Analysing and Detecting Twitter Spam

By

Chao Chen

Submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

Deakin University

Sept 2016

DEAKIN UNIVERSITY
ACCESS TO THESIS - A

I am the author of the thesis entitled

Analysing and Detecting Twitter Spam

submitted for the degree of ***Doctor of Philosophy***

This thesis may be made available for consultation, loan and limited copying in accordance with the Copyright Act 1968.

'I certify that I am the student named below and that the information provided in the form is correct'

Full Name: .....Chao Chen.....
(Please Print)

Signed: 

Date: .....23/03/2017.....

DEAKIN UNIVERSITY
CANDIDATE DECLARATION

I certify the following about the thesis entitled (10 word maximum)

Analysing and Detecting Twitter Spam

submitted for the degree of ___**Doctor of Philosophy**_____

a. I am the creator of all or part of the whole work(s) (including content and layout) and that where reference is made to the work of others, due acknowledgment is given.

b. The work(s) are not in any way a violation or infringement of any copyright, trademark, patent, or other rights whatsoever of any person.

c. That if the work(s) have been commissioned, sponsored or supported by any organisation, I have fulfilled all of the obligations required by such contract or agreement.

I also certify that any material in the thesis which has been accepted for a degree or diploma by any university or institution is identified in the text.

'I certify that I am the student named below and that the information provided in the form is correct'

Full Name: .....**Chao Chen**.....
(Please Print)

Signed: .....

Date: .....**23/03/2017**.....

I dedicate this thesis to my family and friends. A special gratitude to my loved wife, Si Wu, for her support and understanding.

Table of Contents

List of Tables

List of Figures

Acknowledgements

First, I would like to express my deepest gratitude to my principle supervisor Prof. Yang Xiang, for his unreserved and continuous support. His guidance has helped me all the time during my Ph.D study. His advice on both research as well as on my life has been priceless. I would like to thank my associate supervisor Dr. Jun Zhang, who introduced me the fun of doing research, guided me to write my first research paper hand by hand and kept supporting me all the time. Without their help, this thesis would not be finished.

I would also like to thank the academic and general staff of the School of Information Technology, Deakin University, especially to Professor Wanlei Zhou, Dr. Gang Li, Ms. Lauren Fisher, Ms. Alison Carr, and Ms. Kathy Giulieri. I am also thankful to my colleagues, Dr. Yu Wang, Dr. Shen Weng, Dr. Tom Hao Luan, Dr. Xinyi Huang, Dr. Longxiang Gao, Dr. Tianqin Zhu, Dr. Mohammed Sayad Haghighi, Dr. Silvio Cesare, Mr. Xiao Chen, Mr. Yuexin Zhang, Ms. Jiaojiao Jiang, Mr. Donghai Liu, Mr. Di Wu, and many others for your brilliant comments and suggestions.

A special thanks to my family. Words cannot express how grateful I am to my parents for all the sacrifices that they have made for me. At the end, I would like express thanks to my beloved wife Si, for her everlasting love, support, and understanding.

List of Publications

1. Chao Chen, Yu Wang, Jun Zhang, Yang Xiang, Wanlei Zhou, and Minyong Ge "Statistical Features Based Real-time Detection of Drifted Twitter Spam," ***IEEE Transactions on Information Forensics and Security***, vol. 12, no. 4, pp. 914-925, April. 2017.

2. Chao Chen, Jun Zhang, Yi Xie, Yang Xiang, Wanlei Zhou, Mohammad Mehedi Hassan and Abdulhameed AlElaiwi, "A performance evaluation of machine learning based streaming spam tweets detection," ***IEEE Transactions on Computational Social Systems***, vol. 2, no. 3, pp. 65-76, Sept. 2015.

3. Chao Chen, Sheng Wen, Jun Zhang, Yang Xiang, Jonathan Oliver, Mohammad Mehedi Hassan, and Abdulhameed Alelaiwi "Investigating the Deceptive Information in Twitter Spam," ***Future Generation Computer System***, accepted, in press.

4. Chao Chen, Jun Zhang, Yang Xiang, Wanlei Zhou, and Jonathan Oliver, "Spammers are becoming 'smarter' on Twitter," ***IT Professional***, pp. 66-70, March/April 2016 **(Featured Article, Selected by ComputingEdge)**.

5. Chao Chen, Jun Zhang, Xiao Chen, Yang Xiang and Wanlei Zhou, "6 million spam tweets: A large ground truth for timely Twitter spam detection," *2015 IEEE International Conference on Communications (ICC)*, London, 2015, pp. 7065-7070.

6. Chao Chen, Jun Zhang, Yang Xiang and Wanlei Zhou, "Asymmetric self-learning for tackling Twitter Spam Drift," *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Hong Kong, 2015, pp. 208-213.

7. Jun Zhang, Chao Chen, Yang Xiang, Wanlei Zhou, and Yong Xiang, "Internet traffic classification by aggregating correlated naive bayes predictions," ***IEEE Transactions on Information Forensics and Security***, vol. 8, no. 1, pp. 5-15, Jan. 2013.

8. Jun Zhang, Chao Chen, Yang Xiang, Wanlei Zhou, and Athanasios V. Vasilakos, "An effective network traffic classification method with unknown flow detection," ***IEEE Transactions on Network and Service Management***, vol. 10, no. 2, pp. 133-147, 2013.

9. Sheng Wen, Mohammad Sayad Haghighi, Chao Chen, Yang Xiang, Wanlei Zhou, and Weijia Jia, "A Sword with Two Edges: Propagation Studies on Both Positive and Negative Information in Online Social Networks," ***IEEE Transactions on Computers***, vol.64, no.3, pp.640-653, March 2015.

10. Xiao Chen, Chao Chen, Jun Zhang, Yang Xiang and Wanlei Zhou, "A unified aggregation method for network traffic classification," ***Concurrency Computat.: Pract. Exper.***, accepted, in press.

11. Shigang Liu, Yu Wang, Jun Zhang, Chao Chen and Yang Xiang, "Addressing the class imbalance problem in twitter spam detection using ensemble learning", ***Computers and Security***, accepted, in press.

12. Yu Wang, Chao Chen, and Yang Xiang, "Unknown Pattern Extraction for Statistical Network Protocol Identification," *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, Clearwater Beach, FL, 2015, pp. 506-509.

13. Jun Zhang, Chao Chen, Yang Xiang, and Wanlei Zhou, "Robust network traffic

identification with unknown applications," *2013 the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security (ASIACCS)*, Hangzhou, 2013, pp. 405-414.

14. Bailin Xie, Yu Wang, Chao Chen, and Yang Xiang, "Gatekeeping Behavior Analysis for Information Credibility Assessment on Weibo", *in NSS 2016*, Springer International Publishing, 483-496

15. Shigang Liu, Yu Wang, Chao Chen, and Yang Xiang, "An Ensemble Learning Approach for Addressing the Class Imbalance Problem in Twitter Spam Detection," *in ACISP 2016*, Springer International Publishing, 215-228

16. Di Wu, Xiao Chen, Chao Chen, Jun Zhang, Yang Xiang and Wanlei Zhou, "On addressing the imbalance problem: A correlated KNN approach for network traffic classification," *in Network and System Security*, Springer International Publishing, 2014, 138-151.

Abstract

Online Social Networks (OSNs), especially Twitter and Facebook, are becoming an integral part of peoples daily life around the world. People share their activities in OSNs, at the same time, view updates from friends. However, the popularity of OSNs also attracts spammers. Different to Email spam, OSN spam is more challenging due to its short content and streaming characteristic.

In this thesis, we firstly carry out an in-depth analysis Twitter Spam via various aspects, such as deceptive information and three new "smart" spamming strategies. Twitter spam contains deceptive information, such as free voucher and weight loss advertisement to at- tract the interest of victims. A comprehensive analysis on the deceptive information will be of great benefit to the detection of Twitter spam. The analysis is based on a collection of over 550 million tweets with around 6% spam. We find that various deceptive content of spam performs differently in luring victims to malicious sites. We also find the regional response rate to various Twitter spam outbreaks vary greatly. In addition, we find that spammers are becoming more crafty. They now use more complex spamming strategies to avoid being detected. We find three new spamming strategies, *i.e.* "coordinated posting", "finite-state machine based spam template" and "passive spam".

The focus of recent works is on the application of machine learning techniques into Twitter spam detection. However, tweets are retrieved in a streaming way, and Twitter provides the Streaming API for developers and researchers to access public tweets in real-time. There lacks of a performance evaluation of existing machine learning based streaming spam detection methods. In the second work, we bridged

the gap by carrying out a performance evaluation, which was from three different aspects of data, feature and model. For timely spam detection, we further extracted 12 lightweight features for tweet representation. Spam detection was then transformed to a binary classification problem in the feature space and can be solved by conventional machine learning algorithms. We evaluated the impact of different factors to the spam detection performance, which included spam to non-spam ratio, feature discretization, training data size, data sampling, time-related data, and machine learning algorithms. The results show the streaming spam tweet detection is still a big challenge and a robust detection technique should take into account the three aspects of data, feature and model.

In the third, we observe that the statistical properties of spam tweets vary over time, and thus the performance of existing machine learning based classifiers will decrease. This issue is referred to as Twitter Spam Drift. In order to tackle this problem, we propose a novel Lfun scheme. Lfun can discover changed spam tweets from unlabelled tweets and incorporate them into classifiers training process. A number of experiments are performed to evaluate the proposed scheme. The results show that our proposed Lfun scheme can significantly improve the spam detection accuracy to detect spam tweets. The results show that our proposed Lfun scheme can significantly improve the spam detection accuracy in real-world scenarios.

Chapter 1

Introduction

Online Social Networks (OSNs), like Twitter and Facebook, have become increasingly popular in the last few years. Internet users spend more hours on social network sites than any other website [64]. In addition, social network sites have become the main news source for around 30% of Americans according to the survey carried out by Pew Research Center [84]. Moreover, the social networking apps in smartphones make users' access to such sites become ubiquitous (nearly one of five smartphone owners visit social networking sites via their mobile phones [35]). Billions of users spend vast time in OSNs making friends with people who they are familiar with or interested in. After the relation is built, users can receive messages, usually something interesting or recent activities shared by the friends they are connected to, in the terms of tweets, wall posts or status updates.

These connected users and information they shared form a huge social graph with tremendous information spreading in it. These large social networks have attracted

many researchers' interest. Jon Kleinberg *et. al* focus on the analysis of social search and social graph [59, 69, 97, 117, 118], while some researchers are measuring the online social networks [25, 64, 82, 87]. At the same time, Neil *et al.* are researching on predicting links [6, 47, 70] and inference attributes in OSNs [81, 83, 105, 138]. Privacy preserving in social network is also a stream [53, 63, 73, 109, 120]. Moreover, sybil attack in OSNs is also studied by researchers [124, 125, 139]. In the consideration of studying group behaviour of users in OSNs, the community discovery techniques are in research [56, 71, 100, 130]. In addition, some works are on the credibility [20, 23] and dissemination [7, 26, 132] of information. More interestingly, some research are analysing users' sentiment [14, 15, 60], predicting flu trend [2], or even detecting earthquake in OSNs [101], *etc.*

Despite the interest of researchers, the rich information in OSNs has also attracted the attention of cyber-criminals. For example, worms and malware started to spread via OSNs [94, 115]. Such worms in OSNs are using old ideas to apply in the new platforms. Online social network worms make use of victim's friends' list to send themselves to other OSN users, which is similar to email worms like LoveLetter [24]. In addition, [12] demonstrated that it was pretty easy for cyber-criminals to launch an automatic attack to steal the detailed information of a victim's friends by forging a same account of the victim. The stolen information is very invaluable for attackers to spread malware or scams, and personalize victims' online behaviour. Coincidentally,

social bots, which is analogous to network bots [1], also begin to compromise the OSNs [16,32,85,86]. Except these threats, some users are making use of the properties of OSNs to abuse the platforms. A number of politically-motivated attacks have occurred over the past few years. These attacks aim to either mislead public opinion, disseminate false information, or disrupt the conversations of legitimate users [74, 96, 113, 126]. Besides these attacks, URL shortening services are also abused in OSNs [5, 31]. Among all the abuse, online social network spam, which are driving victims to click malicious URLs containing phishing, malware, scams *et al.*, are abusing and polluting these platforms the most.

Why are OSNs so susceptible to spam? To answer this question, characteristics of OSNs must studied. Firstly, information access and sharing are based on the trust between users. Users normally share personal information in OSNs with public access or not. If it is not public, only friends can view his information. However, the authentication of OSNs is not that strong, which makes it very easy for attacker to impersonate a user, forge his identity and enter this victim's network of trust [12, 51]. In addition, users often accept any friend invitation from strangers so as to gain popularity, which may exposure their personal information to malicious users with ulterior motives. A national survey in the U.S. carried out by Harris Interactive showed that at least 83% of social network users received unwanted (unsolicited) friend invitation or spam message in the year of 2008 [54]. Users' security awareness

to threat is also another important characteristic of OSNs. While most users are aware of common Internet threats, they, unfortunately, are not that understanding the hidden threats of social spam. A research study [12] illustrates that approximate 45% of users in OSNs would click the links shared by their social friends, even if they don't know the "friends" in real world. In addition, Chris *et al.* believed that social spam are much more harmful than email spam, with a clickthrough rate of 0.13%, compared to a much lower rate (0.0003% - 0.0006%) for email spam [48]. Social spammers make use of these features to spread spam messages, whose embedded links will lure victims to the sites that are promoting adult content, malware downloading, advertisements and scams.

In this thesis, we take Twitter as the representative of Online Social Networks, and characterise Twitter spam which is defined as unsolicited tweets containing malicious links that directs victims to external sites containing malware downloads, phishing, drug sales, or scams, *etc* [27] , evaluate spam detection algorithms, and address "Spam Drift" issue in following chapters.

1.1 Analysing and Understanding Twitter spam

To begin with, we firstly investigate deceptive information in Twitter spam. What is deceptive information? Spammers always leverage some certain topics, such as "free gift card", "gain followers", *etc.,* to entice victims to click the embedded malicious

link. This kind of information is regarded as "Deceptive Information" in Twitter spam. A better understanding of deceptive information is crucial to spam detection techniques. Therefore, we are motivated to thoroughly study the deceptive information employed by spammers. To study this problem, we collect over 550 million tweets, with 33 million of which are spam tweets. We then cluster the spam tweets into 17 groups according to their content. These 17 groups accounted for 75% of the spam we identified.

By examining the clickthrough data, we find that various deceptive information of spam performs differently in luring victims to malicious sites. In addition, the regional distribution of victims varies due to different types of deceptive information. However, most victims are come from the United States. Our findings reveal that different deceptive information has different click through rates in various countries. This suggests the spam detection system should pay more attention to the tweets that contain the deceptive information mentioned before, as detection efficiency will be greatly improved.

We continue to study the spammers behaviours and expect to see what spammers will do after researchers and companies have proposed a variety of spam detection techniques. Interestingly, we have found that spammers are also becoming more crafty to avoid being detected. They now use more complex spamming strategies. We have identified three new spamming strategies through carrying our an in-depth analysis,

i.e. "coordinated posting", "finite-state machine based spam template" and "passive spam".

1.2 Evaluating ML-based Streaming Spam Detection Algorithms

In order to stop spammers, researchers have proposed a number of mechanisms. The focus of recent works is on the application of machine learning techniques into Twitter spam detection. However, tweets are retrieved in a streaming way, and Twitter provides the Streaming API for developers and researchers to access public tweets in real-time. There lacks of a performance evaluation of existing machine learning based streaming spam detection methods.

In this work, we bridged the gap by carrying out a performance evaluation, which was from three different aspects of data, feature and model. For real-time spam detection, we further extracted 12 lightweight features for tweet representation. Spam detection was then transformed to a binary classification problem in the feature space and can be solved by conventional machine learning (ML) algorithms. We evaluated the impact of different factors to the spam detection performance, which included spam to non-spam ratio, feature discretization, training data size, data sampling, time-related data, and machine learning algorithms. The results show the streaming spam tweet detection is still a big challenge and a robust detection technique should

take into account the three aspects of data, feature and model.

1.3 Addressing "Spam Drift": Lfun approach

Most of Twitter spam detection works rely on machine learning based techniques. In our labelled tweets dataset, however, we observe that the statistical properties of spam tweets vary over time, and thus the performance of existing machine learning based classifiers decreases. This issue is referred to as "Twitter Spam Drift". In order to tackle this problem, we firstly carry out a deep analysis on the statistical features of one million spam tweets and one million non-spam tweets, and then propose a novel *Lfun* (**L**earning **f**rom **un**labelled tweets) scheme. It incorporates two components: Learning from Detected Tweets, and Learning from Human Labelling. The proposed scheme can discover "changed" spam tweets from unlabelled tweets and incorporate them into classifier's training process. A number of experiments are performed to evaluate the proposed scheme. The results show that our proposed Lfun scheme can significantly improve the spam detection accuracy in real-world scenarios.

1.4 Contributions

Spam is plaguing Twitter now. In addition, it entices much more victims than email spam [44]. Spam not only interferes user experience, but also causes damage to users, such as malware downloading, phishing, worm propagation, *etc.* Understanding and

detecting Twitter spam is of great urgency and importance.

In order to achieve it, this thesis firstly provide a through data analysis of spam on Twitter. We demonstrate that various deceptive content of spam performs differently in luring victims to malicious sites and the regional response rate to various Twitter spam outbreaks varies greatly. In addition, spammers are becoming "smarter" by employing more complex spamming strategies to avoid being detected. We then carry out a performance evaluation of streaming spam detection frameworks, which was from three different aspects of data, feature and model. From that, we therefore identified an unseen issue in Twitter spam detection, *i.e.* "Spam Drift". To address this problem, we propose a Lfun scheme, which can learn from unlabelled tweets. Experiments on real-world datasets show that our scheme can greatly improve the detection accuracy. Our contributions of this thesis is summarised as:

- We have collected and labelled a large Twitter data set of around 600 million tweets. After analysing it, we find various deceptive information of spam performs differently in luring victims to malicious sites. In addition, the regional distribution of victims varies due to different types of deceptive information. This suggests the spam detection system could leverage the deceptive information contained in spam tweets to improve detection efficiency. We have also identified three new spamming strategies applied by spammers, which indicates that spammers are also fighting back to avoid being detected. Researchers and

industry should pay more attention to spammers' behaviour, and propose advanced detection systems.

- Research community lacks of a performance evaluation of existing machine learning based streaming spam detection methods. We, thus, evaluate the impact of different factors to performance the streaming spam detection, which include spam to non-spam ratio, feature discretization, training data size, data sampling, time-related data, and machine learning algorithms. The importance of this work is to show that the streaming spam tweet detection is still a big challenge and a robust detection technique should take into account the three aspects of data, feature and model.

- We firstly identify the "Spam Drift" issue in detecting Twitter spam. We then propose a Lfun scheme which can discover "changed" spam tweets from unlabelled tweets and incorporate them into classifier's training process. Our Lfun scheme can effectively detect spam tweets even when they are drifting. Current Twitter spam detection systems can take advantage of our work to further improve their accuracy.

The importance of our work is two-fold. On one hand, the in-depth analysis shed light on that researchers can leverage context information and spammers' behaviour to propose advanced high accurate detection system. On the other hand, researchers also need consider "Spam Drift" issue when they are designing detection systems.

Our proposed Lfun scheme can provide industries an opportunity to re-design their system to wipe as much spam as possible.

This thesis is organized as follows. Chapter 2 reviews recent state-of-art works on spam analysis and characterisation, and spam detection mechanisms. Chapter 3 shows our findings on deceptive information contained in Twitter spam and emerging spamming strategies developed by spammers to avoid being detected. A thorough performance evaluation on machine learning based spam detection, from three different aspects of data, feature and model is carried in Chapter 4. Chapter 5 identifies a new "Spam Drift" issue and provides a solution which learns from unlabelled tweets to solve this issue. Finally Chapter 6 concludes this thesis, and future research directions are pointed out.

Chapter 2

Literature Review

In this chapter, related works will be introduced as two parts, OSN spam characterisation and state-of-art spam detection techniques.

2.1 Spam Characterization in OSNs

Spam messages are flooding in the OSNs, analysing and understanding of spam should be studied before detection. The following works in this section are focusing on the characterization of spam in OSNs.

2.1.1 Analysing Spam with Blacklisted URLs

[48] analysed 25 million URLs from 200 million public tweets, which was collect within one month. After flagging using three blacklists (Google Safebrowsing, URIBL, and Joewein), 3 million tweets are identified as spam. 2 million URLs are regarded as spam (5% of them were malware and phishing and the rest 95% directing victims to

scams), which accounts for 8% of all crawled unique URLs. However, 26% of URLs are directing to spam after they did a manual inspection of a sampled dataset, which indicates that the performance of blacklisting was moderate. They also found that the URLs in the spam messages would be blacklisted after they existed for 4 to 20 days. Nevertheless, 90% victims visited the spam URLs within the first two days of posting. Consequently, blacklists lag-time is too long to prevent the victims from visiting spam URLs. In addition, researchers believed that only 16% spam accounts are fraudulent accounts (which are create explicitly for spamming purpose), and 84% are compromised accounts. By studying the clickthrough of spam campaigns, twitter spam are much more harmful than email spam, with a clickthrogh rate of 0.13%, compared to a much lower rate (0.0003% - 0.0006%) for email spam.

2.1.2 Analysing Suspended Accounts on Twitter

As previous study [48] showed that URL blacklists would miss a large proportion of spam (8% detected by blacklisting, but 26% spam were found by manual inspection), [114] relied on Twitters own detection algorithm to build the ground truth. They collected a dataset of 1.8 billion tweets (sent by 32.9 million accounts in the period of 7 months), with 80 million (from the 1.1 million accounts suspended by Twitter) are spam. Nearly 3.3% accounts in the dataset are suspended by twitter. After validating sampled 100 suspended accounts, researchers found that majority accounts

are fraudulent ones instead of compromised ones, which is opposite to [1, 2]. With deep exploration, Twitters own detection algorithm can only catch 37% of spam accounts. More interestingly, 77% spam accounts are suspended within one day of their first tweet and 92% spam accounts only last within 3 days. Under such pressure, 89% spam accounts are rarely setting up social connections with users (they require less than 10 followers). Instead, 52% accounts make use of unsolicited mention and 17% accounts are hijacking trending topics. In addition, they also studied five large spam campaigns. However, three of them direct victims to reputable online shopping such as Amazon, which blurs the line what constitutes spam on social networks.

