figure 24: Top 20 most popular words in Webb Spam Corpus [2006/2011] vs. percentage of documents that contain them in two datasets.

Taking every web page as document, we adopt a bag of words model [122] to generate document instances in binary features. First, we need to tokenize the documents. Tokenization is the process of splitting the document up into words, phrases, symbols, or other meaningful elements called tokens. The features are the tokens in all documents and the value of feature is false if the token appears in the document or true if not.

For the words in web pages, we first list top 20 most popular words in Webb Spam

Corpus and their appearance as a percentage of documents that contain them, shown in Figure 24.

Figure 24 shows that some words in the top 20 list appear less than in Webb Spam Corpus 2011 such as “free”, “web”, “home”, “search”, and “software”. Some words appear more frequently than in Webb Spam Corpus 2011 such as “information”, “online”, “internet” and “price”. It indicates the trend of spammy words and changes over time.

Besides most popular words, we also look into the discriminative words which distinguish two datasets. We ranked them by the value of their information gain according to the formula and used different labels to mark the instances in two datasets. The result of top 10 words based on information gain is shown in Figure 25.

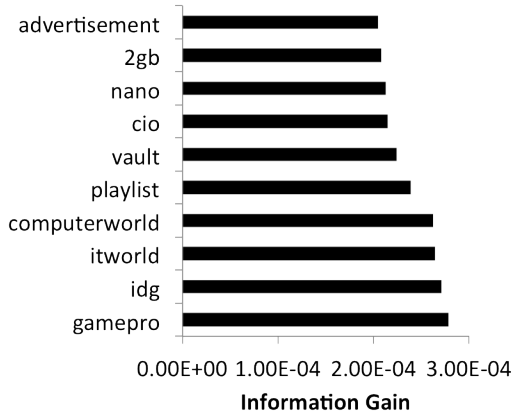


Figure 25: Top 10 words based on information gain.

Figure 25 shows top 10 words based on information gain. We further found that all words except “playlist” appear in Webb Spam Corpus 2006 while only four words including “playlist” , “vault”, “cio”, and “advertisement” present in Webb Spam Corpus 2011. Since we transformed all words into lower case format, words such as “cio” and “itworld” should be “CIO” and “ITworld”. Word “playlist” normally appears in multimedia section of social media. For example, user profile has the embedded radio player which has a playlist for visitors.

Table 21: Top 20 n-gram (n is from 2 to 3) sequences based on frequency in the two datasets (first 20 rows for Webb Spam Corpus 2006).

2-gram	Frequency	3-gram	Frequency
of the	149,029	just a few	26,585
in the	88,505	⟨N⟩ x ⟨N⟩	26,488
V ⟨N⟩	77,254	is just a	26,016
to the	77,050	the links below	25,910
on the	72,948	links below to	25,834
⟨N⟩ A	71,207	for your favorite	25,801
v ⟨N⟩	66,725	a few clicks	25,799
X ⟨N⟩	64,701	the search box	25,750
a ⟨N⟩	63,490	the Web for	25,723
⟨N⟩ x	63,019	looking for is	25,705
⟨N⟩ D	60,603	to search the	25,689
B ⟨N⟩	59,164	search the Web	25,658
x ⟨N⟩	58,455	below to search	25,636
A ⟨N⟩	57,568	few clicks away	25,633
may be	57,522	Use the search	25,632
⟨N⟩ GB	56,328	search box above	25,632
⟨N⟩ a	55,437	above or the	25,625
Price ⟨N⟩	55,424	Whatever you re	25,623
⟨N⟩ B	55,153	or hte links	25,622
⟨N⟩ s	53,330	Web for your	25,619
of the	212,626	w ⟨N⟩ org	138,162
http www	169,180	http www w	126,770
w ⟨N⟩	140,524	www w ⟨N⟩	126,770
⟨N⟩ org	138,247	⟨N⟩ org ⟨N⟩	92,935
Price ⟨N⟩	127,091	org ⟨N⟩ ⟨N⟩	91,219
www w	126,770	mg x ⟨N⟩	73,110
in the	126,273	⟨N⟩ mg x	73,110
USD ⟨N⟩	117,108	⟨N⟩ ⟨N⟩ xmlenc	69,898
Related Searches	103,259	⟨N⟩ USD ⟨N⟩	63,904
Save ⟨N⟩	100,710	Found The doument	58,506
x ⟨N⟩	99,327	Found Found The	58,424
org ⟨N⟩	93,803	⟨N⟩ Found Found	58,424
Privacy Policy	93,328	You Save ⟨N⟩	58,127
to the	91,951	⟨N⟩ You Save	58,103
hair loss	77,774	Price ⟨N⟩ You	56,065
Internet Bellen	77,544	Admin Page Insights	54,726
⟨N⟩ mg	74,103	Retail Price ⟨N⟩	54,400
mg x	73,110	⟨N⟩ Retail Price	54,058
on the	71,891	Download Price ⟨N⟩	54,049
for the	70,177	Price ⟨N⟩ Retail	53,986

Moreover, we compared n-gram (n is from 2 to 3) sequences in the two datasets. After using Perl's Text::Ngrams module[4], we list top 20 n-gram (n is in the range of from 2 to 3) in two datasets based on frequency shown in Table 21.

In Table 21, $\langle N \rangle$ denotes any number sequence. Also we have removed redirections and the grams which only contain number sequences. Webb Spam Corpus 2011 has $22,894,416$ 2-gram sequences and $14,223,621$ 3-gram sequences, compared with $17,049,809$ 2-gram sequences and $6,488,343$ 3-gram sequences in Webb Spam Corpus 2006. Table 28 shows that there are more numeric sequences appearing in 2-gram sequences in Webb Spam Corpus 2006 than in Webb Spam Corpus 2011. 3-gram sequences in Webb Spam Corpus 2006 are more related to links and search while those in Webb Spam Corpus 2011 are more related to price and money.

3.4 Classification Comparison

Previous research [194] shows that web page spam could be detected using HTTP header information efficiently. To further investigate the difference between two datasets, we compare them in terms of classification features and performance. Through the comparison, we try to find out whether the HTTP header information features still are discriminative and how well those classifiers perform on new dataset. To use the results in previous research as control, we adopted similar feature selection and experimental setup in our experiments.

3.4.1 Feature Generation and Selection

In our experiments, we adopt the traditional vector space model [158] (or "bag of words" model) to represent data in a consistent format. Also, this model has been quite effective in previous information retrieval and machine learning research. In the model, it uses a feature vector f of n features: $< f_1, f_2, ..., f_n >$ to represent each

[4]http://search.cpan.org/dist/Text-Ngrams/Ngrams.pm

data instance. Since all of our features are Boolean, we obtain that the feature is present in a given instance if $f_i = 1$; otherwise, the feature is absent. Meanwhile, we borrow three types of feature representations(phrases, n-grams, and tokens) from previous research [194] for each unique HTTP information header value. The feature generation process is: 1) we keep the header value as an uninterrupted phrase; 2) we tokenize the header value using whitespace and punctuation characters as delimiters; 3) we perform n-gram(n is from 1 to 3) generation from the tokens; 4) we prepend the header name to each of feature values. One example of feature representations is illustrated in Table 22 [194].

Table 22: Feature Representations.

Representation	Feature
Phrase	$server\_apache/2.0.52(fedora)$
N-grams	$server\_apache/2052$
	$server\_052fedora$
	$server\_apache/20$
	$server\_052$
	$server\_52fedora$
Tokens	$server\_apache/2$
	$server\_0$
	$server\_52$
	$server\_fedora$

The feature selection process is based on a well-know information theoretic measure called information Gain [67, 204]. Information Gain is defined as follows:

$$IG(f_i, c_j) = \sum_{c \in \{c_j, \bar{c_j}\}} \sum_{f \in \{f_i, \bar{f_i}\}} p(f, c) \cdot log \frac{p(f, c)}{p(f) \cdot p(c)} \tag{6}$$

where f_i is a feature in the feature vector, c_j is one of the classes(i.e., spam or legitimate), $p(f)$ is the probability that f occurs in the training set, $p(c)$ is the probability that c occurs in the training set, and $p(f, c)$ is the joint probability that f and c occur in the training set [194].

Information Gain quantifies the predictive power of features. If feature has a

higher Information Gain value, we say that it has more predictive power. We selected and used the features which have the highest Information Gain scores to train the classifiers.

3.4.2 Classifiers

We perform the classification using the various classifiers implemented in the Weka software package [84]. Weka is an open source collection of machine learning algorithms and has become the standard tool in the machine learning community. The classifiers used in the our experiments include decision trees (e.g., C4.5, Random Forest, etc.), rule generators (e.g, RIPPER, PART, etc.), logistic regression, radial basis function (RBF) networks, HyperPipes, multilayer perceptions, K Star, SMO (an algorithm for training a support vector classifier), Simple Logistic, and naïve bayes.

3.4.3 Classification Setup and Cross validation

We download legitimate web pages from the Standford WebBase project [162], which categorizes its data based on crawling time. To avoid the time factor influence, we only use the web interface provided by the project to fetch web pages on March 2011 the same crawling time as Webb Spam Corpus 2011. We requested 100,000 web pages and obtained about 53,000 legitimate web pages with HTTP status code equals 200 (called WebBase 2011). In the classification, we randomly choose the same amount of legitimate web pages as spam web pages to eliminate any prior probability influence. Therefore, the training dataset used in our experiments is consisting of 53,000 legitimate web pages and 53,000 spam web pages randomly selected from Webb Spam Corpus 2011.

Based on the methodology mentioned in Section 3.4.1, we retain about 5,000 features which have the high predictive power on the basis of information gain. In addition, we employed the machine learning classifiers previously mentioned using 10-fold cross-validation model. Cross validation is a technique for protecting against

over-fitting in a predictive model. Specifically, the data is randomly divided into k groups and the classifier is re-estimated k times, holding back a different group each time. The overall accuracy of the classifier is the mean accuracy of the k classifiers tested.

3.4.4 Result Analysis

To evaluate the performance of the classifiers, we adopt the F-measure and accuracy as evaluation metrics. Previous study shows that there are five algorithms which achieved the best performance in the web spam detection using HTTP header information [194]. The algorithms include C4.5, HyperPipers, Logistic regression, RBF network, and SVM. The results on Webb Spam Corpus 2006 and WebBase 2006 are shown in Table 23 [194]. For comparison, we list the performance results of those five algorithms on our new dataset in Table 24.

Table 23: Classifier performance results for Webb Spam Corpus 2006 and WebBase 2006.

Classifier	TP	FP	F-measure	Accuracy
C4.5	88.5%	4.6%	0.916	91.9%
HyperPipes	88.2%	0.4%	0.935	93.9%
Logistic Regression	88.2%	2.0%	0.927	93.1%
RBFNetwork	87.1%	0.8%	0.927	93.2%
SVM	89.4%	2.3%	0.933	93.6%

Table 24: Classifier performance results for Webb Spam Corpus 2011 and WebBase 2011.

Classifier	TP	FP	F-measure	Accuracy
C4.5	80.1%	0.0%	0.890	90.0%
HyperPipes	73.5%	0.0%	0.847	86.7%
Logistic Regression	80.0%	0.0%	0.889	89.9%
RBFNetwork	44.5%	36.1%	0.494	54.2%
SVM	80.1%	0.0%	0.889	90.0%

By comparing the results in two tables(Table 23 and Table 24), it shows that the overall performance of the five algorithms under-performs on new dataset (Webb

Spam Corpus 2011 and WebBase 2011). HyperPipes is no longer the best algorithm in terms of F-measure and accuracy. C4.5 algorithm which is one kind of decision tree algorithm outperforms others. RBFNetwork algorithm has surprising poor performance which has lowest TP rate and highest FP rate. FP rates for C4.5, HyperPipes, Logistic Regression, and SVM algorithm are nearly zero which means very few legitimate web pages have been mislabeled as spam. However, TP rates for those algorithms are all below 85%. The difference between two classification results indicates that spammers have evolved to avoid the detection based on HTTP header information over time.

Table 25: Top 10 features for Webb Spam Corpus 2011 and WebBase 2011.

Rank	Feature	Rank	Feature
1	*p3p_cp* =	6	*p3p_xmlcp* =
2	*set-cookie_gmt*	7	*link_type* =
3	*link_rel* =	8	*p3p_ind*
4	*p3p_policyref* =	9	*cache-control_no-cache*
5	*p3p_xml*	10	*x-powered-by_php*/5

Table 26: Top 10 features for Webb Spam Corpus 2006 and WebBase 2006.

Rank	Feature	Rank	Feature
1	*accept-ranges_bytes*	6	*expires_0000gmt*
2	*x-powered-by_php*/43	7	64.225.154.135
3	*x-powered-by_php*/4	8	*server_fedora*
4	*content-type_text/html*; *charset* = *utf*-8	9	*pragma_no-cache*
5	*content-type_text/html*; *charset* = *iso*-8859-1	10	*p3p_cp* =

To investigate the feature change over time, we rank the features based on information gain and list top 10 features in Table 25. Comparing with previous top 10 features shown in Table 26, we have the findings as follows: 1) new top 10 features are from the HTTP header fields including "P3P", "LINK", "SET-COOKIE", "CACHE-CONTROL", and "X-POWERED-BY", while the features in previous top 10 list are from the HTTP header fields including "ACCEPT-RANGES","X-POWERED-BY","CONTENT-TYPE","EXPIRES","IP address","SERVER","PRAGMA", and

"P3P"; 2) the features from the header "P3P" show high predictive power on new dataset; 3) only one feature "$p3p\_cp =$" remains in top 10 list.

In those popular HTTP headers, "P3P" header is designed to give users more control of their personal information when browsing. "SET-COOKIE" header stores an HTTP cookie. "LINK" header is used to express a typed relationship with another resource, where the relation type is defined by RFC 5988 [97, 199]. It shows that spammers have changed their attack strategy using new and advanced techniques such as cross-site scripting (XSS) [169] and cookie spoofing, which arise up a serious protection issue to user privacy on the Internet.

3.4.5 Computational Costs

Besides our evaluations on the effectiveness of HTTP session classification, we also consider the computational cost of HTTP session classification on new dataset since the timing requirements of the real-time spam detection system needs efficient classifier.

We perform timing experiments using the 5 classifiers mentioned in Section 3.4.4 to investigate the computational cost of HTTP session classification on Webb Spam Corpus 2011 and WebBase 2011. For each classifier, we compute the training time and per instance classification time to perform a stratified tenfold cross-validation evaluation. The experiments are conducted on a eight process (Intel Xeon 2.67 GHZ) system with 96 GB of memory, and the results for the five classifiers are shown in Table 27.

Table 27: Classifier training and per instance classification times.

Classifier	Training Time (s)	Classify Time Per Instance (s)
C4.5	727.75	0.0068
HyperPipes	10.85	0.0001
Logistic Regression	1,688.7	0.0159
RBFNetwork	275.44	0.0026
SVM	499.96	0.0047

Table 27 shows that HyperPipes is still the most efficient classifier in terms of training time (10.85 s). The training times of C4.5 (727.75 s), Logistic Regression (1,688.7 s), and SVM (499.96 s) are all more than one order of magnitude larger than the corresponding training time for HyperPipes. RBFNetwork still shows efficient in terms of training time (275.44 s) but it has poor performance in classification on new dataset.

3.4.6 Discussion

The finding of our classifications shows that the discriminative power of features in spam detection changes over time and the performance of classifiers differs between two datasets, which inspires our researchers to learn new strategies being adopted by spammers and possible countermeasures. Here, we listed several possible attacks from spammers and countermeasures for those attacks.

3.4.6.1 Possible attacks from spammers

One possible attack is to camouflage the links using short URLs and hidden links. Spammers take advantage of the popularity and fuzzy property of short URLs to lead vulnerable users to spam web pages. As social media are having an enormous growth, short URL spam becomes more serious problem. Also, hidden links behind the text or in the Javascript codes post real challenges to spam detection system. For example, the text for link is the URL to some websites you know well such as Facebook. However, the real destination redirected by hidden links may be a phishing website in the end.

Another possible attack is to embed the URL links in non-text medium such as image or video. It is also the reason why spam web pages show less text in our dataset. Spam URL link may appear in one or many images which is not covered by common text-based spam detection system. Moreover, spam URL links in images require the spam detection system to do image processing first that results in high cost and low

accuracy as well. Those image spam also are spreading in different channels such as user profile in social media and multimedia messages in mobile phones. Spreading spam in video streaming is similar as the network bandwidth is not the bottleneck any more today.

Third one is cross-site scripting (XSS) and cookie spoofing attacks. Spammers inject their malicious codes into web pages viewed by other users in cross-site scripting attack which may help spammers to bypass access controls. Browser cookie is a small piece of data storing the users' previous activity such as logging in or clicking particular buttons. After spammers obtain cookies, they could steal legitimate users' identities for spreading spam which could help them to earn clicks through the trust built among friends.

Of course, we cannot predict all possible attacks from spammers since the spamming strategies are always involving.

3.4.6.2 Countermeasures

To deal with attacks from spammers mentioned above, we may have the following countermeasures:

For camouflage attacks, we need to obtain the redirections and the destination URL through the short URL and hidden links. Meanwhile, other kinds of analysis such as behavior analysis could be used in social media. Spammers may have collective behavior pattern which differs from normal users. For example, spammers repeatedly send the same URL links to popular trending topics using a group of user accounts in Twitter.

For non-text spam attacks, we need to process those non-text media first to extract the embedded text out of them. But it becomes harder and harder to eliminate the noise in the process. Another way to defend against those attacks is simply to disable image showing and let the users make the decisions. However, it is still an open

problem to classify those non-text spam.

For cross-site scripting (XSS) and cookie spoofing attacks, we need to do source code checking and prevent back-doors in our browsers. Also, if possible, we should use SSL instead of normal HTTP requests. Meanwhile, disabling JavaScript may help you prevent those attacks with the cost of not being able to run normal JavaScript programs.

To sum up, due to evolving spamming strategies and growing Internet in terms of size and complexity, we need to spend more research efforts in exploring security vulnerabilities and preventing web spam.

3.5 Related Work

Webb et al. [193] introduced the first large-scale dataset – the Webb Spam Corpus which is a collection of approximately 330,000 web spam pages. It addressed the challenge of the lack of publicly available corpora in previous Web spam research [14, 21, 42, 57, 59]. Further, they conducted intensive experimental study of web spam through content and HTTP session analysis on it [196]. They categorized Web spam into five groups: Ad Farms, Parked Domains, Advertisements, Pornography, and Redirection. Besides, they performed HTTP session analysis and obtained several interesting findings. After that, Webb et al. [194] presented a predicative approach to Web spam classification using HTTP session information (i.e., hosting IP addresses and HTTP session headers). They found that HTTP session classifier effectively detected 88.2% of the Web spam pages with low a false positive rate 0.4%. Our work is to further experimental study on evolution of web spam through content and HTTP session analysis on new Webb Spam Corpus. By comparing the two large-scale datasets in different time ranges, we obtained the trend of Web spam and behavior changes of spammers. Also, we perform the classification experiments on Webb Spam Corpus 2011 and WebBase 2011 and evaluate the performance and computational cost

of classifiers.

Fetterly et al. [66] presented their work on a large-scale study of the evolution of web pages through measuring the rate and degree of web page changes over a significant period of time. They focused on statistical analysis on the degree of change of different classes of pages. Youngjoo Chung [46] studied the evolution and emergence of web spam in three-yearly large-scale of Japanese Web archives which contains 83 million links. His work focus on the evolution of web spam based on sizes, topics and host-names of link farms, including hijacked sites which are continuously attacked by spammers and spam link generators which will generate link to spam pages in the future. Irani et al. [98] studied the evolution of phishing email messages and they classified phishing messages into flash attacks and non-flash attacks and analyzed transitory features and pervasive features. In our work [186, 187], we also studied the evolution of web spam but there are two important ways which are different from his work: First, we focus on redirection techniques, HTTP session information and content not link farms. Second, the majority of the datasets we study on is in English language not in Japanese. It may have common features between them but our datasets are more representative than his dataset in terms of the popularity of web spam in English language.

In previous research, we proposed a social spam detection framework for social networks [184,185,188]. We studied three popular objects in social networks including profile, message, and web page objects. The classification of web page model shows promising results for associative learning. The classification results of web page model also improve the classification of other objects using the relationship between web page model and other models.

We collected new web spam corpus and studied the evolution of web spam. Our work addresses the lack of publicly available dataset for research and shows the trend of web spam in social media. In addition, we investigated the feature and performance

change in web page spam classification over time.

3.6 Conclusion

We introduced new large-scale web spam corpus – Webb Spam Corpus 2011 which is a collection of approximately 330,000 web spam pages. Adopting the automatic web spam collection method [193], we crawled the Internet through more than one million URL links in spam email messages during the time range between May 2010 and November 2010. In data cleansing of new dataset, we found that legitimate URL attacks by spammers are using more URLs in social media and search engine domains.

In addition to introducing new dataset, we also performed intensive study on Webb Spam Corpus 2011 through redirection, HTTP session analysis, and content. In the redirection analysis, we found that fewer redirections appear in the Webb Spam Corpus 2011 (about 70% source URLs returning no redirection). Another observation is Webb Spam Corpus 2011 has less 302 "Found" redirections and location redirection but it has more iFrame redirections. Also Webb Spam Corpus has 4% redirections which are other types of HTTP redirections. For most common host names in redirection chains, we obtained a interesting finding that social networks are used for hosting web profile spam and the widgets and plug-ins of social networks become convenient spamming traps to attract click traffic. Furthermore, we investigated the HTTP session information of Web spam in Webb Spam Corpus 2011. For hosting IP addresses, the percentages of IP addresses in ranges 63.* -69.* and 204.* -216.* have been reduced from 45.4% to 28.1% and from 38.6% to 21.7% respectively. For HTTP session headers, new header P3P appears in top 10 list and old header PRAGMA has been removed from the list. The most popular values for the header SERVER and CONTENT-LENGTH have changed from "microsoft-iis/6.0" to "Apache" and from $1,470$ to 77 respectively. Also we generated the distribution of content length of Web spam and found the content length of web spam text gradually becomes shorter.

Moreover, we analyzed duplications and syntax changes in Webb Spam Corpus 2011. 66.9% web spam pages in Webb Spam Corpus 2011 have the same HTML content as one of 122,618 unique web spam pages, which is much higher than the percentage (42%) in Webb Spam Corpus. Two datasets have very few percentage of exact content duplicates in common (2.0% for Webb Spam Corpus 2011 and 1.3% for Webb Spam Corpus). For content analysis, we listed the trend of top 20 most popular words in Webb Spam Corpus and top 10 words based on information gain to distinguish the two datasets. Also we compared n-gram (2-3) based on frequency in the two datasets.

Also, we have done classification comparison between Webb Spam Corpus 2011 and Webb Spam Corpus 2006 with respect to classifier performance, feature analysis, and computational cost three aspects. Poorer performance results of new experiments imply that spammers have evolved to avoid the detection based on HTTP header information over time. Different top 10 feature list also tells us that spammers have changed their attack strategy using new and advanced techniques. Computational cost computation shows HyperPipes classifier still has the highest efficiency but its accuracy dropped a lot in classification.

To sum up, we collected a new Webb Spam Corpus of approximately 330,000 web pages. We derive insights from this dataset as well as do an evolutionary study by intensive analysis and comparison between Webb Spam Corpus 2011 and Webb Spam Corpus 2006. Also we obtained lots of interesting findings between them.

CHAPTER IV

EVOLUTIONARY STUDY OF EMAIL SPAM

Email is used everyday as a method to communicate, both for individuals and businesses, but also as an information management tool [197]. What started primarily as a person-to-person communication medium has spread widely to one-to-many (e.g. mailing-lists) and many-to-one (e.g. forwarded traffic) communication medium [48]. As social media has grown dramatically, email also enhances the functionality provided by them. For instance, users are sometimes given pseudo-email addresses which can be used to receive email on the social network as well as email can sometimes be used to interact with the social network using specially crafted email addresses.

Spam is unsolicited and unrelated content sent to users, which most commonly is associated with email, but also applies to several different domains including instant messaging, websites, and Internet Telephony [49, 82, 89, 146, 157]. Spam degrades a user's experience as, by definition, it is an annoyance and gets in the way of users consuming non-spam content. In an extreme case, spam can be seen as a denial of information preventing user's from finding non-spam content.

In August 1998, Cranor et al. [50] described the rapidly growing onslaught of unwanted email and since then the volume of spam has grown even more as the amount of all email sent has grown exponentially. Constituting an annoyance, email spam has increased to as much as 90% today [129] from approximately 10% of overall mail volume in 1998, which results in an enormous burden on the thousands of email service providers (ESPs) and millions of end users on the Internet [73].

In addition to being on the receiving side of spam, ESPs need to invest in developing filters to combat the spammers and likewise spammers evolve to avoid spam

filters. The co-evolution nature of spammers and spam filters is an "arms-race", which has resulted in numerous publications employing adversarial strategies to tackle the spam problem [25, 44, 55]. Pu et al. [149] and Fawcett [62] developed techniques for characterization and measurement of email spam trends and researchers have also examined other types of spam including phishing [98] and Web spam [186]. In addition, Guerra et al. [79] compared the effectiveness of old and recent filters over old and recent spam to obtain spam trends on email spam dataset.

In this chapter, we investigate the trends of email spam in terms of content, topics, and sender-receiver network over 15 years by performing an evolutionary study on the Spam Archive dataset [9]. We aim to answer the question of whether the email spam business is dying. More concretely, we make the following contributions:

- First, we perform a long-term evolutionary study on a large email spam dataset, which includes statistical analysis, topic modeling and network analysis.
- Second, we demonstrate the changes of email spam over time with respect to contents and spammer behaviors.
- Lastly, we prove that email spam business is not dying but is becoming sophisticated.