In [129], Wei *et al.* explore the spatial patterns of suspended Twitter users (mainly spammers) to see whether the network structure is impacted or not after removing such users. Authors collected about 74 million tweets sent by 38 million users from April 2010 to November 2013. They evaluate the impact of network structure from these aspects, node and edge statistics, network metrics and metrics ranking of network members. In addition, the impact of of content and sentiment is analysed. They conclude that the removal of suspended accounts has significant impact on influential users and overall topology of mentions, but less impact on what is being talked.

2.1.3 Characterising Spam Campaigns in OSNs

Different to work [48, 114], [44] focuses on the spam in Facebook. By crawling 8 regional networks in Facebook, they retrieved about 187 million message (wall post) from 3.5 million users. Firstly, researchers wanted to detect the users who were spreading the spam. Each wall post is modelled as a ¡description, URL¿ pair, two posts are regarded as similar if they 1) share the similar description or 2) share the same URL. All posts containing URLs are clustered according to the similarity. Malicious clusters are identified by distributed property (the number of users send wall posts in the cluster) and busty property (the absolute time interval between consecutive wall posts). If multiple accounts are sending similar messages or many messages are sent in a short period of time, these clusters are more likely to be spam campaigns. A threshold of (5, 1.5 hr) is used to detect suspicious clusters after initial experiments. Secondly, they began to characterize the identified spam campaigns. Three major spam campaigns are: 1) spammers promise free gifts; 2) spammers trigger victims curiosity by saying someone has a crush on you; 3) spammers describe some products. They also found that 70.3% spam posts directing victims to phishing sites and 35.1% spam posts lead to malware downloading, which is much different to [1], with only 5% spam direct to phishing and malware. Surprisingly, 97% of accounts which are distributing spam are compromised rather than being created by spammers. They also found that around 80% malicious accounts are lasing for less

than one hour and only 10% can be active for more than one day, after analysing the accounts behaviours. In addition, malicious accounts are most active at 3 am. These two facts can suggest that most malicious accounts are compromised accounts which are originally legitimate ones.

2.1.4 Analysing Spammers' Ecosystem

[136] focuses on the analysis of cyber-criminal ecosystem which is composed of criminal account community and criminal supporters community in Twitter. By analysing the sampled criminal account community of 2060 accounts, they find that the inner social relationship of this community is like this: 1) criminal accounts are forming a small world and 2) criminal hubs are more inclined to follow criminal accounts. In order to analyse the outer relationship, authors implemented an Mr.SPA algorithm (malicious relevance score to measure how closely an account to criminal accounts) to extract 5924 supporters, mainly formed by 3 categories: social butterflies (who have large number of followings and followers), social promoters (who have large following-follower ratios to promote their products), and dummies (most likely controlled by criminal accounts). By realizing that checking on each account deeply to determine whether it is a criminal account is impractical, researchers proposed a Criminal Accounts Inference Algorithm to infer more criminal accounts by exploiting criminal accounts social relationships and semantic coordination. This algorithm is based on

these observations: 1) criminal accounts tend to be socially connected and 2) criminal accounts usually share similar accounts. CIA algorithm can infer a large portion of criminal accounts from a small built social graph of known criminals.

Table 2.1 briefly describes the methods, datasets and analysed platforms of the above approaches.

Works	Method	Datasets	Platform
Grier *et al.* [48]	URL Blacklisting	3 million spam tweets collected from Jan, 2010 to Feb, 2010	Twitter
Gao *et al.* [44]	Grouping by similar message or URL, then classify using bursting nature and diverse of accounts	187 million wall posts, including 212,863 spam sent by 57,000 accounts from Jan, 2008 to June, 2009	Facebook
Thomas *et al.* [114]	Nil	1.1 million suspended accounts by Twitter from 17 April 2010 to 4 March 2010	Twitter
Yang *et al.* [136]	Malicious Score calculated by CIA algorithm	2060 detected spammers in [137]	Twitter
Wei *et al.* [129]	Gussian mixture modelling	78 Million tweets from 38 million users in the range of April 2010 to November 2013	Twitter

Table 2.1: Platforms and Datasets used in Characterization Works

2.2 State-of-art Spam Detection Techniques in OSNs

This section provides the detailed review of existing state-of-art social spam detection techniques, based on three categories, URL features based, statistical features based, and text based.

2.2.1 URL Features Based Techniques

As spammers always embed URLs in the spam messages to lure people to the sites which may contain hidden threats, a number of works focused on the detection of URLs contained in the messages to determine whether this message is spam or not. Similar to previous works [55, 75–77] to detect malicious URLs, the reviewed works also applied machine learning on detecting the suspicious spam messages.

2.2.1.1 Real-time URL Spam Filtering in Twitter

Some works focused on the characteristics of spammers accounts to detect social spam. However, this kind of approach has two drawbacks: 1) delay occurs between the account creation and detection due to requirement of collecting history behaviour; 2) compromised accounts are not accurately detected due to its mixed behaviours of benign and malicious accounts. In [112], Thomas *et al.* designed a system named Monarch, which can detect whether a URL directs to spam content. This system contains three components: a crawler to collect URLs from social media, a feature extractor to visit crawled URLs and extract relevant features, and a classification engine to classify the URLs. Authors used Web Browser, DNS resolver, and IP analysis to collect a number of features from URLs captured by email providers spam traps, blacklisted URLs appearing in Twitter and non-spam URLs in Twitter. Features include domain tokens, path tokens, redirects, JavaScript events and so on. In order

to fast train the large-scale datasets, authors applied a stochastic gradient descent for logistic regression. The experimental results show that their system can identify spam with 90.78% accuracy and 0.87% false positives; the classification time is around 6 seconds per instance. They also noted that twitter spamming activities are different from email spamming ones, which indicates that there is no common set of features that can discriminate both spamming activities.

This work [67] provides a real-time Twitter suspicious URL detection system based on the correlations of multiple URL redirect chains that share the same redirection severs. Due to the easy fabrication of account based features and lexical URL features, authors investigated the features extracted from correlated URL redirection chains. If several URL redirection chains share one same intermediate URL (which is call entry point), they are regarded as Correlated URL Redirection Chains. A total number of 14 features are extracted from correlated URL redirection Chains and Tweet context information. The classifier was implemented by using an L2-Regularized L1-loss support vector classification (SVC) algorithm, and the accuracy was 91.87% with 1.13% FP. FP could be further reduced to 0.95% if the weight value of benign samples was set to 1.1. Authors also evaluated the discrimination of features by using F-score, and found that the account creation time, the relative number of source applications, and the relative number of initial URLs are import features. In contrast, the similarity of number of friends and followers and the relative number of Twitter accounts are

less important. However, this system cannot deal with dynamic and multiple URL redirections as the crawler can only process HTTP headers. In addition, the system is not able to handle all the public tweets if it has the 100% sampling access right. [68] is an extended version of this paper.

2.2.1.2 Behaviour Analysis based Spam URL Detection on OSNs

Some works are based on the behaviour analysis of URLs, relying on the fact that behavioural features are more difficult to manipulate than traditional features.

[21] examines the behaviour of spam URLs from two aspects: the posting behaviour and the clicking behaviour. Thus, they extract 15 related features, such as posting count, posting intensity, click dynamics, total number of clicks and so on. They then apply Random Forest on these 15 features of two datasets, and achieve 86% overall accuracy. [123] proposes BEAN to detect spam URL. After analysing spammers' message sending behaviour and their characteristics on Twitter, they define six URL behaviour states. Further, a Markov Chain Model is proposed to detect spam URL. This approach has demonstrated the ability to detect spam which cannot be identified by conventional techniques, such as SVM and TrustRank.

Table 2.2 briefly describes the methods, features, datasets and platforms of the above approaches.

Works	Algorithm	Features	Datasets	Platform
Thomas *et al.* [112]	Logistic Regression with L1-regularization	Source URL, HTML headers, ... URL based features	567,784 spam URLs posted to Twitter, 1.25 million spam URL in emails from Sept, 2010 to Oct, 2010	Twitter and Email
Lee *et al.* [67, 68]	L2-regularized L1-loss support vector classification	content based features, URL redirect chain based features	263,289 accounts suspended by Twitter from April, 2011 to Aug, 2011	Twitter
Cao *et al.* [21]	Random Forest	15 URL based features from posting and clicking behaviour	7 million Bitly-shortened URLs posted on Twitter	Twitter
Wang *et al.* [123]	Markov Chain Model	6 URL behaviour states	2.4 million tweets sent by 900k users in four months	Twitter

Table 2.2: Features and Datasets used in URL based Works

2.2.2 Machine Learning Based Techniques

Inspired by the power of machine learning algorithms, a few researchers apply ML algorithms to detect social spam. This section reviews significant works published on ML-based social spam detection. The key points of each work are discussed in the following subsections and summarised in Table 2.2.2.9

2.2.2.1 Detecting Spam by Using Meta-info Features

Benevenuto *et al.* in 2008 firstly applied Machine Learning to identify video spammers on Youtube by using both user-related and video-related features [9, 10]. Inspired by this work, they later proposed a machine learning based method to detect Twitter spammers in 2010 [8]. Although, Kuak *et al.* have reported twitter spam in their data, they filtered the spam simply using some fixed thresholds [64] instead of using machine learning algorithm. [8] was using a much larger set of both account-based features and tweet context based features to differentiate spammers from normal users. Spammers were defined as the users who posted at least one URL unrelated to the tweet text in this work. A huge dataset of more than 5.4 million users, 1.9 billion links and 1.8 billion tweets were collected at first, then a labelled dataset relating to three trending topics in 2009 were extracted. The dataset was manually labelled by volunteers, which consisted of 335 spammers and 7852 non-spammers. A total number of 62 features from both tweet content and account were used for the classification using SVM with RBF kernel. It can correctly detect approximate 70% (dual behaviour of spammers caused the misclassification of 30% spammers to non-spammers) of spammers and 96% of non-spammers. Features importance was evaluated by using information gain and Chi-Square test. The most discriminative features were URLs in the tweet, and the accounts age. Classifications on content-based features were also carried out, however the FP increased to 7.5% from 3.6%.

Wang *et al.* also proposed a Bayesian classifier based approach to detect spammers on Twitter [121]. Both graph based features (an importance feature, reputation, was included) and content-based features were used in this work. In order to train the classifier, a number of 500 manually labelled users were used, which contained 3% of spammers, e.g. 15 spammers after calculation. 392 out of 25817 users were detected as spammers. 348 suspicious spammers were confirmed to be real spammers after checking manually, with 89% precision. A limitation of this work is obvious: proposed features are easy to be evaded. More robust features were later proposed by Yang *et al.* in [135, 137]

2.2.2.2 Spammer Detection by Using honeypots:

Traditional spam detection classifiers need a lot of human intelligence to manually label a training set. In addition, the learned spam signature disappears quickly as spammers are adopting various evasion techniques. Alternatively, some spam discovery approach relies on the community reporting mechanism (like User S was reported by some other users as spammers). This method, however, can be manipulated themselves to mistakenly label legitimate users to spammers. In security community, researchers often deploy honey pots to observe and analyse malicious activities [61, 93, 106]. Inspired by this, authors in work [66, 107] used honeypots (also called honey profile) to collect spammers in OSNs.

After collecting a number of spammers profiles, Lee *et al.* used an updated classifier to detect spammers [66]. The deployed honeypots collected 1570 spammers profiles and 500 spammers profiles for MySpace and Twitter respectively. Four kinds of features were used for classification: user demographics, user-contributed content, user activity features and user connections. The evaluation of the spam detection was using 10 supervised classifiers from WEKA. The accuracy was above 98% for MySpace dataset, and ranged from 82% to 88% for Twitter dataset. It proved that the deployed social honeypots could attract spammers whose behaviours were strongly discriminative from normal users. In addition, authors also evaluated the effectiveness of large-scale spam detection by using two large datasets: 1.5m profiles in MySpace, and 215,345 user profiles with 4,040,415 tweets in Twitter. (Evaluations were done on sampled small datasets as well)The precision was worse for MySpace, decreasing from 98% to around 70%, but Twitters stayed the same.

Similar to [66], Stringhini *et al.* in [107] deployed 900 honeypots in Facebook, MySpace and Twitter like [66]. These honeypots would accept any friend request from strangers. After monitoring for around one year, they manually identified 173 spammers out of 3831 requests, 8 spammers out of 22 requests, and 361 spammers out of requests for Facebook, MySpace and Twitter respectively. According to different spamming strategy, authors also categorize four classes of spammers: Displayer who only displays spam content in his own profile page, Bragger who post malicious

messages to their own profile, Poster who send messages to victims by posting on their walls, and Whisperer who sends private message to specific victim. Six features (the discriminative power was not examined) were extracted from these spammers. A Random Forest classifier was used to classify spammers. A 10-fold evaluation on the 1000 profiles training set yielded 2% FP rate, and 1% FN rate for Facebook dataset. One the other side, the same evaluation yielded FP rate of 2.5% and FN rate of 3%. However, TP rate was not reported in this work. The analysis on the spam campaigns demonstrated that there were two kinds of campaigns, one was stealthy campaign that was posting mixed malicious and benign messages, the other was greedy campaign that was posting malicious messages only and the rate was fast.

2.2.2.3 Spam Detection in Multiple OSNs

Previous spam detection works mainly focus on one particular social network. Thus, [57] proposed a scalable and online system to deal with this critical security issue in multiple online social networks. Three kind of users, which were spammer, legitimate user, and infected user, were defined by authors. The goal of this system was to detect the messages sent from both spammers and infected users. Different to other works, this system considered detecting spam photos as well, by using image content features such as colour histogram, colour correlation, CEDD, et al. Several features from text content and social network characteristics were also used. After that, existing classifier would be trained with pre-labelled samples and then to classify

unlabelled instances. Classified instances would also be added into the training set to re-train the classifier. It is said this framework is scalable in multiple social networks. This demo, however, was only done in the case of Facebook; performance was not reported. And the adaptability to other social network platforms was not shown.

In addition, Faraz *et al.* proposed a generic statistical approach for spam detection in both Facebook and Twitter [3]. In this work, authors manually labelled a relatively small set of normal and spam profiles from Facebook and Twitter. 165 spam profiles and 155 normal profiles were labelled for Facebook dataset, and the Twitter dataset contained 160 spam profiles and 145 normal profiles. 14 features from four categories (Interaction, Messages, URLs, mentions) were extracted to represent one sample in classification. Nave Bayes, Jrip and J48 were used as supervised classifiers; the results were impressive with more than 95% detection rate. They also evaluate the importance of selected features, measured using Information Gain. Several discriminative features, which related to friends/followers, pages/hashtags, and URLs, were found by the authors. Authors also clustered spammers into campaigns using Markov clustering algorithm. However, the campaign analysis was not done. And the generic approach might only be able to detect Facebook and Twitter spammers, which cannot detect other social network spammers. Furthermore, the experimental dataset was too small which cannot represent the super large online social networks.

Xu *et al.* [134] also noticed that when one spam link appeared in one social network, it was most likely to appear in a another social network, due to the connections within these social networks. They believed that, if the spam detection models have the ability to communicate with each other, it would be effective for spam filtering. As a result, they focused on analysing and extracting spam in one social network, and using the features to detect spam in other social networks. In this work, they collected both Twitter and Facebook datasets. Firstly, they trained a classifier to detect Twitter spam using Twitter dataset. Then, they mixed the Facebook data into Twitter dataset to retrain the model and used it to detect Twitter spam as well. After comparison among various machine learning algorithms, they concluded that similar spam in one social network can benefit the detection of spam in another social network.

2.2.2.4 Designing Robust Features to Fight Social Spammer

A number of previous works [8,121] has applied machine learning algorithms to detect Twitter spammers by using discriminative features. However, some feature were easily to be evaded by buying more followers, posting more tweets, mixing normal and spam tweets, *etc.* Consequently, authors in [135,137] intended to import some robust features, including 3 graph-based features, 3 neighbour-based features, 3 automation-based features and 1 timing-based features. Authors also built a quantitative model

to analyse the robustness of the features, results showed that, age, betweenness centrality, clustering coefficient were the most robust features. After comparing with three similar work [66, 107, 121], this work had the lowest False Positive Rate, along with the highest Detection Rate and F-measure. In addition, after removing the 10 new features, the detection rate would decrease about 10%. This also proved the effectiveness of the new proposed features. The evaluation dataset was small, which only consisted 500 spammer accounts, and 5000 non-spammer accounts. It may have sampling bias. Furthermore, some graph-based features were also very expensive to collect.

[122] also designs the Twitter spam detection system, which only relies on the tweet-inherent features. The benefit of tweet-only based features can facilitate the timely detection of spam tweets, and the extraction of such features is near real-time. In this work, Wang *et al.* proposed 17 content features, such as No. of words, Number of characters, N-gram features, and sentiment features. Based on these features, they conducted spam detection tasks under five different machine learning algorithms. They have achieved encouraging results when compared with existing detection frameworks with costly features.

2.2.2.5 Social Relation Based Techniques

User account features based approaches have two significant limitations in spam detection in online social networks: 1) features can be easily fabricated; 2) it has a delay

between spam creation and detection for the reason that features cannot be collected until numerous malicious activities are done. In order to solve these limitations, [104] proposed a social relation feature based detection framework, which can avoid the evasion of features by spammers and detect spam at the same time when a receiver receives a message. They proposed two new relation features in this work: distance and connectivity. While distance was the length of shortest path, the connectivity was measured by min-cut and random walk. From the evaluation of distribution of spammers and non-spammers, they found that most spam came from users at a distance of more than three. In addition, the connectivity between spammers and non-spammers was different from that between non-spammers. In the consideration of measuring these two relation features, a directed subgraph G which was part of the whole social graph G was generated under four conditions. To evaluate this method, they crawled a dataset containing 148,371 profiles, 267,551 tweets. Finally, they got 308 spammers and 10,000 spam messages by checking with the official @spam account which was used by users to report spam. Five supervised classifiers implemented in Weka were tested on 10-fold cross validation. The True Positive rate of the best classifier, Bagging, varied from 93.3% to 95.1% when using only distance and random walk and using distance, random walk, and min-cut. Authors also imported 11 user based features in previous works, and found that the True Positive rate was improved up to 99.7%, while the False Positive rate was reduced to 0.5% from about 5%. The

improvement was substantial when combing both the relation based features and user based features. Limitations of this work include: 1) message from new user would be classified as spam; 2) spam message from compromised users could be labelled as benign message.

Spammers prefer to use compromised account to spread spam, as these accounts already have well-established social connections. As a result, Egele *et al.* [40] and Ruan *et al.* [98] proposed approaches to detect compromised accounts. Egele *et al.* used statistical models to characterise seven features, Time, Message Source, Message Text, Message Topic, Links in Message, Direct User Interaction and Proximity. Then, authors leverage anomaly detection techniques to identify sudden changes in users' behaviour. This approach can effectively detect compromisation in accounts in Social Networks. Ruan *et al* [98] noticed that, although spammers can hack legitimate users' accounts, they cannot easily mimic the users' social behaviour patterns as spammers know little about those hacked users. In addition, spammers tend to use accounts to massively distribute spam messages, while legitimate users tend to entertain with friends. This lead more difficulty for spammers to mimic legitimate users' behaviour. Thus, it is possible to detect account compromisation by checking the compliance of the account's new behaviour with the authentic patterns. After analysing users' clickstream behaviour, [98] proposed some new social behaviour features to quantify the users' difference. By converting these features into eight vectors, they firstly

calculated the Euclidean distance vector, and then a user's behaviour was modelled as the mean difference between each pair of profiles. When the behaviour profile diverges, compromised activity can be detected. Their evaluation on Facebook users achieved high accuracy detection of compromised accounts

2.2.2.6 Spam Campaign Detection on OSNs

Account based spam detection is not sufficient to fight the spamming activities in online social networks, since adversaries were using compromised accounts to spread spam [44, 48]. The long latency and low efficiency in [48] and [44] prevent their usability for online detection. Consequently, authors in [43] proposed a campaign based technology which was capable to detect spam online. Instead of directly inspecting each single message, authors grouped incoming messages into campaigns based on similar text or same embedded URL. Similar texts were determined if the resemblance score (defined as the ratio of shared shingles to all unique shingles) of them is 0.5. For an incoming message, it firstly created a cluster which only contained itself. Then all clusters that had the similar message should be found and merged with the new cluster (called incremental clustering by authors).After that, features were extracted from the formed clusters. Despite general features like, Cluster Size, Average Time Interval, Average URL Number per Message, and Unique URL Number, authors also proposed two OSN specific features: Sender Social Degree and Interaction History. At last a trained supervised classifier using C4.5 Decision Tree was used for

classification. After tuning the ratio of spam to legitimate messages, the TP rate could reach 80.9% with 0.19% FP rate (class imbalance problem). At last, testing on the sever confirmed the low latency (21.5ms) and high throughput (1580 message per second), which can be deployed online (though no demo is shown).

Zhang *et al.* [142, 143] also considered detecting promoting campaigns along with the detection of social spam campaigns. Motivated by that current account or message based methods cannot detect all spam at one shot, authors proposed a campaign based detection scheme like [44]. While [44] applied similar message or same URL to cluster campaigns, [143] used Shannon Information Theory to estimate the similarity between two accounts. All the similar accounts were linked to form a graph, and cohesive campaigns (defined in [127]) were extracted using the intuition in [65]. After that, they extracted 9 features from these found campaigns. SVM with RBF kernel was used to do the classification. The dataset used was from Tweets2011, and 844 campaigns were extracted. By carefully checking the campaigns manually, 375 regular campaigns, 278 promoting ones and 140 spam ones were labelled. The classifier was trained by 200 samples (out of 844, distribution was not shown), and the classification results were good.

Xiao *et al.* in [133] proposed a scalable approach to detect clusters of fake accounts. They aimed at fast detection even before the fake accounts established connections with normal users. Their approach consisted three modules, cluster builder, profile

featurizer and Account Scorer. Cluster builder took all accounts, and built clusters based on IP address and date. Profile featurizer was then used to extract basic distribution features, pattern features and frequency features. At last, account scorer was trained and used to detect clusters of fake accounts. This system was implemented in Java, Hive and R, and successfully detected over 250,000 fake LinkedIn accounts.

2.2.2.7 Detecting Spam and Spammers Together

The above works were either detecting spam or spammers, however, [30] proposed a framework to jointly detect web spam and spammers at the same time, which was called co-classification in this work. The problem of detecting web spam and spammers was formalized to learn a pair of classifiers which can differentiate web spam from non-spam and spammers from non-spammers accurately. Authors used the extensions of least-square support vector machine proposed by Suykens et al. to classify spam and spammers. The model parameters were estimated from the training data. In order to test effectiveness of the proposed algorithm, a real-world dataset from delicious.com was used. The whole dataset consisted of about 3 million users and 110,000 bookmarks. They labelled the bookmarks with the URLs from spam benchmark data in [128]; a user would be labelled as spammer if he posted to at least one spam bookmarks. A sampled dataset contained 20,000 bookmarks and 20,000 users (spam/non-spam ration is 1/4) was used for experiments. Results showed that F-measure of their co-classification approach was at least 5% higher than linear and

non-linear SVM in terms of classifying both spam bookmarks and spammers. Their work is also applicable for other social media sites where the following information is available: links between users, links between users and their submitted web content, and content based features derived from the web content.