The remainder of the chapter is organized as follows. We motivate the problem further in Section 4.1. Section 4.2 introduces the Spam Archive dataset used in our study. Section 4.3 presents the analysis performed on the dataset and findings derived from the results. Section 4.4 discusses the future of email spam business and the limitations of our study. We talk about related work in Section 4.5 and conclude the chapter in Section 4.6.

4.1 Motivation

The chapter is inspired by an article by Kaspersky labs [5] named "The dying business of email spam" [155], which stated that "Spam email is on the wane. And no one on God's green Earth is going to miss it". The conclusions were based on their annual report [78] citing that the share of spam in email traffic decreased steadily throughout 2012 to hit a five year low.

We are excited by the decline in the volume of email spam but it also raises the question as to whether the email spam business is dying and will continue to decline. Besides the volume change, we also consider the quality of email spam and the impact, which may be constituting a new trend of email spam business. For instance, spammers may post email spam in a more complicated way using spoofed email addresses and changing email relay servers. Those kind of email spam may slip away under the inspection of spam filters. Thus, it motivated us to investigate the evolution of email spam using advanced techniques such as topic modeling and network analysis. We try to find out the real trend of email spam business through email content, meta information such as headers, and sender-to-receiver network over a long period of time.

4.2 Data Collection

In this section, we introduce the Spam Archive dataset and show the overview of the dataset used in our study.

Spam Archive dataset [9] is collected by Bruce Guenter since early 1998 using honey-pot addresses. The project is still ongoing with monthly releases of new email spam. Since it provides a continuous long-term email spam data source from a consistent source, it is an excellent dataset for our investigation into spam trends. The volume of email messages received over the 15 years is shown in Figure 26, with the date on the x-axis and log-scale volume of email messages received per month on the

y-axis. From the figure we see that email spam volume grows steadily over time. For the spike of email spam during 2006, Bruce Guenter has attributed this to one of the spam traps having a wild-card address which received increasingly large amounts of spam which was subsequently disabled after 2006, since most of the spam was duplicates of other spam received.

Besides showing the trend of overall volume of email spam (shown in Figure 26), we also present the volume changes monthly for different years in Figure 27, with the month of the year on the x-axis and the log-scale volume of spam messages per month on the y-axis. It shows volume trends over the previous 15 years. The volume of email spam is not always increasing over time such as the email spam volume changes during 1999. Some years' volumes also shows fluctuations over time. For instance, during 2002, the volume first went up in May and decreased dramatically afterward until July. Several factors may have contributed to this change such as new strategies used by spammers (e.g. image spam is introduced in emails), improved spam filters (e.g. URL analysis tool is adopted) and even political influence from governments (e.g. Electronic Communications and Transactions Act, 2002 [2]). We investigate the details and potential reasons of these changes in more detail in the following sections.

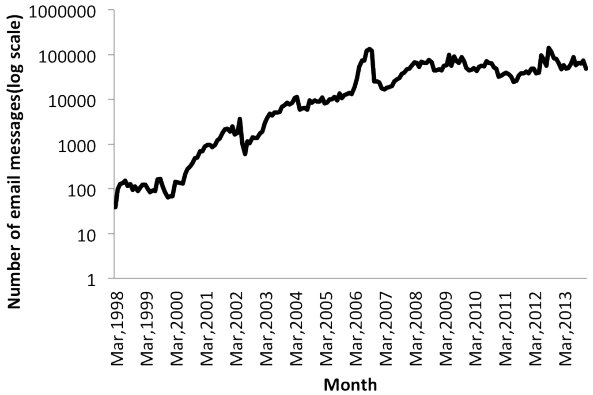


Figure 26: Number of email messages (per month) over time

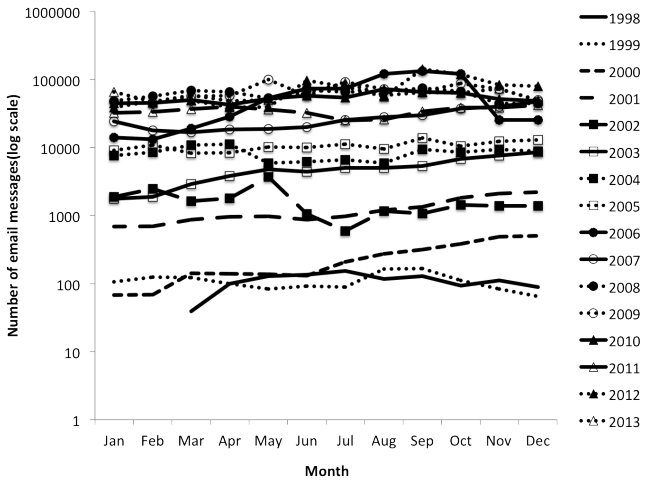


Figure 27: Number of email messages in month order for different years

4.3 Data Analysis

In this section, we start with content analysis of Spam Archive dataset, followed by topic modeling and network analysis.

4.3.1 Content Analysis

The two main types of email message content are "Text" and "Multipart". Messages in type "Text" are simple text messages while messages in type "Multipart" have parts arranged in a tree structure where the leaf nodes are any non-multipart content type and the non-leaf nodes are any of a variety of multipart types [1]. To have a better sense of the distribution of main types in email spam, we show the main type distribution in different years in Figure 28.

Figure 28 demonstrates that the distribution of two main types in our dataset changed over time. For instance, before 2003, more email spam had the message format in the main type "Text". After that, the two main types almost occupied the same percentage until 2010. The new trend is that email spam is using more messages in main type "Text" (e.g. the percentage of email spam in main type "Text" is over 80% for the half year of 2013).

Next thing we are interested in is the embedded items in email spam such as

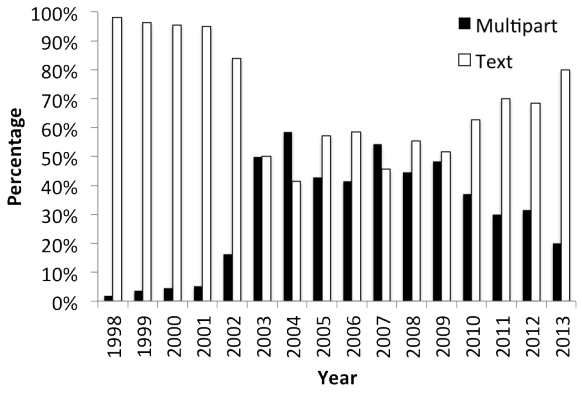


Figure 28: The distribution of main types of message content

HTML web page, images, and URL links. After scanning all email spam in our dataset, we present the distribution of embedded items in email spam over time in Figure 29.

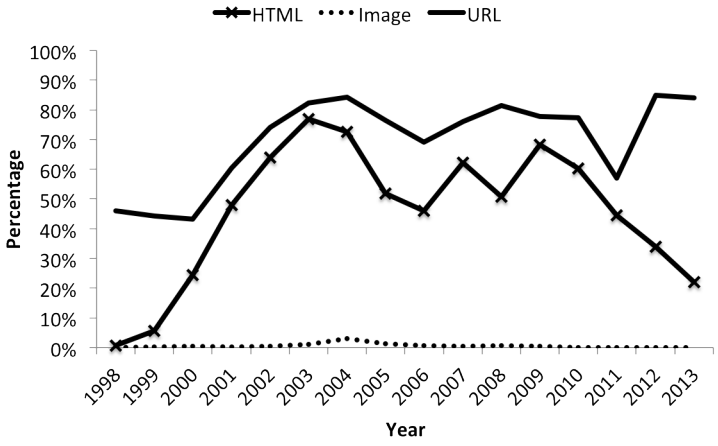


Figure 29: The distribution of embedded items in email spam over time

Figure 29 shows that low percentage of email spam, which was always less than 5% in our dataset, contained image attachments. On the contrary, more email spam had embedded HTML web pages and URL links. But the percentages of email spam containing HTML web pages and URL links changed dramatically over time. Several peaks and valleys appeared over 15 years in the Figure 29. For instance, HTML pages had peaks in 2003, 2007, and 2009 and valleys in 2006 and 2008. While for URL links,

peaks appeared in 2004, 2008 and 2012 and valleys appeared in 2006 and 2011. Since HTML page normally carries URL links, they should have similar fluctuations along the time. However, we observe that an exception occurred after 2011. The percentage of email spam containing HTML web pages decreased suddenly after 2009. While the percentage of email spam containing URL links dropped down along with HTML web pages until 2011 and it increased sharply afterwards. One possible reason is that more URL camouflage techniques, which are quite efficient in avoiding spam filters, appeared such as shortened URLs and hidden URLs in recent years. To investigate further the trend of URL links, we aggregate all URL links on a yearly basis for email spam that contain URL links and show the cumulative distribution of URL links in email spam in Figure 30 (1998 – 2012). The data of 2013 is not included due to that it only contains half year data.

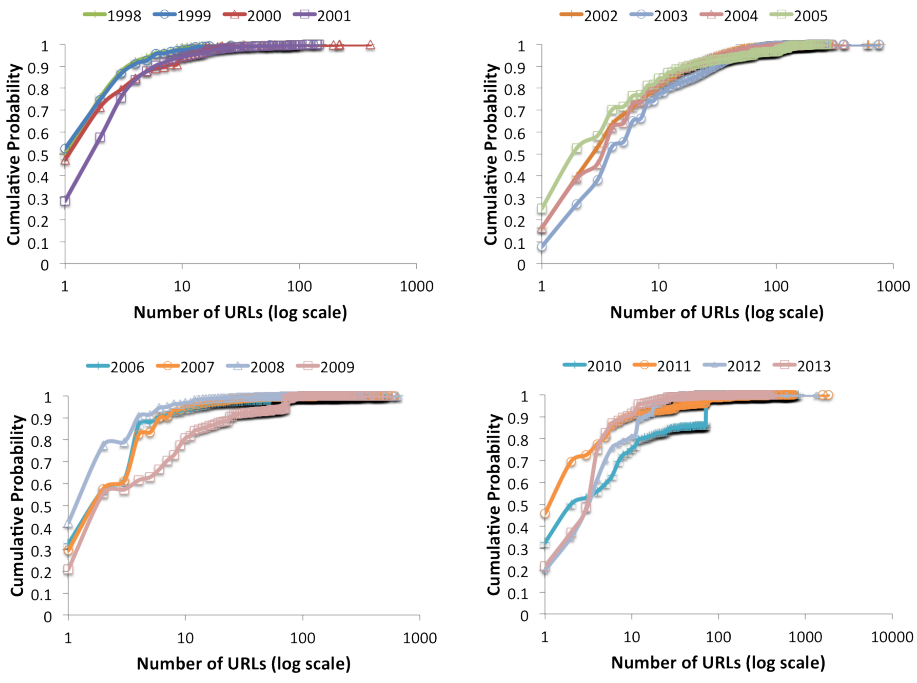


Figure 30: Cumulative distribution of URL links in different years

Figure 30 shows the number of URL links for the majority of email spam is below 10. Only a small portion of email spam have more than 1,000 URL links which may

be embedded in different depths of email messages. Even though the densities of URL links in email spam changed variously, email spam contained more and more URL links over time.

Table 28: List of top-10 n-grams every 5 years on a monthly basis (n ranges from 1 to 3)

textbfJune, 2003	**June, 2008**	**June, 2013**
$\langle N \rangle$	$\langle N \rangle$	$\langle N \rangle$
click	euro	important
email	dass	garden
information	online	class
bait	http	email
mail	mail	media
free	original	screen
message	super	dark
work	time	right
please	active	registration
$\langle N \rangle$ $\langle N \rangle$	$\langle N \rangle$ $\langle N \rangle$	$\langle N \rangle$ $\langle N \rangle$
email bait	euro euro	garden $\langle N \rangle$
august $\langle N \rangle$	super active	$\langle N \rangle$ garden
market information	active euro	media screen
world leader	tabs doses	important media
auction records	kinder dass	important important
remove email	autopilot dass	dark skin
reply message	original stress	screen class
link work	stress angst	class important
leader market	angst dass	rights reserved
$\langle N \rangle$ $\langle N \rangle$ $\langle N \rangle$	$\langle N \rangle$ $\langle N \rangle$ $\langle N \rangle$	garden $\langle N \rangle$ garden
world leader market	euro euro euro	$\langle N \rangle$ garden $\langle N \rangle$
leader market information	active euro euro	$\langle N \rangle$ $\langle N \rangle$ $\langle N \rangle$
case link work	super active euro	important media screen
demander plus figurer	dass kinder dass	media screen class
allow mail removed	dass autopilot dass	important important media
removed thank operation	dass dass kinder	class important media
modifier sera effective	autopilot dass dass	screen class important
message modifier sera	original stress angst	limited become member
effective coop demander	kinder dass super	become member soon

In addition to looking into embedded items, we also investigate the top n-grams in email spam over time. The tool we used for obtaining n-grams of email spam is

Perl's module Text::Ngrams [8]. First, we need to clean our dataset by filtering out stop words and striping out HTML tags. And then we calculate top-10 n-grams (n ranges from 1 to 3) on a monthly basis over 15 years. Due to space limit, we only list the top-10 n-grams starting from June 2003 to June 2013, which is shown in Table 28.

In Table 28, $\langle N \rangle$ denotes any number sequence. Top-10 n-grams set contained different words or word sequences along the time, showing different topics as well. For instance, The n-grams set in June 2003 was about marketing and market leaders leading people to click external URL links. The n-grams set in June 2008 was about DASS (Defensive Aids Sub System) [3] which is a fighter system from European countries. After checking the original email, it is a trap news or game to attract the email receivers to enter into. The n-grams set in June 2013 was more related to new media announcement and membership registration. Moreover, the differences indicate the topic drift in email spam over time (e.g. from fake advertising to fake registration services). To learn more about the topic drift of email spam, we will apply topic modeling on the dataset next.

4.3.2 Topic Modeling

Topic modeling is defined as a technique that looks for patterns in the use of words and it is an attempt to inject semantic meaning into vocabulary, in which a "topic" consists of a cluster of words that frequently occur together [135]. The tool we used in our topic modeling is a machine learning toolkit for language named "MALLET" [135]. It provides an efficient way to build up topic models based on Latent Dirichlet Allocation model (LDA) [28].

To simplify the illustration, we set up the number of topics to 10 in the data processing. After the calculation, we obtain the word (also called term) lists associated with topics and topic composition for different months over time, which is shown in Table 29 and Figure 31.

Table 29: List of topics and associated terms

Topic Name	Samples of Most Related Terms
Account Information	email important pass check account address information
Order Information	click message privacy online policy information address view order receive required
Business News	click information price free professional time link business work
Sales News	price life money make time today offer year online real world women retail deal credit people
Adult Product	world price penis back people product degree patch life make great experience enlarge
Software Product	price professional click software company copy softwares read suite online site office
Official News	united states world state national city view university government international people population
Free Product	online pills price click quality save products email item prices service offer free
Medical Product	generic save price time products medications order pharmacy home service product
Newsletter	mail click email privacy newsletter message receive view offers link receiving policy subscribed

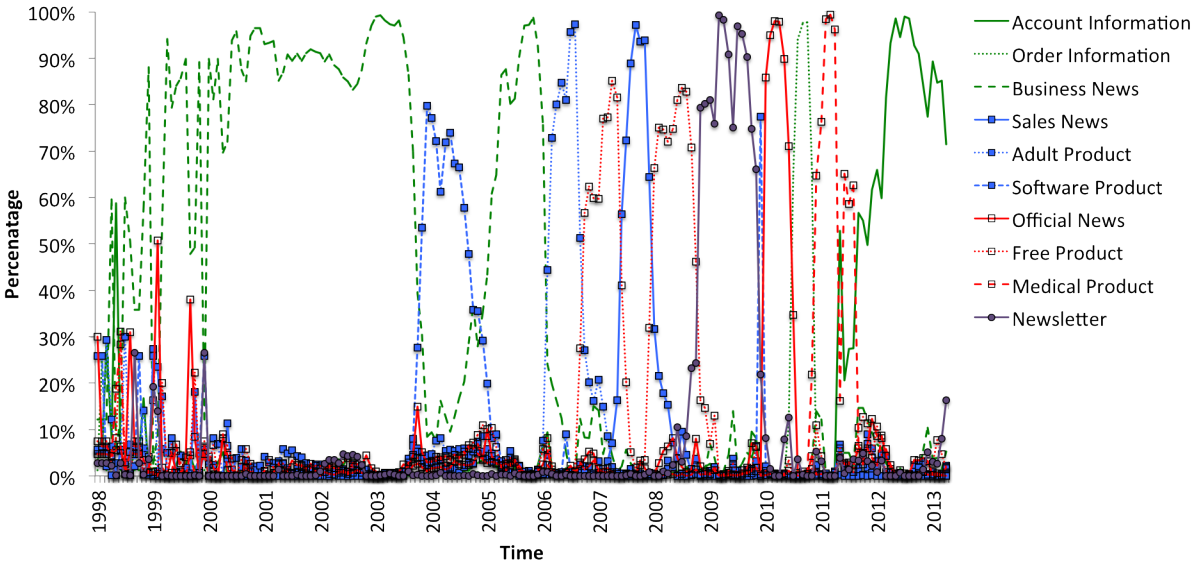


Figure 31: Topic drift in time order (time unit: month)

In Table 29, it shows the topic name and the samples of most related terms. After the topic modeling, we only have the word or term clusters for each topic which has not been labeled. Based on associated terms with each topic and experience with email spam, we label the topics as "Account Information", "Order Information", "Business News", "Sales News", "Adult Product", "Software Product", "Official News", "Free Product", "Medical Product", and "Newsletter" separately. Due to the space limit, we just list sample of most related terms for each topic in Table 29.

Figure 31 shows the topic drift in our dataset. We observe that the popular topics drifted along the time. Before 2004, the topic "Business News" was the most popular topic in email spam. After that, the most popular topic changed more frequently than before. First, the most popular topic changed to "Software Product" for around a year. And then it changed back to the topic "Business News" again. And later on, the most popular topic changes happened in the following order: "Adult Product", "Free Product", "Sales News", "Free Product", "Newsletter", "Official News", "Order Information", "Medical Product", and "Account Information". For each topic, it contains certain features that are attractive to certain group of users. For instance, topic "Free Product" is more attractive to users who like free stuff. Topic "Medical Product" is more attractive to users who need medical service or special medical products. Topic "Sales News" and "Order Information" are more attractive to users who like shopping. Meanwhile, as social media have interfaces with email systems normally and gain increasing popularity, email spam which have the content related to social media are growing rapidly. For instance, by investigating the content of email messages which belonging to the recent most popular topic "Account Information", we observe that a lot of email spam have associations with social media. One example is that social media account registration email spam which contains spam URLs that camouflaged as confirmation URL links. Another example is social media account notifications. For example, it informs you that your account has been changed by

someone and needs immediate action to reset the password, followed by the spam URL links. Thus, one possible reason why the topic "Account information" becomes popular is that a lot of spammers try to impersonate the support team of social media to steal sensitive information, such as credential and credit information, or lead users to spam or phishing web pages for further actions.

4.3.3 Network Analysis

Besides content analysis and topic modeling, we also try to find out the sending behavior changes of spammers over time through analyzing the routing network between sender and receiver. Before entering into the detail of network analysis, we will talk about data processing and some findings during the process.

For the data processing, we need to process the headers of email message to obtain the information about routing between sender and receiver. The headers which are related to the routing info are "From", "To", "CC", "BCC" and "Received". The header "From" and "To" provide the sender and receiver email addresses. The header "CC" and "BCC" show the recipient lists in carbon copy and blind carbon copy mode. The header "Received" contains routing information from sender and receiver. First, we look into the headers "From" and "To" and intend to use them to extract the sender-to-receiver network. However, the fact is that we cannot use them in our study since most of the messages in the dataset contain forged "From" headers in one form or another, which is also mentioned in the Spam Archive dataset homepage. Although "From" header should not be trusted, we still extract top-10 domains from the "From" header to find out what are those popular domains used by spammers to set up social engineering traps for users. It is hard for users to recognize fake senders based on senders' email address especially when the email address is belonging to the domains they trust. The list of top-10 domains is shown in Table 30.

Table 30: List of top-10 domains

1998	**1999**	**2000**	**2001**
hotmail.com	yahoo.com	yahoo.com	hotmail.com
yahoo.com	hotmail.com	hotmail.com	yahoo.com
msn.com	aol.com	earthlink.net	excite.com
usa.net	usa.net	aol.com	msn.com
earthlink.net	ibm.net	usa.net	aol.com
worldnet.att.net	msn.com	excite.com	btamail.net.cn
aol.com	iname.com	mail.com	earthlink.net
mailexcite.com	hotbot.com	bigfoot.com	mail.com
juno.com	bigfoot.com	email.com	pacbell.net
prodigy.com	mailcity.com	postmark.net	mail.ru
2002	**2003**	**2004**	**2005**
yahoo.com	yahoo.com	yahoo.com	yahoo.com
hotmail.com	hotmail.com	hotmail.com	hotmail.com
aol.com	aol.com	msn.com	msn.com
msn.com	msn.com	yahoo.co.kr	yahoo.co.kr
excite.com	artauction.net	aol.com	gmail.com
link2buy.com	earthlink.net	attbi.com	yahoo.co.jp
eudoramail.com	excite.com	yahoo.co.jp	163.com
flashmail.com	artaddiction.com	excite.com	msa.hinet.net
netscape.net	juno.com	seznam.cz	mail.com
btamail.net.cn	artists-server.com	netscape.net	126.com
2006	**2007**	**2008**	**2009**
yahoo.co.jp	yahoo.com	dyndns.org	dyndns.org
hotmail.com	dyndns.org	yahoo.com	homeip.net
mail.ru	hotmail.com	adelphia.com	lists.untroubled.org
0451.com	yahoo.co.jp	hotmail.com	gmail.com
em.ca	paran.com	gmail.com	hotmail.com
yahoo.com	gmail.com	wikipedia.org	yahoo.com
0733.com	163.com	earthlink.net	untroubled.org
aol.com	msn.com	att.net	ezmlm.org
infoseek.jp	msa.hinet.net	163.com	em.ca
msn.com	so-net.ne.jp	cox.net	mail.ru
2010	**2011**	**2012**	**2013**
dyndns.org	yahoo.com	yahoo.com	yahoo.co.jp
yahoo.com	dyndns.org	garden.md	li-brooz.jp
homeip.net	ymail.com	yahoo.co.jp	yahoo.com
untroubled.org	gmail.com	ageha.cc	mixi1mega.biz
lists.untroubled.org	mail.ru	peach.6060.jp	netstar-inc.co.uk
ezmlm.org	msn.com	ts5558.com	garden.md
em.ca	bk.ru	momoiro.cc	for-dear-2013.mobi
comcast.net	qip.ru	koikoilkoii.com	wakuwaku06.info
gmail.com	list.ru	wakuwaku-happy.net	greemmix.info
pfizer.com	aol.com	get-c.com	docomo.ne.jp

From Table 29, we observe that several popular email domains are used by spammers such as "yahoo.com", "hotmail.com", "msn.com", and "gmail.com". Also some top domains are related to receiver domains such as "untroubled.org" and "dyndns.org". It reveals that spammers were camouflaging themselves coming from the same domains as the users' domains. In addition, some domains in the top-10 list are from countries outside US such as "163.com" which is the largest email service domain in China. In 2013, the top domains list contains more special domains such as ".biz" which is intended for registration of domains to be used by businesses and ".mobi" which is used by mobile devices for accessing Internet resources via the Mobile Web. It indicates that spammers were spoofing the sender addresses targeting business and mobile users. Meanwhile, it proves that spammers recognize the trend of information flow in the Internet and evolve to take advantage of the trending.

Next, we investigate the header "CC" and "BCC" in email message to know whether spammers use those functions to spread email spam. The trends of "CC" and "BCC" are shown in Figure 32.

Figure 32 shows that spammers used more "CC" and "BCC" in the early years (1999-2004) and less in the recent years (2011-2013). One possible reason is that most spam filters have taken the number of "CC" and "BCC" as important features

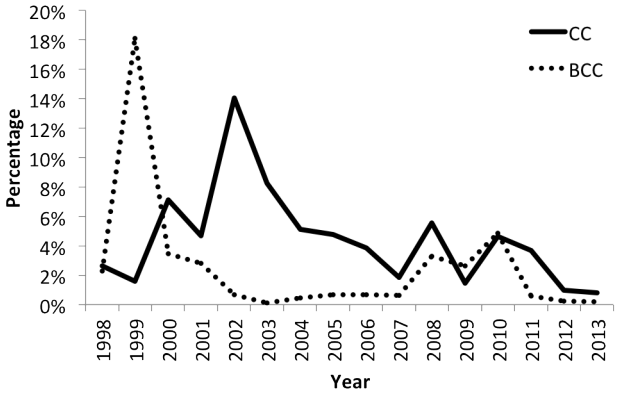


Figure 32: Cc and Bcc trends

to detect spam. Meanwhile, people become alert to email message which contains a long recipient list in the header "CC" and "BCC" so that this type of email spam lost markets gradually.

Based on observations above, we realize that the header "From", "To", "CC", and "BCC" are not helpful in extracting routing network from email spam. To better understanding the changes in terms of spammers' behaviors, we still need to find a way to extract the real sender and the routing information. The study of header "Received", which is much harder to be forged, provides us the routing information such as hops' IP addresses between sender and receiver. Therefore, we will use the header "Received" to extract sender-to-receiver IP routing information and construct routing network. The tool we used in extraction is the email module in Python [7] and the network analysis tool is the open source network visualization software Gephi [4].

During the process of extracting networks, we also collect two extra features: average hops between sender and receiver and the Geolocation distribution of sender IP addresses. The list of average hops and the Geo-location distribution of sender IP addresses over time are shown in Figure 33 and Figure 34 respectively.

Figure 33 presents the trend of average hops between sender and receiver. We observe that the number of hops was increasing over time. For instance, the average hops for 1998 was only two while it became almost eight in 2013. One possible reason

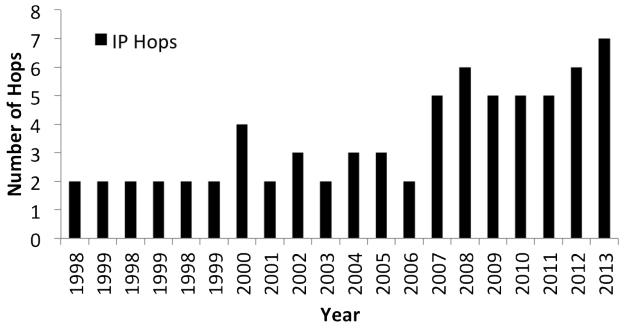


Figure 33: Average hops between sender and receiver

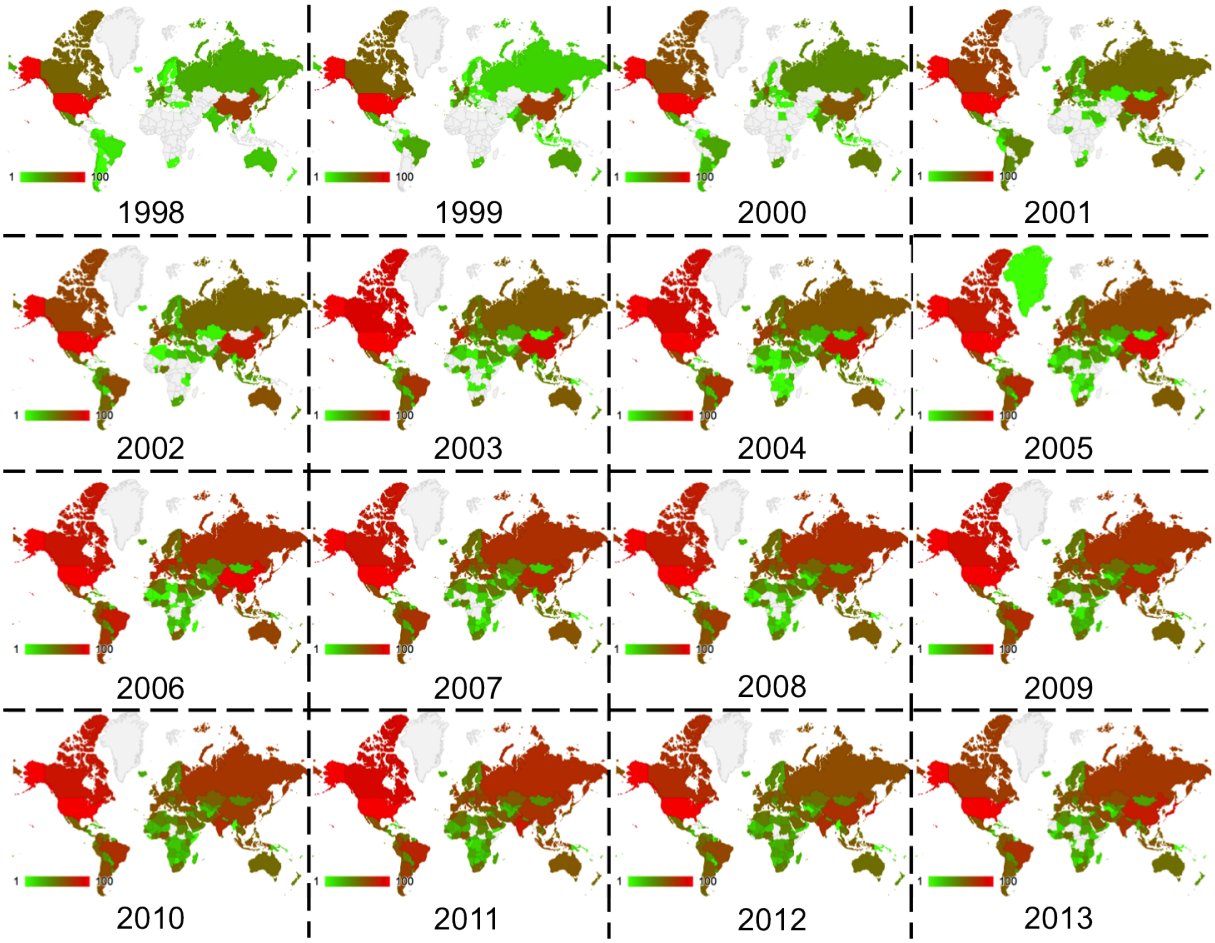


Figure 34: Geo-location distribution of senders' IP addresses every two years (in log scale and normalized)

is that it increased the cost for spam filters to detect or trace back the senders of email spam as spammers used more hops through intermediate proxies. It also indicates that the sender-to-receiver network becomes more complicated.

Figure 34 shows the Geo-location distribution of senders' IP addresses over time. Due to space limit, we only present the Geo-location maps every two years based on the normalized number of IP addresses coming from different countries. We use the GeoIP service provided by MaxMind [6] to do the mapping between IP address and Geo-location. Also, we employ Google Geo Chart APIs [10] to implement the map drawing. The number of IP addresses from different countries has been put into log scale and then normalized into the same range from 1 to 100. Also we use

green color to label countries who had the fewest sender IP address and red color to label countries who had the more sender IP addresses. White color means that no sender IP address came from the country. Observing the maps, we have the following findings in our dataset: 1) the sender IP addresses almost come from all over the world; 2) United States has the largest number of sender IP addresses along the past fifteen years; 3) Besides United States, the distribution of sender IP addresses shows dynamic changes over time. For instance, the number of sender IP addresses coming from China kept increasing until 2007 and grew again in 2013. Also, some countries had sudden increase of sender IP addresses in particular years. For example, Canada and France had sudden increase in 2003. India had sudden increase in 2011. And Japan had sudden increase in 2013. It indicates that spammers used global email service servers and also kept changing the traffic from different countries.

Next, we extract the networks from our dataset for each year and use three major metrics to measure the complexity of them. The three metrics are network diameter (the longest of all the calculated shortest paths in a network), average degree (average number of edges connected to or from one node), and average clustering coefficient (a measure of degree to which nodes in a network tend to cluster together). The result of measurement is shown in Figure 35. Since the data for 1998 and 2013 does not cover the whole year, we only list the results from 1999 to 2012.

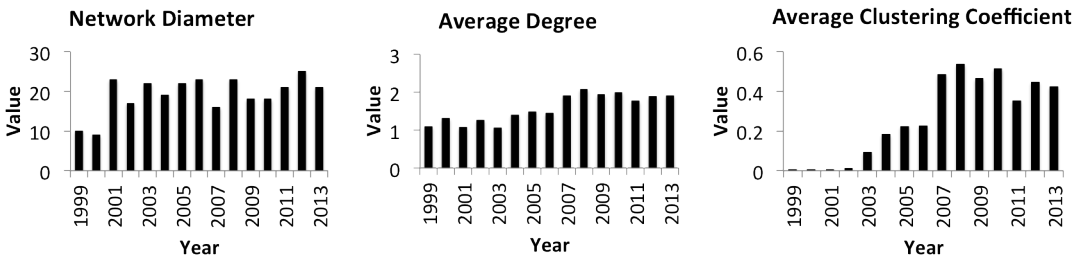


Figure 35: The comparison of three metrics from 1999 to 2012

Figure 35 shows the three metrics comparison from 1999 to 2012. The values of them have the increasing trend overall but fluctuations existed along the time.

Network diameter became more stable after 2007 and it is the same for the metrics average degree and average clustering coefficient. Those metrics kept staying at high value in terms of complexity of network.

For the purpose of better visualization, we remove those nodes whose degree is lower than certain threshold. And also due to the space limit, we only present the network graph every five years (1998, 2003, 2008, and 2013) in Figure 36. For 1998 and 2003, we keep the nodes whose degree is greater than 3. While for 2008 and 2013, we keep the nodes whose degree is greater than 10. The reason is that too many node overlaps occur if we choose the threshold 3 for 2008 and 2013.

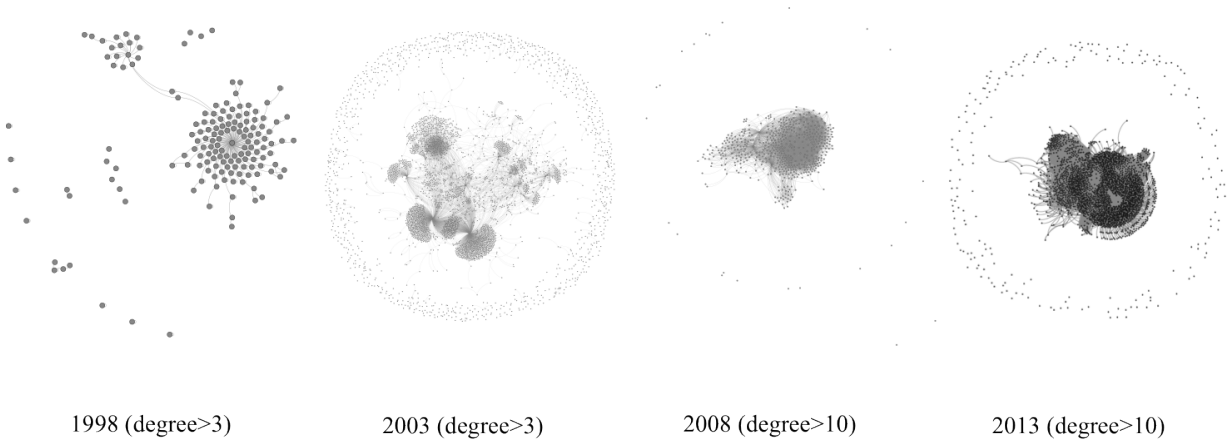


Figure 36: Sender-to-receiver routing networks every five years from 1998 to 2013

Figure 36 shows the sender-to-receiver routing network based on the IP addresses extracted from email header "Received". We observe that the complexity of graph increases explicitly along the time. For 2013, even that it only contains half year data, the routing network has already shown much more complicated than the routing network in 2008.

4.4 Discussion

Our large-scale evolutionary study on email spam dataset in a long period of time shows the trend of email spam business. Although the volume of email spam had a

slight drop in recent years, we cannot conclude that email spam business is dying and email spam filters have won the battle against spammers. Through intensive analysis including content analysis, topic modeling and network analysis, we demonstrated that the battle is still ongoing and even worse since spammers become more sophisticated and capricious. Moreover, our study still have the following limitations and future work to do.

The dataset we used does not cover all the email spam over the fifteen years, which may influence the accuracy of our results, especially for the portion in the early years such as 1998-2000 that contains small number of email spam. Also, the bait email addresses used in data collection may cause some biases in the dataset. For example, the domain of the email address may result in that spammers forge their email addresses to the same domain.

Besides the limitation on dataset, we also have limitation on our analysis. In the topic modeling analysis, we set up the number of topics to 10 that may influence the result of topic modeling . If we change the number of topics to larger value, the result may be more accurate and fine-grained. But it should not conflict with our conclusion that the topic drift occurs frequently over time. We will take the fine-grained analysis as future work. Additionally, in the network analysis, we used the study of the header "Received" to extract sender-to-receiver network. But we cannot guarantee that no forged information exists in the header "Received". Spammers also have some techniques to spoof the header "Received" but the portion of forged headers is low since it costs spammers a lot and has certain strict requirements to meet. We will also look into the further validation work in the future.

4.5 Related Work

Our work mainly involves three lines of research work: email spam detection, analysis approach on email data, and evolutionary study of spam.

Email spam detection has been studied by lots of researchers in different directions. For instance, Carreras et al. [36] applied boosting trees to filter out email spam. Wang et al. [192] used heuristic feature selection techniques to improve the performance of email spam filtering. Chan et al. [41] co-trained with a single natural feature set in email classification. Liu et al. [125] adopted multi-field learning for email spam classification. Sculley et al. [159] used relaxed online SVMs for email spam filtering. Besides those machine learning techniques, more researchers tried other kinds of detection methods. Attenberg et al. [17] introduced collaborative email spam filtering with the hashing trick. Balakumar et al. [18] offered ontology based classification of email. Dasgupta et al. [56] combined similarity graphs to enhance email spam filtering. Jung et al. [105] used DNS black lists and spam traffic to detect email spam. Ramachandran et al. [154] filtered email spam with behavioral blacklisting. Clayton et al. applied extrusion detection in stopping email spam by observing distinctive email traffic patterns. Xie et al. [202] provided an effective defense approach against email spam laundering. Additionally, researchers also have used email spam to help detecting other types of spam. For instance, Zhuang et al. [210] developed an approach to map botnet membership using traces of spam email. Webb et al. [193] identified an interesting link between email spam and Web spam and used it to extract large Web spam samples from the Web. Wang et al. [185] demonstrated the relationship among different formats of social spam including user profile spam, message spam and Web spam, in which message spam contain email spam.

Analysis approach on email data is another focus of researchers. Bird et al. [26] constructed social networks of email correspondents to address some interesting questions such as the social status of different types of participants and the relationship of email activity and other activities. McCallum et al. [134] illustrated experimental study on Enron and academic email to discover topic and role in social networks

from emails, in which the model builds on Latent Dirichlet Allocation (LDA) and the Author-Topic (AT) model. Culotta et al. [52] presented an end-to-end system that extracts a user's social network and its members contact information given the user's email inbox.

Research work on evolutionary study of spam is close to this chapter. Pu et al. [149] presented a study on dataset collected from Spam Archive and focused on two evolutionary trends: extinction and existence. Irani et al. [98] studied the evolution of phishing email messages and classified them into two groups: flash attacks and non-flash attacks. Wang et al. [186] compared two large Web spam corpus: Webb spam corpus 2006 and Webb spam corpus 2011 and shown the trending of Web spam. Chung et al. [46] and Fetterly et al. [66] also have done intensive study on evolution of web spam. Guerra et al. [79] investigated how the popularity of spam construction techniques changes when filters start to detect them and determined automatically techniques that seemed more resistant than others.

The evolution of spamming techniques shows the increasing sophistication of spammers. Our work focuses on tactics changes of email spam over time and inspires more researchers to work on email spam detection collaboratively.

4.6 Conclusion

We introduced a long-term evolutionary study on large scale email spam corpus – Spam Archive dataset, which contains over 5 million email messages from 1998 to 2013. Besides content analysis of email spam including n-grams analysis, we adopted topic modeling and network analysis techniques to investigate topic drift and increasing complexity of sending behaviors of spammers.

In the topic modeling, we clustered our dataset based on LDA model and categorized them into ten topics: "Account Information", "Order Information", "Business News", "Sales News", "Adult Product", "Software Product", "Official News", "Free

Product", "Medical Product", and "Newsletter" based on the most related terms associated. The result shows spammers changed topics over time and also made the topics attractive to users. In the network analysis, we presented social engineering attacks from spammers by observing senders' domains. After studying the header "Received", we extracted sender IP addresses and the sender-to-receiver routing networks from the dataset. The Geolocation distribution of senders' IP addresses shows that spammers employed the servers all over the world and dynamically switched locations among different countries. Moreover, we chose three metrics: network diameter, average degree, and average clustering coefficient to measure the complexity of routing networks, showing that the sending behaviors of spammers are becoming more complicated and harder to track.

To sum up, email spam business is becoming more sophisticated along the time and the spammers behind it evolve into more capricious in the ongoing battle with spam filters.

CHAPTER V

CLICK TRAFFIC ANALYSIS OF SHORT URL FILTERING ON TWITTER

Twitter, a popular social network, has over 400 million members and it allows them to post 140-character tweets (messages) to their network of followers. Given the limited length of a tweet, URL shorteners have quickly become the de facto method to share links on Twitter [77]. Twitter, due to its large audience and information reach, attracts more and more spammers [22,23,69,77,164,173,183,205]. Even though spammers have limited flexibility with the 140-character limit for a tweet, they utilize URL shorteners to camouflage their spam links [43, 71, 142, 166]. This enables spammers to hide the true domain of the URL, thereby might prevent Twitter from effectively applying blacklists to filter out such spam.

The popular URL shortener websites such as Bit.ly (henceforth referred to as Bitly) provide interfaces that allow users to convert long URLs into short URLs [15, 130]. After receiving a long URL, the services typically use a hash function to map the long URL to a short string of alphanumeric characters, which is then appended to the domain name of the shortener and returned as the short URL. For instance, the long URL `http://www.google.com` might be shortened as `http://bit.ly/olDmsz`. The hash function takes into account several factors, such as whether the long URL has already been mapped to a short URL. In this case, the shorteners typically return the existing short URL rather than generating a new one for the input long URL.

In this chapter, we perform an analysis on short URL spam by investigating their click traffic with the following goals. First, we aim to determine the feasibility of efficiently collecting the click traffic of short URLs. This is important because a social

network typically contains a massive number of short URLs and an efficient mechanism is needed to collect their click traffic. Second, we aim to discover significant patterns in the click traffic of a given set of spam short URLs. Third, we aim to determine the feasibility of detecting short URL spam effectively. This is particularly important because spam can lead to loss and damage [104, 185].

The highlights of our work can be summarized as follows:

- We generate a large-scale click traffic dataset for short URL spam;
- We obtain several findings about short URL spam through an in-depth analysis of creators and click sources of short URLs;
- We demonstrate the feasibility of detecting short URL spam by classification based on the click traffic features.

The remainder of the chapter is organized as follows. We motivate the problem further in Section 5.1. Section 5.2 introduces the approach developed for collecting the short URLs and the datasets used in the experiments. Section 5.3 provides the results of the statistical analysis of the short URLs. Section 5.4 describes our approach of classifying short URLs based on their click traffic features. Section 5.5 presents the evaluation metrics and classification results using different classifiers. Section 5.6 discusses the limitations and challenges of our approach. We survey the related work in Section 5.7 and conclude the chapter in Section 5.8.

5.1 Motivation

Existing studies have focused on URL spam detection and revolved primarily around blacklists and domain reputation. Blacklists are typically built for previously-classified URLs, domains, or IP addresses, and incoming URLs are simply checked against them [75]. These techniques do not work effectively when spammers employ short URLs. This is because blacklists based on domains and IP addresses incorrectly flag

the short URL generated by the URL shortening service instead of the long, malicious URL behind by the short URL, and furthermore, spammers generate a new short URL as soon as the previous one is blacklisted. One solution to this problem might be to resolve each shortened URL and fetch the web page associated with it. Previous studies [64,144] on web page content classification have shown high accuracy, however these techniques, although highly accurate, result in high classification cost and incur significant delay due to the fact that they need to download the content. Additionally, these techniques do not work for some malware customized web pages that are capable of dynamically changing content to confuse the content-based spam filters.

Similar to traffic analysis approach used in network anomaly detection [115], we aim to investigate short URL click traffic in order to detect patterns for short URL spam. Also, Las-Casas et al. [117] have efficiently used network metrics to detect spammers at the source network instead of content. In this chapter, we assume that spammers propagate spam URL in different way from legitimate users and that it should be less probable for people choosing to click spam URL. Next, we describe how we obtained the short URLs and their click traffic using public APIs.

5.2 Data Collection

In this section, we first describe the approach used for data collection and the properties of the datasets. And then, we discuss the ground truth labeling of the datasets.

5.2.1 Collection Approach

To collect data, we use two APIs: Twitter APIs [176] and Bitly APIs [201]. Twitter APIs provide two types of objects: a user profile and a tweet. We extract URL links from the tweets and filter out all the short URLs. Bitly APIs provide four major types of meta-data for each short URL: *info*, *clicks*, *countries*, and *referrers*. *Info* contains the properties of the short URL, such as the long URL behind the short

URL. *Clicks* contains the total amount of clicks for the short URL. *Countries* and *referrers* record the number of clicks from various countries and referrers, respectively. Here, "referrers" correspond to the applications or web pages that contain the short URLs.

5.2.2 Datasets

The details of our datasets are as follows:

Twitter Dataset: We collected data of over 900,000 Twitter users, about over 2.4 million tweets, fetching any links in the tweets. The tweets were gathered by querying the top trending topics every minute and they represent about 600 topics over the span of November 2009 to February 2010. Twitter users or tweets marked as suspended or removed due to terms of service violations are explicitly marked as spam in the dataset. There are over 26,000 such users and 138,000 such tweets.

We extracted all the short URLs from the Twitter dataset and then ranked the short URL providers based on the total number of URLs created by the providers. The result is shown in Table 31.

Table 31: Top-10 short URL providers in the Twitter dataset

Short URL Provider	*Count*
Bit.ly	641,423
t.co	129,677
Tiny.com	62,488
Ow.ly	42,838
Is.gd	14,664
Goo.gl	13,122
j.mp	8,963
Su.pr	3,764
Twurl.nl	2,807
Migre.me	2,788

We observed that Bitly generated the majority of the short URLs in the dataset, achieving about 57% of the total URLs. As Bitly also has public APIs that enabled

us to download click traffic meta-data of the short URLs, we decide to focus on the Bitly short URLs and generate the Bitly dataset as follows.

Bitly Dataset: We extracted all the Bitly short URLs from the Twitter dataset and fetched their click traffic using the Bitly APIs. The click traffic dataset consists of four types of meta-data:

- **Basic information** containing five attributes: id (identification number of the short URL), url (short URL address), long_url (URL that the short URL points to), title (title of the web page), created_by (creator of the short URL);
- **Number of user-level and global clicks** containing three attributes: url_id (identification number of the short URL), user_clicks (total number of clicks received), global_clicks (total number of clicks received globally[1]);
- **Country click distribution** containing three attributes: url_id (identification number of the short URL), countries (list of countries), and clicks (number of clicks from the countries);
- and **Referrer click distribution** containing three attributes: url_id (identification number of the short URL), referrers (web pages or applications that referred the short URL), clicks (number of clicks from the referrers). The total number of short URLs is $641,423$, including $18,496$ spam short URLs and $622,927$ legitimate short URLs.

We have to admit that Bitly is no longer the default link shortener on Twitter [12]. However, our work is independent to the shortening service provider since almost all shortening service providers could provide the similar information above for each short URL. Therefore, our research methods could be adapted easily by any other shortening URL providers. Next, we explain the labeling of the short URLs.

[1]There may be multiple short URLs pointing to the same long URL. This attribute records the total number of clicks received for all the short URLs pointing to the same long URL.

5.2.3 Data Labeling

As mentioned earlier, the tweets marked as removed due to terms of service violations are explicitly marked as spam in the Twitter dataset. We utilized this information as ground truth and assumed that these tweets contain malicious content, hence we labeled them as spam tweets.

The average account suspension rate is about 3% in the Twitter dataset. Account suspension by creation date is shown in Figure 37.

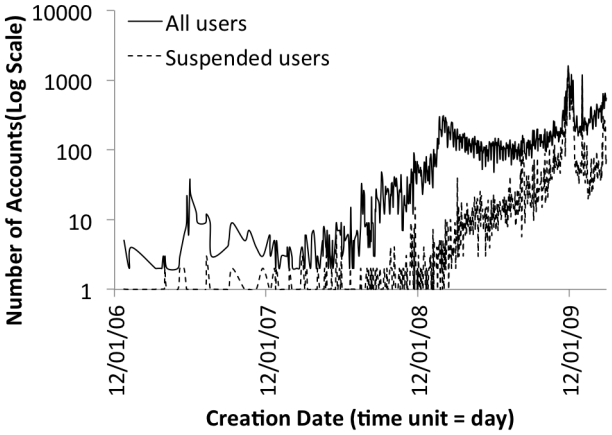


Figure 37: Account suspension by creation time in the Twitter dataset (time unit = day)

Based on previous work which has shown that URL links in spam messages have a high probability to be spam URLs [193], we labeled the short URLs in the spam tweets as spam short URLs.

We also checked the short URLs (including the final URLs and URLs in the redirection chain) against several public blacklists to validate the ground truth. The public blacklists included Google Safe Browsing, McAfee SiteAdvisor, URIBL, SURBL, and Spamhaus [75, 133, 165, 170, 179]. Google Safe Browsing allows users to query URLs against Google's constantly updated lists of phishing and malware pages. McAfee SiteAdvisor provides safety test results for the websites and shows a warning

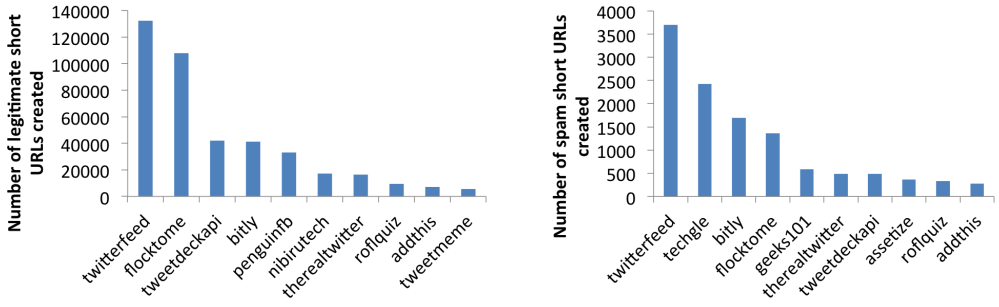
when the URL links to spam. URIBL, SURBL and Spamhaus are using similar mechanisms; they all contain suspicious websites appeared in spam emails. If the URLs were listed in any of the blacklists, we labeled them as spam. Since there was a delay between the time we generated the dataset and the time we labeled the dataset, the lag effect of blacklist validation was not a problem.

5.3 Data Analysis

In this section, we start with our analysis by focusing on the various attributes in the Bitly dataset, performing a statistical creator analysis and click source analysis.