Wu *et al.* also proposed a co-classification approach to detect spammers and spam messages together in microblogging sites [131]. In this work, authors found three observations which could approve that detecting spammers and spam messages together achieves better results than performing single task. First one was user-message relation, which indicates that spammers tend to post more messages than legitimate users. Second one was user-user relation, which underlies that spammers are usually followed by spammers to get collaborated. Third one was message-message relation, which observes that massive spam messages share same topic as spammers worked together to spread spam. Based on these three observations, authors proposed a ADMM-based [18] model to co-detect spammers and spam message on Sina Weibo.

2.2.2.8 Detecting Spam in location-based Social Network

Despite the popularity of traditional online social networks like Facebook and Twitter, Location Based Social Networks (LBSNs) are attracting users in exponential rates. Like spamming activities in other online social networks, LBSNs also need to deal with spammers who are posting unrelated or even malicious tip comments about locations. Helen et al. in work [35] addressed the problem of detecting tip

spam in Apontador that was a Brazil location based social network site. The dataset was provided by Apontador, which contained 1260 spam tips and 1260 non-spam tips manually labelled from 15^{th} to 22^{rd} September 2011. Based on the information contained in the labelled dataset, authors also crawled the places information, users and their neighbours information, which formed a weak social graph consisted of 137,464 users. They considered four feature sets (41 features in total) to differentiate the characteristics of spam and non-spam tip, including content features, user features, place features, and social features. A state of art algorithm in machine learning, Random Forest, was used to detect spam tips. After running 5-fold cross validation ten times, they obtained 0.84 TP rate and 0.918 FP rate. The experimental results further showed that, the classification accuracy could reach 82.6%. However, the dataset was not big enough to avoid bias as well.

2.2.2.9 Unsupervised Spam Detection

There are also some works using unsupervised method to detect social spam. [110] proposed UNsupervised socIal networK, UNIK, which relies on the social graph. They firstly built a social graph based on posted URLs and the connections of users. Based on the graph, Tan *et al.* calculated the node degree, and flagged users as spammers whose node degree exceeded the pre-defined threshold. This work heavily relied on social graph and may missed some spammers who did not post URLs.

In [119], authors used a matrix to represent users and their social behaviour features. PCA was then used to extract the principal components from the matrix. The top-k components represented the normal users' behaviour, and the rest represented anomalies. After that, authors calculated the bounds on the L^2 norm. If one user's the L^2 norm exceeded a user-defined threshold, it would be flagged as anomaly.

The previous two works need human intervention to set up an optimal threshold. To address this shortcoming, Fathaliani *et al.* proposed a unsupervised scheme to automatically distinguish spammers and legitimate users [41]. They firstly constructed a feature vector of each user to represent his behaviour and interactions with others. Then they used Dirichlet mixture [17, 78] to create a statistical framework to model the users' behaviour. After estimating the probability density, the Dirichlet components were calculated. The Dirichlet component which contains vectors with smallest values corresponded with spammers.

Works	Algorithm	Features	Datasets	Platform
Benevenuto *et al.* [8]	SVM with RBF kernel	39 content based features, such as No. of words per tweets, No. of URLs per works *etc.* and 23 user based features, such as No. of followers, No. of followees, No. of tweets, *etc.*	1065 manually labelled users sampled in three trending topics: Michael Jackson's death, Susan Boyle's emergence, and #musiccommonday from a big crawled dataset August 2009	Twitter
Wang *et al.* [121]	Naive Bayes	graph based features, such as No. of friends, No. of followers, and reputation *etc.* and content based features, such as Duplicate of Tweets, HTTP Links, Replies and Mentions, and Trending Topics	500 manually labelled accounts with 3% (15) spammers	Twitter

continued on next page

continued from previous page

Works	Algorithm	Features	Datasets	Platform
Lee *et al.* [66]	Top 10 supervised classifiers implemented in WEKA (Decorate, SimpleLogistic, FT, ...)	user demographics: age, gender, location, *etc.*; user contributed content: "About Me" text, posts, comments, *etc.*; user activity features: posting rate, tweet frequency ; user connections: No. of friends, or followers in the social network	1570 spammers' profiles in MySpace (Oct 2008 Jan 2008) and 500 spammers' profiles in Twitter (Aug 2009 Sept 2009)	MySpace and Twitter

continued on next page

continued from previous page

Works	Algorithm	Features	Datasets	Platform
Stringhini *et al.* [107]	Random Forest	FF ratio, URL ratio, Message Similarity, Friend Choice, Message Sent, and Friend Number	1000 samples training, 790,951 profiles for detection in Facebook, 135,834 in Twitter June 2009 June 2010	Facebook, MySpace and Twitter
Ahmed *et al.* [3]	Naive Bayes, C4.5 and JRIP	14 features from 4 categories: Interaction, Post/Tweets, URLs, Tags/@mentions	manually labelled 320 Facebook profiles and 302 Twitter Profiles	Facebook and Twitter
Xu *et al.* [134]	Random Forest, Bagging, J48, Random Tree, BayesNet, Logistic	word features	1937 spam tweets and 10942 non-spam tweets on Twitter, 1338 spam posts and 9285 non-spam posts on Facebook	Facebook and Twitter
Yang *et al.* [135, 137]	Supervised Classifiers (RF, DT, BN, DE)	24 features from 6 categories: Profile based, Content based, Graph based, Neighbour based, Automation based and Timing based	2060 detected spammers using URL blacklisting services	Twitter

continued on next page

continued from previous page

Works	Algorithm	Features	Datasets	Platform
Wang *et al.* [122]	Naive Bayes, KNN, SVM, Decision Tree, and Random Forest	User features, Content features, N-gram features, and Sentiment features	Social Honeypot dataset [66] and 1KS-10KN dataset [138]	Twitter
Song *et al.* [104]	Naive Bayes, KNN, SVM, Decision Tree, and Random Forest	distance, connectivity	148,371 profiles, 267,551 tweets	Twitter
Gao *et al.* [43]	C4.5	Obsolete features used in email spam detection, such as message size, network based features, and OSN specific features, such as sender social degree, interaction history, and General features, such as cluster size, average time interval	187 million Facebook wall posts (Jan 2008 June 2009) 17 million tweets (June 2011 July 2011)	Facebook and Twitter

continued on next page

continued from previous page

Works	Algorithm	Features	Datasets	Platform
Zhang *et al.* [143]	SVM with RBF kernel	Average Posting Interval, UTNum, UTFrequency, URLNum, CampaignDensity, DomainNum, VU-ratio, DTSimilarity, and Blacklisted Number	844 candidate campaigns extracted using the algorithm in [127]	Twitter
Xiao *et al.* [133]	Random Forest, Logistic Regression, SVM	Basic distribution features, Pattern features, and Frequency features	over 500K accounts collected from LinkedIn	LinkedIn
Jin *et al.* [57]	Scalable Active Learning	Image content features, such as colour histogram, colour correlogram, CEDD, and Text features, such as caption, description, comments, and Social Network Features, cluster size, average time interval	Popular Facebook pages with more than 500,000 fans	Cross-platform (but demonstrated only in Facebook)

continued on next page

continued from previous page				
Works	Algorithm	Features	Datasets	Platform
Chen *et al.* [30]	A variant of maximum margin classifier (the extensions of LS-SVM)	User based, Bookmark based, Tag based, Post based, and Fan based	A real-world data set obtained from delicious.com, which contains 3 million users	Social Media Websites
Wu *et al.* [131]	ADMM [18]	User-Message relation, User-User relation, and Message-Message relation	5090 users together with 53,484 messages from Sina Weibo	Sina Weibo
Costa *et al.* [35]	Random Forest	Content based features, User Attributes, Place Attributes and Social Attributes	manually labeled 1260 tip spam from 15^{th} to 22^{rd}, September 2011 in Apontador	Apontador, a Brazilian Location Based Social Network

Table 2.3: Summary of Reviewed Works in Section 2.2.2

2.2.3 Text Based Techniques

Due to the rich linguistic information contained in tweets, researchers [4, 33, 72] begin to leverage text information for Twitter spam detection.

Clark *et al.* proposed a natural language approach to detect automated accounts

(*i.e.* spammers). After analysing linguistic characteristics of users' tweets, they proposed two novel features, average pairwise tweet dissimilarity and word introduction decay rate. The first one is based on the fact that, legitimate accounts usually have very dissimilar tweets, while automated ones have very similar contents. The second feature based on the fact that the two types of accounts have different number of unique word types introduced over time from a given sample. Experiments show that these two features can discriminate automated and legitimate accounts effectively.

Based on the finding that spammers use trending topics to disseminate malicious tweets in [79], Antonakaki [4] study the trending topics from 6.5 million tweets, and find that the number of trending topics has the highest divergence between spammers and legitimate users. They then train a decision tree regression classifier, which can correctly identify 73.5% spammers with 0.25% false positive rate.

In [72], Liu *et al.* observe that legitimate users focus on limited number topics, while spammers concentrate on a wide range of topics. In addition, they find that legitimate users and spammers have different interested topics. Based on these observation, they use Latent Dirichlet Allocation (LDA) [13], a topic model to compute two topic-based features. One is Local Outlier Standard Score (LOSS), which reveals users' interest on various topics. Another is Global Outlier Standard Score (GOSS), which indicates the users interest on certain topics compared with other users. These

two features can distinguish spammers and legitimate users effectively. When incorporating these two features with others in work [65], the classification results are further improved.

2.3 Discussions

The persistently unsolved challenges in social spam detection field will be outlined in this section, by using the terminology and context provided above.

2.3.1 Public Data and Ground Truth

The most obvious difficulty in social spam detection is a persistent problem for researchers: the lack of shared datasets to serve as the test data as well as ground truth (*i.e.* the flagged message to indicate whether it is spam or not) for validation. How to balance between individual privacy against other needs, such as security, critical infrastructure protection, or even science, has long become a challenge or law enforcement, policy makers and scientists. It is good news that laws or legislations prevent unauthorised parties from examining the normal users' online social activities. Current policy, however, makes it hard or even impossible to provided researchers with the data they needed to study the OSNs. The current situation is that, laws intended to protect individual's privacy leaves the researchers exploring the OSNs' ecosystem in the dark. Benevenuto *et al.* shared their crawled data which contains huge users'

tweets in 2009. However, the data was unavailable now according to Twitter's explicit request. They are only sharing the anonymised topology of the Twitter social network which makes little sense to detect spammers on OSNs.

To solve the data sharing problem, researchers in other fields have proposed a untested alternative is to "move code to the data", where the data providers run the code and send researchers back the results. There are, however, some difficulties to do this, since the data providers may not have the resources and incentives to review the code to ensure it runs correctly.

Researchers has also explored the possibility to share anonymised data with ground truth [109]. Unfortunately, there are no tools for labelling the data currently. A large number of works are manually labelling the data before experiments [3, 8, 121]. While manual inspection is very expensive for human labour, it is not realistic to label a large set of spam messages. Some researchers were using honeypots to collect spammers [66, 107]. Unfortunately, the amount of collected spammers was relatively small, even after collecting for a long period [66]. In our work, we apply a commercial labelling tool provided by our research partner to solve this issue.

2.3.2 Limitations of Current Approaches

Most of the works which use machine learning are focusing on the supervised approaches. Supervised algorithms, however, have their own inherent limitations. First

of all, labelled training data is needed for supervised approaches. As pointed out before, the labelling for spam message is very human labour cost. In addition, the labelling work must be done repetitively to maintain effectiveness for spam detection given the volatility of the spam content and some spam posting pattens.

More importantly, we have identified a "Spam Drift" issue, which leads the classifier become inaccurate gradually as time goes on. Details are cover in Chapter 5.

2.4 Summary

The increasing popularity of Online Social Networks not only attracts research interest, but also malicious activities. Traditional spammers in email have transferred to OSNs due to its huge user base and easy-suspicious. In order to tackle the spamming activities, researchers have proposed a number of significant works in a short time period.

Motivated by the power of machine learning algorithms, a set of works use machine learning algorithms, such as C4.5, SVM, Naive Bayes, *etc* to detect Twitter spam. Works using social relation were also proposed. More recent works are focusing on the early detection of spam so as to quickly mitigate threats. Text based detection techniques are very promising as they only extract information from tweets. We provide a comprehensive review of all these works.

As a new and promising field, there is still a lot of room for further research. While most of the techniques build supervised classification frameworks, unsupervised approaches can be evaluated. Each ML algorithm may perform differently toward different datasets, and may require different parameter configurations. The use of a combination of classification models is worth investigating. Parallel processing for real-time detection may be useful when the classifiers need to cope with millions of concurrent messages simultaneously. In addition, the early detection of spam is very important to reduce spam's harm.

In the next chapter, we will provide an in-depth study from two aspects to better understand Twitter spam.

Chapter 3

Analysing and Understanding Twitter Spam

3.1 Introduction

As Twitter's user base is growing, it has also become more attractive to spammers. The study [48] confirmed the existence of large scale spam in Twitter. Spam can entice victims to malicious sites and pollute Twitter platform. It not only interferes with real time search and the statistics retrieved by tweet mining tools, but also wastes human attention [80]. Many spam filtering technologies rely on blacklists to block spam, but this kind of filtering only suppresses spam links that are blacklisted at the time of posting. Moreover, most spam was posted on Twitter in the form of short URLs. This technique generally makes the task of identifying spam on Twitter more difficult. Therefore, the authors in [48], after examining 400 million public tweets and 25 million URLs, concluded that blacklists used in Google SafeBrowsing were ineffective in the detection of spam.

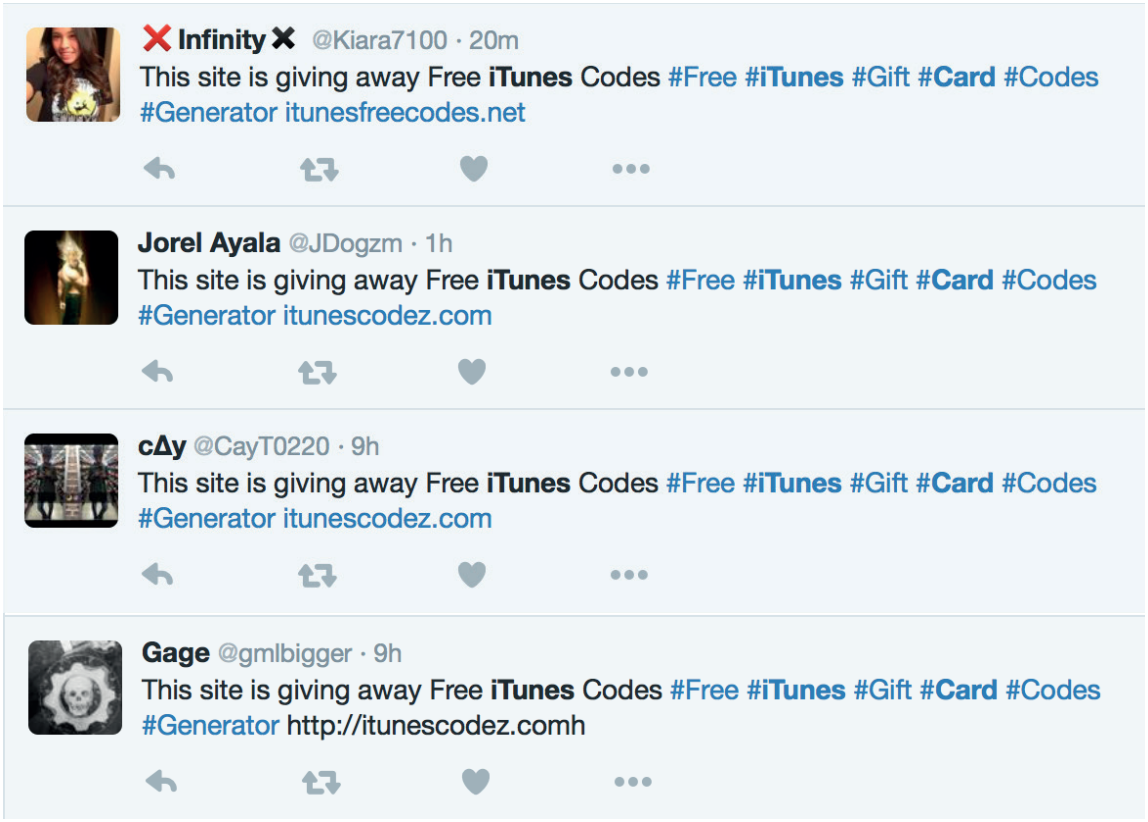


Figure 3.1: An Example of Twitter Spam

Current security experts suggest the best defence against spam is to educate Internet users to never click suspicious links in tweets. However, spammers leverage some attractive deceptive topics, such as "gain followers", "cracked games", *i.e.* to lure users to click their malicious links. We refer this kind of information as deceptive information. Take Figure 3.1 for example, the claimed "iTunes gift card" attracts a number of victims fallen into the trap. In this case, "iTunes gift card" is a piece of deceptive information.

The deceptive information is one of the key factors to the spreading efficiency of

spam on Twitter. A better understanding of deceptive information is crucial to spam detection techniques. Therefore, we are motivated to thoroughly study the deceptive information employed by spammers. We used Trend Micro's Web Reputation Technology to conduct the spam labelling on our collection of real data (over 568 million tweets from two weeks' capture, including 5.8% spam tweets were identified). We then used a graphical algorithm to infer more spam. We clustered the spam tweets into 17 groups according to their content. These 17 groups accounted for 75% of the spam we identified. We also inspected users' clicks on the deceptive information. The results showed there was great variability in the effectiveness of various spam topics.

In addition, We have found that spammers are becoming "smarter" on Twitter. While researchers are developing methods to detect spam, spammers continuously invent new spamming strategies to bypass the detection. Thus, we also provide a detailed study of spammers' new spamming strategies.

This chapter consists two main sections: one is about investigating deceptive information, another is about "smarter" spamming strategies.

Our contributions in this chapter are summarised as below:

- We collected a big dataset for the research on Twitter spam, which contains around 600 million tweets with more than 33 million spam tweets. Based on the dataset, we present an in-depth analysis of deceptive information contained in Twitter spam.

- We used graph clustering techniques to infer more spam with the help of identified spam.
- We studied the clicks per tweet for four kinds of deceptive topics on our dataset, and found that the clicks per tweet vary according to different topics.
- We also examined the victims' country distribution corresponding to different deceptive information. Interestingly, most victims still clicked the malicious URLs embedded in tweets, even if they do not speak the language which is used to write the tweets.
- We have identified "smarter" spammers with new spamming strategies.

3.2 Investigating Deceptive Information on Twitter Spam

3.2.1 Big Dataset and Spam Labelling

We collected a complete Twitter feed with URLs for the two-week period from 24th September 2013 to the 8th October 2013 (We understand that the study period overlapped with a significant spam outbreak, which has been confirmed by Twitter). While it is possible to use Twitter to send spam and other messages without using URLs, the majority of spam and other malicious messages on the Twitter platform contain URLs [39]. In the thousands of spam tweets which were inspected by hand

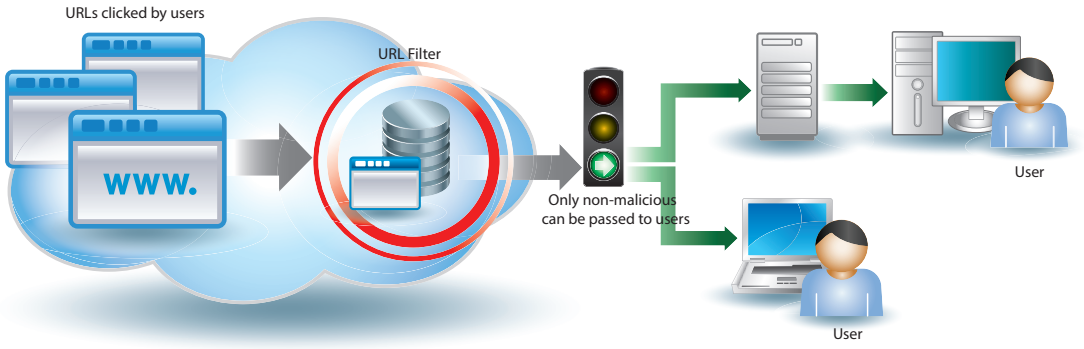


Figure 3.2: Trend Micro WRT System Framework

during the research, we found only a handful of tweets without URLs that could be considered as spam. In addition, spammers mainly use embedded URLs to make it more convenient to direct victims to their external sites to achieve their goals, such as phishing, scams, and malware downloading [143]. Therefore, we restricted this research to tweets with URLs.

Currently researchers are using two ways to label spam, manual inspection [8,121] and blacklists filtering, *e.g.* google safebrowsing, [44, 48, 135, 136]. While manual inspection can label a small amount of training data, it is very time- and resource-consuming. A large group of people is needed to help during the process. Although HIT (human intelligence task) websites can help to label the tweets, it is also costly and sometimes the results are doubtful [23]. Others apply existing blacklisting service, such as Google SafeBrowsing to label spam tweets. Nevertheless, these services' API limits make it impossible to label a large amount of tweets.

We applied a different way to label spam. There were two steps involved in this

Table 3.1: Collected Data

Date	Tweets with URLs	Spam Tweets labelled by WRT	Spam Tweets labelled by Cliques	% Spam Tweets
25 Sept 2013	39,257,353	1,871,502	420,986	5.8%
26 Sept 2013	47,252,411	2,602,228	588,372	6.8%
27 Sept 2013	49,465,975	3,545,467	402,048	8.0%
28 Sept 2013	37,806,326	1,809,205	209,730	5.3%
29 Sept 2013	nil	nil	nil	nil
30 Sept 2013	nil	nil	nil	nil
1 Oct 2013	48,778,630	2,125,149	386,340	5.1%
2 Oct 2013	51,728,355	3,174,904	564,693	7.2%
3 Oct 2013	51,638,205	3,343,961	588,225	7.6%
4 Oct 2013	49,230,861	2,992,197	406,329	6.9%
5 Oct 2013	44,165,664	1,995,249	298,290	5.2%
6 Oct 2013	45,089,730	1,689,424	317,023	4.4%
7 Oct 2013	50,457,403	2,032,507	273,287	4.6%
8 Oct 2013	42,031,232	973,573	178,366	2.7%
9 Oct 2013	16,612,318	448,971	89,162	3.2%
Total	573,514,463	28,604,517	4,722,851	5.8%

process. The first step applied Trend Micro's Web Reputation Technology [89] (refer to Figure 5.3) to identify which URLs were deemed malicious. Trend Micro's WRT

maintains a large dataset of URL reputation records, which are derived from Trend Micro's customer opt-in URL filtering records. WRT is dedicated to collecting the latest and most popular URLs, analysing them, and then providing its customers with real-time protection while they are surfing the web. According to the lasted testing report from AV-Comparatives [34], the block rate of WRT is 99.8%, and false positive rate is also very low. Hence, we rely on Trend Micro's WRT to check whether a URL is malicious. Realising the fact that WRT may miss some malicious URLs, we also used a clustering algorithm to label more malicious URLs in the second step.

We can see from Table 3.1 that, about 5 million more spam tweets is identified by our clustering algorithm. The method is well explained in the next section. We define those tweets which contain malicious URLs as Twitter spam. During this period, we collected a total of 573.5 million tweets and identified 33.3 million malicious tweets (28.6 million by WRT and 4.7 million by Cliques), which accounted for approximately 5.8% of all tweets. As can be seen from Table 3.1, the daily spam ratio ranges from 2.7% to 8.0%.

3.2.2 Inferring and Grouping Twitter Spam

As mentioned before, we applied two steps to identify twitter spam. One was using Trend Micro's WRT. Although the false positive rate of WRT is very low, it may also miss some spam tweets. In addition, our research goal is to have high level

understanding of the various deceptive topics used in Twitter spam. So, the second clustering step was involved. The advantages are two fold: 1) by clustering unlabelled tweets and spam tweets into groups, we can find more spam tweets since only similar tweets can fall into the spam group; 2) it would be more useful to analyse spam groups, *e.g.* studying the behaviour of the spamming group, rather than understanding a huge mass of spam tweets.

We used a graphical clustering approach which made use of bipartite cliques, instead of machine learning algorithms to group the spam tweets. In order to identify bipartite cliques [92, 108], we firstly extracted the domains of URLs embedded in tweets along with the senders of the tweets. Then, we constructed a graph where the Twitter users were nodes on one side of the graph while the domains in sent tweets were nodes on the other side. For each tweet from user U that contains a link with domain D, we connected this user U to domain D in the graph. Once the graph was fully connected as in Figure 3.3, a bipartite clique was formed. The rationale for identifying bipartite cliques is that if we can find groups of accounts that have sent tweets with spam domains in one clique then it is very likely any account sending all the domains in this clique is sending spam as well. Figure 3.3 gives an example of a bipartite clique found in the data consisting of 11 domains and 727 users, that is each user in the clique has sent tweets which have all the spam domains, and each spam domain is included in the tweets that all users send.

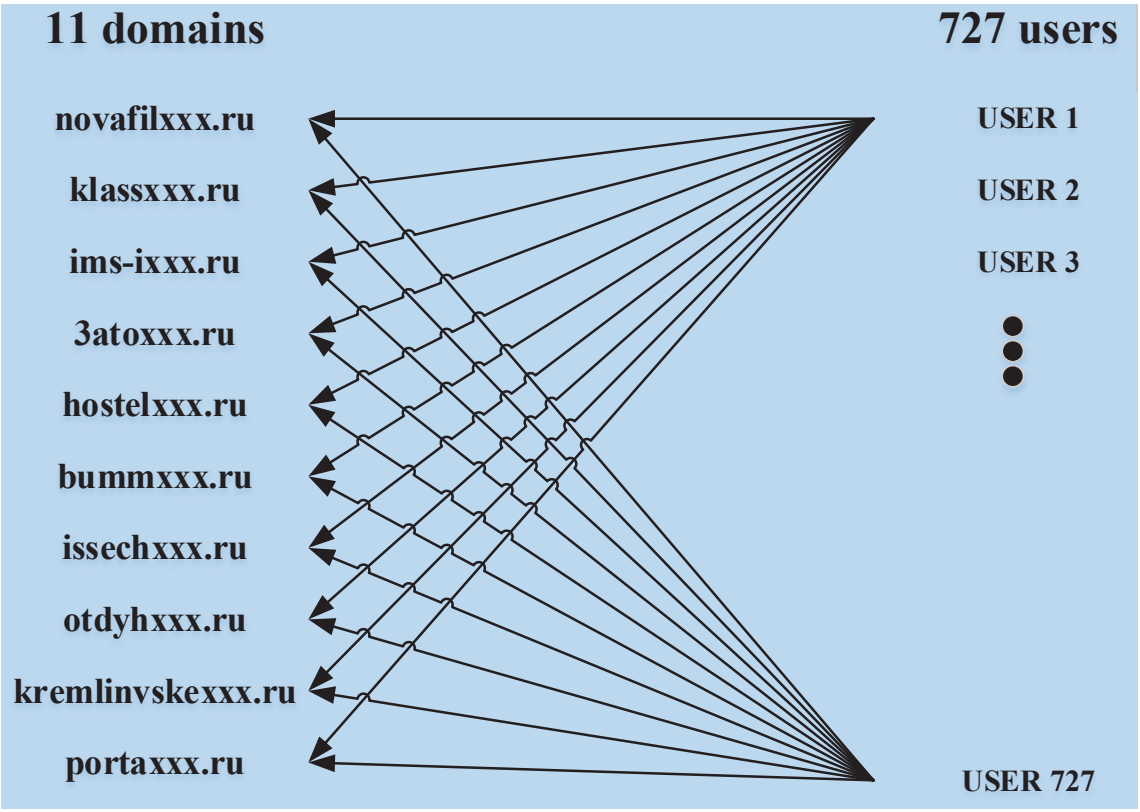


Figure 3.3: An Example of Bipartite Clique

Through this way, we not only found more spam tweets, but also successfully identified 17 cliques of Twitter spam in our collected data, each of which accounts for 1% or more of the total spam. Table 3.4 gives a description of each of the clique generated. Specifically, Clique G is Twitter follower spam which accounts for 2.5% of the Twitter spam [91]. The first column of Table 3.4 gives a description of the deceptive information of spam cliques. The second column stands for the percentage of tweets out of 28 million spam tweets. The "Senders" column represents the number of confirmed senders in each clique. A confirmed sender has sent tweets to all the

domains in the corresponding clique. The "Domain" column lists the number of different domains used by spammers, and the last column indicates the percentage of accounts within each clique that have been suspended by Twitter after we checked their status in December 2013 (two months after the collection period).

Table 3.2: Spam Breakdown

Clique	Description	% Spam Tweets	Senders	Domains	% Suspended
A	Edu spam, etc	27.28%	797	24	10.3%
B	Cracked software, games spam	8.11%	578	20	31.5%
C	Edu spam	6.26%	539	20	19.7%
D	Cracked software	6.19%	9509	21	12.0%
E	Cracked software spam	4.39%	727	11	11.6%
F	Printer / mobile spam	3.72%	12275	3	89.1%
G	Twitter follower spam	2.54%	59205	1	2.1%
H	Video / Mobile / Cracked Software/ games spam	2.23%	8987	50	95.2%

continued on next page

continued from previous page					
Clique	Description	% Spam Tweets	Senders	Domains	% Suspended
I	Games, computer spam	2.04%	608	19	97.9%
J	Edu spam, etc	1.99%	284	14	47.9%
K	Shirt-spam	1.91%	1699	5	74.7%
L	Games, mobile printer spam	1.81%	1197	18	98.8%
M	Computer / Printer spam	1.77%	26603	60	42.3%
N	Games / Hardware spam	1.53%	2514	70	90.0%
O	Computer game / mobile device spam	1.41%	1491	73	94.7%
P	Credit card spam and edu spam	1.08%	8541	32	72.5%
Q	Cracked software and games spam	1.02%	9066	4	98.6%
	Other spam	24.74%	Nil	Nil	Nil

We have drawn some observations from the results of Table 3.2 as follows:

- The 17 cliques account for 75% of the spam we identified on Twitter.
- Twitter responds relatively effectively to some spam outbreaks. For example they have identified and suspended over 95% of spam accounts in Clique H, I, L and Q. However, some spam Clique are not being detected effectively. For example, Clique A, which accounted for over 27% of the total spam, had only

approximate 10% of its accounts suspended by Twitter.

- There are a minority of spammers (around 24.74%) who can hardly been grouped in our study. We categorised them as "Others" in Table 3.2.

3.2.3 Category of Deceptive Topics

Note that some groups in Table 3.2 are sharing the same deceptive information but they belong to different cliques. In this part, we further characterised them into four categories:

1. Malware: content in spam which distributes websites containing malware, such as games, cracked software, and hardware drivers.

2. Phishing: content in spam which distributes phishing sites, purporting to be a trusted party such as a financial institution.

3. Twitter follower scam: content in spam which entices users to install an app which is granted authorisation to their Twitter Accounts (Group G in Table 3.4).

4. Advertising: content in spam that distributes sales of education assignments, shirts, videos, *etc.*

The four categories are formed according to different deceptive information contained in spam groups. We will later investigate the users' clicks per tweet for each

Table 3.3: Spam Categories

Spam Categories	Relative Click Through Rate
Malware	0.03065 %
Phishing	0.00959 %
Advertising	0.00239 %
Twitter Follower Scam	0.00112 %

category.

3.2.4 Users' Clicks on Deceptive Information

Study [58] examined the click through rates of email spam and found click through rates (the number of people who arrive at the website having clicked the email) vary from 0.003% to 0.02%. However, [48] estimated the overall click through rate for Twitter spam as 0.13%, suggesting the click through rate for Twitter spam was two orders of magnitude higher than for email spam.

Distinguished from the studies [48] and [58], we further inspect users' clicks on the deceptive information of each group in order to examine how attractive those topics are. We make use of Trend Micro's WRT to calculate the clicks per tweet. A part of this service is a feedback system for those people who opt in for malicious feedback. The feedback is anonymous. We examined the feedback data to determine which spam URLs were being clicked on from tweets. From the feedback, we can see the

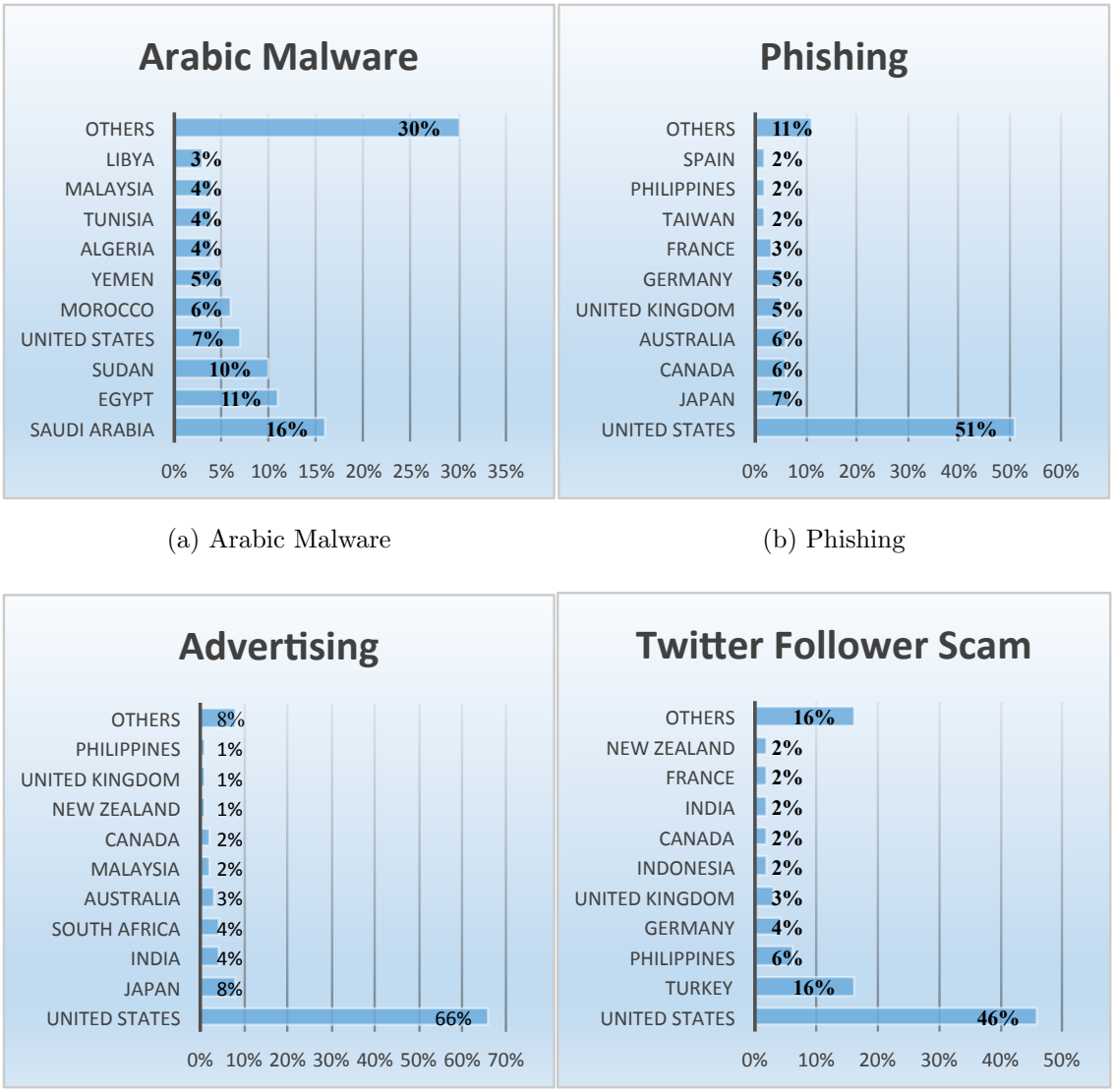
number of web hits on malicious URLs in one category. Then clicks per tweet equal to the total number of web hits divided by the number of tweets in this category. Please note that the clicks/per could be higher as the number of web hits is larger when considering all the users rather than Trend Micro users only. The clicks per tweet listed in TABLE 3.3 for "Malware", "Phishing", "Advertising" and "Twitter Follower Spam" are 0.03065, 0.00959, 0.00239, and 0.00112 respectively. This data can be used to compare the effectiveness of various deceptive spam. However, calculating the absolute click through rate would require a more global perspective or access to Twitter's infrastructure.

From the results listed above, we can see that "malware" has the largest clicks per tweet among the deceptive information. It is understandable because people are seeking "cracked software" or "free games" on Twitter. The downloaded software from those links also contains malware. If they install the so-called "cracked software", malware will also be installed. Once the victim is infected by malware, the victim's computer will become one of the "zombies" in a botnet, which will recursively contribute to the spam campaigns. "Phishing" is also a major topic of deceptive information which can attract many victims' interest as spammers are applying more enticing information to attract victims to click. For instance, the spam tweet may come with a sentence like "Some one is saying something bad about you" to entice victims. It seems that users are becoming more wise as the "If you follow me, you

will get five more followers" spam ("Twitter follower spam") is much less effective than others. The "clicks per tweet" data clearly indicates there is great variability in the effectiveness of various deceptive information.

3.2.5 Who Clicked the Spam

We also investigate the countries of victims who clicked the spam we identified. As we can access the feedback data of WRT, IP addresses of clickers can be retrieved. We then use the geolocation database to map IP address with its originating country. While some IP addresses may be associated with inaccurate locations (wrong addresses within the same city), the broad geographic area such as cities, is correct. During the study period, there was an outbreak of Arabic tweets that led to a malware site. We used this outbreak incident as an example to investigate the click through rate of malware. The majority of respondents were in Saudi Arabia, Egypt, and Sudan followed by the United States. As shown in Figure 3.4a, this malware site attracted victims from almost all the countries in the Arabic world, wherein Saudi Arabia contributed the largest portion of the victims with 16%. For the "Phishing" category, we can see from Figure 3.4b that the majority of victims were from the U.S., with a percentage of 51%, followed by Japan, Canada, Australia, *etc.* Victims from the United States also accounted for the majority of both "Advertising" and "Twitter Follower" spam. The results are shown in Figure 3.4c and 3.4d. Similar to



(a) Arabic Malware

(b) Phishing

(c) Advertising

(d) Twitter Follower Scam

Figure 3.4: Regional Distribution of Victims

email spam targets, the U.S. is also the prime target country of Twitter spam.

We also noticed that landing pages of URLs in many spam tweets were written in Russian, thus we further investigated the victim who clicked on this kind of spam

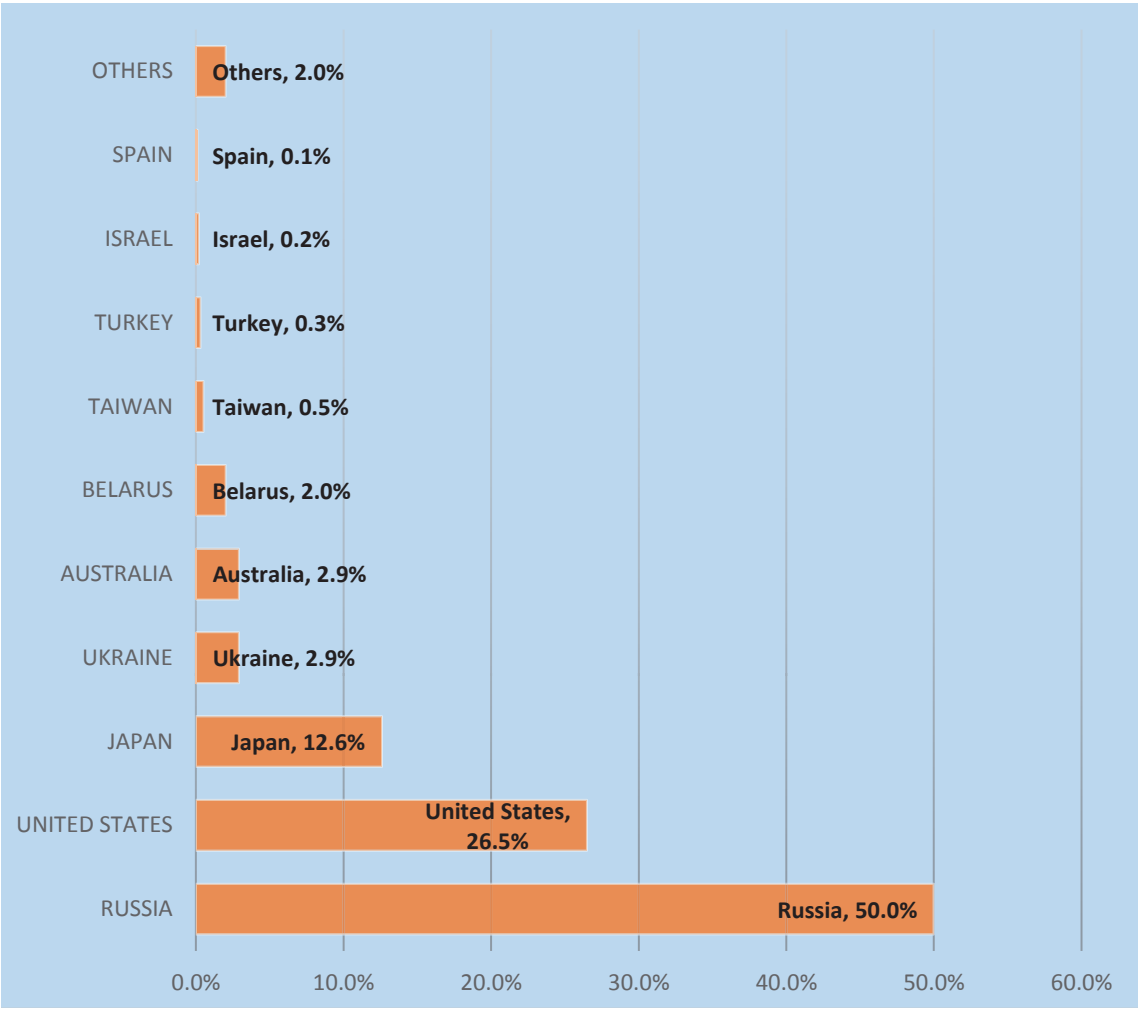


Figure 3.5: Russian Spam

(named "Russia Spam" in Figure 4). As shown in Figure 3.5, the majority of victims who clicked on "Russia Spam" were in Russia, with a percentage of 50%. However, victims from many other countries who did not speak Russian also clicked on this kind of spam. We theorise that the content advertised in this spam was sufficiently appealing to some users (cracked software and games, free movies, cracks for mobile

devices, exam answers and homework, *etc.*), so they used tools such as computer translation software to access the inappropriate content.

3.2.6 Discussions

In the study conducted by [48], they believe the blacklists, *e.g.* Google SafeBrowsing, are no longer suitable for detecting Twitter spam. According to the findings presented in this work, we have been inspired to think again about whether the blacklists are indeed useless in the detection of spam on Twitter.

To answer this question, we have examined the response rates of various types of spam on Twitter in previous sections, and found the response rate could vary widely depending on the content and the regional factors about the spam. We practically apply a blacklist technique to Trend Micro's WRT system with varied priorities and suspicious rankings to different spam content and regions, and found the detection performance was reasonably effective to identify around 6% spam (refer to Table 3.1) in a collection of 573 million tweets. Therefore, we conclude the previous work [48] which quotes a single response rate for Twitter spam is inadequate. In fact, it is important to quote response rates for different types of spam involved.

Our findings are also beneficial to the development of spam detection techniques. There are over 400 million tweets posted by users every day, 25% of which contain URLs. To monitor such a large volume of tweets and remove those with spam links

is computationally too expensive to be implemented in the real world. Our findings reveal that different deceptive information has different click through rates in various countries. This suggests the spam detection system should pay more attention to the tweets that contain the deceptive information mentioned before, as detection efficiency will be greatly improved.

3.3 Spammer Are Becoming "Smarter" on Twitter

As we discussed before, spam is a problem throughout the Internet, and Twitter is not immune. In addition, Twitter spam is much more successful compared to email spam [48]. Various methods have been proposed by researchers to deal with Twitter spam, such as identifying spammers based on the tweeting history [8] or social attributes [104], abnormal behaviour detection [39], and classifying tweet-embedded URLs [68]. Although researchers, as well as Twitter itself, have attempted to combat spam, the percentage of spam in the whole platform is still high. We hypothesise that this is because spammers are becoming more cunning on Twitter. While researchers are developing methods to detect spam, spammers continuously invent new spamming strategies to bypass the detection.