5.3.1 Creator Analysis

We first focus on the creators of the short URLs. On Bitly, a creator is either a regular user or an enterprise user who generates the short URL for its services [27]. Through the creator analysis, we try to find out whether it could reveal the real spammers behind the scene. In our dataset, the total number of creators are $32,452$. Figure 38(a) depicts the top-10 creators of legitimate short URLs. Of these creators, *twitterfeed* is a utility allowed to feed content to Twitter, *tweetdeckapi* is the API service for tweetdeck, which is a social media dashboard application for management of Twitter, *penguinfb* is a service for sending status updates to Twitter from Facebook, *niburutech* is the software company behind a widely used Twitter client, *roflquiz* is a website offering funny Twitter quizzes, *addthis* is a social bookmarking service for Twitter, and *tweetmeme* is a service that determines the popular links on Twitter. We were not able to obtain accurate information about *flocktome* and *therealtwitter*. Similarly, Figure 38(b) depicts the top-10 creators of the spam short URLs. Of these creators, *assetize* is an advertising network for Twitter. We were not able to obtain accurate information about *techgle* and *geeks101*. We further observe that the total number of spam URLs created by the top-3 legitimate creators (i.e., twitterfeed, bitly, and tweetdeckapi) accounts for more than 31% of all the spam short URLs in



(a) Top-10 creators of legitimate short URLs (b) Top-10 creators of spam short URLs

Figure 38: Creators of the short URLs

the dataset.

We subsequently computed the percentage of the spam short URLs created by each creator. We observe that the number of creators who have created 80% or more of short URL spam in all URLs is 344. This corresponds to over 1% of the creator population. The total number of short URLs created by these creators corresponds to more than 31% of all the spam short URLs in the dataset.

Also, we notice that some creators that are assumed to be legitimate (e.g., twitterfeed) created spam short URLs. One reason for this might be the fact that these legitimate creators are not individual creators in the sense that they automatically shorten the URLs posted by the Twitter users. Moreover, it tells us that we cannot determine whether the creator is a spammer based on spam URLs they may create when the creator is an enterprise user. However, If the creator is individual user who generates spam URLs, we could track it back through the URLs and block it away. Furthermore, if the enterprise user has the mapping record which indicates who is the original creator for the spam URL, the enterprise user could cooperate with shortening service provider to lock down the culprit.

Another thought about creators is that whether we could determine that they are spammers by calculating out the percentage of spam URLs in all URLs that have been created by particular creators. Generally speaking, we believe that legitimate

users should create more legitimate URLs than spam URLs. Therefore, we rank the creators based on the percentage of spam URLs and the total number of spam URLs they created in the decreasing order.

Table 32: Top-10 creators that created only spam short URLs

Creator	*Spam URLs / All URLs*
dailypiff187	150/150
golfhonchonow	72/72
headlinehoncho	63/63
newswatchphilly	56/56
mskaya4u	56/56
golfhonchotoo	50/50
golfhoncho	48/48
breakingnewssource	47/47
onlinenewsblast	47/47
portlandtimestribune	46/46

Table 32 lists the top-10 creators that created only spam short URLs, sorted by the total number of short URLs created. We observe that the short URLs created by these 10 creators account for more than 3.4% of all the spam short URLs, which is a significant portion considering the total number of creators in the dataset. We were not able to obtain additional information about these creators from Bitly in order to decide whether all the short URLs they created are spam or not, nonetheless we believe that this kind of ranking is useful to classify whether a user of a URL shortening service is a spammer or not.

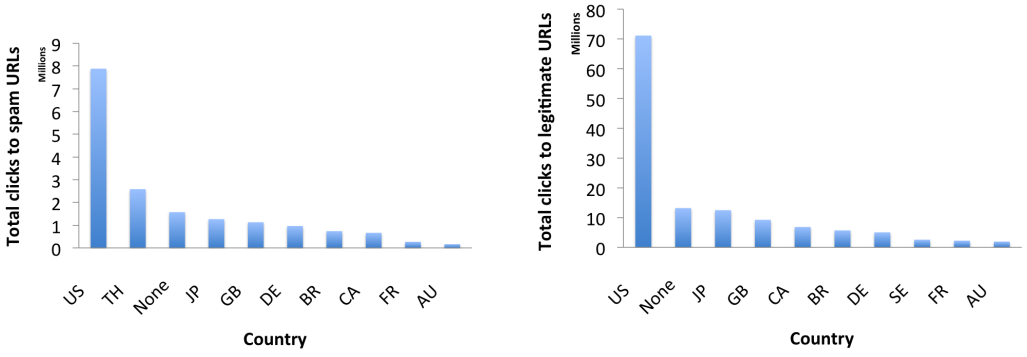
Meanwhile, prior research [112] has concluded that Twitter is mostly a news propagation network, with more than 85% of trending topics reflecting headline news. It is also interesting to see that many creators in Table 32 have news-related names (e.g., *newswatchphilly*, *breakingnewssource* and *onlinenewsblast*), showing that spammers are likely using this fact in order to increase their success rates. In addition, it means that spammers may employ URL shortening services by registering as enterprise users with trustworthy business names, which is really hard for those URL shortening service providers to distinguish them from legitimate enterprise users.

5.3.2 Click Source Analysis

In addition to creator analysis, we also investigated the sources of click traffic which may shed light on special patterns in click traffic of spam URLs. There are two types of click sources available in the Bitly dataset, namely the country click source and the referrer click source. Each short URL is associated with a country list and a referrer list, which contain click distributions coming from the countries and referrers, respectively. For example, short URL `http://bit.ly/olDmsz` has country list (US: 9 clicks) and referrer list (direct (email, IM, apps): 8 clicks and bitly.com: 1 click). This short URL is created for the long URL `http://www.google.com` for test purpose.

5.3.3 Country Source Analysis

The country list of short URL tells us the clicks from each country, which will help us find out those countries which generate high click traffic for spam URLs and legitimate URLs as well. We first aggregate the country click source by country name and list the top-10 countries based on the clicks to spam URLs and legitimate URLs. The results are shown in Figure 39. Here, "None" country source means that the country source is unknown to URL shortening service provider – Bitly in this case.



(a) Top-10 countries based on clicks to spam URLs

(b) Top-10 countries based on clicks to legitimate URLs

Figure 39: Country click sources of the short URLs

From Figure 39(a) and Figure 39(b), we have the following observations: (i) United States (US), Japan (JP), Great Britain (GB), Germany (DE), Brazil (BR), Canada (CA), France (FR), and Australia (AU) are in the lists for both spam and legitimate URLs, (ii) Thailand (TH) is ranked the second in the list for spam URLs but it is not in the list for legitimate URLs, and (iii) Concerning the relative order of the countries, Canada swaps positions with Germany in the list for legitimate URLs compared to that for spam URLs. The others remain in the same order.

For the second observation, after checking the clicks from Thailand, we determined that the reason is the spam URL *http://www.uggdiscount.co.uk/*, for which the total number of clicks is $2,554,121$. The creator of the short URL in Bitly is *mysocial*. It shows that spam URL may generate heavy traffic using URL shortening service.

Therefore, the observations shows some differences in terms of click traffic from source countries between spam URLs and legitimate URLs. But from spam detection perspective, we also need to look into click distribution of countries for each URL. Sophisticated spammers may spread spam URLs across the world that results in that more country codes in the list. We will take the distribution into account in our classification section.

5.3.4 Referrer Source Analysis

Referrer is the web page or application that contain the link to the web page pointed by the short URL. A referrer must have generated click traffic so as to appear in the Bitly dataset. The referrer list shows how many clicks come up from each referrer after the short URL is posted on them. We aggregate the referrer click source by referrer and list the top-10 referrers of the spam URLs and legitimate URLs based on clicks in Tables 33 and 34, respectively. Here, "direct" referrer means that referrers such as email messages, instant messages, and Apps.

From Tables 33 and 34, we make the following observations: (i) The majority of the

Table 33: Top-10 referrers of spam URLs based on clicks

Referrer	*Clicks*
direct	11,392,281
http://twitter.com/	2,619,560
http://twitter.com/home	229,628
http://td.partners.bit.ly	155,050
http://iconfactory.com/twitterrific	138,392
http://www.facebook.com/home.php	132,627
http://real-url.org	114,789
http://www.youtube.com/watch	105,056
http://www.facebook.com/	89,988
http://untiny.me	80,359

Table 34: Top-10 referrers of legitimate URLs based on clicks

Referrer	*Clicks*
direct	44,149,149
http://twitter.com/	10,947,917
http://td.partners.bit.ly	1,421,585
http://twitter.com/home	1,154,206
http://www.facebook.com/1.php	1,120,563
http://www.facebook.com/home.php	994,012
http://iconfactory.com/twitterrific	931,254
http://www.facebook.com/	774,080
http://twitter.com/ricky_martin	395,698
http://www.youtube.com/watch	385,082

clicks are from direct sources such as email clients, instant messages and applications, and (ii) The spammers utilize popular social media such as Twitter and Facebook for short URL spam to attract more attention.

The observations shows that short URLs are very popular not only on social media but also on other kinds of media such as traditional emails and mobile phones.The reason is that social networking sites connect those media together like the prediction that everything will be connection in twenty years [116] . Moreover, spammers should know that all those media are connected and propagate spam across them through short URLs.

The same to the country source analysis, we need to look into click distribution of

referrers for each URL as well. We believe that spammers try to use different channels as many as possible. And we will take it into account in the classification section. In addition, the referrer list exposes the places where spammers post short URL spam. Thus, after short URL spam are detected, spam detection team on social media could use the referrer list to track all the places having the short URL spam except direct source since it does not or cannot provide specific addresses in the URI form.

5.4 Classification

In this section, we first introduce the features used in short URL classification . Then, we describe the data filtering process and the classifiers used in our classification framework.

5.4.1 Classification Features

As mentioned in Section 5.3.2, there are four types of meta-data available in the Bitly dataset. We keep most of the attributes in the tables as features and additionally add aggregate features for classification. All following features are Twitter-independent features so that our classifier could be easily adapted to detect short URL spam on any other social media.

For each short URL, we have chosen the following features for classification:

- **Clicks:** user_clicks (total number of clicks received), global_clicks (total number of clicks received globally), and the ratio between user_clicks and global_clicks;
- **Countries:** country count and features of click distribution (mean and standard derivation);
- **Referrers:** referrer count and features of click distribution (mean and standard derivation).

Clicks features provide us the quantitative measure of click traffic over the lifetime of short URLs in big picture. Countries features show us click distribution from source

countries and referrers features give us click distribution from source referrers. We use those features to test our assumption that spammers are propagating spam across multiple countries using many referrers as they can. Also, we try to find out which feature could express the most discriminative power in short URL classification.

5.4.2 Machine Learning Classifiers

We use the various classifiers implemented in the Weka software package [84]. Weka is an open source collection of machine learning algorithms and has become the standard tool in the machine learning community. The classifiers used in the classification framework include Random Forest, Decision Table, Random Tree, K*, SMO (an algorithm for training a support vector classifier), Simple Logistic, and Decision Tree. The reason why we choose them is that they are popular and also represent different categories of classification algorithms. Through those algorithms, we could find out which algorithm is the best fit for our short URL classification.

5.4.3 Classification Setup and Cross Validation

We know that click distributions from countries or referrers have no meaning if the user clicks is less than 2. Thus, our experiments will only focus on short URLs having that user clicks value is larger than or equals to 2 at least.

Several reasons have caused low clicks of URLs in our dataset. One reason for this is that the URL may be created recently compared with our dataset creation time so that our dataset is not able to collect more clicks. Also no interest from people and URL filtering of websites may cause low clicks as well. If spam URL attracts few clicks or is filtered by spam detection engine of website, that means this kind of URL is easy to distinguish or detect. We will not focus on this kind of URL in this chapter.

In addition, we believe that the click traffic pattern of short URL spam appear more evidently as the increasing of user clicks. To prove that, we process dataset

into 7 groups based on the value range of user clicks: ≥ 2 , ≥ 5, ≥ 10, ≥ 20, ≥ 30, ≥ 40, and ≥ 50. The reason for breaking-down the data into seven groups is as follows: first, group ≥ 2 serves the baseline group. After that, we want to increase the threshold by the same increase interval starting with the threshold 10. Due to that more than 30% short URLs are between threshold 2 and 10, the group ≥ 5 is added into the list. We try to show more accurate results by split the range between 2 and 10. For each group, we randomly choose the same amount of legitimate URLs as spam URLs to eliminate any prior probability influence.

We employed the machine learning classifiers previously mentioned using 10-fold cross-validation model. Cross validation is a technique for protecting against over-fitting in a predictive model. Specifically, the data is randomly divided into k groups and the classifier is re-estimated k times, holding back a different group each time. The overall accuracy of the classifier is the mean accuracy of the k classifiers tested.

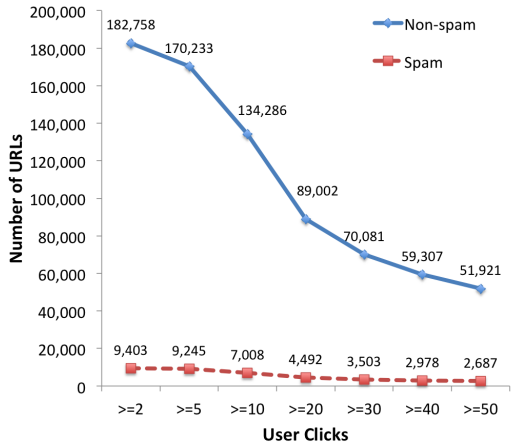
5.5 Evaluation

In this section, we first introduce the evaluation metrics for short URL classification. Then, we evaluate two major metrics that are the F-measure and accuracy of the classification framework based on the ground truth dataset.

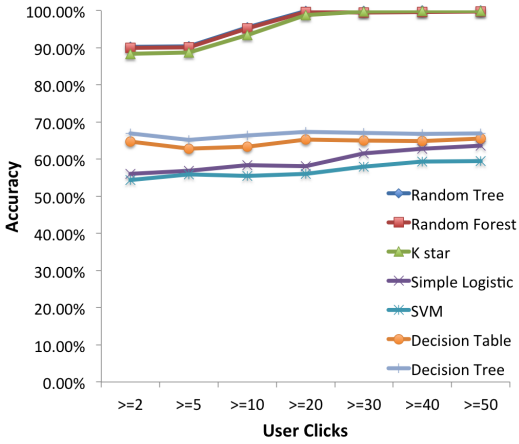
5.5.1 Evaluation Results

We use two major metrics including the F-measure and accuracy to evaluate the performance of the classifiers. After performing all the classification experiments on seven groups (≥ 2 , ≥ 5, ≥ 10, ≥ 20, ≥ 30, ≥ 40, and ≥ 50) in our dataset, we present the results of cross validation in Figure 40. The results show that Random Tree, Random Forest, and K star algorithms outperform other four algorithms including Decision Tree, Decision Table, Simple Logistic, and SVM algorithms in terms of accuracy, F-measure and false positive rate. Especially, Random Tree algorithm performs the best among all seven algorithms. Meanwhile, as the number of user clicks

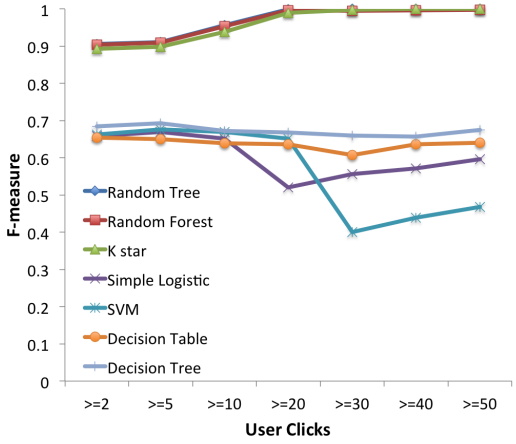
increases, the performance of Random Tree, Random Forest, and K star classifiers has improved.



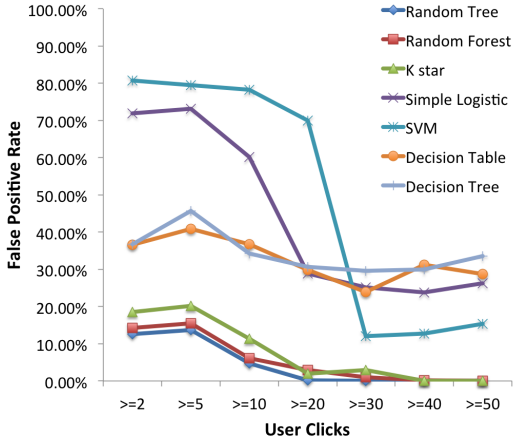
(a) Spam and Non-spam URLs in Datasets



(b) Accuracy



(c) F-measure



(d) False Positive Rate

Figure 40: Experimental Results of Cross Validation

We also observe that the performance of some algorithms such as SVM has sharp drop in some groups like ≥ 20 and ≥ 30 groups. One possible reason is that the number of spam URLs has decreased a lot as we increase the threshold, which may result in over-fitting in the classification, especially when the classifier is sensitive to the size of training dataset. However, other classifiers such as Random Tree and K star are stable as the increase of the threshold.

For the dataset in which user clicks number is larger than or equals to 2, we

list four metrics for evaluating the classification performance sorted on accuracy in Table 35. It shows that the best classification performance is from the Random Tree algorithm but it still has high FP rate.

Table 35: Results of classification for short URL spam detection based on click traffic features

Algorithm	*TP*	*FP*	*F-measure*	*Acc.*
Random Tree	0.959	0.143	0.913	90.81%
Random Forest	0.946	0.134	0.910	90.6%
KStar	0.949	0.171	0.895	88.88%
Decision Tree	0.806	0.372	0.740	71.69%
Decision Table	0.622	0.342	0.634	64.03%
Simple Logistic	0.657	0.528	0.601	56.43%
SVM	0.886	0.807	0.658	53.93%

By investigating the errors, we attribute them to the following possible reasons:

- Lack of features: only 9 click traffic features are used in classification, and
- Some mislabeled short URLs: Spammers might try to appear legitimate by mixing in spam and legitimate URLs in their posts. This might be as a result of previous spam fighting efforts by Twitter and following evolution of spammers.

In future work, we plan to investigate additional features for short URL classification and confirm labeling of a small sample of tweet URLs manually. And we will discuss this in more details later.

In classification, some features play a more important role than others during classification. Thus, by using information gain in feature ranking, we listed all sorted features in the decreasing order of information gain value (shown in Table 36). It evaluates the discrimination weight each feature has. The top 3 discriminative features are user clicks, standard deviation among country click sources, and standard deviation among referrer click sources. It indicates there exists differences between spam and non-spam URLs in terms of distributions of country click sources and referrer click sources. Number of referrers, number of countries, and ratio between user

Table 36: Ranked features based on information gain

Feature Name	*Information Gain*
User clicks	0.0392
Standard deviation among country click sources	0.0387
Standard deviation among referrer click sources	0.0364
Mean among referrer click sources	0.0356
Global clicks	0.0308
Mean among country click sources	0.0289
Number of referrers	0.0219
Number of countries	0.019
Ratio between user clicks and global clicks	0.0174

clicks and global clicks are the last 3 features. It implies that they did not help a lot in short URL spam classification.

By further investigating the correlation from those features to spam URLs, we looked into the value distributions of features in spam URLs and legitimate URLs. We found out that with the same mean among referrer click sources, there are more short URL spam when short URLs have low standard deviation among referrer click sources. It implies that spammers spread spam on a lot of referrers and the click traffic from those referrers show not much difference. Moreover, this phenomenon becomes more evident when the short URL obtains high click traffic.

5.6 Discussion

Although our classification shows good results in applying click traffic analysis on short URL spam detection, there are still many limitations and challenges we need to face.

5.6.1 Limitations

We have two major limitations in our classification with respect to two aspects: dataset collection and data labeling.

First, our click traffic dataset is based on APIs provided by URL shortening service providers. Thus, data collection is limited by those APIs. If service providers

block the API access or modify their APIs, our data collection will need to modify accordingly. In addition, it is also hard for us to obtain other kinds of click traffic features outside the APIs such as daily clicks from country click sources and referrer click sources. Given daily click traffic features, we will be able to use them in classification and explore more special patterns in click traffic of spam URLs.

Second, we have used several public blacklists such as Google safe browsing, SURBL and URIBL in data labeling which provide strong validation of ground truth spam labels. But it is still possible that some URL spam in the dataset are mislabeled. One possible reason is that those blacklists keep updating and also removing old items based on their own policies. Thus, some URLs supposed to be spam URLs may be labeled as legitimate instead as the blacklists removed them from the list. Additionally, those mislabeling URLs exert more influences on the results of classification when the size of training dataset is small. Therefore, we need to obtain more validation resources for data labeling especially when our classification is deployed in real-time.

5.6.2 Challenges and Possible Countermeasures

In addition to limitations above, we are also facing several challenges in terms of performance and effectiveness in practice.

One challenge is that our classifiers work when the short URLs have click traffic (at least 2 clicks for each short URL). For those short URLs with less than 2 clicks, we ignored them since either no one is interested in the content or people recognize it as spam easily based on content. But only using our algorithm may be not enough for preventing spamming activities. Combining with other layers of spam detection may make a better result. For example, we analyze them based on content and user behaviors for those URLs with very few clicks. As they attract more clicks, we could combine our classification with behavior analysis like work done by Maia et al. [131]

and content analysis [144].

Another challenge is that a spammer can setup a new URL shortener in several minutes to the same or another long URL spam after being detected. If the long URL is the same URL as the previous long URL or appears in the redirection chain to the previous long URL, we could store the previous long URL and the redirection chain to the previous long URL on our URL blacklist after we classified the short URL as spam URL. In such way, it will force spammers to create completely new domain to avoid detection. Our method could increase the cost of spamming activities of spammers at least.

Moreover, another challenge is that spammers could create click traffic for spam URL to confuse our classifier after they know our algorithm. We need to adopt methods for click fraud detection in our data pre-filtering process to eliminate the noise, which will be considered as future work.

5.7 Related Work

Social network spam has been investigated in several recent papers. Zhang et al. [207] proposed a method for detecting instances of automated Twitter accounts using the publicly available timestamp associated with each tweet. The work revealed that automated accounts exhibit distinct timing patterns that are detectable using statistical inference. Similarly, Castillo et al. [38] discussed how to assess the level of social media credibility of newsworthy topics from a given set of tweets, classifying those topics as credible or non-credible. Benevenuto et al. [24] have addressed the issue of detecting video spammers and promoters. Other recent works on spam detection in Twitter include Wang et al. [183] that proposed a classification approach to automatically identify suspicious users by (i) a directed social graph model that captures the follower and friend relationships in the network, and (ii) content-based

and graph-based features extracted based on the spam policy of Twitter, and Benevenuto et al. [23] that introduced machine learning techniques to identify spammers based on number of followers, number of followees, and other social interactions such as the number of times the user was mentioned and the number of times the user was replied to.

The popularity of short URLs has immensely increased over the years due to micro-blogging platforms such as Twitter, where message space is limited by 140 characters. Antoniades et al. [15] has recently explored this emerging phenomenon and presented a characterization of short URLs. Their analysis was performed on a dataset collected by crawling Twitter for short URLs from Bitly and Owly URL shortening services. Specifically, their results showed that (i) the maximum access to short URLs come from emails and online social media, (ii) the click distribution of short URLs is approximately a log-normal curve, and (iii) a large percentage of short URLs are not ephemeral and 50% of short URLs live for more than three months.

Recently, several papers have investigated spamming on social networks via short URLs. Grier et al. [77] presented a characterization of spam on Twitter. Their analysis showed that (i) Twitter spam is more effective than email spam with an overall click-through rate of 0.13%, (ii) blacklists are no optimal solution for fighting spam on Twitter as they are too slow at identifying new threats, and (iii) spammers use URL shortener services to obfuscate their links in tweets, negating any potential gains even if blacklist delays were reduced. Chhabra et al. [43] analyzed phishing attacks on Twitter using URL shorteners. Phishing is one form of spam, where the goal is to steal personal information from users for fraudulent purposes. This work concluded that (i) phishers use URL shorteners to hide their identity, (ii) online social media brands such as Twitter are targeted by phishers more than traditional brands such as eBay, and (iii) phishing URLs which are referred from Twitter are more likely to attract victims. Klien et al. [108] studied usage logs of a URL shortener service

that had been operated by the authors for more than a year. Their results showed that (i) different countries differ significantly with regard to the usage of their service, (ii) around 80% of URLs shortened by their service lead to spam-related content, and (iii) spamming attacks via short URLs cross national borders. Maggi et al [130] measured two years of short URLs and provided some countermeasures but it did not offer efficient short URL spam detection approach.

Our work differs from the aforementioned research along three dimensions: (i) we analyze a comprehensive dataset containing over $600,000$ short URLs; (ii) we consider spam in general and do not restrict the analysis to a specific form of spam such as phishing, and most importantly; (iii) we attempt to classify short URLs as to whether they lead to spam or not using their click traffic information.

5.8 Conclusion

We conducted the first large-scale experimental study of short URLs through creator and click source analysis on the Bitly dataset - a collection of 641,423 short URLs. We first analyzed the creators of the short URLs and determined that the legitimate creators in Bitly generate short URL spam as well. As future work, we plan to uncover spam creators after short URL classification. We then examined the clicks to the short URLs and found that the majority of the clicks are from direct sources such as email clients and that the spammers utilize popular websites such as Facebook to attract more attention. We finally performed classification of short URL spam based on click traffic and analyzed performance change of classifiers as the increase of user clicks. Random Tree, Random Forest, and K start algorithms outperform other algorithms. Of them, the Random Tree algorithm achieved the best performance with an accuracy of 90.81% and an F1-measure value of 0.913. We believe some of the classification errors might have been caused by the lack of features and some mislabeling in the dataset.

Our analysis and classification work can be considered as a new approach to classification in the ongoing battle of short URL spam detection. An interesting direction for future research involves combing our click traffic analysis process with these existing analysis techniques to create a multi-layered defense against short URL spam.

CHAPTER VI

BEAN: A BEHAVIOR ANALYSIS APPROACH OF URL SPAM FILTERING IN TWITTER

URL spam is a major problem because of its popular use and redirection function. Existing filtering methods are not efficient in detecting URL spam in collective spamming activities. Public URL blacklists have large time delay in URL spam detection in spite of their high accuracy. Content-based analysis has high false-positive rate in addition to the length limit of content in some cases (for example, Twitter message has the limit of 140 characters). Moreover, web page analysis that detects URL spam through classifying downloaded web pages behind the URL links has performance shortcomings due to the costs of downloading the URL content and, typically, text classification.