In this section, we will first briefly introduce the well-known ways that spammers used to avoid or reduce the chance to be caught on Twitter. After that, we will

show that spammers are now using more advanced spamming strategies, namely 'Coordinated Posting Behaviour', 'Finite-state machine based Spam Template' and 'Passive Spam'.

Table 3.4: Spam Breakdown: we name spam type according to the content of it, *e.g.* "Cracked Software spam" is about cracked software.

Group	Spam Type	% Spam Tweets
A	Edu spam, etc	27.28%
B	Cracked software, games spam	8.11%
C	Edu spam	6.26%
D	Cracked software	6.19%
E	Cracked software spam	4.39%
F	Printer / mobile spam	3.72%
G	Twitter follower spam	2.54%
H	Video / Mobile / Cracked Software/ games spam	2.23%
I	Games, computer spam	2.04%
J	Edu spam, etc	1.99%
K	Shirt-spam	1.91%
L	Games, mobile printer spam	1.81%
M	Computer / Printer spam	1.77%
N	Games / Hardware spam	1.53%
O	Computer game / mobile device spam	1.41%
P	Credit card spam and edu spam	1.08%
Q	Cracked software and games spam	1.02%
	Other spam	24.74%

3.3.1 Well-known Spamming Strategy

At the most basic level, spammers make use of various Twitter functions such as @ and hash (#) tags to engage victims. Spammers can use @ to make spam tweets appear on the victim's feed without being a follower of this victim; for example, a spam tweet will appear on Obama's timeline, if it is written with @obama. By embedding popular

hashtag keywords, one spam tweet can become part of a trending topic that can then be viewed by a victim who is interested in that topic. For example, a spam tweet with #007 will be disseminated to victims who are browsing the popular book and film series. Spammers also use other functions of Twitter, such as 'Reply', 'Favourite', and 'Following' to spread spam [111]. Fortunately, researchers can also make use of these features (such as the number of followers or the number of hashtags) to detect Twitter spam [8].

To bypass such detection systems, spammers apply evasion tactics, such as gaining more followers, posting more tweets and so on [135]. They will not be exposed by the simple detection systems described above, because their activity mimics that of legitimate users. To combat this, researchers propose robust social graph-based features, such as local clustering coefficient, betweenness centrality [135], distance/connectivity [104] to detect those fabricated by spammers.

Spammers mainly use embedded URLs to direct victims to external sites. To evade the domain names, spammers usually use URL shortening service [48]. Even though, they use a long direction chain to be less traceable. URL based detection approaches [68] can successfully combat with such spam.

In addition to the aforementioned spamming strategies, our Twitter spam analysis reveals that spammers are now using more advanced methods (described below).

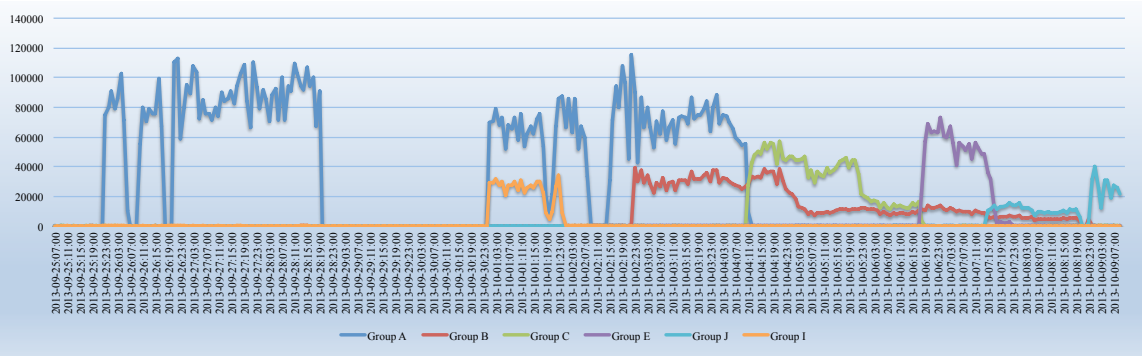


Figure 3.6: The number of spam tweets sent by the six groups

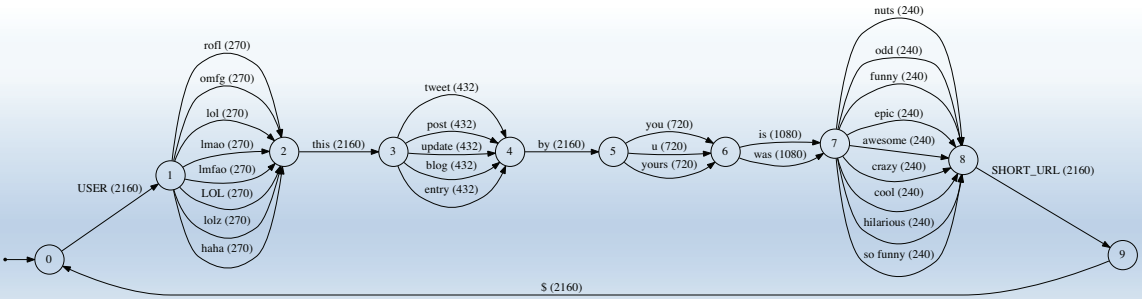


Figure 3.7: FSM based Spam Template

3.3.2 Coordinated Posting Behaviour

We collected a dataset of over 570 million tweets with URLs from 25 Sept 2013 to 09 Oct 2013. Within this dataset, we identified around 33 million spam tweets by using TrendMicro's Web Reputation Technology [88], which accounts for 5.8% of the total tweets. We then clustered the spam tweets using bipartite cliques into 17 groups as shown in Table 3.4. 17 groups dominate over 75% of the spam, while 'others' accounts for less than 25%, indicating that in general, spam was sent by groups. The details of data collection and spam labelling is discussed in Chapter 3.

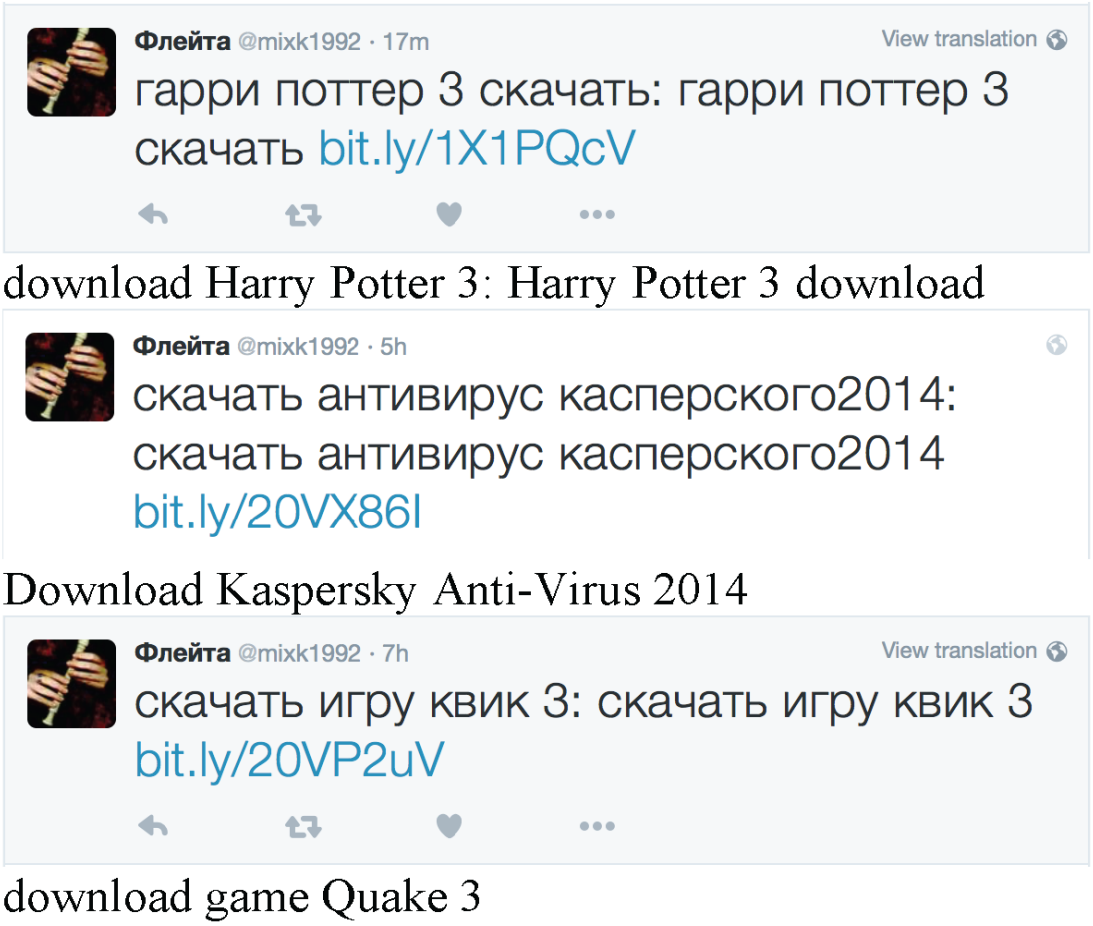


Figure 3.8: Passive Spam (Please note, these Russian spam tweets are just examples, it does not mean that this strategy is only applied by Russian spammers.)

We also found that six groups in Table 3.4 (*i.e.,* Groups A, B, C, E, I and J, the bold and italic letters in the "Group" column) had some common features:

- The URLs embedded in the tweets tend to use a .ru (server of Russian origin) domain.
- The content of the landing pages are written in Russian.

- The URLs tend to end with a Unix timestamp, *e.g.,* http://xxxx.ru/xxxx-1380642617.html.

To study the spamming behaviour of these six groups, we counted the tweets sent per hour by each group. Group A spread spam actively from 26 September (Fig. 1; note: data were lost for 29 and 30 September). When Group A stopped sending spam, Group C started to send spam on 4th of October. Groups C & E, and E & J also displayed this type of spamming behaviour. We regard this behaviour as 'coordinated posting behaviour, a phenomenon in which one group of spam tweets disappears and another group is being sent at the same time. This kind of posting behaviour is more difficult to detect, because spammers change the groups of accounts to abuse Twitter.

Although these groups have the "coordinated behaviour", they were not spreading the same spam. For example, Group C was spreading spam talking about education (we named such spam as "edu spam"), while Group E was distributing "Cracked software spam". This may indicate that these spamming groups were employed to perform different spamming tasks.

3.3.3 Finite-state machine based Spam Template

Some have found that most spam is generated using specific templates [45], which is logical because it is very expensive for spammers to write each tweet manually.

However, the template is often simple [45], for example: 'celebrity name' + 'an eye-catching action' + URL. Therefore, researchers can extract the templates and match tweets to them to detect spam.

We found that spammers are now using more complex templates to generate spam. Surprisingly, spammers are using finite-state machines to generate what we have named 'finite-state machine-based spam templates' (Fig. 3.7). One finite-state machine has a number of states and each edge of it is donated by one word. If we travel from the beginning to the end, we can have one full sentence, such as "lol, this tweet by you is funny + SHORT_URL" in the finite-state machine. By using one finite-state machine based spam template, spammers can generate many different tweets. Take the finite-state machine in Fig. 3.7 for example; it has $8 \times 5 \times 3 \times 2 \times 9 = 2160$ different routes from start to end. This means that, spammers can use this template to generate 2160 different spam tweets with little effort. For example, spammers can write a script which randomly chooses one option from each node to generate one spam tweet. Relying on simple string signatures to match spam tweets will allow most of these finite-state machine-based template spam tweets to escape detection.

3.3.4 Passive Spam

As previously described, traditional spam is distributed by using Twitter functions such @ and #. However, we also found that much spam does not use any tags. As

a result, such spam cannot be identified by machine learning based spam detection that makes use of these features. Contrary to traditional spam, which tries to be involved as much as possible with victims, this spam is only viewed by victims when they search for specific key words. Consequently, we call this 'passive spam'. None of these spam tweets have tags embedded (Fig. 3.8), and they are mostly promoting cracked games, software or pirate movies.

We found that of the victims who clicked on this kind of spam [88], 50% were in Russia. However, victims from many other non-Russian-speaking countries also clicked on this kind of spam. Assuming these users did not speak Russian, we hypothesise that the content advertised in this spam was sufficiently enticing for victims to use translation software to access the inappropriate content. We also found that the suspended rate of this type of spam by Twitter is much lower than others, because they have much lower interaction with users, allowing spammers to make use of this strategy successfully.

3.4 Summary

In this chapter, we collect and detect large number of spam tweets (5.8% out of 560 million tweets). After dividing the spam into 17 groups, we further study the deceptive information in Twitter spam and find that various deceptive content of spam performs differently in luring victims to malicious sites. We also find the regional response rate

to varies Twitter spam outbreaks vary greatly. These factors are of great significance to both academia and industry in the field of Twitter spam detection.

We also note that, while researchers and industry are devoted to developing detection and mitigation approaches to combat Twitter spam, spammers can thwart their efforts with ever-evolving spamming techniques. We have identified and described three complex spamming strategies: 'coordinated posting', 'finite-state machine based spam template' and 'passive spam'. The war with spammers is becoming fiercer and is far from over; we should therefore continue to analyse spammers behaviour and propose robust spam detection systems to make a safe Twitter environment for all users.

In order to stop spammers, researchers have proposed a number of mechanisms. The focus of recent works is on the application of machine learning techniques into Twitter spam detection. However, tweets are retrieved in a streaming way, and Twitter provides the Streaming API for developers and researchers to access public tweets in real-time. There lacks of a performance evaluation of existing machine learning based streaming spam detection methods. In next chapter, we bridged the gap by carrying out a performance evaluation, which was from three different aspects of data, feature and model. We evaluated the impact of different factors to the spam detection performance, which included spam to non-spam ratio, feature discretization, training data size, data sampling, time-related data, and machine learning algorithms.

The results show the streaming spam tweet detection is still a big challenge and a robust detection technique should take into account the three aspects of data, feature and model.

Chapter 4

A Performance Evaluation of Machine Learning Based Streaming Spam Tweets Detection

4.1 Introduction

The research community, as well as Twitter itself, has proposed some spam detection schemes to make Twitter as a spam-free platform. For instance, Twitter has applied some "Twitter Rules" to suspend accounts if they behave abnormally. Those accounts, which are frequently requesting to be friends with others, sending duplicate content, mentioning others users or posting URL-only content, will be suspended by Twitter [114]. Twitter users can also report a spammer to the official @spam account. To automatically detect spam, machine learning algorithms have been applied by researchers to make spam detection as a classification problem [8, 22, 28, 35, 39, 46, 57, 64, 68, 104, 107, 110, 112, 121, 135, 136, 140, 143]. Most of these works classify a user is spammer or not by relying on the features which need

historical information of the user or the exiting social graph. For example, the feature, "the fraction of tweets of the user containing URL" used in [8], must be retrieved from the users' tweets list; features such as, "average neighbours' tweets" in [135] and "distance" in [104] cannot be extracted without the built social graph. However, Twitter data is in the form of stream, and tweets arrive at very high speed [11]. Despite that these methods are effective in detecting Twitter spam, they are not applicable in detecting streaming spam tweets as each streaming tweet does not contain the historical information or social graph that are needed in detection.

Alternatively, classifying a streaming tweet instead of a Twitter user to spam or non-spam is more realistic in the real world [8,123]. In this scenario, only information available in a tweet that is captured by Twitter's Streaming API can be used for classification. In order to better understand ML algorithms' power in classifying streaming spam tweets, we provided a fundamental evaluation in this work. To achieve this goal, we have collected a huge amount of tweets. This data contains more than 600 million tweets, in which we further labelled 6.5 million spam tweets by using Trend Micro's Web Reputation Service [89]. We also extracted some straightforward features for each tweet and examined some ML algorithms' performance on the detection of spam from various aspects. In summary, our contributions of this chapter are follows:

- We created a big ground-truth for the research on spam tweet detection. We reported the impact of the data related factors, such as spam to non-spam ratio,

training data size and data sampling, to the detection performance.

- We extracted 12 lightweight features for streaming tweet spam detection and found feature discretization is important to spam detection performance. A new finding is that the features of spam tweets are time varying.

- We investigated six machine learning algorithms to build up the tweet spam detection model and reported the behaviour of these models under different experiment settings.

4.2 A Big Dataset of Streaming Spam Tweets

A dataset with *ground-truth* (annotated instances with class labels for referencing) is needed to perform a number of challenging machine learning based streaming spam tweets detection tasks. However, we found no datasets are publicly available specially for our task. Although there are a few dataset published by some researchers [8,135], the labelled instances are spammers instead of spam tweets. As a result, we decided to collect streaming tweets and generate the ground-truth. We will also make this dataset available for others researchers to use. In this section, we will describe our large data set with over 600 million tweets, including more than 6.5 million spam tweets.

4.2.1 Collection Procedure

We used Twitter's Streaming API [65] to collect tweets with URLs. The public Streaming API provides real-time access to 1% of all the public tweets, but no access to the tweets sent by protected accounts or direct messages. A tweet is retrieved as JSON format (See Fig. 4.1 for a incomplete tweet JSON example), which is very simple and easy to be parsed as each line of this format represents an object [11]. The returned tweet by the Streaming API contains many attributes of the tweets, such as the text, "the number of retweets", "contained hastags, URLs", *etc.*, and associated Twitter user, such as "the number of tweets", "account generated time", "the number of friends", *etc* [116].

While it is possible to use Twitter to send spam and other messages without using URLs, the majority of spam and other malicious messages on the Twitter platform contain URLs [39]. In the thousands of spam tweets which were inspected manually during the research, we found only a few tweets without URLs which could be considered as spam. In addition, spammers mainly use embedded URLs to make it more convenient to direct victims to their external sites to achieve their goals, such as phishing, scams, and malware downloading [143]. Therefore, we restricted this research to tweets with URLs. During the collection period, we collected a total of over 600 million tweets with URLs [27].

```
{
    "contributors": null,
    "coordinates": null,
    "created_at": "Mon Jan 05 21:16:32 +0000
    2015", "entities": {
        "hashtags": [
            {
                "indices": [
                    91,
                    99
                ],
                "text":
            }       "litmags"
        ],
        "symbols":
        [ ],
        "trends":
        [ ],
        "urls": [
            {
                "display_url": "artsandletters.gcsu.edu/issue-29/",
                "expanded_url": "http://artsandletters.gcsu.edu/
                issue-29/", "indices": [
                    68,
                    90
                ],
                "url": "http://t.co/
            }       siYL9N3LJE"
        ],
        "user_mentions" :
        [    {
                "id": 357729794,
```


Figure 4.1: A Tweet JSON Object

4.2.2 Ground Truth

Currently researchers are using two ways to generate groundtruth, manual inspection [8, 121] and blacklists filtering, *e.g.* google safebrowsing, [44, 48, 135, 136]. While manual inspection can label a small amount of training data, it is very time- and resource-consuming. A large group of people is needed to help during the process. Although HIT (human intelligence task) websites can help to label the tweets, it is also costly and sometimes the results are doubtful [23]. Others apply existing blacklisting service, such as Google SafeBrowsing to label spam tweets. Nevertheless, these services' API limits make it impossible to label a large amount of tweets.

We used Trend Micro's Web Reputation Service to identify which URLs were deemed malicious tweets. Trend Micros WRS maintains a large dataset of URL reputation records, which are derived from Trend Micro customer opt-in URL filtering records. WRS is dedicated to collecting the latest and the most popular URLs, to analysing them, and then to providing Trend Micro customers with real-time protection while they are surfing the web. The maintaining team of WRS is using many frontier technologies to analysing and labelling URL. They will even manually visit the URL if necessary. WRS is trusted by Trend Micros large user base. According to a third party investigation carried out recently, the protection rate of WRT is 99.8%. Thus, the results is trustworthy, as well as the analysis. Hence, through checking URLs with the WRS service, we are able to identify whether a URL is malicious and