In this chapter, we explore the URL spam detection methods focusing on a popular social website, namely Twitter. More specifically, we focus on collecting and analyzing data from Twitter trending topics primarily because we observe that Twitter trending topics attract more spammers because of their high openness and popularity. Using the above data, we make the following contributions:

- We investigate spammers behaviors in social media through an in-depth study on URL spam in Twitter trending topics and make the following observations: Spammers post multiple messages that contain the same URL in multiple trending topics. They also create or hack different user accounts to post multiple messages that contain the same URL.

- We create behavior analysis approach BEAN derived from Markov Chain model

to detect URL spam in collective spamming activities in Twitter. Applying BEAN only requires a small number of parameters, which will be detailed later.

- We demonstrate the efficiency and scope of BEAN by an empirical study on real data. Namely, we compare its performance with two baselines (SVM and TrustRank), where BEAN shows better performance in terms of several metrics including precision, re-call, and F-measure.

The remainder of the chapter is organized as follows. Section 6.1 introduces preliminaries of our research target – Twitter trending topics. Section 6.2 describes data collection and analysis. Section 6.3 defines our problem and gives an overview of our methodology. Section 6.4 explains detailed methodology of the behavior analysis approach – BEAN and Section 6.5 presents our evaluation results. Section 6.6 summarizes related work. Finally, Section 6.7 concludes the chapter.

6.1 Preliminaries

In this section, we introduce the background of our work including the objects in Twitter trending topics and the relationships among those objects. Most of the objects commonly appear in message sending activities in social media. Thus, it makes our approach easy to be adopted by other social media as well.

6.1.1 Objects in Twitter Trending Topics

First, we look into three major objects in Twitter trending topics (see examples in Figure 41):

- Trend: it is a popular topic in Twitter. Normally, you could see the top 10 trending topics on Twitter homepage or your personal profile. Twitter users could create a trend using hashtag (#) in messages they posted.

- Status: it is a message in Twitter and is also known as tweet . One status may contain multiple URLs (or short URLs).

- User: it is a registered user account in Twitter.

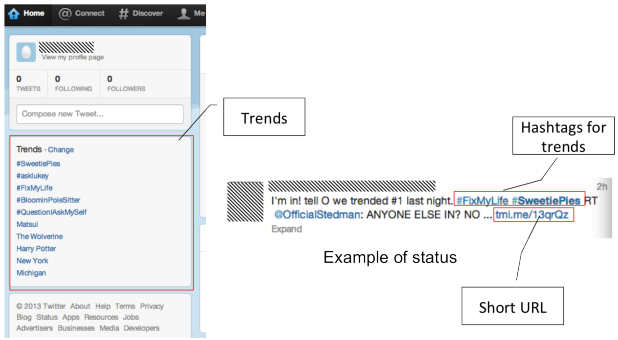


Figure 41: Examples of trends and status in Twitter

In Figure 41, it also shows a short URL example. Due to 140 characters limit of Twitter message, people prefer using short URL instead of original long URL in sending messages. Here, we define short URL as the URL generated by URL shortening service providers such as Bitly and TinyURL. It has a corresponding expanded URL or long URL, which is the original URL of short URL. Next, we will talk about the relationships among those objects in Twitter.

6.1.2 Relationships among Objects

We draw the communication model for Twitter trending topics shown in Figure 42 and list the four major relationships as follows. 1) Trend – Status: one to many; 2) Status – Short URL: many to many; 3) Short URL – Expanded URL: one to one; 4) User – Status: one to many.

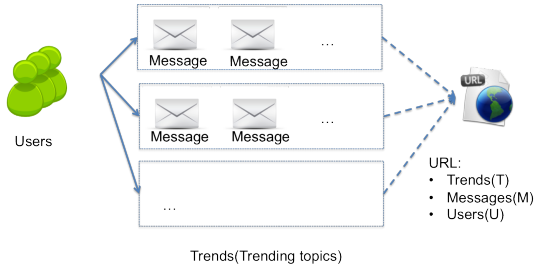


Figure 42: Communication model in Twitter trending topics

6.2 Data Collection and Analysis

We collect our Twitter data through Twitter APIs and the details of our dataset are as follows:

Twitter Profile, Message, and Web Page Datasets: It consists of over 900,000 Twitter users, about over 2.4 million tweets, and all URL links in the tweets. The tweets were gathered by querying the top ten trending topics every minute and they represent about 600 topics over the span of November 2009 to February 2010.

6.2.1 Statistics of URLs

To investigate the usage of URLs in Twitter, we aggregate the URLs by the URL domain names that are shown in Table 37.

Table 37: URL domains in Twitter Dataset

URL domain	Count of URLs
Bit.ly	641,423
t.co	129.677
Tiny.com	62,488
Ow.ly	42,838
Is.gd	14,664
Goo.gl	13,122
j.mp	8,963
Su.pr	3,764
Twur.nl	2,807
Migre.me	2,788
Wp.me	2,704
Post.ly	2,556
Tiny.cc	2,520
Dlvr.it	2,325
3.ly	679
Adf.ly	107

We observe that the majority of URLs in our dataset are short URLs. Bitly is the most popular URL shortening service provider in our Twitter data. Also a bunch of other URL shortening service providers are active in Twitter. The next section will show the labeling method of our Twitter dataset.

6.2.2 Data Labeling

The time delay between data collection and data labeling allows us to label the dataset according to public blacklists. To obtain the ground truth, we automatically check the URLs including the final URLs and URLs in the redirection chain against a list of several public blacklists containing Google Safe Browsing, McAfee SiteAdvisor, URIBL, SURBL, and Spamhaus [75, 133, 165, 170, 179] using public APIs or web services. Google Safe Browsing has Google's constantly updated lists of phishing and malware pages to allow users to query URLs against. McAfee SiteAdvisor shows a warning when the URL links to spam by providing safety test results for the websites. URIBL, SURBL and Spamhaus used similar mechanism to update their spam URL blacklists. We label the URL as spam if it is listed in any of the blacklists.

6.2.3 Analysis of URL Spam

After finishing the data labeling, we further analyze the spammy property of URLs in correlation with trends, statuses, and users. Next, we start with introducing the data processing, followed by the result analysis.

Data Processing: We organize the statuses in our Twitter dataset by different trends and extract "user $\rightarrow$ status" pairs from the trends. Specifically, a trend T consists of a sequence of k statuses, i.e., $T = S_1, S_2, ..., S_k$. Each status has a corresponding Twitter user and also may contain URLs.

Before illustrating the details of the processing, we outline the following relevant terms based on one unique URL: T_n: the number of trends that include the statuses having the unique URL; S_n: the number of statuses that contain the unique URL; U_n: the number of users who send the statuses having the unique URL. To keep URL unique, we track back to the original long URLs for short URLs and maintain an existing URL list.

Each unique URL has one vector: $\langle T_n, S_n, U_n \rangle$. We aggregate the dataset based

on the three terms to see the characteristics of URL spam. The data processing for T_n has the following steps:

Step 1: Group URLs by the values of T_n.

Step 2: Calculate the percentage of URL spam for each group.

Step 3: Visualize the trend of percentage of URL spam as the increasing of T_n.

For S_n and U_n, we repeat the same process to see the correlations. We observe that the percentage of URL spam in the group cannot show the trend accurately when the size of URL group is too small. Thus, we select URL groups who contain 3 or more URLs for illustration in the result analysis section.

Result Analysis: Figure 43 shows the percentage of URL spam in groups versus the number of trends. It shows that the percentage of URL spam increases greatly as the increasing of the number of trends. The percentage of URL spam achieves 100% for most groups with 80 or more trends. It means that those groups only contain URL spam.

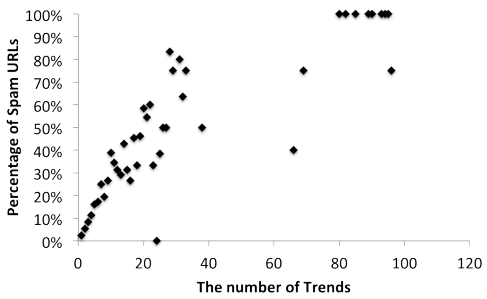


Figure 43: Percentage of URL spam in groups vs. number of trends

Figure 44 shows the percentage of URL spam in groups versus the number of statuses. It has the linear increasing trend overall. But it becomes sparser and more fluctuated when the number of statuses is beyond 20. One possible reason is that there are a small number of URLs in those groups.

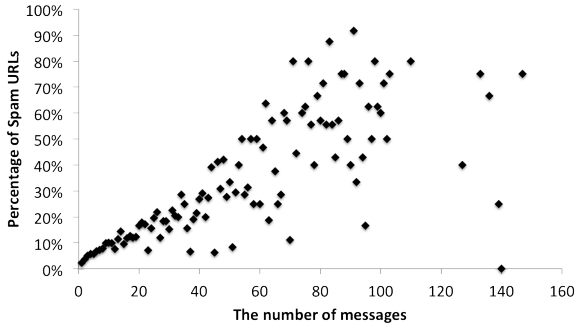


Figure 44: Percentage of URL spam in groups vs. number of statuses

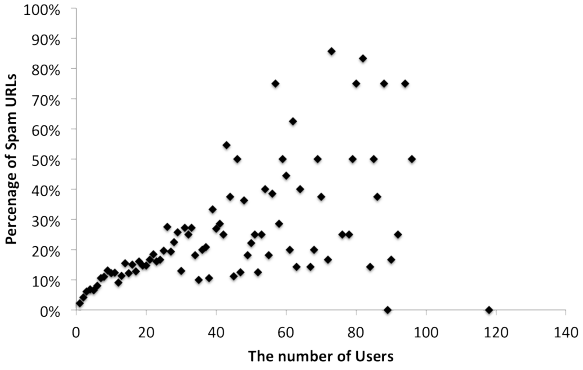


Figure 45: Percentage of URL spam in groups vs. number of users

Figure 45 shows the percentage of URL spam in groups versus the number of users. It has the linear increasing trend as well. But after the number of users is beyond 30, the percentage of spam URLs is no longer stable and does not fit into the linear increasing trend. One possible reason is that many popular legitimate URLs also have lots of associated users.

Table 38: Four types of sending events

Name	Denotation	Explanation
Seeding event	S_e	New user sends new message to new trending topic
User joining event	U_e	New user joins sending new message to existing trending topic set
Topic spreading event	T_e	Existing user sends new message to new trending topic
Message repeating event	M_e	Existing user sends new message to existing trending topic set

Table 39: Six URL behavioral states

No.	State	Attributes $\langle T_n, S_n, U_n \rangle$	Explanation
1	First-time sending	$\langle 1, 1, 1 \rangle$	One user sends one status to one trend
2	Multiple messages	$\langle 1, k, 1 \rangle$	One user sends multiple statuses to one trend
3	Multiple topics	$\langle k, 1, 1 \rangle$	One user sends one status to multiple trends
4	Multiple messages across multiple topics	$\langle k, k, 1 \rangle$	One user sends multiple statuses to multiple trends
5	Group user sending	$\langle 1, k, k \rangle$	Multiple users send multiple statuses to one trend
6	Group user sending across multiple topics	$\langle k, k, k \rangle$	Multiple users send multiple statuses to multiple trends

Through the analysis, we have the following observations:

- Spammers post multiple statuses that contain the same URL in multiple trends.
- Spammers create or hack different user accounts to post multiple statuses that contain the same URL.

6.3 Overview

The section above shows that lots of collective spamming activities have happened in Twitter trending topics. In this section, we discuss problem definition and present the overall procedure of our solution.

6.3.1 Problem Definition

URL spam in collective spamming activities is hard to be detected by normal spam detection approaches. We propose a behavior analysis approach – BEAN to detect URL spam in those activities by capturing the anomalous message sending behaviors of spammers. Our main research interest is to study how to exploit message sending behaviors and their characteristics to detect URL spam. For ease of discussion, we have the following definitions:

Definition 1. **Sending event.** An observable occurrence of message sending associated with one unique URL.

Definition 2. **URL associated sending behavior.** The range of sending events associated with one unique URL in conjunction with its environment.

Sending event is the base unit of our behavior analysis approach. We will use symbols to denote sending events. Let Σ denote the set of sending event symbols. We have four types of events: seeding event, user joining event, topic spreading event, and message repeating event. We list the major notations about sending events and their explanations in Table 38. All events here are for the same URL spreading in Twitter. The sending event trace of one unique URL represents URL associated sending behavior of the URL.

Besides, the three variables (trend, status, and user) associated with one unique URL determine the behavioral states of URLs that in fact influence the spam detection strategies. Next, we will discuss the details of URL behavioral states.

URL Behavioral States: As mentioned in 3rd section, each URL has one vector: $\langle T_n, S_n, U_n \rangle$. Based on values of the vector, we could have six different URL behavioral states. Table 39 shows the explanations of them. We also show the transition among URL behavioral states in Figure 46.

Figure 46 shows the dynamic transitions among different behavioral states. To investigate spamming behaviors in our dataset, we show the distribution of spam URLs in different URL behavioral states in terms of total amount and percentage in Figure 47(a) and the distribution of message spam associated with those URL spam in Figure 47(b) .

Comparing Figure 47(a) with Figure 47(b), we have the following observations:

- A large amount of spam URLs are in "First-time sending" state but the percentage of spam URLs over total URLs in the "First-time sending" state is about

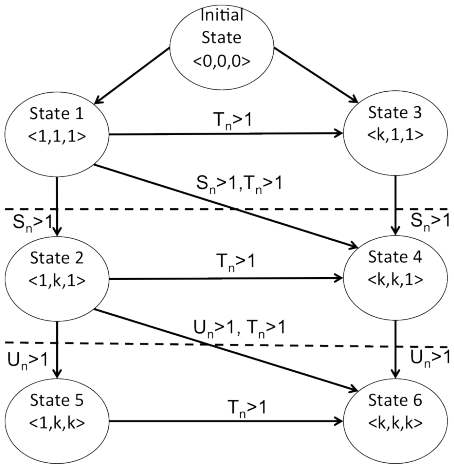
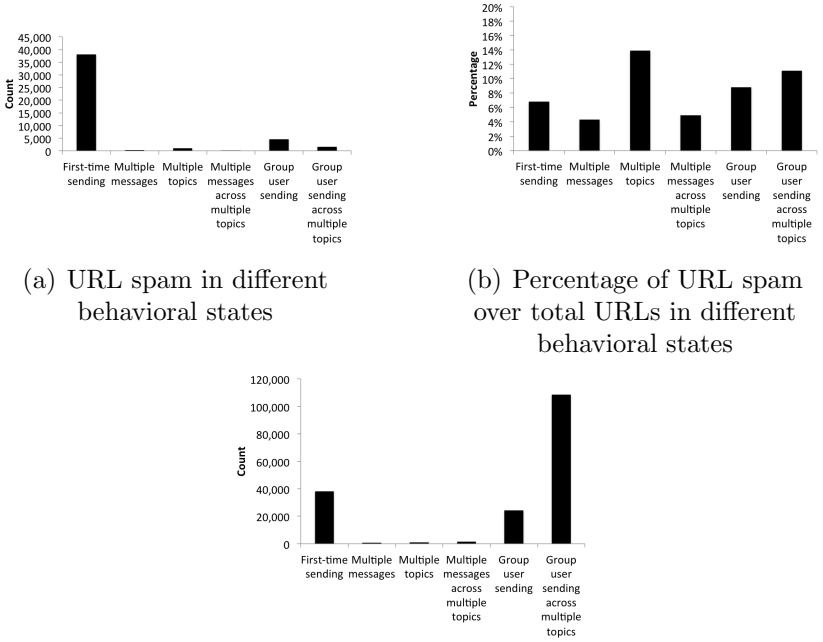


Figure 46: Transitions among URL behavioral states



(a) URL spam in different behavioral states

(b) Percentage of URL spam over total URLs in different behavioral states

(c) Message spam associated with URL spam in different behavioral states

Figure 47: Distribution of URL spam in different behavioral states and associated message spam

6%. While a small number of spam URLs are in "Multiple topics" state but the percentage of spam URLs over total URLs in the "Multiple topics" state is higher than 15%. It indicates that spammers try to attract more traffic by adding multiple topics to messages.

- In terms of percentage of URL spam in behavioral states, the descending order is as follows: "Multiple topics", "Group user sending across multiple topics", "Group user sending", "First-time sending", "Multiple messages across multiple topics" and "Multiple messages". It further indicates that spammers are becoming more sophisticated than before. Instead of using one user account to post multiple messages containing the same URL, they adopt collective spamming activities to avoid detection.

Since message sending event is the base of our behavior analysis approach, we also investigate the distribution of message spam associated with URL spam in different behavior states as shown in Figure 47(c).

Figure 47(c) shows the distribution of message spam associated with URL spam in different behavioral states . "Group user sending across multiple topics", "Group user sending", and "First-time sending" states are three states having a large number of message spam associated. It also shows the seriousness of collective spamming activities in Twitter.

6.3.2 Overall Procedure

First, we have our URL blacklist to filter out previous detected URL spam. Then, according to different behavioral states, we will use different strategies to detect new URL spam:

- For URLs in "First-time sending" state, we adopt content analysis and web page analysis through downloading. This strategy will not be discussed in details in this chapter.

- For URLs in "Group user sending across multiple topics" and "Group user sending" states, we detect URL spam through our behavior analysis approach BEAN.

- For URLs in other states, they could be easily detected and restricted by Twitter. We will use BEAN if they transit to "Group user sending" and "Group user sending across multiple topics" states later.

All detected URL spam will be added into URL blacklist.

6.4 Detailed Methodology

In this section, we describe the behavior analysis approach BEAN in details and also illustrate how to use it effectively in URL spam detection.

6.4.1 Markov Chain Model

Since our behavior analysis approach BEAN is built on Markov Chain model, we will introduce the Markov Chain model first. Suppose the Markov Chain Model is denoted by 3-tuple (S, P, s_0), where S is the set of states, $P : (S \times S) \rightarrow R$ denotes the transition probabilities, and $s_0 \in S$ is the initial state. The probability of a transition (s, s') is denoted by $P(s, s')$. The following equality should hold for all states to keep P a valid measure:

$$\sum_{s' \in SUCC(s)} P(s, s') = 1 \tag{7}$$

Where $SUCC(s)$ denotes the set of successors of s. The trace of length w (also called as window size) associated with the state s is denoted by $\sigma(s)$. We define the term event trace as a finite sequence of event symbols over event symbol set Σ. The set of finite event traces over Σ is denoted by Σ^*. The empty trace is denoted by ϵ. The set of traces of length n is denoted by Σ_n. Next, we will talk about how we associate the sending event with Markov Chain model.

State set is the fundamental component of Markov Chain model. In our behavior analysis, a state in the Markov Chain model is associated with a sending event trace of length w over the sending event symbol set $\Sigma \cup \{\phi\}$. A state transition is a pair of states. The pair (s, s') denotes a transition from s to s'. The initial state of the

Markov Chain (denoted by s_0) is associated with sending event trace of length w consisting of all null symbols Φ, e.g., if $w = 2$, s_0 is associated with the sending event trace $[\Phi, \Phi]$. We illustrate an example of state transition in which the window size w is equal to 2 in Figure 48.

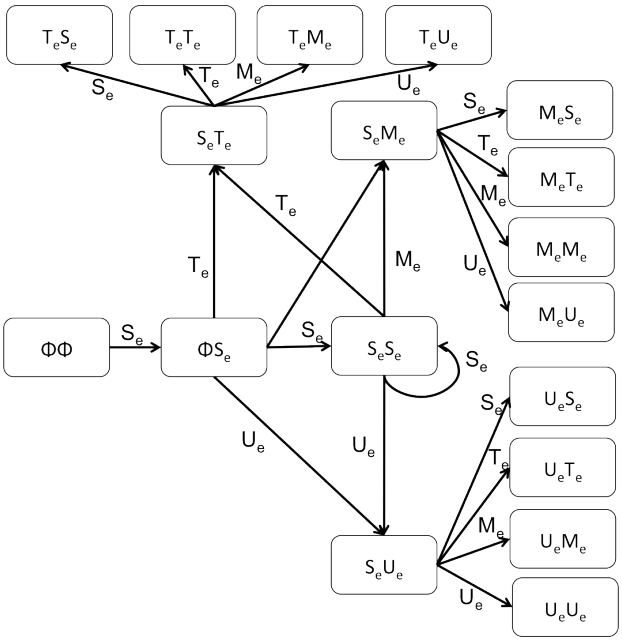


Figure 48: Example of state transitions

Figure 48 shows the example of state transitions when the window size equals 2. S_e, U_e, T_e, and M_e denote four sending events: seeding event, user joining event, topic spreading event, and message repeating event respectively. The state set grows from the initial state $[\Phi, \Phi]$ as new sending events happen. Since different probabilities are associated with the state transitions, the final state will gain different probabilities by multiplying the transition probability along the sending event trace.

6.4.2 Converting to Classifier from Markov Chain Model

We cannot directly use Markov Chain Model to detect URL spam. The conversion from Markov Chain model to classifier is needed in our behavior analysis approach. Derived from the conversion method introduced in [102], we design the conversion algorithm in our approach.

Algorithm 1 Calculate $\mu(\alpha)$

Require: $Y = 0; X = 0; i = 0$
  while $i \leq m$ **do**
    if $s_i \rightarrow s_{i+1}$ is a valid transition **then**
      $Y = Y + F_y(s, (s, s'))$
      $X = X + F_x(s, (s, s'))$
    else
      if $s_i \rightarrow s_{i+1}$ is not a valid transition **then**
        $Y = Y + z$
        $X = X + 1$
      end if
    end if
    $i = i + 1;$
  end while
Ensure: $X > 0$
  $\mu(\alpha) = \frac{Y}{X}$

In Algorithm 1, we define a metric $\mu(\alpha)$ corresponding to a sending event trace α considering $\alpha \in \Sigma^*$. m is the length of the sending event trace α. Function $F_y(s, (s, s'))$ and $F_x(s, (s, s'))$ are the adjustment function in the conversion algorithm that we will introduce the details of them later. The metric $\mu(\alpha)$ measures how well the Markov Chain model predicts the sending event trace α.

Given a threshold $r \in R$, a classifier f can be constructed from the metric μ as follows:

$$f(\alpha) = \begin{cases} 1 & \text{if } \mu(\alpha) \geq r \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Here, when $f(\alpha) = 1$, it means that the sending event trace is from a URL spam. For the threshold setting, we will discuss in details in experimental evaluation section.

6.4.3 Common Functions

Depending on the choice of the common functions, we will obtain different classifiers. Also, according to the classifier function above, sending event traces for URL spam obtain higher value than ones of legitimate URLs in our classifiers. We will use three different metrics for common functions F_y and F_x as follows:

- The miss-probability metric

  $F_y(s,(s,s')) = \sum_{s_1 \in SUCC(s) \wedge s' \neq s_1} P(s,s_1)$

  $F_x(s,(s,s')) = \sum_{s_1 \in SUCC(s)} P(s,s_1) = 1.$

- The miss-rate metric

  $P_{max}(s) = \max_{s' \in SUCC(s)} P(s,s')$

  $F_y(s,(s,s')) = (P(s,s') \neq P_{max}(s))$

  $F_x(s,(s,s')) = 1$

- The local-entropy-reduction metric

  Local entropy:

  $LE(s) = \sum_{s' \in SUCC(s)} -P(s,s')log(P(s,s'))$

  $F_y(s,(s,s')) = LE(s) + P(s,s')log(P(s,s'))$

  $F_x(s,(s,s')) = LE(s)$

In the miss-probability metric, $F_y(s,(s,s'))$ equals the sum of all the probability of the transitions from the state s that are not equal to (s,s'). It means that $F_y(s,(s,s'))$ has a higher value if a sending event trace has more state transitions with low transition probability. In the miss-rate metric, every transition that is not equal to the "maximal" transition will be penalized by 1. In the local-entropy-reduction metric, $LE(s)$ denotes the "residual" local entropy of the state s after removing the transition (s,s') or the local entropy reduction due to taking the transition (s,s'). The sending event trace that has higher information gain will have higher $\mu(\alpha)$ in the end.

For a better understanding of the common functions, we illustrate the calculation of F_x and F_y choosing different metrics with example URLs (one is URL spam and the other is legitimate URL). Suppose that the sliding window size $w = 2$, the state transitions in Markov Chain model are shown in Figure 48. Besides, the sending event trace of example URL spam $\sigma_s(s)$ is S_eT_e and the sending event trace of example

legitimate URL $\sigma_l(s)$ is S_eU_e. Also we suppose that the state transition probabilities obtained from training dataset (legitimate URLs) for $[\Phi, \Phi] \rightarrow [\Phi S_e]$, $[\Phi, S_e] \rightarrow [S_eT_e]$, $[\Phi, S_e] \rightarrow [S_eM_e]$, $[\Phi, S_e] \rightarrow [S_eS_e]$ and $[\Phi, S_e] \rightarrow [S_eU_e]$ are 1.0, 0.1, 0.2, 0.3, and 0.4 respectively. For state transition $s_0 \rightarrow s_1$, F_x and F_y are the same for the two examples. So we just listed out the values of F_x and F_y for the state transition $s_1 \rightarrow s_2$ for those three metrics as follows:

- *Miss-probability metric:*

  $\sigma_s(s_1)$: $F_x = 1.0$ and $F_y = 0.2 + 0.3 + 0.4 = 0.9$

  $\sigma_l(s_1)$: $F_x = 1.0$ and $F_y = 0.1 + 0.2 + 0.3 = 0.6$

- *Miss-rate metric:*

  $\sigma_s(s_1)$: $F_x = 1.0$ and $F_y = 1.0$

  $\sigma_l(s_1)$: $F_x = 1.0$ and $F_y = 0.0$

- *Local-entropy-reduction metric:*

  $LE(s_1) = -0.1log(0.1) - 0.2log(0.2) - 0.3log(0.3) - 0.4log(0.4) \approx 0.5558$

  $\sigma_s({}_1s)$: $F_x = LE(s_1) \approx 0.5558$ and $F_y = LE(s_1) + 0.1log(0.1) \approx 0.4558$

  $\sigma_l(s_1)$: $F_x = LE(s_1) \approx 0.5558$ and $F_y = LE(s_1) + 0.4log(0.4) \approx 0.3967$

6.5 Experimental Evaluation

In this section, we evaluate the effectiveness of the proposed approach for URL spam filtering in Twitter. Precision, Recall and F-measure are the three major metrics to evaluate the performance of the classifiers.

6.5.1 Tuning Parameters

In our approach, we need to tune four parameters in experiments. These parameters are: 1) The window size w; 2) Functions $F_y(s, (s, s'))$ and $F_x(s, (s, s'))$; 3) A real number Z; 4) A threshold r. The intuition of tuning parameters is to choose the parameters that maximize the difference between $\mu_l(\alpha)$ (the average $\mu(\alpha)$ for sending

event traces of all legitimate URLs) and $\mu_s(\alpha)$ (the average $\mu(\alpha)$ for sending event traces of all URL spam).

Common functions $F_y(s,(s,s'))$ and $F_x(s,(s,s'))$ and real number Z (must be bigger than 1) can be determined by iterating different function options and values since they have limited choices. Suppose $F_y(s,(s,s'))$, $F_x(s,(s,s'))$, and Z are determined, we discuss how to decide the values of the parameters w and r. We decide the window size by keeping increasing it until the difference between $\mu_l(\alpha)$ (the average $\mu(\alpha)$ for sending event traces of all legitimate URLs) and $\mu_s(\alpha)$ (the average $\mu(\alpha)$ for sending event traces of all URL spam) archives the highest value. And the threshold r will be the average of $\mu_l(\alpha)$ and $\mu_s(\alpha)$.

6.5.2 Dataset and Experimental Setup

In our experiments, the training dataset consists of 111,312 unique legitimate URL entries, which are randomly chose and occupy 70% of overall legitimate URLs. The training dataset will be used to obtain the transition probabilities in BEAN. While test dataset contains two parts:

> **Part 1**: legitimate URLs dataset consisting of 47,286 unique URL entries, which are the rest of original legitimate URLs.
>
> **Part 2**: URL spam dataset containing 19,589 unique URL entries.

The dataset will be updated as the changing of the window size w. For instance, if the window size is 2, the dataset will filter out any URL the length of whose sending event trace is less than 2.

Since common functions $F_y(s,(s,s'))$, $F_x(s,(s,s'))$ and Z are independent from other parameters, we try the three options mentioned in 5th section for common functions and increase Z by the step of 0.1 starting from 1.0 to 10.0 . We obtain the optimal setup when the difference between $\mu_l(\alpha)$ and $\mu_s(\alpha)$ achieves the highest value

with the same setup for other parameters. We find that the best result happens when the common functions are using the miss-rate metric and real number Z equals 5.0.

After that, we increase the window size w from 1 to 10. We find that the best result achieves when the window size w equals 3. We choose the window size that maximizes the difference between $\mu_l(\alpha)$ and $\mu_s(\alpha)$. We set up initial threshold $r_0 = (\mu_l(\alpha) + \mu_s(\alpha))/2$. For our data, the initial threshold r_0 equals 0.355. The threshold will keep being updated as the training dataset is updated in real-time scenario.

6.5.3 Performance Comparison

To compare our approach with traditional approaches, we use two baselines in the experiments. The first one is a machine learning classification based method. SVM-light [171] is used as the classifier to combine features from the Twitter trending topics. We choose overall 24 features based on trends, statuses, and user profiles information in Twitter dataset: the number of trending topics associated, the number of messages associated, the number of users associated, the frequency of URLs, the statistical features of distribution of messages sent by users (mean, average, max, and min), the statistical features of distribution of messages appeared in trending topics (mean, average, max, and min), the statistical features of distribution of URLs sent by users (mean, average, max, and min), the number of retweets in associated messages, the number of replies in associated messages, total number of followers of associated users, total number of followings of associated users, the statistical features of distribution of ratio between followers and followings for associated users (mean, average, max, and min), Also we use 10-fold cross-validation to prevent over-fitting in SVM classification. The second baseline is the TrustRank method based on Web link graph. Specifically, we employ the trust rank checker developed by seomastering.com team [172], which is based on the TrustRank algorithm introduced in [83], to obtain the classification results

For simplicity, we denote the above two baselines and our proposed method as *SVM*, *TrustRank*, and *BEAN*. The results of our experiments are listed in Table 40.

Table 40: Performance on spam detection

Method	Precision	Recall	F-measure
SVM	0.73	0.81	0.77
TrustRank	0.78	0.65	0.71
BEAN	0.91	0.88	0.89

Table 40 shows that our approach *BEAN* performs the best among the three approaches. *TrustRank* fails to identify a significant proportion of URLs in our dataset as spam since the URLs in our dataset do not have very typical link patterns that are detectable. We regard our proposed approach BEAN as a complementary approach to the state-of-the-art anti-spam methods.

6.5.4 Anomalous patterns

By comparing the event sequence patterns of length 3 for URL spam and legitimate URLs, we try to investigate the anomalous patterns in sending event traces of URL spam. The event sequence patterns are shown in Figure 49.

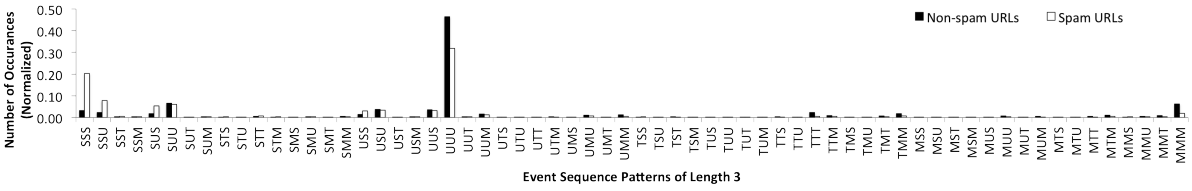
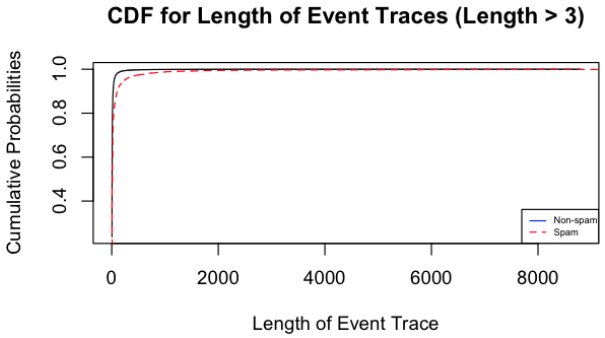


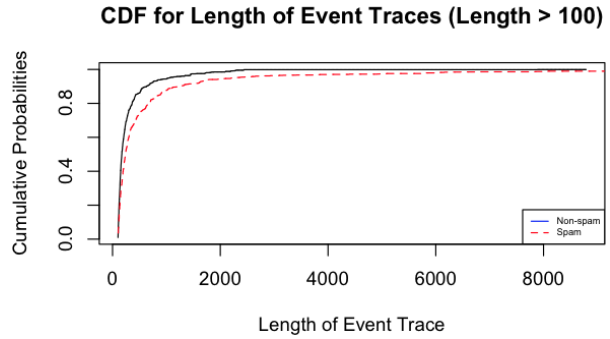
Figure 49: Event sequence patterns of length 3 for spam and non-spam URLs

Figure 49 shows that only a few patterns gain popularity such as pattern *UUU*, *SSS* and *SSU*. Here, for simplicity, S denotes seeding event, U denotes user joining event, T denotes topic spreading event, and M denotes message repeating event. Meanwhile, we observe that there exists different pattern distribution for spam URLs and legitimate URLs. For instance, the sending event traces of spam URLs contain

more *SSS*, *SSU*, *SUS*, and *USS* patterns. While the sending event trace of legitimate URLs contain more *UUU* and *MMM* patterns.



(a) Length > 3



(b) Length > 100

Figure 50: CDF for length of event traces

Moreover, we compare the length distributions of sending event traces of spam URLs and legitimate URLs of as shown in Figure 50(a) (Length > 3) and Figure 50(b) (Length > 100). We observe that there are more URL spam with a long sending event trace than legitimate URLs in terms of percentage.

6.6 Related Work

Lots of researchers work on detecting different kinds of spam such as email spam [148], web spam [137] [20] [11] [54], blog spam [109], and review spam [140], just to name a few. As URL spam has high impact on the quality of online communities, the research communities have begun to look into the detection problem more than a decade ago [37]. Popular URL spam detection algorithms include TrustRank [83],

BadRank [200] and SpamRank [21].

Also, many researchers work on social spam detection [132] [95] [85] [23]. Lee et al. [120] [121] had a long-term study on content polluters in Twitter. Zhang [207] detected spam through analyzing automated activity in Twitter. Thomas et al. [173] presented a real-time system Monarch that detects URLs. Ghosh et al. [71] investigated link farming in the Twitter network and then explored mechanisms to discourage the activity. Benevenuto et al. [164] approached the problem of detecting trending-topic spammers who include unrelated URLs with trending words in tweets. Since short URL is used often on social media recently, spammers also adopted it for camouflaging the spam URLs [15]. Klien et al. [108] proposed geographical analysis of spam in short URL spam detection. Chhabra et al. [43] presented the method to detect phishing through short URLs. Maggi et al. [130] presented security threats and countermeasures of short URLs in Twitter.

Besides, our work is related to behavior analysis in information security area. Some researchers have proposed behavior analysis approach to detect intrusion such as [102]. It is also applicable to spam detection in some cases. Liu et al. [126] proposed a spam page detection algorithm based on Bayesian Learning. Tang et al. [167] used support vector machines and random forests modeling for spam senders behavior analysis. Zhu et al. [209] proposed a novel approach to discover spammers in social networks by exploiting both social activities as well as users' social relations. Aggarwal et al. [13] detected spam tipping behaviors on Foursquare and Lim et al. [124] detected product review spammers using rating behaviors.

Compared with those related work, our work differs from them as follows: 1) our work focuses on behavior analysis instead of content analysis and link analysis in previous URL spam detection; 2) our approach only needs a few parameters derived from users' message sending behaviors, while many detection approaches normally require complex feature set; 3) our behavior analysis approach BEAN defines sending

events and formulates URL spam detection problem into Markov Chain model. Also it uses classification converter to create suitable classifiers.

6.7 Conclusion

We intensively analyzed URL spam in Twitter trending topics to explore useful methods to prevent URL spam pollution in social media. Our study shows spammers evolve to use collective spamming techniques in spreading URL spam. We defined our detection problem and explained sending events and URL associated behaviors in social media. By investigating URL behavioral states, we provided a solution that uses different strategies to detect URL spam in different behavioral states. For URLs in "Group user sending across multiple topics" and "Group user sending" states, we proposed our behavior analysis approach BEAN derived from Markov Chain model. Through the implementation and discussion of the approach, we demonstrated the feasibility of BEAN in URL spam detection and discovered anomalous patterns in sending event traces of URL spam. Our experimental results have shown that BEAN can detect a lot of URL spam that cannot be detected by conventional anti-spam approaches such as SVM and TrustRank. Therefore, our approach BEAN is a good complement to existing anti-spam solutions, especially for detecting URL spam in collective spamming activities.

CHAPTER VII

INFORMATION DIFFUSION ANALYSIS OF RUMOR DYNAMICS OVER A SOCIAL-INTERACTION BASED MODEL

In this chapter, we will introduce the information diffusion analysis on rumor dynamics in social networks. Rumor is also one kind of low quality information, which exploits trust relationships built among users and may cause serious consequences such as the widespread panic in the general public [143]. For instance, we observed the misinformation with fake satellite photos about hurricane Sandy blackout in 2012 [177] and the false report that President Obama unveils new American flag in 2013 [178].

In addition, microblogging services such as Twitter and Weibo support real-time propagation of information to a large group of users since tweets can be posted by a wide range of services: email, SMS text-messages, and Apps on smartphones. Therefore, microblogging services provides an ideal environment for the dissemination of breaking-news directly from the news sources [38]. This rapid process may not be able to separate true information from rumors. Especially, for some political or government related news, it requires investigation time to verify the news by authorities, which allows rumors to reach a larger audience.

More concretely, we make the following contributions:

- We create social-interaction based social graph model FAST which constructs a directed and weighted social graph considering social interactions. In the model, we define and quantitatively analyze four kinds of properties of social interactions including familiarity, activeness, similarity, and trustworthiness.

- We create a new metric Fractional and Directed Power Community Index (FD-PCI) to identify influential spreaders in information diffusion on microblogging services. By comparing with baselines like degrees, k-core index, and μ-PCI, we demonstrate that FD-PCI has the best performance in terms of correlation with spreading ability of nodes in real data.

- We explore influential features to detect rumors in microblogs through real data analysis. By comparing rumor dynamics with real news dynamics, it shows significant differences in the values of influential features, which indicates that those features could be used to efficiently detect rumors in real world.

The remainder of the chapter is organized as follows. Section 7.1 summarizes related work of our research. Section 7.2 describes our social-interaction based model FAST. Section 7.3 introduces the metrics FD-PCI for identification of influential spreader and shows the comparison results with baselines. Section 7.4 presents influential features in rumor detection. Finally, the section 7.5 concludes the chapter.

7.1 Related Work

Many researchers have studied on rumor or misinformation detection in social networks [34,38,96,139,143,151,203]. Castillo et al. [38] proposed automatic methods for assessing the credibility of a given set of tweets based on features from users' posting and re-posting behavior. Qazvinian et al. [151] explored the effectiveness of 3 categories of features: content-based, network-based, and microblog-specific memes for correctly identifying rumors. Budak et al. [34] addressed the problem of influence limitation where a "bad" campaign starts propagating from a certain node in the network and use the notion of limiting campaigns to counteract the effect of misinformation. Nguyen et al. [143] analyzed and presented solutions including inapproximability result, greedy algorithms that provide better lower bounds on the number of selected nodes, and a community-based heuristic method for the Node Protector problems.

Morris et al. [139] presented survey results regarding users' perceptions of tweet credibility and showed that users are poor judges of truthfulness based on content alone, and instead are influenced by heuristics such as user name when making credibility assessments. Huang et al. [96] studied the rumor spreading process with denial and skepticism. Yang et al. [203] studied the problem of information credibility on Sina Weibo, China's leading micro-blogging service provider.

Information diffusion analysis enables us to understand the properties of underlying media and communication patterns. The information diffusion model also determines how effectiveness of the approaches in rumor detection. Jin et al. [103] used the SEIZ enhanced epidemic model that explicitly recognizes skeptics to characterize eight events across the world and spanning a range of event types. Apolloni et al. [16] used a probabilistic model to decide whether two people will converse about a particular topic based on their similarity and familiarity. Wang et al. [190] proposed to use partial differential equations(PDEs) to characterize temporal and spatial patterns of information diffusion over online social networks. Guille et al. [80] proposed a practical solution which aims to predict the temporal dynamics of diffusion in social networks, which is based on machine learning techniques and the inference of time-dependent diffusion probabilities from a multidimensional analysis of individual behaviors. Myers et al. [141] presented a model in which information can reach a node via the links of the social network or through the influence of external sources.

How to identify the influential spreaders in rumor dynamics is another research problem attracting researchers' attention [19, 31, 206]. Most studies of influential spreaders have focused on their linkage with other nodes. Kitsak et al. [106] found that the degree of a node is a bad indicator of its ability to spread a message to a sufficiently large part of the network. Kitsak's team argued that a better way of quantifying influential spreaders is the node's position in a k-shell decomposition of the network's graph, and verified this hypothesis in the context of disease propagation.

However, subsequent research by Borge-Holthoefer et al. [30] proved that a node's spreading capabilities in the context of rumor spreading do not depend on its k-core index. Later on, the same team [29] introduced two mechanisms with the aim of filling the gap between theoretical and experimental results. The first model introduces the assumption that spreaders are not always active whereas the second model considers the possibility that an ignorant is not interested in spreading the rumor. In both cases, results from numerical simulations show a higher adhesion to real data than classical rumor spreading models.

7.2 Social-interaction based Model

In this section, we will discuss the social-interaction based model for information diffusion in social networks. The goal here is to find the most close-to-reality social graph model by exploring the real historical data from social networks domain.

7.2.1 Beyond Simple Graph

Due to limited resources, most of previous studies used simple graph model as the base model. In the simple graph model, they assumed that the social graph is unweighted and undirected. It means that the links of any user pair have the same strength and are based on mutual relationships. But in reality, the assumption cannot stand for most cases. Taking Twitter as an example, the links between any two users have low chance to have the same weight considering differences of users' properties and relationships between them. Also, the relationships among users are not always mutual such as the following relationship in Twitter. So if user A follows B, it does not necessarily make B to follow A in return. Therefore, simple graph model is definitely not a good model for our research purpose and we need a better model beyond it.

7.2.2 FAST Model

Based on the discussion above, we realize that simple graph model is not enough for studying problems in reality. A close-to-reality model should be a model taking the social interactions among users into account. Social interactions define the strength of the links among users and the directions of relationships. For instance, user A is familiar with user B through many conversations in the format of comments and replies. They follow each other to receive updates. But user A does not follow user C and has no direct conversation with user C even though user C is following A. Obviously, the link between user A and user B is stronger than the link between user A and user C. Thus, the social-interaction based model could distinguish different relationships among users by giving weights and directions to the links. Meanwhile, social-interaction based model is a dynamic model since the social interactions are not static but always changing over time. Based on the observations in real data, we propose a social-interaction based model named FAST (shown in Figure 51) considering four major properties of social interaction: Familiarity, Activeness, Similarity, and Trustworthiness. We will illustrate each of them in the following sections.

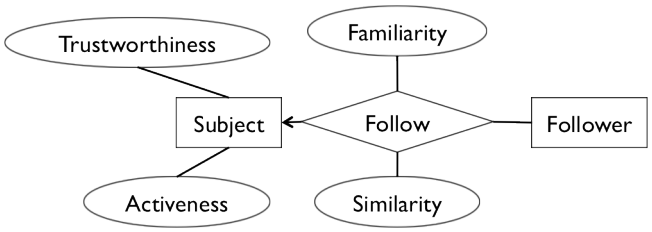


Figure 51: FAST Social Graph Model

7.2.2.1 Familiarity

Familiarity is the property to measure how familiar it is for a user toward the user's neighborhood. It is not mutual since you cannot say user A is familiar with user B if user B is familiar with user A. For example, suppose that user B is a celebrity or a famous expert in some fields and user A is a big fan of user B, user A is quite familiar

with user B's information such as interests and favorites. But user B may or may not familiar with user A. Besides this, in the context of social networks, familiarity is more related to the communication frequency between two users. The conversations between two users will increase the familiarity as we have mentioned before. Therefore, we quantify the familiarity property by using the number of contacts including comments, replies and retweets (for Twitter) over the total number of contacts for each user.

7.2.2.2 Activeness

Activeness is the property to measure how active it is for each user using the account in social networks. It is hard to measure based on users' log in time and log out time since you do not know whether the user is really active during the whole time. Thus, we use a different way to measure the activeness by counting how many days that the user has sent out message or done any other action over a period of time. For instance, if a user sends out a message, we will increase the count by one. But if the user repeats sending out messages on the same day, we will not increase the count.

7.2.2.3 Similarity

Similarity is the property to measure how similar it is between two users. It is a mutual property so two directed linked users share the same value on similarity. To quantify this property, we need to look into the features of user demographical background. The assumption is that if two directed linked users have the same gender coming from the same geo-location, they probably have high similarity than other user pairs. Simply speaking, if we only consider two factors : gender and location, the value of similarity between two users will be increased by 0.5 if one factor is the same for both users. If there is no common factor for two directed linked users, the value of similarity will be zero.

7.2.2.4 Trustworthiness

Trustworthiness is the property to measure the level of trust for each user. Many researchers have proposed reputation systems to rank users' trustworthiness. But here, we do not want to complicate our model by adopting an embedded reputation system. Thus, we simply use the verified features of users to quantify the trustworthiness. In our dataset from Sina Weibo, a user has a special feature named verified (also called Big V) that means the user has been verified by Sina Weibo service provider through specific procedure such as uploading ID card, portrait photos, and mobile phone binding. Therefore, in our model, the value of trustworthiness will be one if the user account is verified. Otherwise, it is zero.

7.2.2.5 Mathematical Model

In our FAST model, we use symbol F, A, S, and T to denote familiarity, activeness, similarity and trustworthiness respectively. Suppose we give them equal weights in the model, the final weight on each link denoted as W could be calculated in the following formula:

$$W_{ij} = F_{ij} + A_i + S_{ij} + T_i \tag{9}$$

where, W_{ij} is the weight of the link from user i to user j. $F_{i,j}$ is the value of familiarity for the link from user i to user j. A_i is the value of activeness for user i. S_{ij} is the value of similarity between user i and user j. T_i is the value of trustworthiness for user i.

For each property, we have the following formula to calculate:

$$F_{ij} = n_c/n_t \tag{10}$$

$$A_i = t_d/t_p \tag{11}$$

$$S_{ij} = f_s/f_t \tag{12}$$

$$T_i = v \tag{13}$$

where, n_c denotes the number of contacts between user i and user j through the link from i to j and n_t denotes the number of total contacts from the user i. t_d and t_p denote the number of days and the number of days in a period of time respectively. f_s and f_t represent the number of factors with the same value and the total number of common factors respectively. v is defined as boolean value whether the user i is verified or not.

7.3 Influential Spreaders in Information Diffusion

In information diffusion process, it is critical to identify influential spreaders so that we can perform some measures to prevent further information propagation. Social graph model is the fundamental part to simulate the process and obtain the accurate ranking information of influential spreaders. In the following sections, we will introduce our new metrics Fractional and Directed Power Community Index (FD-PCI) based on FAST to compare with existing metrics such as K-core index, μ-PCI, and PageRank. Also, we will illustrate the performance of our FAST model comparing with simple graph model.

7.3.1 Existing Metrics

Some researchers from physics and computer science graph theory areas have already proposed and compared many existing metrics to identify influential spreaders in information diffusion. For example, to define k-core index, we need to know the k-core concept first. The k-core of a graph G is the largest subgraph of G for which every node has a degree of at least k within the sub-graph. The coreness of a vertex is k if it belongs to the k-core but not to the (k+1)-core [1]. The approach has two major shortcomings: one is significant computational overhead, rendering it unsuitable for dynamic networks; the other is impossible to guarantee a monotonic relationship

[1] http://igraph.sourceforge.net/doc/R/graph.kcores.html

between the k-core index and a node's spreading capability.

Another metric called μ-PCI, which is short for μ power community index, is defined as follows: the μ-PCI of a node v is equal to k, such that there are up to μk nodes in the μ-hop neighborhood of v with degree greater than or equal to k, and the rest of the nodes in that neighborhood have a degree less than or equal to k. This method balances the principles of betweenness centrality and the transitive network density implied by the coreness measure. Since the calculation of the metric only depends on the neighborhood, it requires less computational overhead than k-core index.

Moreover, PageRank is an algorithm used by Google Search to rank websites in their search engine results, which also can apply for ranking user influences. Here, we will use those algorithms as the baselines to evaluate our metrics. One thing we should mention is that k-core index and μ-PCI currently can only apply for simple social graph model that does not consider the weights of each edge. Thus, we have to generate simple social graph by ignoring the weights and directions.

7.3.2 FD-PCI: Fractional and Directed PCI

We create a new metric based on μ-PCI by extending it to weighted and directed social graph model FAST. And we named it as FD-PCI that is short for Fractional and Directed PCI. Fractional means the weight on each edge may not necessarily be integer type. It is actually a floating number with fraction part in our model. Thus, we define our metrics as follows:

The FD-PCI of a node v is equal to k, such that there are up to k nodes in the 1-hop neighborhood of v with weight greater than or equal to k, and the rest of the nodes in that neighborhood have a weight less than or equal to k. Here, the weight of a node is the sum of all weights of edges connecting to this node. And k is an integer by rounding the value of weights.

7.3.3 Spreading Capability

Spreading capability of a user is the spreading influence that a use can reach. We used the average size of the network's infected area as a performance measure. To quantify $inf(s)$, the influence of a singe spreader s, we use the average size of the network infected from the spreader. It is computed as the average ratio between the number of users joining propagating the same information and total number of users in the social graph.

$$inf(s) = \frac{1}{n}\sum_{i=1}^{n} area_i \tag{14}$$

where, n is the number of information spreading process simulation. $area_i$ is the influence area of a single spreader s.

In the context of microblogging services, when a use posts or retweets a message to its neighborhood, there exist different probabilities that the neighboring users will retweet or continue to propagate the message. Therefore, we use probabilistic model to simulate the information spreading process with single information source.

In the probabilistic model, we take the retweeting probabilities as the information propagation rates for links among users. The retweeting probabilities are computed by the following formula:

$$P_r^{ij} = n_r^{ij}/n_t^{ij} \tag{15}$$

where, P_r^{ij} denotes the retweeting probability from user i to user j. n_r^{ij} and n_t^{ij} denote the number of user i's messages retweeted by user j and the number of user i's messages received by user j respectively. The calculation is based on the real dataset from Sina Weibo in the year of 2012.

The detailed steps to compute the spreading capabilities are as follows:

Step 1: Calculate the retweeting probabilities for all links.

Step 2: Iteratively compute each user node's spreading capability taking each user node as single information source.

Step 3: Repeat step 2 for 100 times to achieve statistically unbiased results.