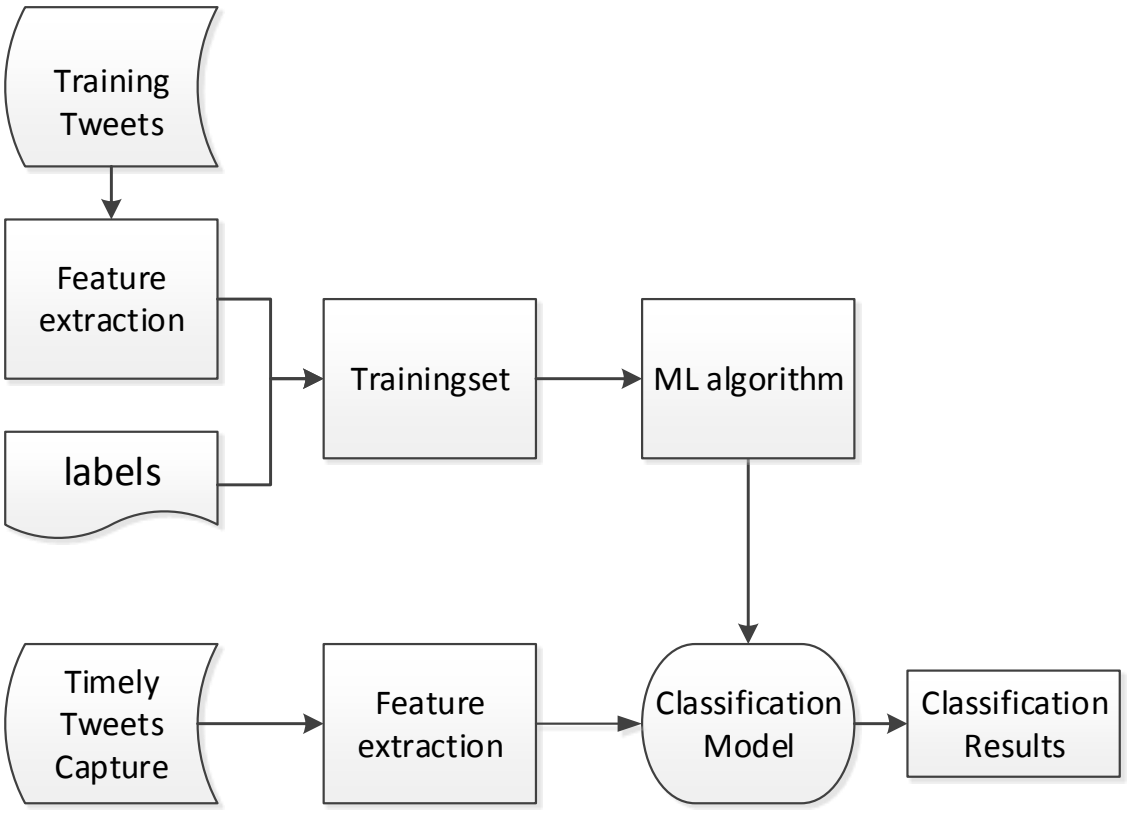
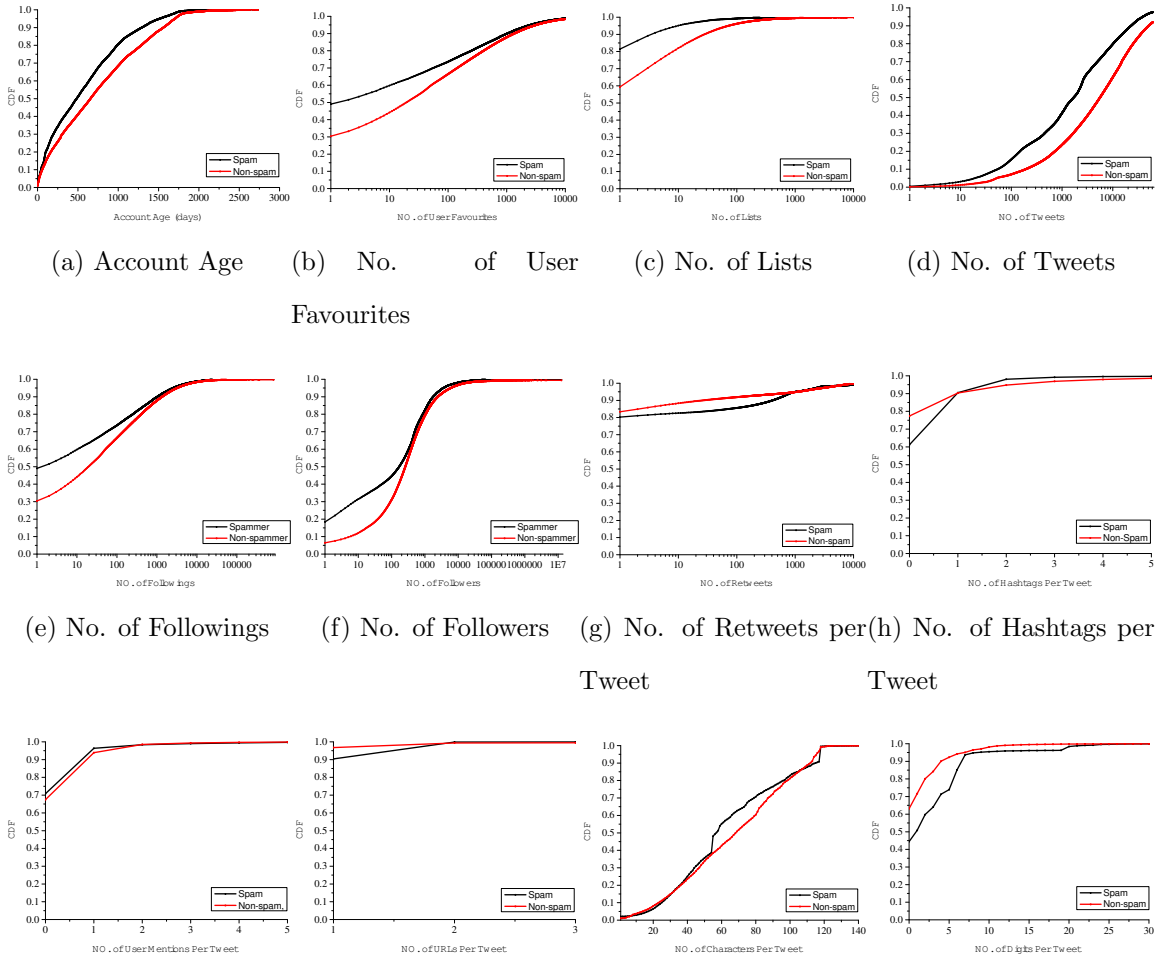


Figure 4.2: ML based Spam Detection Process

the categories a URL belongs to.We define those which contain malicious URLs as Twitter spam. In our data set of 600 million tweets, we identified 6.5 million malicious tweets, which accounted for approximately 1% of all tweets.

4.2.3 Features

After labelling the spam tweets, we further extracted features from them. Since Twitter's Public Streaming API only returned random public tweets and they were not socially connected, we were not able to build a social graph from the data. As a



(a) Account Age (b) No. of User Favourites (c) No. of Lists (d) No. of Tweets

(e) No. of Followings (f) No. of Followers (g) No. of Retweets per Tweet (h) No. of Hashtags per Tweet

(i) No. of User Mentions per Tweet (j) No. of URLs per Tweet (k) No. of Characters per Tweet (l) No. of Digits per Tweet

Figure 4.3: Cumulative Distribution Functions of Features

result, it is not possible for us to extract social graph based features such as Local Clustering Coefficient, Betweenness Centrality [135] and distance [104]. Such expensive features are not suitable to be used in real-time detection, despite that they have more discriminative power in separating spam and non-spam tweets. Moreover, we

Table 4.1: 12 Extracted Lightweight Features

Feature Name	Description
account_age	The age (days) of an account since its creation until the time of sending the most recent tweet
no_follower	The number of followers of this twitter user
no_following	The number of followings/friends of this twitter user
no_userfavourites	The number of favourites this twitter user received
no_lists	The number of lists this twitter user added
no_tweets	The number of tweets this twitter user sent
no_retweets	The number of retweets this tweet
no_hashtag	The number of hashtags included in this tweet
no_usermention	The number of user mentions included in this tweet
no_urls	The number of URLs included in this tweet
no_char	The number of characters in this tweet
no_digits	The number of digits in this tweet

are specially focusing on detecting the streaming spam tweets; features which can be straightforwardly computed from the tweet itself are preferred. We have totally extracted 12 features from our data set as listed in TABLE 5.1.

According to the object where the features were extracted, the 12 features can be divided into two categories, user-based features and tweet-based features. *User-based features* were extracted from the JSON object "user", such as account_age, which can be calculated by using the collection date minus the account created data. Other user-based features, like no_of followers, no_of followings, no_userfavourites, no_lists, and no_tweets, can be directly parsed from the JSON structure. *Tweet-based features*

includes no_retweets, no_hashtags, no_usermentions, no_urls, no_chars, and no_digits. While no_chars and no_digits needs a little computing, *i.e.* counting them from the tweet text, others can also be straightforwardly extracted.

4.2.4 Feature Statistics

To look into the characteristics of these features, we plotted the Cumulative Distribution Function (CDF) of them, as shown in Fig. 4.3.

We can see from Fig. 4.3(c) that spammers are involved in more lists than normal users, so as to be exposed more to the public. Naturally, in order to spread more spam tweets, spammers send more tweets compared to non-spammers, as shown in Fig. 4.3(d). In terms of "number of followings", Fig. 4.3(e) shows that, spammers do like to follow more users than non-spammers. The aim is also to attract more attentions from victims to click their spam links.

As Fig. 4.3(h) shows, non-spammers use less hashtags than spammers. There are about 80% non-spam tweets do not have hashtags embedded in their sent tweets, while the ratio in spam tweets is only 60%. When it comes to the feature "Number of Characters Per Tweet", there is not much difference between spam tweets and non-spam tweets. The reason could be that spammers begin to imitate the posting behaviour of normal users. Fig. 4.3(i) shows that spammers tend to use less digits than non-spammers. Due to the limit of pages, we only describe six features'

Table 4.2: Sampled Datasets

Dataset	Sampling Method	NO. of Spam Tweets	NO. of Non-spam Tweets
I	Continuous	5000	5000
II	Continuous	5000	95000
III	Non-continuous	5000	5000
IV	Non-continuous	5000	95000

characteristics here. In general, the analysis of these features has showed us their discriminative power to detect Twitter spam.

4.3 Fundamental Evaluation of ML based Streaming Spam Tweets Detection

In this section, we evaluate the spam detection performance on our dataset by using six machine learning algorithms, *Random Forest, C4.5 Decision Tree, Bayes Network, Naive Bayes, k Nearest Neighbour, and Support Vector Machine*. We also sampled several different data sets to conduct the experiments. The datasets are listed in TABLE 4.2.

In TABLE 4.2, we can see that the spam to non-spam ratio is 1:1 in Dataset I and III while the ratio is 1:19 in Dataset II and IV. In previous works, most of the datasets are nearly evenly distributed; the spam to non-spam ration is nearly 1:1. However,

Twitter has around 5% spam tweets of all existing tweets in the real world [48]. The evenly distributed dataset cannot represent the Twitter sphere. Consequently, we sampled Dataset II and IV which has a spam ration of 1:19 to simulate the real world scenario.

All of the four datasets are randomly selected from the whole 600 million tweets. However, the datasets can be divided into two groups based on the sampling method: Dataset I and II are both randomly selected from the whole dataset, but the tweets were sent in a certain continuous time frame. On the other hand, the tweets in Dataset III and IV were not sent continuously. Instead, those tweets were totally independent from each other.

4.3.1 The Process of ML based Twitter spam detection

This subsection describes the process of Twitter spam detection by using machine learning algorithms. Fig. 5.3 illustrates the steps involved in building a supervised classifier and detecting Twitter spam. Before classification, a classifier which contains the knowledge structure should be trained with the pre-labelled tweets. After the Classification Model gains the knowledge structure of the training data, it can be used to predict a new incoming tweet. The whole process consists of two steps: learning and classifying. Firstly, features of tweets will be extracted and formatted as a vector $\vec{F} = \{f_1, f_2, \ldots, f_n\}$. The class labels (spam or non-spam) could be get via some other

approaches (like manual inspection). Features and class label will be combined as one instance for training. One training tweet can then be represented by a pair containing one feature vector which represents a tweet, and the expected result $(\vec{F}, label)$, and the training set is the vector $\vec{TS} = \{(\vec{F_1}, label_1), (\vec{F_2}, label_2), (\vec{F_n}, label_n)\}$. The training set is the input of machine learning algorithm, the classification model will be built after training process. In the classifying process, timely captured tweets $T = \{F_1, F_2, \ldots, F_n\}$ will be labelled by the trained classification model.

4.3.2 Performance Metrics

In order to evaluate the performance of spam detection approaches, some metrics are imported from Information Retrieval are widely used by the researchers.

4.3.2.1 Positives and Negatives

Suppose there is a tweet t and the spam class S. The output of the classifier is whether t belongs to S or not. A common way to evaluate the classifier's performance is to use **True Positives, False Positives, False Positives, False Negatives**. These metrics are defined as following:

- True Positives (TP), tweets of class S correctly classified as belonging to class S.
- False Positives (FP), tweets not belonging to class S incorrectly classified as

Table 4.3: Evaluation Metrics

		Predicted	
		Sapm	Non-sapm
True	Spam	TP	FN
	Non-spam	FP	TN

belonging to class S.

- True Negatives (TN), tweets not belonging to class S correctly classified as not belonging to class S.

- False Negatives (FN), tweets of class S incorrectly classified as not belonging to class S.

The relations of TP, FP, TN and FN in social spam detection are shown in Table 4.3

In order to measure the ability to detect spam, we also import True Positive Rate (TPR), and False Positive Rate (FPR).

- TPR is defined as the ratio of those spam tweets correctly classified as belonging to class *spam* to the total number of tweets in class *spam*, it can be calculated by

$$TPR = \frac{TP}{TP + FN} \tag{4.3.1}$$

- FPR is defined as the ratio of those non-spam tweets incorrectly classified as

belonging to spam class S to the total number of non-spam tweets

$$FPR = \frac{FP}{FP + FN} \tag{4.3.2}$$

4.3.2.2 Precision, Recall and F-measure

Literature also uses Precision, Recall, and F-measure to evaluate per-class performance.

- Precision is defined as the ratio of those tweets that truly belong class S to those identified as class S, it can be calculated by

$$Precision = \frac{TP}{TP + FP} \tag{4.3.3}$$

- Recall (which is also known as Detection Rate in the detection scenario) is defined as the ratio of those tweets correctly classified as belonging to class S to the total number of users in class S, it can be calculated by

$$Recall = \frac{TP}{TP + FN} \tag{4.3.4}$$

- F-measure is a combination of precision and recall, it is a widely adopt metric to evaluate per-class performance, it can be calculated by

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.3.5}$$

Table 4.4: Performance Evaluation on Dataset I and II

Unit: %	Dataset I			Dataset II		
Classifier	TPR	FPR	F-measure	TPR	FPR	F-measure
RandomForest	92.9	5.6	93.6	92.9	7.1	56.6
C4.5	92.4	8.4	92	92.4	10.9	46.2
BayesNetwork	75.3	8.7	81.9	75.3	9.8	41.6
Naive Bayes	97.3	77.1	70.9	97.3	78.8	11.5
Knn	91.9	11.1	90.5	91.9	15.9	37.3
SVM	79.1	18.9	79.9	79.1	19.5	28.8

4.3.3 The Impact of Spam to Non-spam Ratio

In this section, We evaluate the impact of spam to non-spam ratio of the above-mentioned machine learning algorithms on Dataset I and II. Each classifier in this set of experiments was trained with a dataset of 1000 spam tweets and 1000 non-spam tweets. Then these trained classifiers were used to detect spam in the four sampled datasets. As in [135], we also used True Positive Rate (TPR), False Positive Rate (FPR) and F-measure to evaluate the performance of these classifiers.

As seen in TABLE 4.4, most of the classifiers can achieve more than 90% TPR, expect Bayes Network and SVM, on both datasets. These classifiers can also reach satisfactory F-measure on Dataset I. However, the F-measures decrease dramatically when evaluating on Dataset II, *i.e.* when the spam to non-spam ration is 1:19.

To figure out why F-measure drops on Dataset II, TABLE 4.5 outputs the confusion matrix of Random Forest when evaluated on both datasets. Since the classifiers were trained by the same dataset, we can see that, there was no impact on the True

Table 4.5: Confusion Matrix of Random Forest on Both Datasets

classified as –>	spam	non-spam		spam	non-spam
spam	4645	355		4645	355
non-spam	282	4718		6766	88234
	Dataset I			Dataset II	

Positives and False Negatives of spam class when the spam to non-spam ratio was changed, so Recall, which is define as the ratio of the number of tweets classified correctly as spam to the total number of real spam tweets, stayed the same. However, when more non-spam tweets were involved in the test, the number of False Positives increased exponentially. Thus, the precision, which is define as the ratio of the number of tweets classified correctly as spam to the total number of predicted spam tweets, decreased. As a result, F-measure, which is combination of precision and recall, decreased dramatically due the decrease of precision. Generally, we find that the F-measure of machine learning based classifiers is quite low as there are much more non-spam tweets than spam tweets.

4.3.4 The Impact of Feature Discretisation

In this subsection, the impact of feature discretisation of selected classifiers, such as Naive Bayes, kNN, and SVM when on discretised and non-discretised Dataset I and II, is evaluated.

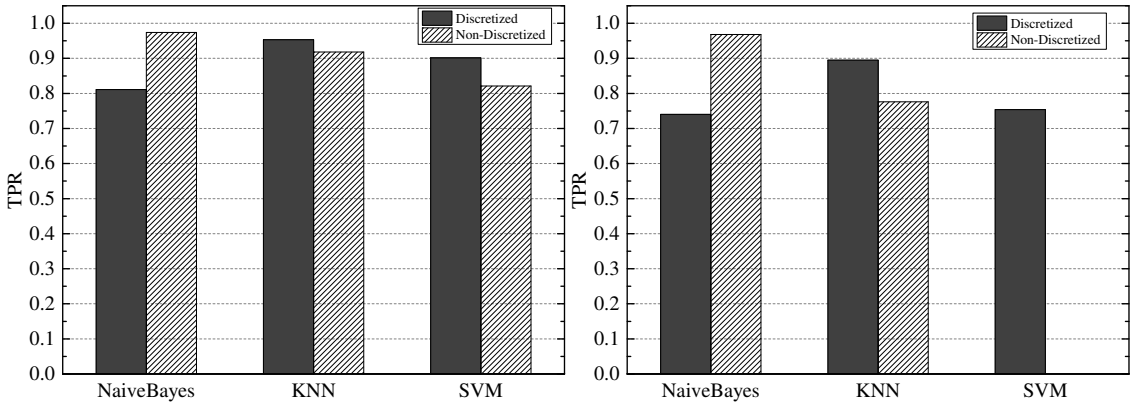
Fig. 4.4 - 4.6 shows the True Positive Rate, False Positive Rate, F-measure and

classification speed of spam detection on Dataset I and II. We can see that, the False Positive Rate of Naive Bayes decreases dramatically after discretisation, from 80% to 20% on Dataset I. Similar on Dataset II, the FPR declined from 45% to less than 5%. However, the performance of Naive Bayes also decreases in terms of True Positive Rate. The TPR of Naive Bayes drops from 94.5% to 88% and 74.5% to 58%, respectively on Dataset I and II. When it comes to F-measure, the performance of Naive Bayes increases around 3% and over 20% on Dataset I and II. Overall, feature discretisation has positive impact for Naive Bayes, especially when on Dataset II. Similarly, feature discretisation can help to improve performance for kNN and SVM on both datasets. For example, the F-measure has been improved 5% for kNN and 10% for SVM on Dataset I. We also notice that, although SVM can achieve 75% F-measure on Dataset I, it becomes useless on Dataset II with less than 5% F-measure without feature discretisation. However, SVM can achieve over 80% F-measure after discretising features. In general, feature discretisation can improve performance of classifiers for Twitter spam detection.

4.3.5 The Impact of Increasing Training Data

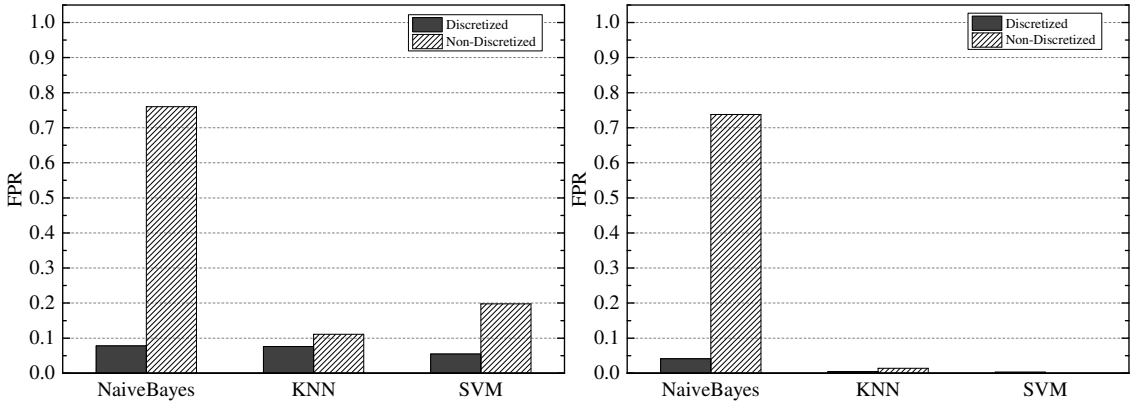
We evaluate the performance of all six classifiers with training data varying from 100 samples to 1000 samples in this subsection.

Fig. 4.7 shows the spam detection performance with increasing training samples



(a) TPR on Dataset I (b) TPR on Dataset II

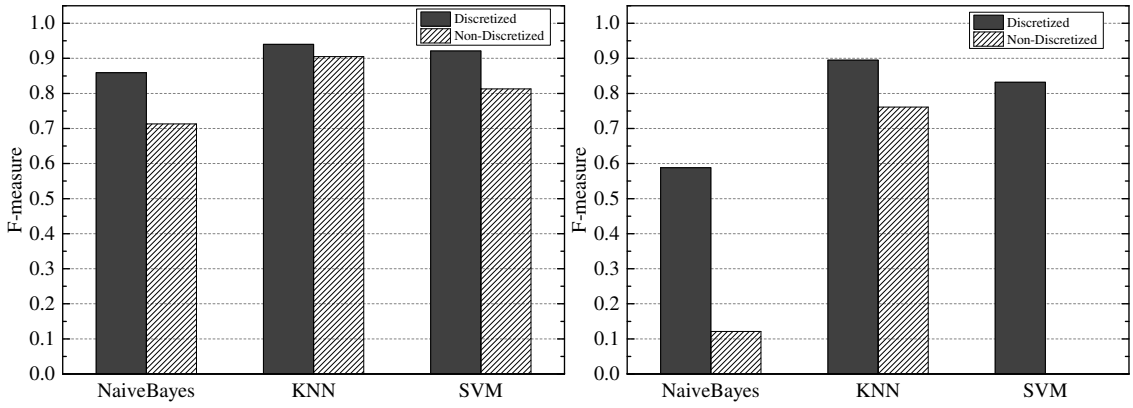
Figure 4.4: True Positive Rate on Spam



(a) FPR on Dataset I (b) FPR on Dataset II

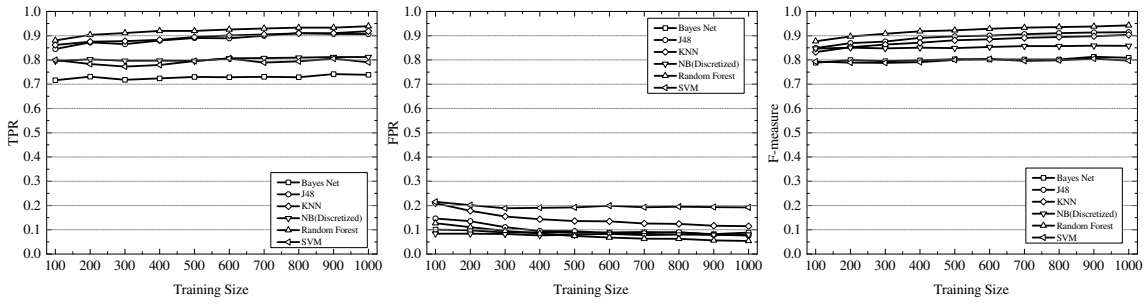
Figure 4.5: False Negative Rate on Spam

on Dataset I. In Fig. 4.7a, one can find that Random Forest outperforms all the other classifiers with TP rate ranging from 78% to 85%, followed by kNN. However, Navie Bayes with discretisation has the lowest FP rate, while SVM has the highest FP rate. When it comes to F-measure, Random Forest still ranks as number one among all



(a) F-measure on Dataset I (b) F-measure on Dataset II

Figure 4.6: F-measure on Spam

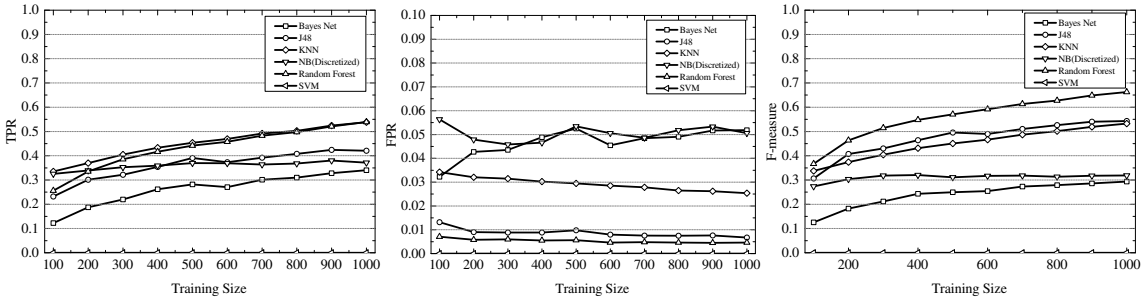


(a) TP Rate (b) FP Rate (c) F-measure

Figure 4.7: Spam detection with increasing training size on Dataset I

classifiers, with a range from 70% to 75%.