7.3.4 Experimental Evaluation

In this section, we will illustrate the Sina Weibo dataset first. And then we will talk about the results of simulation and compare our FD-PCI metric with baseline metrics.

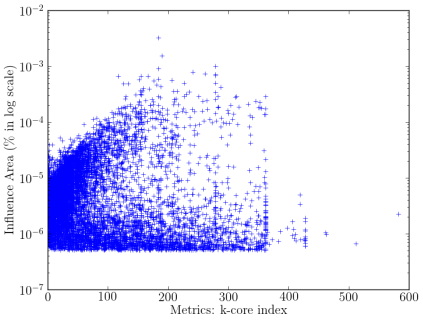
7.3.4.1 Dataset

The dataset we used is collected between January 1 and December 31, 2012 from Sina Weibo (microblogs in English) which is one of the largest microblogging service provider in China. It contains around 200 million Weibo messages and 14 million users information. We obtained the dataset from Weiboscope project hosted by the University of Hong Kong [182]. They made use of the Sina Weibo Open APIs to access the microblog data. Through User Search API, they constructed a list of popular microbloggers who have 1,000 or more followers. For those followers, they also generated a list of about 350,000 microbloggers. All the messages posted by those users are collected through the User Timeline API function. You could find more details about data collection in their work.

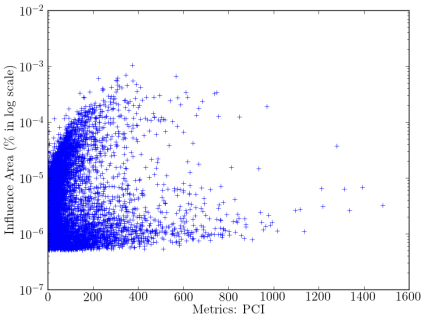
7.3.4.2 Comparison Results

Upon finished downloading the Sina Weibo dataset, we are able to compare our metric FD-PCI on the FAST model with three major baselines including k-core index, PCI, and PageRank.

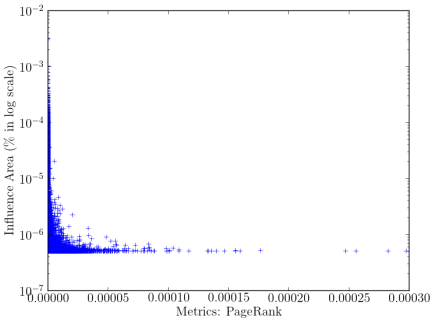
Figure 52 shows k-core index, PCI and PageRank metrics on simple social graph model. For k-core index, it has two different trends: one trend is monotonic increasing trend as the increasing of k-core index; the other is no correlation with the value of



(a) User Influence vs. K-core Index



(b) User Influence vs. μ-PCI

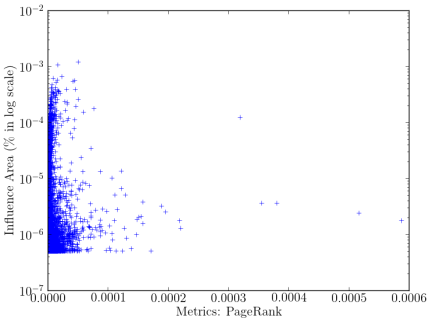


(c) User Influence vs. PageRank

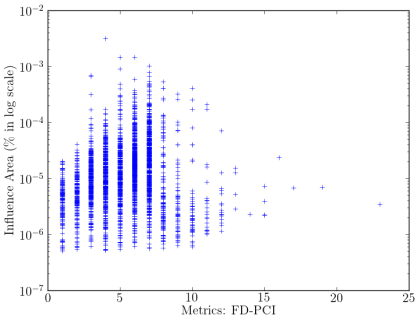
Figure 52: Performance of Metrics on Simple Social Graph Model

k-core index. For μ-PCI, it also has two trends as the increasing of μ-PCI and one trend has higher slope than the other. However, PageRank shows very low correlation with the user influence.

Next, we performed the same comparison with PageRank and FD-PCI on our FAST model. The results are shown in Figure 53.



(a) User Influence vs. PageRank



(b) User Influence vs. FD-PCI

Figure 53: Performance of Metrics on FAST Model

Figure 53 shows the results of user influence versus PageRank score and FD-PCI index on FAST model. For PageRank, it shows better correlation with user influence than the result on simple social graph model. For FD-PCI, It outperforms all other metrics in terms of correlation with user influence.

Therefore, the experiments demonstrate two major points: one is that FAST model is more practical and closer to the reality; the other is that FD-PCI is a better metric to identify influential spreaders in social networks compared with the baselines.

7.4 Influential Features in Rumor Detection

Previous section shows that our social graph model FAST is more suitable for usage in reality. We try to explore the influential features in rumor detection based on FAST model. Before that, we need to collect rumor information from Sina Weibo dataset. Thus, we will talk about the process of rumor collection, followed by influential features introduction.

7.4.1 Rumor Collection

Sina Weibo provides one special service named "piyao" that announces rumors in Sina Weibo acting as a rumor remover. Due to this service, we are able to extract many rumors with high impact from the dataset. Since the dataset covers the whole year of 2012, we looked into the announcements in 2012 from "piyao" service and categorized the rumors based on topics (shown in Table 41).

Table 41 shows 21 rumors identified by "piyao" service in the year of 2012. Based on the announcements and the original messages, we categorized the rumors into 5 major topics: Politics / Law / Government, Crime, Celebrities, Disaster, and Food Safety. We found that more than half of them contain fakery images which are either real but irrelevant images or fake images. Meanwhile, we approximately counted the lasting times of rumors on different topics. It is easy to observe that only topic Food Safety lasts more than 5 days and the others are all less than 2 days. The main

Table 41: Categories of Rumors

Topic	**Count**	**Rumors with Fakery Image**	**Approximate Average Lasting Time (Hours)**
Politics / Law / Government	10	5	24
Crime	9	6	36
Celebrities	2	0	36
Disaster	1	1	12
Food Safety	1	1	120

reason is that the rumor on food safety requires more investigation time by officials for verifying the information. Based on the samples from those rumors, we will talk about the influential features which shows difference between rumors and real news.

7.4.2 Influential Features

To distinguish rumors from real news, we need to figure out influential features to do classification and labeling. Our FAST model specifies four major properties of social graph which could be used as features for rumor detection. We listed possible features for testing on examples in Table 42.

Table 42: Notations and Descriptions of Possible Features

Notation	**Description**
C_s	Count of information sources/users
C_v	Count of verified information sources/users
$\overline{A}$	Average activeness
$\overline{S}$	Average similarity
$\overline{T}$	Average trustworthiness

Here, information sources means independent users who have sent messages originally not retweeted others' messages. Meanwhile, users in the information sources are not direct connected. In data processing, we extracted the information sources based on the type of messages and independence with existing information source set.

If the message is not a retweeted message and has no "RT" in the body, we consider it as original message. And the sender, who has no direct relationship with existing information source set, will be included in our information source set. Due to that, we do not use average familiarity as one possible feature.

After selecting possible features, we extracted example rumors from 21 rumors identified by "piyao" service and also took real news from Sina Weibo dataset as control. The criteria for selecting examples are: one is that it should have more than 500 relevant messages; the other is that it should be propagated more than 2 days. Since the original messages are in Chinese language, we list a few examples of their translations in English language as follows:

Example rumor: Attention! Friends who like eating ShaXian steamed wonton or dumplings. Some netizens saw those people made wonton or dumplings stuffing with pork tumor. It is the tumor lesions or lymph nodes in pigs with the virus, which is not edible. In fact, more than ShaXian, many stores use pork tumors for stuffing because of low cost and high interest! Dirty money! (No wonder ShaXian food is so cheap yet while the prices are soaring).

Example real news: February 6, 2012, the former vice mayor of Chongqing, Wang Lijun entered the U.S. consulate in Chengdu. He stayed there for a day and was under investigation by China government afterwards.

Based on possible features, we calculated the feature values of example rumors and real news (shown in Table 43).

Table 43: Comparison results between example rumor and real news

Feature	**Example Rumor**	**Example real news**
C_s	41	544
C_v	10	413
$\overline{A}$	0.7463	0.6833
$\overline{S}$	0.4451	0.4430
$\overline{T}$	0.2439	0.7592

Table 43 shows the results of comparison between example rumors and example

real news. It shows that rumor has fewer information sources and verified users. Additionally, rumor has lower value on the average trustworthiness feature than real news. But it has higher value on average activeness and average influence area features. They does not show much difference on average similarity feature. Since average influence area requires computing based on the whole social graph which is not practical in reality, the potential influential features for rumor detection should be count of information sources, count of verified information sources, average activeness and average trustworthiness.

7.5 Conclusion

We have proposed a new social graph model called FAST by considering the four major properties of social interactions: Familiarity, Activeness, Similarity and Trustworthiness. Through the experiments on real dataset from Sina Weibo, we demonstrated that our social interaction based model FAST outperforms simple social graph model in terms of closeness to reality. Meanwhile, we create a new metric called FD-PCI based on PCI index to identify influential spreaders on weighted and direct social graph. Taking k-core index, PCI, and PageRank as baselines, we showed that FD-PCI has high correlation and monotonic relationship with users' information spreading capability. While k-core index and PCI are not suitable for weighted and direct social graph model. PageRank has low performance in terms of correlation with users' information spreading capability. Moreover, we have explored possible features to distinguish rumors from real news and found that information source features including count of information sources, count of verified information sources, average activeness and average trustworthiness show big differences for rumors and real news. Thus, we could use them as influential features to detect rumors in social networks.

CHAPTER VIII

CONCLUSIONS

In this dissertation, I have provided viable approaches to analyze and detect low quality information in social networks. More specifically, I have contributed in three major parts: analytics and detection framework of low quality information, evolutionary study of low quality information, and detection approaches of low quality information.

In Part I, I have proposed a social spam detection framework for multiple social networks. It deals with various types of objects in social networks by transforming them into uniform schemata. Through the uniform schemata, we are able to perform cross-domain classification and associated classification to significantly improve the performance of social spam detection. The results of experiments based on real datasets show that our classification improved accuracy and F-measure by 7% - 10%.

In Part II, I have performed evolutionary study of web spam and email spam. The reasons why I choose web page spam and email spam are as follows: 1) web page spam is the most popular and common spam on Internet. And it is still popular in social networks appearing as URL link spam. 2) email system is the fundamental communication system supporting social networks and the trend of email spam shows the direction of low quality information to some extent. Through intensive study on large-scale and long term real spam data including content analysis of spam and behavior analysis of spammer, it shows that content and topics changes over time, and spammers become more sophisticated and capricious. Only one particular analysis such as content-based analysis is not sufficient to detect spam any more. We need evolution-resistant spam detection approaches.

In Part III, based on observations in real data, I have designed three novel detection approaches to detect low quality information in social networks: click traffic analysis of short URL spam, behavior analysis of URL spam and information diffusion analysis of rumor. They involve monitoring and analysis of collective activities of legitimate users and malicious users in social networks using machine learning and intrusion detection techniques. Click traffic analysis shows that URL spam has different click traffic patterns from legitimate URLs. And it also demonstrates that random forest algorithm has the best performance in the classification comparing to other popular algorithms. Behavior analysis investigates users' sending message behaviors in social networks. By modeling users' behaviors based Markov Chain model, we are able to classify URL spam through the event traces of URL links. It indicates that spammers have abnormal behavior patterns being captured by our classifier. Comparing to two baselines: SVM machine learning algorithm and TrustRank link graph analysis algorithm, our approach outperforms both of them in terms of accuracy and F-measure. Thus, BEAN approach is considered as a good complement to existing URL spam filtering techniques. Information diffusion analysis creates a new social graph model FAST based on four properties in person-to-person communication including familiarity, activeness, similarity, and trustworthiness. Also by simulating the user influences in the information propagation based on real data from Sina Weibo, we are able to compare our new metric FD-PCI to existing metrics μ-PCI, K-core index, and PageRank. Our metric shows higher correlation to user influences, which implies it has better performance in identifying influential spreaders. Meanwhile, FD-PCI is suitable to weighted and directed social graph model that cannot be applied by some other metrics. Furthermore, I have demonstrated useful features in distinguishing rumors from real news in terms of information sources.

REFERENCES

[1] "Multipurpose internet mail extensions (mime) part one: Format of internet message bodies." http://tools.ietf.org/html/rfc2045, 1996.

[2] "Electronic communications and transactions act, 2002." http://www.internet.org.za/ect_act.html, 2002.

[3] "Defensive aids sub system (dass)." http://www.eurofighter.com/capabilities/technology/sensor-fusion/defensive-aids-sub-system.html, 2013.

[4] "Gephi, an open source graph visualization and manipulation software." http://gephi.org, 2013.

[5] "Kaspersky lab." http://usa.kaspersky.com/, 2013.

[6] "Maxmind - ip geolocation and online fraud prevention." http://www.maxmind.com/en/home, 2013.

[7] "Python: email an email and mime handling package." http://docs.python.org/2/library/email, 2013.

[8] "Text::ngrams - flexible ngram analysis (for characters, words, and more)." http://search.cpan.org/dist/Text-Ngrams/Ngrams.pm, 2013.

[9] "Untroubled website." http://untroubled.org/spam/, 2013.

[10] "Visualization: Geochart - google charts google developers." https://developers.google.com/chart/interactive/docs/gallery/geochart, 2013.

[11] Abernethy, J., Chapelle, O., and Castillo, C., "Web spam identification through content and hyperlinks," in *AIRWeb*, pp. 41–44, 2008.

[12] About Twitter's Link Service. http://support.twitter.com//entries/109623, 2013. Accessed on August. 1, 2013.

[13] Aggarwal, A., Almeida, J. M., and Kumaraguru, P., "Detection of spam tipping behaviour on foursquare," in *WWW (Companion Volume)*, pp. 641–648, 2013.

[14] Amitay, E., Carmel, D., Darlow, A., Lempel, R., and Soffer, A., "The connectivity sonar: detecting site functionality by structural patterns," in *In Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*, pp. 38–47, ACM Press, 2003.

[15] Antoniades, D., Polakis, I., Kontaxis, G., Athanasopoulos, E., Ioannidis, S., Markatos, E. P., and Karagiannis, T., "we.b: the web of short urls," in *Proceedings of the 20th international conference on World wide web*, WWW '11, pp. 715–724, 2011.

[16] Apolloni, A., Channakeshava, K., Durbeck, L., Khan, M., Kuhlman, C., Lewis, B., and Swarup, S., "A study of information diffusion over a realistic social network model," in *Computational Science and Engineering, 2009. CSE '09. International Conference on*, vol. 4, pp. 675–682, 2009.

[17] Attenberg, J., Weinberger, K., and Dasgupta, A., "Collaborative Email-Spam Filtering with the Hashing Trick," in *CEAS*, pp. 1–4, 2009.

[18] Balakumar, M. and Vaidehi, V., "Ontology based classification and categorization of email," in *Proceedings of Signal Processing, Communications and Networking*, pp. 199–202, 2008.

[19] Basaras, P., Katsaros, D., and Tassiulas, L., "Detecting influential spreaders in complex, dynamic networks," *Computer*, vol. 46, no. 4, pp. 24–29, 2013.

[20] Becchetti, L., Castillo, C., Donato, D., Baeza-Yates, R. A., and Leonardi, S., "Link analysis for web spam detection," *ACM Trans. Web*, vol. 2, no. 1, 2008.

[21] Benczur, A. A., Csalogany, K., Sarlos, T., Uher, M., and Uher, M., "Spamrank - fully automatic link spam detection," in *In Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005.

[22] Benevenuto, F., Haddadi, H., and Gummadi, K., "The World of Connections and Information Flow in Twitter," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, pp. 991–998, July 2012.

[23] Benevenuto, F., Magno, G., Rodrigues, T., and Almeida, V., "Detecting spammers on twitter," in *Proceedings of the Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, 2010.

[24] Benevenuto, F., Rodrigues, T., Veloso, A., Almeida, J. M., Gonçalves, M. A., and Almeida, V. A. F., "Practical detection of spammers and content promoters in online video sharing systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 42, no. 3, pp. 688–701, 2012.

[25] Biggio, B., Fumera, G., and Roli, F., "Evade hard multiple classifier systems," in *Applications of Supervised and Unsupervised Ensemble Methods* (Okun, O. and Valentini, G., eds.), vol. 245 of *Studies in Computational Intelligence*, pp. 15–38, Springer Berlin Heidelberg, 2009.

[26] Bird, C., Gourley, A., and Devanbu, P., "Mining email social networks," in *the 2006 international workshop on Mining software repositories*, pp. 137–143, 2006.

[27] Bitly Enterprise. `http://www.enterprise.bitly.com/`, 2013. Accessed on August. 1, 2013.

[28] Blei, D. M., Ng, A., and Jordan, M., "Latent dirichlet allocation," *JMLR*, vol. 3, pp. 993–1022, 2003.

[29] Borge-Holthoefer, J., Meloni, S., Gonçalves, B., and Moreno, Y., "Emergence of influential spreaders in modified rumor models," *Journal of Statistical Physics*, vol. 151, no. 1-2, pp. 383–393, 2013.

[30] Borge-Holthoefer, J. and Moreno, Y., "Absence of influential spreaders in rumor dynamics," *CoRR*, 2011.

[31] Borge-Holthoefer, J., Rivero, A., and Moreno, Y., "Locating privileged information spreaders during political protests on an online social network," *CoRR*, vol. abs/1111.4181, 2011.

[32] Bosma, M., Meij, E., and Weerkamp, W., "A framework for unsupervised spam detection in social networking sites," in *ECIR 2012: 34th European Conference on Information Retrieval*, (Barcelona), pp. 364–375, 2012.

[33] Boykin, P. and Roychowdhury, V., "Personal email networks: An effective anti-spam tool," in *arXiv preprint cond-mat/0402143*, vol. 90095, 2004.

[34] Budak, C., Agrawal, D., and El Abbadi, A., "Limiting the spread of misinformation in social networks," in *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pp. 665–674, 2011.

[35] Byun, B., Lee, C., Webb, S., Irani, D., and Pu, C., "An anti-spam filter combination framework for text-and-image emails through incremental learning," in *Proceedings of the the Sixth Conference on Email and Anti–Spam (CEAS 2009)*, 2009.

[36] Carreras, X. and Marquez, L., "Boosting trees for anti-spam email filtering," *arXiv preprint cs/0109015*, 2001.

[37] Castillo, C., Donato, D., Becchetti, L., Boldi, P., Leonardi, S., Santini, M., and Vigna, S., "A reference collection for web spam," *SIGIR Forum*, vol. 40, pp. 11–24, Dec. 2006.

[38] Castillo, C., Mendoza, M., and Poblete, B., "Information credibility on twitter," in *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pp. 675–684, 2011.

[39] Caverlee, J., Liu, L., and Webb, S., "Socialtrust: tamper-resilient trust establishment in online communities.," in *Proceedings of the 8th ACM/IEEE-CS joint conference on Digitial libraries*, 2008.

[40] Caverlee, J. and Webb, S., "A large-scale study of MySpace: Observations and implications for online social networks," *Proceedings of the International Conference on Weblogs and Social Media*, vol. 8, 2008.

[41] Chan, J., Koprinska, I., and Poon, J., "Co-training with a Single Natural Feature Set Applied to Email Classification," in *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)*, pp. 586–589, Ieee, 2004.

[42] Chandrinos, K., Androutsopoulos, I., Paliouras, G., and Spyropoulos, C. D., "Automatic web rating: Filtering obscene content on the web," in *Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries*, ECDL '00, (London, UK, UK), pp. 403–406, Springer-Verlag, 2000.

[43] Chhabra, S., Aggarwal, A., Benevenuto, F., and Kumaraguru, P., "Phi.sh/$ocial: the phishing landscape through short urls," in *Proceedings of the Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, 2011.

[44] Chinavle, D., Kolari, P., Oates, T., and Finin, T., "Ensembles in adversarial classification for spam," in *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, (New York, NY, USA), pp. 2015–2018, ACM, 2009.

[45] Chinchor, N., "Muc-4 evaluation metrics," in *The Fourth Message Understanding Conference*, 1992.

[46] Chung, Y., *A Study on the Evolution and Emergence of Web Spam.* PhD thesis, Univ. of Tokyo, Tokyo, Japan, 2011.

[47] Clayton, R., "Stopping Spam by Extrusion Detection.," in *Proceedings of the First Conference on Email and Anti-Spam (CEAS 2004)*, (Mountain View, CA, USA), July 2004.

[48] Clayton, R., "Email traffic: a quantitative snapshot," in *the 4th Conference on Email and Anti-Spam (CEAS 2007)*, (Mountain View, CA, USA), 2007.

[49] Cournane, A. and Hunt, R., "An analysis of the tools used for the generation and prevention of spam," *Computers & Security*, vol. 23, no. 2, pp. 154 – 166, 2004.

[50] Cranor, L. F. and LaMacchia, B. A., "Spam!," *ACM Communications*, vol. 41, pp. 74–83, Aug. 1998.

[51] Cullity, B. D., *Introduction to Magnetic Materials*. Reading, MA: Addison-Wesley, 1972.

[52] Culotta, A., Bekkerman, R., and McCallum, A., "Extracting social networks and contact information from email and the web," in *Proceedings of the First Conference on Email and Anti-Spam (CEAS 2004)*, 2004.

[53] Cutts, M., "Google blog: Using data to fight webspam." `http://googleblog.blogspot.com/2008/06/using-data-to-fight-webspam.html`, 2013.

[54] Dai, N., Davison, B. D., and Qi, X., "Looking into the past to better classify web spam," in *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb '09, pp. 1–8, 2009.

[55] Dalvi, N., Domingos, P., Mausam, Sanghai, S., and Verma, D., "Adversarial classification," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, (New York, NY, USA), pp. 99–108, ACM, 2004.

[56] Dasgupta, A., Gurevich, M., and Punera, K., "Enhanced email spam filtering through combining similarity graphs," in *Proceedings of the fourth ACM international conference on Web search and data mining*, (New York, NY, USA), p. 785, ACM Press, 2011.

[57] Davison, B. D., "Recognizing nepotistic links on the web," in *In AAAI-2000 Workshop on Artificial Intelligence for Web Search*, pp. 23–28, AAAI Press, 2000.

[58] Delorme, F. and others, "Butt-jointed DBR laser with 15 nm tunability grown in three MOVPE steps," *Electron. Lett.*, vol. 31, no. 15, pp. 1244–1245, 1995.

[59] Drost, I. and Scheffer, T., "Thwarting the nigritude ultramarine: learning to identify link spam," in *In Proceedings of the 16th European Conference on Machine Learning (ECML*, pp. 233–243, 2005.

[60] Drucker, H., Wu, D., and Vapnik, V., "Support vector machines for spam categorization," *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 1048–1054, 1999.

[61] Fallows, D., "Spam. how it is hurting email and degrading life on the internet," technical report, the Pew Internet & American Life project, Washington, DC, USA, October 2003.

[62] Fawcett, T., ""in vivo" spam filtering: a challenge problem for kdd," *SIGKDD Explor. Newsl.*, vol. 5, pp. 140–148, Dec. 2003.

[63] Fazeen, M., Dantu, R., and Guturu, P., "Identification of leaders, lurkers, associates and spammers in a social network: context-dependent and context-independent approaches," *Social Network Analysis and Mining*, vol. 1, no. 3, pp. 241–254, 2011.

[64] Fetterly, D., Manasse, M., and Najork, M., "Spam, damn spam, and statistics: using statistical analysis to locate spam web pages," in *Proceedings of the 7th International Workshop on the Web and Databases: colocated with ACM SIGMOD/PODS 2004*, WebDB '04, 2004.

[65] Fetterly, D., Manasse, M., and Najork, M., "Detecting phrase-level duplication on the world wide web," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, 2005.

[66] Fetterly, D., Manasse, M., Najork, M., and Wiener, J., "A large-scale study of the evolution of web pages," in *Proceedings of the 12th international conference on World Wide Web*, WWW '03, (New York, NY, USA), pp. 669–678, 2003.

[67] Forman, G., "An extensive empirical study of feature selection metrics for text classification," *J. Mach. Learn. Res.*, vol. 3, pp. 1289–1305, Mar. 2003.

[68] Friedman, J., Hastie, T., and Tibshirani, R., "Additive logistic regression: a statistical view of boosting," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[69] Gao, H., Hu, J., Wilson, C., Li, Z., Chen, Y., and Zhao, B. Y., "Detecting and characterizing social spam campaigns," in *Proceedings of the Internet Measurement Conference*, 2010.

[70] Garcia-Molina, H., Ullman, J. D., and Widom, J., *Database System Implementation*. China Machine Press, 2002.

[71] Ghosh, S., Viswanath, B., Kooti, F., Sharma, N. K., Korlam, G., Benevenuto, F., Ganguly, N., and Gummadi, K. P., "Understanding and combating link farming in the twitter social network," in *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pp. 61–70, 2012.

[72] Giatsidis, C., Malliaros, F. D., and Vazirgiannis, M., "Advanced graph mining for community evaluation in social networks and the web," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pp. 771–772, 2013.

[73] Goodman, J., Cormack, G. V., and Heckerman, D., "Spam and the ongoing battle for the inbox," *ACM Communications*, vol. 50, pp. 24–33, Feb. 2007.

[74] "Google opensocial api," 2011. http://code.google.com/apis/opensocial/.

[75] Google Safe Browsing API. https://developers.google.com/safe-browsing/, 2013. Accessed on August. 1, 2013.

[76] Gosier and Guadeloupe, "Social networks as an attack platform: Facebook case study," in *Proceedings of the Eighth International Conference on Networks*, 2009.

[77] Grier, C., Thomas, K., Paxson, V., and Zhang, C. M., "@spam: the underground on 140 characters or less," in *Proceedings of the ACM Conference on Computer and Communications Security*, 2010.