Fig. 4.8 reports the spam detection performance with increasing training samples on Dataset II. Unsurprisingly, Random Forest also performs the best in terms of all three metrics with more than 40% TP rate and less then 1% FP rate. In addition, the increment of F-measure from 100 training samples to 1000 training samples is more

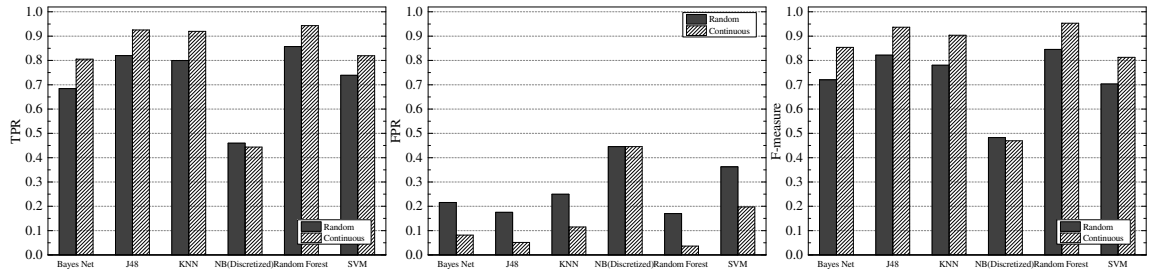


(a) TP Rate (b) FP Rate (c) F-measure

Figure 4.8: Spam detection with increasing training size on Dataset II

than 10% for all classifiers except for Naive Bayes. Especially for Random Forest, the F-measure increases from 36% to 65%, with an increment of over 30%.

One would expect that the performance of the classifiers will increase with additional training data [50]. However, we find that the performance is relatively stable even with more training data. In Fig. 4.7c, we can find that F-measure of these classifiers can reach as high as 80%. However, it cannot be improved further by simply increasing the training data. Specifically, the F-measure rises slightly (less than 3%) for Random Forest, C4.5 Decision Tree, and kNN, after the training samples number of 500. There is no growth for Bayes Network and SVM in terms of F-measure. Particularly, F-measure of Navie Bayes even drops with more training samples. This phenomenon also happens with Dataset II. For instance, the F-measure of Naive Bayes stays around 30% despite the growth of training samples. We conclude that there is little benefit by simply increasing the training data when the training size



(a) TP Rate (b) FP Rate (c) F-measure

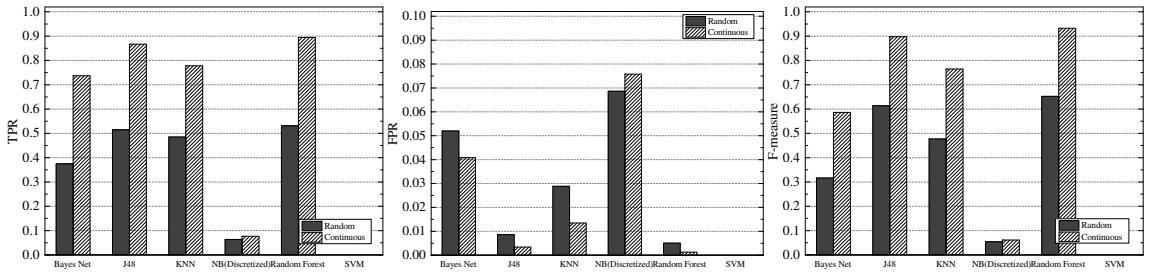
Figure 4.9: Spam detection on Dataset I VS Dataset III

has reached a certain size. More pre-processes, such as developing more discriminant features or cleaning training data [144], should be done to further improve the performance.

4.3.6 The Impact of Different Sampling Method

During our study, we also notice that classifiers' performance is better on the dataset where the tweets are sampled from a continuous period of time than that where the tweets are randomly selected. To further study this, Dataset III and IV are sampled. The samples in Dataset I and II are randomly selected, while those in Dataset III and IV are continuous. We also perform 10-fold cross-validation on both datasets. The results are shown in Fig. 4.9 and 4.10.

The results in Fig. 4.9a indicate that the TP rates of all classifiers on Dataset III arise around 10% compared to the performance on Dataset I, expect Naive Bayes. For example, the TP rates of C4.5 Decision Tree and kNN are 12% higher on Dataset

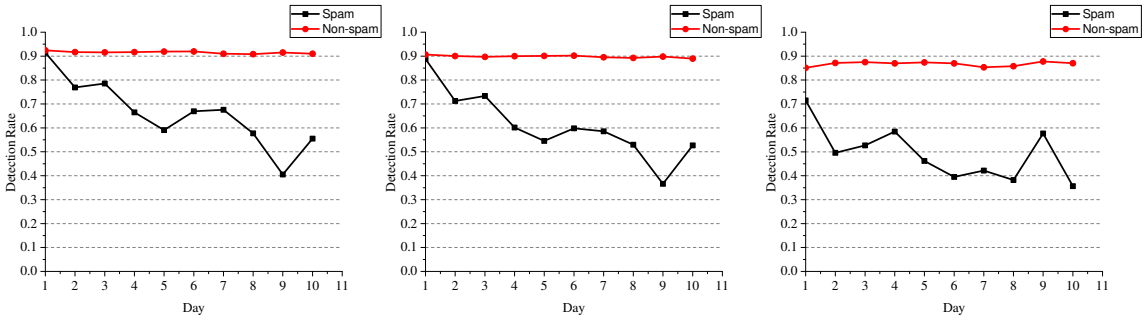


(a) TP Rate (b) FP Rate (c) F-measure

Figure 4.10: Spam detection on Dataset II VS Dataset IV

III than those on Dataset I. In addition, most of these classifiers can reach 80% TP rate; some of them, such as C4.5 Decision Tree, kNN and Random Forest can even have over 90% TP rates when evaluated on Dataset III. Similarly, the FP rates on Dataset III drops significantly, especially for SVM, it drops from nearly 40% to less than 20%, with an decrease of 20%. Most of the classifiers have a FP rate of less than 10%. In terms of F-measure, all classifiers evaluated on Dataset III except Navie Bayes outperform those on Dataset I. Furthermore, several classifiers can have more than 90% F-measure, which is very effective in detection Twitter spam.

Fig. 4.10 shows the TP rates, FP rates and F-measures of all the classifiers evaluated on Dataset II and IV. The difference of TP rates on Dataset II and 4 is significantly huge, which is around 30% to 40%. When it comes to the metric of F-measure, the same difference exists. For instance, the F-measure of Random Forest evaluated on Dataset IV can reach as high as 95%, which is 30% higher than it on Dataset II. In this set of experiments, we find that Naive Bayes and SVM work badly



(a) Random Forest (b) C4.5 Decision Tree (c) Bayes Network

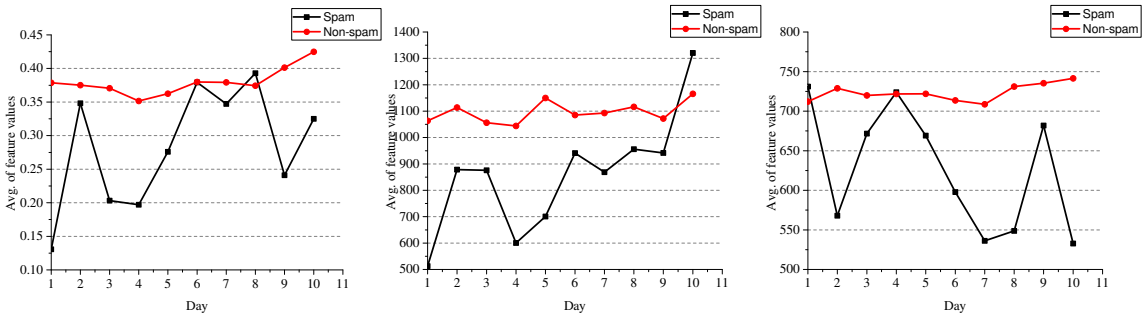
Figure 4.11: Trend of Detection Rate

when on the datasets with 1:19 spam to non-spam ratio. Naive Bayes can only detect less than 10% spam tweets, while SVM miss all the spam tweets. We will put the problem why Naive Bayes and SVM cannot work well on imbalanced datasets as a future work.

In this subsection, we evaluate the performance of different classifiers on two kinds of datasets (randomly sampled and continuously sampled), and find that classifiers have much better performance in detection spam tweets on the continuous datasets. We will further investigate this in Section 4.3.7.

4.3.7 The Investigation of Time-Related Data

As discussed in the above section, the performance varies when in differently sampled datasets. We believe that "time" plays an important role in this difference. In this section, a serious of experiments are conducted from various kinds of views to



(a) Account Age (b) No. of Followings (c) No. of User Mentions per Tweet

Figure 4.12: Changes of Average Values of Features

Table 4.6: KL Divergence of Spam and Nonspam Tweets of two Consecutive Days

	D1 VS D2		D2 VS D3		D3 VS D4		D4 VS D5		D5 VS D6		D6 VS D7		D7 VS D8		D8 VS D9		D9 VS D10	
f1	0.36	0.04	0.34	0.03	0.44	0.04	0.24	0.03	0.26	0.03	0.27	0.03	0.29	0.05	0.26	0.03	0.34	0.04
f2	0.24	0.1	0.22	0.1	.26	0.1	0.19	0.1	0.21	0.1	0.21	0.1	0.17	0.1	0.38	0.1	0.35	0.1
f3	0.28	0.07	0.22	0.07	0.32	0.07	0.15	0.07	0.22	0.07	0.2	0.07	0.2	0.08	0.26	0.08	0.23	0.08
f4	0.16	0.07	0.13	0.07	0.14	0.08	0.14	0.07	0.17	0.07	0.19	0.07	0.13	0.07	0.27	0.08	0.19	0.08
f5	0.02	0.01	0.02	0.01	0.03	0.01	0.02	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.05	0.01	0.05	0.01
f6	0.98	0.35	0.52	0.35	0.63	0.35	0.36	0.35	0.45	0.34	0.4	0.34	0.45	0.35	0.5	0.35	0.52	0.36
f7	0.1	0.04	0.08	0.03	0.04	0.04	0.04	0.04	0.05	0.03	0.07	0.04	0.06	0.04	0.1	0.04	0.08	0.04
f8	0.19	0	0	0	0.04	0	0.03	0	0.02	0	0.03	0	0.01	0	0.04	0	0.02	0
f9	0.09	0	0.03	0	0.01	0	0.02	0	0.01	0	0.01	0	0	0	0.04	0	0.01	0
f10	0	0	0.03	0	0.03	0	0.01	0	0.1	0	0	0	0.01	0	0.32	0	0.27	0
f11	0.26	0.01	0.06	0.01	0.06	0.01	0.11	0.01	0.1	0	0.09	0	0.26	0.01	0.28	0.01	0.2	0.02
f12	0.04	0	0	0	0.02	0	0.03	0.01	0.03	0	0.04	0	0.04	0	0.46	0	0.46	0

investigate the "time-related" issue in detecting streaming spam. In order to perform such evaluation, we sampled a new dataset which is constituted by 10 consecutive days' tweets, while each day contains 100k spam tweets and 100k non-spam tweets.

4.3.7.1 From the view of Detection Rate

We perform a serious of experiments in this section to show how Detection Rates of spam and non-spam changes while testing on different days. As in [135], we use Detection Rate to show the classifier's performance.

During our experiments, Day 1 data is divided into two parts, half for training pool where training data can be extracted from, and another half for testing purpose. We create a classifier by using a supervised classification algorithm, and train it with 10k spam and 10k non-spam tweets which are randomly sampled from the training pool of Day 1. Then the classifier is used to classify the testing data in Day1, as well as the testing samples in Day 2 to Day 10. In order to make the results more fair, we only use half of the samples for testing in Day 2 to Day 10.

Fig. 5.5 shows the Detection Rate of both spam and non-spam tweets on three classifiers, Random Forest, C4.5 Decision Tree and Bayes Network. We can see that, the DR of non-spam is very stable, it keeps above 90% for Random Forest and C4.5 Decision Tree, and near 90% for Bayes Network, despite the change of testing data. However, when it comes to spam tweets, the DR fluctuates dramatically, and the overall trend is decreasing. The DRs for Random Forest and C4.5 Decision Tree are 90% in the first day, but they could decrease to less then 40% in the 9th day. This phenomenon also applies with Bayes Network, the DR decreases from 70% on 1^{st} day to less than 50% for most of the other testing days. From this, we can see that the

Detection Rate is decreasing when training data and testing data are from different period of time.

4.3.7.2 From the view of average values of features

To further investigate the reason why performance decreases when training and testing data are from different days, we calculate the average value of each feature in all tweets of each day, and find that the average value of features from spam tweets varies while that is more stable in terms of non-spam tweets.

Fig. 5.1 shows the changing trend of average value of three features for two classes in 10 days. In general, the vary of average value of feature from spam tweets is greater than that of non-spam tweets. Fig. 5.1a shows that, the average value of Account Age for spam tweets ranges from 530 to 730, and the variation is dramatic. However, it deviates from 710 to 740 for non-spam tweets. We infer that spammers are creating a large number of new accounts to send spam once their old account are blocked, which leads the decrease of average age for spammers. Naturally, spammers tend to keep following new friends as they want to be exposed to public more frequently, whereas for non-spammers, their number of followings are not changing too much once they have built their friend circle, as we can see from Fig. 5.1b. Due to the page limit, we excluded the figures of other features. However, most of the other features have the same trend as expected: the average value of one feature varies for spam tweets, while it is stable for non-spam tweets. Consequently, the detection of one classifier

become inaccurate, as the statistical features of the testing data varies.

4.3.7.3 From the view of KL Divergence of two days' feature distribution

Previously, we simply compared the some representative statistics, such as the mean values of features to show the reason why classifiers' performance decreases while training and testing are done in different days. To further illustrate the changing of the statistical features in a dataset, a natural approach is to model the distribution of the data [37]. One of the most common measure to compute the distance of distributions is Kullback-Leibler (KL) Divergence [37, 102]. The suitability of KL Divergence to be used in measuring distributions can be found in [37].

We compute the KL Divergence of each feature of spam and nonspam tweets in consecutive two days, which is listed in TABLE 5.2. The shadowed ones are the KL Divergence of features of nonspam tweets, while the other are the KL Divergence of features of spam tweets. KL Divergence indicates the dissimilarity of two distributions. The larger the value is, the more dissimilar the two distributions are. As shown in Table 5.2, the KL Divergence of spam tweets in two consecutive days are much larger than that of the nonspam tweets for more than half the features. Taking f1 for example, the KL Divergence of spam between Day 1 and Day 2 is 0.36, while it is only 0.04 for non-spam, which indicates that the distribution of f1 of spam in Day 1 is much different to it in D2, compared with nonspam tweets' distribution. From these KL Divergence values, we can see that the distribution of spam tweets'

features is changing unpredictably from day to day. Nevertheless, the distribution of training data is unchanged. So, the knowledge structure which learns from the unchanged training data is not updated while being used to classify new incoming tweets. That's why the performance of classifiers becomes inaccurate.

4.4 Summary

In this chapter, we provide a fundamental evaluation of ML algorithms on the detection of streaming spam tweets. In order to perform this evaluation, we firstly collected a large number of 600 million public tweets. Then we applied Trend Micro's Web Reputation System to label as many as 6.5 million spam tweets. We also extracted 12 light-weight features which are able to differentiate spam tweets and non-spam tweets from this labelled dataset. Furthermore, we used CDF figures to illustrate the characteristics of extracted features. We leveraged these features to machine learning based spam classification later in our experiments. To investigate the ability of spam detection of different classifiers, we sampled four different datasets to simulate various scenarios. In our evaluation, we found that classifiers' ability to detect Twitter spam reduced when in a near real-world scenario since the imbalanced data brings bias. We also identified that Feature discretisation was an important pre-process to ML based spam detection. Secondly, increasing training data only cannot bring more benefits to detect Twitter spam after a certain number of training samples. We should try to

bring more discriminative features or better model to further improve spam detection rate. Thirdly, classifiers can detect more spam tweets when the tweets were sampled continuously rather than randomly selected tweets.

From the third point, we thoroughly analysed the reason why classifiers' performances reduced when training and testing data were in different days from three point of views. We conclude that the performance decreases due to the fact that the distribution of features changes of later days' dataset, while the distribution of training dataset stays the same. This problem will exist in streaming spam tweets detection, as the new tweets are coming in the forms of streams, but the training data set is not updated. We will work on this issue in the future. This problem is referred as "Spam Drift", which will be further studied and addressed in Chapter 6.

Chapter 5

Addressing "Spam Drift": Lfun approach

5.1 Introduction

Research shows that blacklist fails to protect victims from new spam due to its time lag [48]; more than 90% victims may visit a new spam link before it is blocked by blacklists [114]. In order to address the limitation of blacklists, researchers have proposed some machine learning based schemes which can make use of spammers' or spam tweets' statistical features to detect spam without checking the URLs [43, 136].

Machine Learning (ML) based detection schemes involve several steps. First, statistical features, which can differentiate spam from non-spam, are extracted from tweets or Twitter users (such as account age, number of followers or friends and number of characters in a tweet). Then a small set of samples are labelled with class, *i.e.* spam or non-spam, as training data. After that, machine learning based classifiers are trained by the labelled samples, and finally the trained classifiers can be

used to detect spam. A number of ML based detection schemes have been proposed by researchers [8, 107, 135, 143].

However, the observation in our collected data set shows that the characteristics of spam tweets are varying over time. We refer to this issue as "Twitter Spam Drift". As previous ML based classifiers are not updated with the "changed" spam tweets, the performance of such classifiers are dramatically influenced by "Spam Drift" when detecting new coming spam tweets. Why do spam tweets drift over time? It is because that spammers are struggling with security companies and researchers. While researchers are working to detect spam, spammers are also trying to avoid being detected. This leads spammers to evade current detection features through posting more tweets or creating spam with the similar semantic meaning but using different text [104, 135].

In this chapter, we firstly illustrate the "Twitter spam drift" problem through analysing the statistical properties of Twitter spam in our collected dataset and then its impact on detection performance of several classifiers. By observing that there are "changed" spam samples in the coming tweets, we propose a novel **Lfun** (**L**earning **f**rom **un**labelled tweets) approach, which updates classifiers with the spam samples from the unlabelled incoming tweets. In summary, our contributions are listed below:

- We collect and label a real-world dataset, which contains 10 consecutive days' tweets with 100k spam tweets and 100k non-spam tweets in each day (2 million

tweets in total). This dataset is available for researchers to study Twitter spam [1].

- We investigate the "Twitter Spam Drift" problem from both data analysis and experimental evaluation aspects. To the best of our knowledge, we are the first to study this problem in Twitter spam detection.

- We propose a novel Lfun approach which learns from unlabelled tweets to deal with "Twitter Spam Drift". Through our evaluations, we show that our proposed Lfun can effectively detect Twitter spam by reducing the impact of "Spam Drift" issue.

The rest of this chapter is organized as follows. In Section 5.2, the collection and labelling of the data used in our work is introduced. Meanwhile, the "Spam Drift" problem is illustrated and justified. Then we introduce our Lfun approach in Section 5.3, and analyse the performance benefit of our approach. Section 5.4 evaluates our Lfun approach and compares it with four traditional machine learning algorithms. Finally, Section 5.6 concludes this work and introduces our future work.

Table 5.1: Extracted Features

Feature Name	Description
account_age	The age (days) of an account since its creation until the time of sending the most recent tweet
no_follower	The number of followers of this twitter user
no_following	The number of followings/friends of this twitter user
no_userfavourites	The number of favourites this twitter user received
no_lists	The number of lists this twitter user added
no_tweets	The number of tweets this twitter user sent
no_retweets	The number of retweets this tweet
no_hashtag	The number of hashtags included in this tweet
no_usermention	The number of user mentions included in this tweet
no_urls	The number of URLs included in this tweet
no_char	The number of characters in this tweet
no_digits	The number of digits in this tweet

5.2 Problem of Twitter Spam Drift

5.2.1 10-day groundtruth

A labelled dataset is important for classification tasks, such as Twitter spam detection. In this work, we used Twitter's Streaming API to collect tweets with URLs in a period of 10 consecutive days. While it is possible to send spam without embedding URLs on Twitter, the majority of spam contains URLs [33,39,45]. We have inspected hundreds of spam tweets by hand and only find a few tweets without URLs which

[1]You can download our dataset from http://nsclab.org/nsclab/resources/

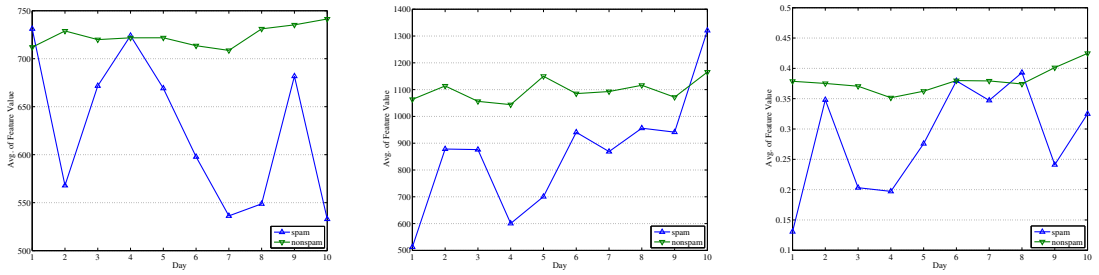
could be considered as spam. In addition, spammers mainly use embedded URLs to make it more convenient to direct victims to external sites to achieve their goals, such as phishing, scams, and malware downloading [143]. Therefore, we only focus on spam tweets with URLs.