[78] Gudkova, D., "Kaspersky security bulletin: Spam evolution 2012." http://www.securelist.com/en/analysis/204792276/Kaspersky_Security_Bulletin_Spam_Evolution_2012, 2012.

[79] Guerra, P. and Guedes, D., "Exploring the spam arms race to characterize spam evolution," in *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS 2010)*, (Redmond, Washington USA), July 2010.

[80] Guille, A. and Hacid, H., "A predictive model for the temporal dynamics of information diffusion in online social networks," in *Proceedings of the 21st International Conference Companion on World Wide Web*, WWW '12 Companion, pp. 1145–1152, 2012.

[81] Gupta, R. K. and Senturia, S. D., "Pull-in time dynamics as a measure of absolute pressure," in *Proc. IEEE International Workshop on Microelectromechanical Systems (MEMS'97)*, (Nagoya, Japan), pp. 290–294, Jan. 1997.

[82] Gyöngyi, Z. and Garcia-Molina, H., "Web spam taxonomy," in *Proceedings of 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, (Chiba, Japan), May 2005.

[83] Gyöngyi, Z., Garcia-Molina, H., and Pedersen, J., "Combating web spam with trustrank," in *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pp. 576–587, VLDB Endowment, 2004.

[84] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I., "The WEKA data mining software," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[85] Hameed, S., Fu, X., Hui, P., and Sastry, N. R., "Lens: Leveraging social networking and trust to prevent spam transmission," in *ICNP*, pp. 13–18, 2011.

[86] Han, B. and Baldwin, T., "Lexical normalisation of short text messages: makn sens a #twitter," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, (Stroudsburg, PA, USA), pp. 368–378, Association for Computational Linguistics, 2011.

[87] Han, J. S. and Park, B. J., "Efficient detection of content polluters in social networks," in *ICITCS*, pp. 991–996, 2012.

[88] Hao, S., Syed, N. A., Feamster, N., Gray, A. G., and Krasser, S., "Detecting spammers with snare: Spatio-temporal network-level automatic reputation engine," in *Proceedings of the 18th Conference on USENIX Security Symposium*, SSYM'09, (Berkeley, CA, USA), pp. 101–118, 2009.

[89] Hayati, P., Potdar, V., Talevski, A., Firoozeh, N., Sarenche, S., and Yeganeh, E., "Definition of spam 2.0: New spamming boom," in *Proceedings of the 4th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, pp. 580–584, 2010.

[90] Hayati, P. and Potdar, V., "Evaluation of spam detection and prevention frameworks for email and image spam," in *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, (New York, NY, USA), p. 520, ACM Press, 2008.

[91] He, Q., Zhuang, F., Li, J., and Shi, Z., "Parallel implementation of classification algorithms based on mapreduce," in *Rough Set and Knowledge Technology*, vol. 6401 of *Lecture Notes in Computer Science*, pp. 655–662, 2010.

[92] Heymann, P., Koutrika, G., and Garcia-Monlina, H., "Fighting spam on social web sites: A survey of approaches and future challenges," *IEEE Internet Computing*, vol. 51, no. 4, pp. 36–45, 2007.

[93] Hirai, J., Raghavan, S., Garcia-Molina, H., and Paepcke, A., "WebBase: A repository of web pages," *Computer Networks*, vol. 33, no. 1-6, pp. 277–293, 2000.

[94] "HOOTSUITEsocial media dashboard," 2011. http://hootsuite.com/.

[95] Hu, X., Tang, J., Zhang, Y., and Liu, H., "Social spammer detection in microblogging," in *IJCAI*, 2013.

[96] Huang, W., "On rumour spreading with skepticism and denial," tech. rep., Shanghai Jiao Tong University, 2010.

[97] Internet Engineering Task Force (IETF), "Rfc 5988." `http://tools.ietf.org/html/rfc5988`, 2013.

[98] Irani, D., Webb, S., Giffin, J., and Pu, C., "Evolutionary study of phishing," *eCrime Researchers Summit, 2008*, pp. 1–10, 2008.

[99] Irani, D., Webb, S., Pu, C., and Li, K., "Study of trend-stuffing on twitter through text classification," in *Proceedings of the Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, CEAS '10, 2010.

[100] Jensen, D., Neville, J., and Gallagher, B., "Why collective inference improves relational classification," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pp. 593–598, 2004.

[101] Jesse Alpert and Nissan Hajaj, "Google blog: We know the web was big." `http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html`, 2013.

[102] Jha, S., Tan, K., and Maxion, R. A., "Markov chains, classifiers, and intrusion detection," in *Proceedings of the 14th IEEE workshop on Computer Security Foundations*, CSFW '01, (Washington, DC, USA), IEEE Computer Society, 2001.

[103] Jin, F., Dougherty, E., Saraf, P., Cao, Y., and Ramakrishnan, N., "Epidemiological modeling of news and rumors on twitter," in *Proceedings of the 7th Workshop on Social Network Mining and Analysis*, SNAKDD '13, pp. 8:1–8:9, 2013.

[104] Jin, X., Lin, C. X., Luo, J., and Han, J., "Socialspamguard: A data mining-based spam detection system for social media networks," in *Proceedings of the International Conference on Very Large Data Bases*, 2011.

[105] Jung, J. and Sit, E., "An empirical study of spam traffic and the use of DNS black lists," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, (New York, NY, USA), p. 370, ACM Press, 2004.

[106] Kitsak, M., Gallos, L., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H., and Makse, H., "Identification of influential spreaders in complex networks ," *Nature Physics*, vol. 6, pp. 888–893, Aug 2010.

[107] Kleinberg, J. M., "Distributed knowledge networks. design, implementation, and applications," in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[108] Klien, F. and Strohmaier, M., "Short links under attack: Geographical analysis of spam in a url shortener network," in *Proceedings of the ACM Conference on Hypertext and Hypermedia*, 2012.

[109] Kolari, P., Java, A., Finin, T., Oates, T., and Joshi, A., "Detecting spam blogs: A machine learning approach," in *AAAI*, pp. 1351–1356, 2006.

[110] Kreibich, C., Kanich, C., Levchenko, K., Enright, B., Voelker, G., Paxson, V., and Savage, S., "On the spam campaign trail," in *Proceedings*

of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, pp. 1–9, USENIX Association, 2008.

[111] Kullback, S. and Leibler, R. A., "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[112] Kwak, H., Lee, C., Park, H., and Moon, S. B., "What is twitter, a social network or a news media?," in *Proceedings of the International Conference on World Wide Web*, 2010.

[113] Lakhina, A., Crovella, M., and Diot, C., "Characterization of network-wide anomalies in traffic flows," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, IMC '04, pp. 201–206, 2004.

[114] Lakhina, A., Crovella, M., and Diot, C., "Diagnosing network-wide traffic anomalies," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '04, pp. 219–230, 2004.

[115] Lakhina, A., Crovella, M., and Diot, C., "Mining anomalies using traffic feature distributions," in *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '05, 2005.

[116] Landwehr, C. E., Boneh, D., Mitchell, J. C., Bellovin, S. M., Landau, S., and Lesk, M. E., "Privacy and cybersecurity: The next 100 years," *Proceedings of the IEEE*, vol. 100, no. Centennial-Issue, pp. 1659–1673, 2012.

[117] Las-Casas, P. H., Guedes, D., Almeida, J. M., Viviani, A., and Marques-Neto, H. T., "Spades: Detecting spammers at the source network," *Computer Networks*, vol. 57, no. 2, pp. 526 – 539, 2013.

[118] Learmonth, M., "Twitter getting serious about spam issue," 2010. http://adage.com/article/digital/digital-marketing-twitter-spam-issue/142800/.

[119] Lee, K., Caverlee, J., Kamath, K. Y., and Cheng, Z., "Detecting collective attention spam," in *Proceedings of the 2Nd Joint WICOW/AIRWeb Workshop on Web Quality*, WebQuality '12, (New York, NY, USA), pp. 48–55, 2012.

[120] Lee, K., Caverlee, J., and Webb, S., "Uncovering social spammers: social honeypots + machine learning," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pp. 435–442, 2010.

[121] Lee, K., Eoff, B. D., and Caverlee, J., "Seven months with the devils: A long-term study of content polluters on twitter," in *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2011.

[122] Lewis, D., "Naive (bayes) at forty: The independence assumption in information retrieval," in *Proceedings of 10th European Conference on Machine Learning (ECML-98)*, (Springer Verlag, Heidelberg, DE), pp. 4–15, August 1998.

[123] Lex, E., Seifert, C., Granitzer, M., and Juffinger, A., "Efficient cross-domain classification of weblogs," *International Journal of Intelligent Computing Research*, vol. 1, no. 1, pp. 36–45, 2010.

[124] Lim, E.-P., Nguyen, V.-A., Jindal, N., Liu, B., and Lauw, H. W., "Detecting product review spammers using rating behaviors," in *CIKM*, pp. 939–948, 2010.

[125] Liu, W. and Wang, T., "Multi-field learning for email spam filtering," in *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), p. 745, ACM Press, 2010.

[126] Liu, Y., Cen, R., Zhang, M., Ma, S., and Ru, L., "Identifying web spam with user behavior analysis," in *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, AIRWeb '08, (New York, NY, USA), pp. 9–16, ACM, 2008.

[127] Liu, Y., Zhang, M., Ma, S., and Ru, L., "User behavior oriented web spam detection," in *Proceedings of the 17th international conference on World Wide Web*, WWW '08, 2008.

[128] Ma, Y., Wang, L., and Li, L., "A parallel and convergent support vector machine based on mapreduce," in *Computer Engineering and Networking*, vol. 277 of *Lecture Notes in Electrical Engineering*, pp. 585–592, Springer International Publishing, 2014.

[129] MAAWG, "Email Metrics Report 2011," tech. rep., November 2011.

[130] Maggi, F., Frossi, A., Zanero, S., Stringhini, G., Stone-Gross, B., Kruegel, C., and Vigna, G., "Two years of short urls internet measurement: security threats and countermeasures," in *Proceedings of the 22nd international conference on World Wide Web*, WWW '13, (Republic and Canton of Geneva, Switzerland), pp. 861–872, International World Wide Web Conferences Steering Committee, 2013.

[131] Maia, M., Almeida, J., and Almeida, V., "Identifying user behavior in online social networks," in *the 1st workshop on Social network*, (New York, New York, USA), pp. 1–6, ACM Press, 2008.

[132] Markines, B., Cattuto, C., and Menczer, F., "Social spam detection," in *Proceedings of the Fifth International Workshop on Adversarial Information Retrieval on the Web (AIRWeb 2009)*, pp. 41–48, 2009.

[133] McAfee SiteAdvisor. http://www.siteadvisor.com/, 2013. Accessed on August. 1, 2013.

[134] McCallum, A., Wang, X., and Corrada-Emmanuel, A., "Topic and role discovery in social networks with experiments on enron and academic email.," *J. Artif. Intell. Res.(JAIR)*, vol. 30, pp. 249–272, 2007.

[135] McCallum, A. K., "Mallet: A machine learning for language toolkit." http://mallet.cs.umass.edu, 2002.

[136] McNamara, T. J., *Key concepts in mathematics: strengthening standards practice in grades 6-12 (2nd ed.)*. Corwin Press Inc., 2007.

[137] Metaxas, P. T. and DeStefano, J., "Web spam, propaganda and trust," in *AIRWeb*, pp. 70–78, 2005.

[138] Modi, S., "Relational classification using multiple view approach with voting," *International Journal of Computer Applications*, vol. 70, pp. 31–36, May 2013. Published by Foundation of Computer Science, New York, USA.

[139] Morris, M. R., Counts, S., Roseway, A., Hoff, A., and Schwarz, J., "Tweeting is believing?: Understanding microblog credibility perceptions," in *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pp. 441–450, 2012.

[140] Mukherjee, A., Liu, B., Wang, J., Glance, N. S., and Jindal, N., "Detecting group review spam," in *WWW (Companion Volume)*, pp. 93–94, 2011.

[141] Myers, S. A., Zhu, C., and Leskovec, J., "Information diffusion and external influence in networks," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pp. 33–41, 2012.

[142] Neumann, A., Barnickel, J., and Meyer, U., "Security and privacy implications of url shortening services," in *WEB 2.0 Security & Privacy Workshop(W2SP)*, 2011.

[143] Nguyen, N. P., Yan, G., Thai, M. T., and Eidenbenz, S., "Containment of misinformation spread in online social networks," in *Proceedings of the 3rd Annual ACM Web Science Conference*, WebSci '12, pp. 213–222, 2012.

[144] Ntoulas, A., Najork, M., Manasse, M., and Fetterly, D., "Detecting spam web pages through content analysis," in *Proceedings of the 15th international conference on World Wide Web*, WWW '06, 2006.

[145] Pan, S. J., Ni, X., Sun, J.-T., Yang, Q., and Chen, Z., "Cross-domain sentiment classification via spectral feature alignment," in *Proceedings of the 19th international conference on World wide web*, WWW '10, pp. 751–760, 2010.

[146] Park, S. Y., Kim, J.-T., and Kang, S.-G., "Analysis of applicability of traditional spam regulations to voip spam," in *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference*, vol. 2, pp. 3 pp.–1217, 2006.

[147] Pfleeger, S. and Bloom, G., "Canning spam: Proposed solutions to unwanted email," in *Security & Privacy, IEEE*, 2005.

[148] Preece, J., Lazar, J., Churchill, E., de Graaff, H., Friedman, B., and Konstan, J., "Spam, spam, spam, spam: How can we stop it," in *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, (New York, NY, USA), pp. 706–707, 2003.

[149] Pu, C. and Webb, S., "Observed trends in spam construction techniques: a case study of spam evolution," in *Proceedings of the Third Conference on Email and Anti-Spam (CEAS 2006)*, 2006.

[150] Pu, C., Webb, S., Kolesnikov, O., Lee, W., and Lipton, R., "Towards the integration of diverse spam filtering techniques," in *Proceedings of the IEEE International Conference on Granular Computing (GrC06)*, pp. 17–20, 2006.

[151] Qazvinian, V., Rosengren, E., Radev, D. R., and Mei, Q., "Rumor has it: Identifying misinformation in microblogs," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pp. 1589–1599, 2011.

[152] Quinlan, J. R., *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.

[153] Radlinski, F., "Addressing malicious noise in clickthrough data," in *Proceedings of the 3rd international workshop on adversarial information retrieval on the web (AIRWeb)*, 2007.

[154] Ramachandran, A., Feamster, N., and Vempala, S., "Filtering spam with behavioral blacklisting," in *Proceedings of the 14th ACM conference on Computer and communications security*, (New York, NY, USA), p. 342, ACM Press, 2007.

[155] Rapoza, K., "The dying business of email spam." `http://usa.kaspersky.com/about-us/press-center/in-the-news/dying-business-email-spam`, 2012.

[156] Rosen, D., Barnett, G. A., and Kim, J.-H., "Social networks and online environments: when science and practice co-evolve," *Social Network Analysis and Mining*, vol. 1, no. 1, pp. 27–42, 2011.

[157] Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E., "A bayesian approach to filtering junk e-mail," *Learning for Text Categorization: Papers from the 1998 Workshop*, vol. 62, pp. 98–105, 1998.

[158] Salton, G., Wong, A., and Yang, C. S., "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, pp. 613–620, Nov. 1975.

[159] Sculley, D. and Wachman, G., "Relaxed online SVMs for spam filtering," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 415–422, April 2007.

[160] Sebastiani, F., "Text categorization," in *Text Mining and its Applications to Intelligence, CRM and Knowledge Management*, pp. 109–129, WIT Press, 2005.

[161] Spirin, N. and Han, J., "Survey on web spam detection: principles and algorithms," *SIGKDD Explor. Newsl.*, vol. 13, pp. 50–64, May 2012.

[162] Stanford University, "The stanford webbase project." http://dbpubs.stanford.edu:8091/~testbed/doc2/WebBase/, 2013.

[163] Stein, T., Chen, E., and Mangla, K., "Facebook immune system," in *Proceedings of the forth ACM EuroSys Workshop on Social Network Systems(SNS2011)*, 2011.

[164] Stringhini, G., Kruegel, C., and Vigna, G., "Detecting spammers on social networks," in *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pp. 1–9, 2010.

[165] SURBL. http://www.surbl.org/, 2013. Accessed on August. 1, 2013.

[166] T, K. C., Ponnapalli, H., Herts, D., and Pablo, J., "Analysis and Detection of Modern Spam Techniques on Social Networking Sites," in *2012 Third International Conference on Services in Emerging Markets*, pp. 147–152, IEEE, Dec. 2012.

[167] Tang, Y., Krasser, S., He, Y., Yang, W., and Alperovitch, D., "Support vector machines and random forests modeling for spam senders behavior analysis," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pp. 1–5, 2008.

[168] The Harris Interactive Survey, "The harris interactive survey–trends and tudes 2009." http://www.harrisinteractive.com/vault/HI_TrendsTudes_2009_v08_i04.pdf, 2013.

[169] The Open Web Application Security Project(OWASP), "Cross-site scripting (xss)." https://www.owasp.org/index.php/Cross-site_Scripting_(XSS), 2013.

[170] The Spamhaus Project. http://www.spamhaus.org/, 2013. Accessed on August. 1, 2013.

[171] The SVMLigth. http://svmlight.joachims.org/, 2013. Accessed on Dec. 1, 2013.

[172] The TrustRank checker. http://www.seomastering.com/trust-rank-checker.php, 2013. Accessed on Dec. 1, 2013.

[173] Thomas, K., Grier, C., Ma, J., Paxson, V., and Song, D., "Design and evaluation of a real-time url spam filtering service," in *IEEE Symposium on Security and Privacy*, pp. 447–462, 2011.

[174] "Tweetdeck by twitter," 2011. http://tweetdeck.com/.

[175] Twitter Blog, "Twitter blog: Shutting down spammers." http://blog.twitter.com/2012/04/shutting-down-spammers.html, 2013.

[176] Twitter Developers. https://dev.twitter.com/, 2013. Accessed on August. 1, 2013.

[177] Urban Legends at About.com. http://urbanlegends.about.com/od/naturalwonders/ss/Fake-Hurricane-Sandy-Photos_11.htm, 2014. Accessed on Jan. 1, 2014.

[178] Urban Legends at About.com. http://urbanlegends.about.com/od/Fake-News/ss/Obama-Unveils-New-American-Flag.htm, 2014. Accessed on Jan. 1, 2014.

[179] URIBL. http://www.uribl.com/, 2013. Accessed on August. 1, 2013.

[180] V., H., L., M., and J., W., "Distributed knowledge networks. design, implementation, and applications," in *Proceedings of the IEEE Information Technology Conference*, 1998.

[181] Voorhees, E., Harman, D., of Standards, N. I., and (US), T., *TREC: Experiment and evaluation in information retrieval*. MIT press USA, 2005.

[182] wa Fu, K., hong Chan, C., and Chau, M., "Assessing censorship on microblogs in china: Discriminatory keyword analysis and the real-name registration policy," *IEEE Internet Computing*, vol. 17, no. 3, pp. 42–50, 2013.

[183] Wang, A. H., "Don't follow me: Spam detection in twitter," in *Proceedings of the International Conference on Security and Cryptography*, 2010.

[184] Wang, D., "Analysis and detection of low quality information in social networks," in *Proceedings of Ph.D. Symposium at 30th IEEE International Conference on Data Engineering (ICDE 2014)*, (Chicago, IL, United States), 2014.

[185] Wang, D., Irani, D., and Pu, C., "A social-spam detection framework," in *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS 11)*, (Perth, Australia), pp. 46–54, September 2011.

[186] Wang, D., Irani, D., and Pu, C., "Evolutionary study of web spam: Webb spam corpus 2011 versus webb spam corpus 2006," in *Proceedings of the 8th International Conference on Collaborative Computing: Networking, Applications and Work-sharing (CollaborateCom)*, (Pittsburgh, PA, USA), pp. 40–49, October 2012.

[187] Wang, D., Irani, D., and Pu, C., "A perspective of evolution after five years: A large-scale study of web spam evolution," *International Journal of Cooperative Information Systems*, vol. 23, no. 2, 2014.

[188] Wang, D., Irani, D., and Pu, C., "Spade: a social-spam analytics and detection framework," *Social Network Analysis and Mining*, vol. 4, no. 1, pp. 1–18, 2014.

[189] Wang, D., Navathe, S. B., Liu, L., Irani, D., Tamersoy, A., and Pu, C., "Click traffic analysis of short url spam on twitter," in *Proceedings of 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pp. 250–259, 2013.

[190] Wang, H., Wang, F., and Xu, K., "Modeling information diffusion in online social networks with partial differential equations," *CoRR*, vol. abs/1310.0505, 2013.

[191] Wang, P., Domeniconi, C., and Hu, J., "Cross-domain text classification using wikipedia," *IEEE Intelligent Informatics Bulletin*, vol. 9, no. 1, pp. 36–45, 2008.

[192] Wang, R., Youssef, A., and Elhakeem, A., "On Improving the Performance of Spam Filters Using Heuristic Feature Selection Techniques," in *Proceedings of 23rd Biennial Symposium on Communications, 2006*, pp. 227–230, Ieee, 2006.

[193] Webb, S., Caverlee, J., and Pu, C., "Introducing the webb spam corpus: Using email spam to identify web spam automatically," in *Proceedings of the Third Conference on Email and Anti-Spam (CEAS 2006)*, 2006.

[194] Webb, S., Caverlee, J., and Pu, C., "Predicting web spam with http session information," in *Proceedings of the Seventeenth Conference on Information and Knowledge Management (CIKM 2008)*, (Napa Valley, CA, USA), October 2008.

[195] Webb, S., Caverlee, J., and Pu, C., "Social honeypots: Making friends with a spammer near you," in *Proceedings of the Fifth Conference on Email and Anti-Spam (CEAS 2008)*, 2008.

[196] Webb, S., Caverlee, J., and Pu, C., "Characterizing web spam using content and http session analysis," in *Proceedings of the Fourth Conference on Email and Anti-Spam (CEAS 2007)*, (Mountain View, CA, USA), pp. 84–89, August 2007.

[197] Whittaker, S., Bellotti, V., and Gwizdka, J., "Email in personal information management," *ACM Communications*, vol. 49, pp. 68–73, Jan. 2006.

[198] Wolfe, A. W., "Anthropologist view of social network analysis and data mining," *Social Network Analysis and Mining*, vol. 1, no. 1, pp. 3–19, 2011.

[199] World Wide Web Consortium (W3C). `http://www.w3.org/`, 2013.

[200] Wu, B. and Davison, B. D., "Identifying link farm spam pages," in *Proceedings of the 14th International World Wide Web Conference*, pp. 820–829, ACM Press, 2005.

[201] WWW::Shorten::Bitly - Interface in Perl. `http://search.cpan.org/~pjain/WWW-Shorten-Bitly-1.17/lib/WWW/Shorten/Bitly.pm`, 2013. Accessed on August. 1, 2013.

[202] Xie, M., Yin, H., and Wang, H., "An effective defense against email spam laundering," in *Proceedings of the 13th ACM conference on Computer and communications security*, (New York, NY, USA), p. 179, ACM Press, 2006.

[203] Yang, F., Liu, Y., Yu, X., and Yang, M., "Automatic detection of rumor on sina weibo," in *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, MDS '12, pp. 13:1–13:7, 2012.

[204] Yang, Y. and Pedersen, J. O., "A comparative study on feature selection in text categorization," in *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, (San Francisco, CA, USA), pp. 412–420, 1997.

[205] Yardi, S., Romero, D. M., Schoenebeck, G., and Boyd, D., "Detecting spam in a twitter network," *First Monday*, vol. 15, no. 1, 2010.

[206] Zeng, A. and Zhang, C.-J., "Ranking spreaders by decomposing complex networks," *Physics Letters A*, vol. 377, no. 14, pp. 1031 – 1035, 2013.

[207] Zhang, C. M. and Paxson, V., "Detecting and analyzing automated activity on twitter," in *Proceedings of the Passive and Active Measurement Conference*, 2011.

[208] Zhen, Y. and Li, C., "Cross-domain knowledge transfer using semi-supervised classification," in *AI 2008: Advances in Artificial Intelligence*, vol. 5360 of *Lecture Notes in Computer Science*, pp. 362–371, Springer Berlin Heidelberg, 2008.

[209] Zhu, Y., Wang, X., Zhong, E., Liu, N., Li, H., and Yang, Q., "Discovering spammers in social networks," in *AAAI Conference on Artificial Intelligence*, 2012.

[210] Zhuang, L., Dunagan, J., Simon, D., and Wang, H., "Characterizing Botnets from Email Spam Records.," in *Proceedings of the first USENIX workshop on large-scale exploits and emergent threats (LEET 08)*, 2008.

[211] Zinman, A. and Donath, J., "Is britney spears spam?," in *Proceedings of the Fourth Conference on Email and AntiSpam(CEAS 2007)*, (New York, NY), pp. 112–117, ACM, 2007.

[212] Zou, M., Wang, T., Li, H., and Yang, D., "A general multi-relational classification approach using feature generation and selection," in *Advanced Data Mining and Applications* (Cao, L., Zhong, J., and Feng, Y., eds.), vol. 6441 of *Lecture Notes in Computer Science*, pp. 21–33, Springer Berlin Heidelberg, 2010.

VITA

De Wang was born and brought up in Anqing, a city of Anhui province in China. He received his bachelor's degree in Software Engineering with outstanding honor and his master's degree in Computer Software and Theory from Jinan University, Guangdong, China in 2007 and 2010 respectively. After that, He moved to Atlanta to pursue a Ph.D. in Computer Science at the College of Computing at Georgia Institute of Technology. De pursued his dissertation research in the area of cyber security with applications in systems and data analytics under the guidance of Prof. Calton Pu in the Distributed Data Intensive Systems Lab (DISL) and Center for Experimental Research in Computer Systems (CERCS) . He contributed to a multitude of projects including DOI (Denial of Information), GT spam common, and GRAIT-DM.