Currently, researchers use two ways to build ground-truth, manual inspection and blacklists filtering. While manual inspection can label a small number of training data, it is very time- and resource-consuming. A large group of people are needed to check tens of thousands of tweets. Although HIT (human intelligence task) websites can help label the tweets, it is also costly and sometimes the results are doubtful [23]. Others apply existing blacklisting services, such as Google SafeBrowsing and URIBL [44] to label spam tweets. Nevertheless, these services' API limits make it impossible to label a large amount of tweets.

We apply Trend Micro's Web Reputation Technology to identify which tweets are deemed spam [27]. Trend Micro's WRT system maintains a large dataset of URL reputation records, which are derived from their customers' opt-in URL filtering records. WRT system is dedicated to collecting the latest and the most popular URLs, to analysing them, and then to providing Trend Micro customers with real-time protection while they are surfing the web. Hence, through checking URLs with the WRT system, we are able to identify whether a URL is malicious or not. We define those which contain malicious URLs as Twitter spam. WRT system is reliable

as the protection rate of it is 100%, as stated in AV Comparatives' testing report. In addition, we have done a manual inspection of hundreds of tweets to confirm the reliability of WRT. In our collected data, we labelled one million spam tweets and one million non-spam tweets for 10 days, with 100k spam tweets and 100k non-spam tweets for each day.

Feature extraction is a key component in machine learning based classification tasks [141]. Some studies [8, 107, 121] have applied a few features which make use of historical information of a user, such as tweets that the user sent in a period of time. While these features may be more discriminative, it is not possible to collect them due to the restrictions of Twitter's API. Other researchers [104, 135] applied some social graph based features, which are hard to be evaded. Nevertheless, It is significantly expensive to collect those features, as they cannot be calculated until the social graph is formed. Thus, those expensive features are not suitable for real-time detection, despite that they have more discriminative power in separating spammers and legitimate users. The longer time a spam tweet exists, the more chance it can be exposure to victims. Thus, it is very important to detect spam tweets as early as possible. To reduce the loss caused by spam, real-time detection is in demand. Consequently, we only focus on extracting light-weight features which can be used for timely detection as in [49]. These features can be straightforwardly extracted from the collected tweets' JSON data structure [90] with little computation. We have



(a) Account Age (b) No. of Followings (c) No. of User Mentions per Tweet

Figure 5.1: Changes of Average Values of Features

totally extracted 12 features from our dataset as listed in TABLE 5.1.

5.2.2 Problem Statement

In the real world, the statistical features of spam tweets are changing in unpredicted ways over time. As a result, machine learning based detection system becomes inaccurate. The issue is referred to as "Spam Drift" problem in our previous paper [28]. Here, we present an investigation of "Spam Drift" problem from the aspect of the change of mean value of each feature from day to day.

Fig. 5.1 shows the changing trend of average value of each feature for two classes in 10 days. In general, the variation of average value of feature from spam tweets is greater than that of non-spam tweets. Fig. 5.1a shows that, the average value of Account Age for spam tweets ranges from 530 to 730, and the variation is dramatic. However, it deviates from 710 to 740 for non-spam tweets, which is relatively stable.

It is due to the fact that spammers are creating a large number of new accounts to send spam once their old account are blocked. For instance, we have 3 spammers with account age of 2 days, 6 days, 10 days in the first day, the average value of Account Age is (2 +6 +10) / 3 = 6 days. In the second day, if the spammer whose account age is 2 days is detected and removed, the average value of Account Age is (6+10) / 2 = 8 days, which increases. In addition, spammers may also generate new accounts with 0 day Account Age to spread spam after some of their accounts are block, which can lead the decrease of average value of Account Age. That is why the average value of Account Age is fluctuating. Naturally, spammers tend to keep following new friends as they want to be exposed to public more frequently, whereas for non-spammers, their number of followings are not changing too much once they have built their friend circle, as we can see from Fig. 5.1c. As expected, most of the other features have the same trend: the average value of one feature varies for spam tweets, while it is stable for non-spam tweets.

To sum up, the characteristics of spam tweets is varying from day to day, while that of non-spam tweets is not changing much, as we see from Fig. 5.1. "Spam Drift" is a crucial issue in Twitter spam detection, which is in great need to be solved.

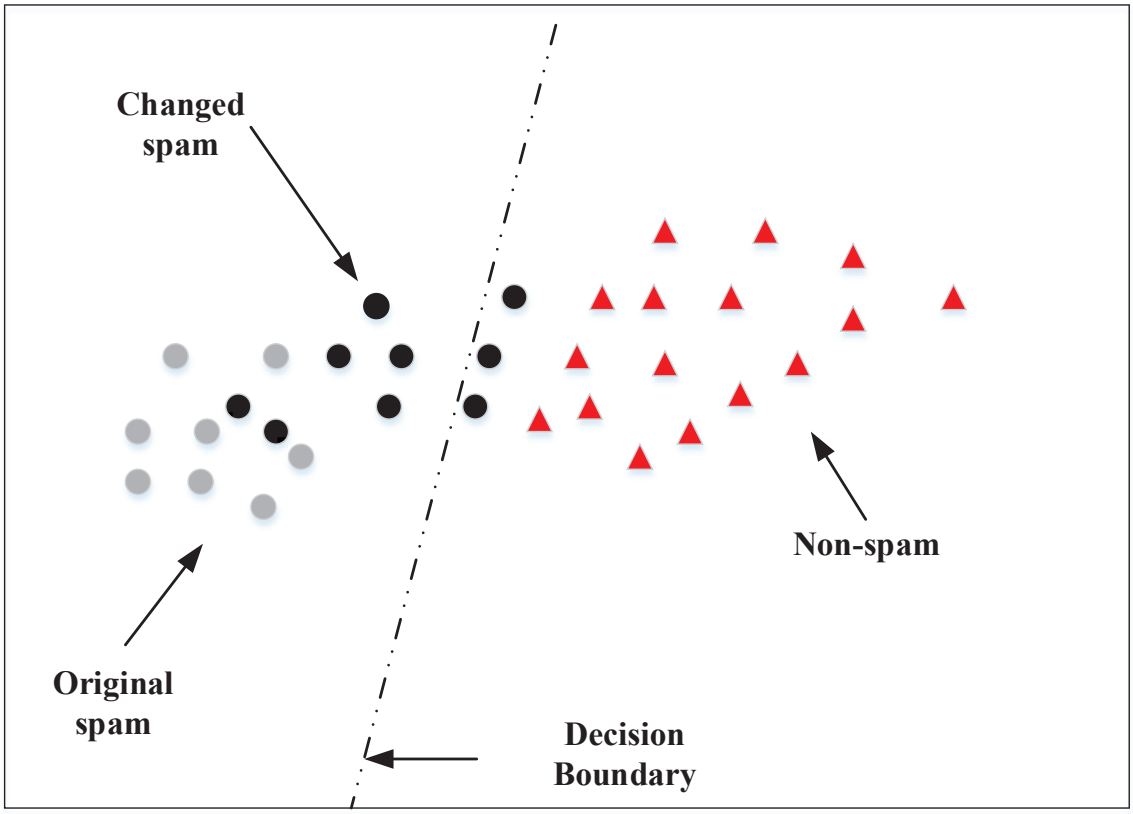


Figure 5.2: Illustration of "Spam Drift"

Table 5.2: KL Divergence of Spam and Nonspam Tweets of two Consecutive Days

	D1 VS D2		D2 VS D3		D3 VS D4		D4 VS D5		D5 VS D6		D6 VS D7		D7 VS D8		D8 VS D9		D9 VS D10	
f1	0.36	0.04	0.34	0.03	0.44	0.04	0.24	0.03	0.26	0.03	0.27	0.03	0.29	0.05	0.26	0.03	0.34	0.04
f2	0.24	0.1	0.22	0.1	.26	0.1	0.19	0.1	0.21	0.1	0.21	0.1	0.17	0.1	0.38	0.1	0.35	0.1
f3	0.28	0.07	0.22	0.07	0.32	0.07	0.15	0.07	0.22	0.07	0.2	0.07	0.2	0.08	0.26	0.08	0.23	0.08
f4	0.16	0.07	0.13	0.07	0.14	0.08	0.14	0.07	0.17	0.07	0.19	0.07	0.13	0.07	0.27	0.08	0.19	0.08
f5	0.02	0.01	0.02	0.01	0.03	0.01	0.02	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.05	0.01	0.05	0.01
f6	0.98	0.35	0.52	0.35	0.63	0.35	0.36	0.35	0.45	0.34	0.4	0.34	0.45	0.35	0.5	0.35	0.52	0.36
f7	0.1	0.04	0.08	0.03	0.04	0.04	0.04	0.04	0.05	0.03	0.07	0.04	0.06	0.04	0.1	0.04	0.08	0.04
f8	0.19	0	0	0	0.04	0	0.03	0	0.02	0	0.03	0	0.01	0	0.04	0	0.02	0
f9	0.09	0	0.03	0	0.01	0	0.02	0	0.01	0	0.01	0	0	0	0.04	0	0.01	0
f10	0	0	0.03	0	0.03	0	0.01	0	0.1	0	0	0	0.01	0	0.32	0	0.27	0
f11	0.26	0.01	0.06	0.01	0.06	0.01	0.11	0.01	0.1	0	0.09	0	0.26	0.01	0.28	0.01	0.2	0.02
f12	0.04	0	0	0	0.02	0	0.03	0.01	0.03	0	0.04	0	0.04	0	0.46	0	0.46	0

5.2.3 Problem Justification

In previous section, we simply compare some representative statistics, such as the mean values of features to show the "Spam Drift" problem. To further illustrate the changing of the statistical features in a dataset, a natural approach is to model the distribution of the data [37]. There are two kinds of approaches: parametric and non-parametric. Parametric approaches are very powerful when the specific distribution of the dataset, like Normal Distribution, is already known. However, the distribution of the Twitter spam data is unknown, thus it is not possible to apply parametric approaches. Consequently, non-parametric methods, such as statistical tests, which make no assumptions of the dataset distributions are used by researchers [42].

The statistical tests are to compute the distance of two distributions to determine the change. One of the most common measures to compute the distance of distributions is Kullback-Leibler (KL) Divergence [37, 102]. The suitability of KL Divergence to be used in measuring distributions can be found in [37]. In [79], Juan *et al.* also use KL Divergence to model language models of tweets. KL Divergence, which is also known as relative entropy is defined as

$$D_{kl}(P\|Q) = \sum_i P(i) log \frac{P(i)}{Q(i)}.$$

It is used to compare two probability distributions. We need to map data points into distributions to apply the formula. According to [36], let $\mathbf{s} = \{x_1, x_2, \ldots, x_n\}$ be a multi-set from a finite set $\mathbf{F}$ containing numerical feature values, and denote $N(x|\mathbf{s})$

the number of appearances of $x \in \mathbf{s}$, thus the relative proportion of each x is donated by

$$P_{\mathbf{s}}(x) = \frac{N(x|\mathbf{s})}{n}.$$

However, the ratio of p/q is undefined if $Q(i) = 0$. As suggested by [62], the estimate $P_{\mathbf{s}}$ is replaced as,

$$P_{\mathbf{s}}(x) = \frac{N(x|\mathbf{s}) + 0.5}{n + |F|/2}.$$

when $|F|$ is number of elements in the finite set $\mathbf{F}$. The distance between two day's tweets, $D1$ and $D2$ is,

$$D(D1\|D2) = \sum_{x \in \mathbf{F}} P_{\mathbf{D1}}(x) log \frac{P_{\mathbf{D1}}(x)}{P_{\mathbf{D2}}(x)}.$$

We compute the KL Divergence of each feature of spam and non-spam tweets in two adjacent days, which is listed in TABLE 5.2. The shadowed ones are the KL Divergence of features of non-spam tweets, while the others are the KL Divergence of features of spam tweets. KL Divergence indicates the dissimilarity of two distributions. The larger the value is, the more different the two distributions are. As shown in Table 5.2, the KL Divergence of spam tweets in two adjacent days are much larger than that of the non-spam tweets for more than half the features. Taking f1 ("account_age") for example, the KL Divergence of spam between Day 1 and Day 2 is 0.36, while it is only 0.04 for non-spam, which indicates that the distribution of f1 of spam in Day 1 is much different to it in D2, compared with non-spam tweets'

distribution. From these KL Divergence values, we can see that the distribution of spam tweets' features is changing unpredictably from day to day. Nevertheless, the distribution of training data is unchanged. As the knowledge structure which learns from the unchanged training data is not updated while being used to classify new incoming tweets, the performance of classifiers becomes inaccurate. As it is illustrated in Fig. 5.2, while the spam changes, the decision boundary is not updated. Consequently, more spam tweets are misclassified as non-spam.

5.3 Proposed Scheme: *Lfun*

Existing machine learning based spam detection methods suffer from the problem of "Spam Drift" due to the change of statistical features of spam tweets as time goes on. When "spam drifts", the old classification model is not updated with "changed" spam samples, as a result, the classification results will gradually become inaccurate. To solve this problem, obtaining the "changed" samples to update the classification model is very important. By observing that there are such samples in the unlabelled incoming tweets which are very easy to collect, we propose a scheme called "Lfun" to address "Spam Drift" problem.

This section presents our Lfun scheme to deal with the drift problem in Twitter spam detection. Fig. 5.3 illustrates the framework of our proposed scheme. There are two main components in this framework: LDT is to learn from detected spam tweets

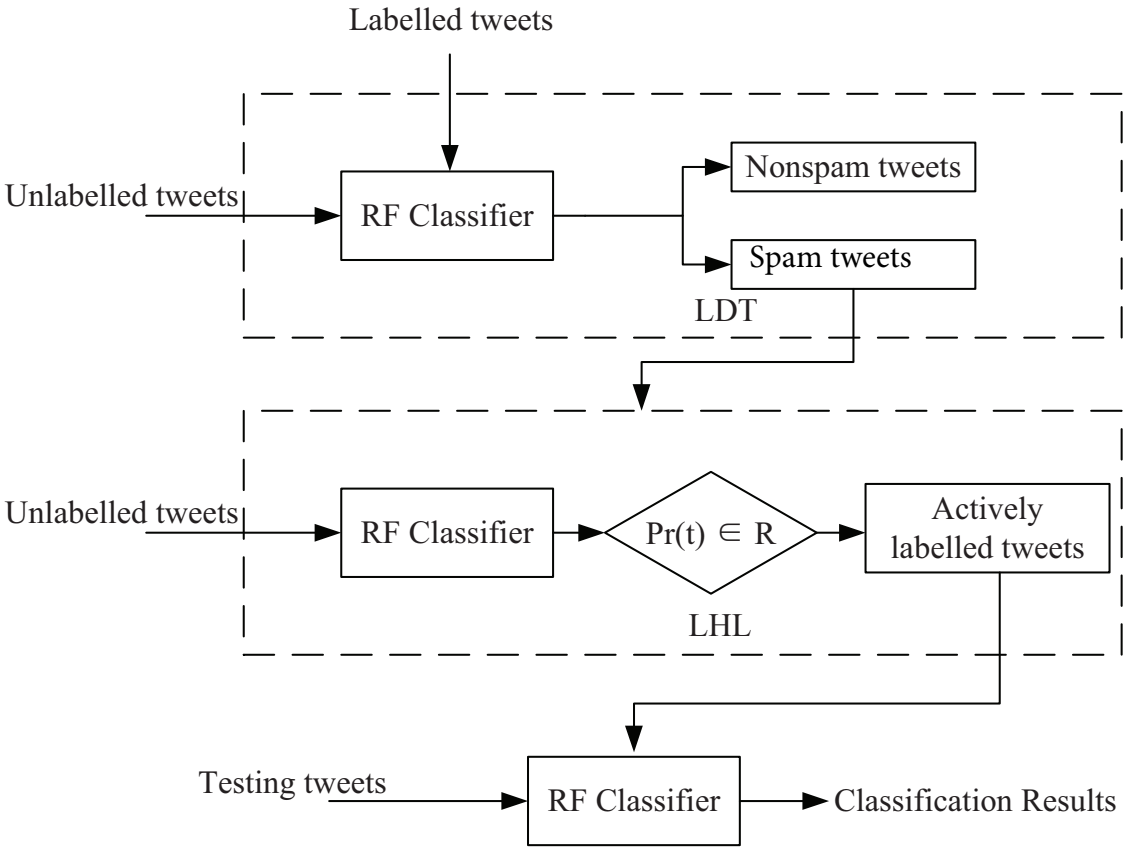


Figure 5.3: Lfun Framework

and LHL is to learn from human labelling. In "Drifted Spam Detection" scenario, we have already got a small amount of labelled spam and non-spam tweets. However, there are not enough samples of "changed" spam. It is extraordinary expensive to have human label a large amount of "changed" tweets. Consequently, we make use of the above mentioned two components to automatically extract "changed" spam tweets from a set of unlabelled tweets, which are very easy to collected from Twitter. Once getting enough labelled "changed" spam tweets, we implement the scheme which

employs a sufficiently powerful algorithm, Random Forest, to perform classification. Our Lfun scheme is summarised in Algorithm 1.

5.3.1 Learning from Detected Spam Tweets

LDT is used to deal with a classification scenario where there is a sufficiently robust algorithm, but in lack of more data [95]. By learning from a large number of unlabelled data, LDT can obtain sufficient new information, which can be used to update the classification model.

In a LDT learning scenario, we are given a labelled data set $T_l = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, containing m labelled tweets, where $x_i \in \mathbb{R}^k (i = 1, 2, \ldots, m)$ is the feature vector of a tweet, $y_i \in \{spam, non-spam\}$ is the category label of a tweet. We are also given a large data set $T_u = \{x_{m+1}, y_{m+1}), (x_{m+2}, y_{m+2}), \ldots, (x_{m+n}, y_{m+n})\}$ containing n unlabelled tweets ($n >> m$). Then a classifier φ is trained by T_l. φ can be used to divide T_u into spam T_{spam} and non-spam $T_{non-spam}$. Labelled spam tweets from T_u will be added into the labelled data set T_l to form a new training data set.

The basic of LDT is to find a function $\varphi : \mathbb{R}^k \longrightarrow \{spam, non-spam\}$ to predict the label $y \in \{spam, non-spam\}$ of new tweets when trained by T_{l+spam}, which is the combination of the labelled data set T_l and spam tweets T_{spam} identified from T_u. Particularly, the unlabelled data set T_u used in LDT does not have to share the same distribution with the labelled data set T_l [52]. In addition, only detected

spam tweets will be added into the training data. The reason is that, we've already gained sufficient information of non-spam tweets, as the statistical properties are not changing for non-spam tweets. It is not necessary for us to gain more information about non-spam tweets.

However, the spam tweets detected by the classifier that is trained using T_l also have the same or similar distribution of old spam. We need samples from "changed spam" to calibrate the classifier. We then use LHL (in Section 5.3.2) to get "changed spam" samples.

5.3.2 Learning from Human Labelling

In a supervised spam detection system, a learning algorithm, such as Random Forest, must be trained by sufficient labelled data to obtain more accurate detection results. However, labelled instances are very expensive and time-consuming to obtain. Fortunately, we have a huge number of unlabelled tweets which can be easily collected. The LHL in our Lfun is best suited where there are numerous unlabelled data instances, and human annotator anticipating to label many of them to train an accurate system [103]. LHL aims to minimize the labelling cost by using different learning criteria to select most informative samples from unlabelled data to be labelled by a human annotator [145]. We also import active learning in our Lfun scheme.

Now let us define our learning component in a formal way. In supervised Twitter

Algorithm 1 Lfun Algorithm

Require: labelled training set $\{\psi_1, ..., \psi_N\}$,

unlabelled tweets $T_{unlabelled}$,

a binary classification algorithm Φ,

Ensure: manually labelled selected tweets T_m

$T_{labelled} \leftarrow \bigcup_{i=1}^{N} \psi_i$

// Use Φ to create a classifier Cls from $T_{labelled}$:

$Cls \leftarrow \Phi : T_{labelled}$

// $T_{unlabelled}$ is classified as T_{spam} and $T_{non-spam}$:

$T_{spam} + T_{non-spam} \leftarrow T_{unlabelled}$

// Merge spam tweets T_{spam} classified by Cls into $T_{labelled}$:

$T_{ex} \leftarrow T_{labelled} + T_{spam}$

// use T_{ex} to re-train the classifier Cls :

$Cls \leftarrow \Phi : T_{ex}$

// determine the incoming tweet's suitability for selection:

$U \leftarrow \emptyset$

for $i = 1$ **to** k **do**

  if U_i meet the selection criteria S **then**

    $U \leftarrow (U \cup U_i)$

  end if

end for

// manually labelling each u_i in U

$T_m \leftarrow \emptyset$

for $i = 1$ **to** k **do**

  manually label each u_i

  $T_m \leftarrow (T_m \cup u_i)$

end for

spam detection, we are given a labelled training data set $T_{training} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, containing m labelled tweets, where $x_i \in \mathbb{R}^k (i = 1, 2, \ldots, m)$ is the feature vector of a tweet, $y_i \in \{spam, non - spam\}$ is the category label of a tweet. The label y_i of a tweet x_i is donated as $y = f(x)$. The task is then to learn a function $\hat{f}$ which can correctly classify a tweet to spam or non-spam. We use generalisation

error to measure the accuracy of the learned function:

$$Error(\hat{f}) = \sum_{x \in T_{training}} \mathcal{L}\Big(f(x), \hat{f(x)}\Big) P(x).$$

In practice, $f(x)$ is not available for testing data instances. Therefore, it is usual to estimate the generalisation error by the test error:

$$Error(\hat{f}) = \sum_{x \in T_{testing}} \mathcal{L}\Big(f(x), \hat{f(x)}\Big) P(x),$$

where $T_{testing}$ refers to the testing tweets, and prediction error can be measured by a loss function $\mathcal{L}$, such as mean squared error (MSE) [99]:

$$\mathcal{L}_{MSE}\Big(f(x), \hat{f(x)}\Big) = \Big(f(x) - \hat{f(x)}\Big)^2.$$

The learning criteria is set to select the most useful instances $X_{selected}$ and add them to the training set $T_{training}$ for achieving some certain objectives. Let us consider this objective as the minimization of generation error of a learned function trained by $T_{training}$. So the learning criteria can be donated as

$$Error(T_{training} \cup \{X_{selected}\}).$$

The goal of this kind of learning is to select instances $X_{selected}$ which can minimize the generalisation error $Error(X_{selected})$:

$$argmin\ Error(X_{selected}).$$

As a result, good selection criteria must be estimated to minimize the error. In Lfun scheme, we apply the selection criteria, called "Probability Threshold Filter

Model", to select the most informative tweets to tackle "Spam Drift". In order to achieve this, Random Forest (RF) is used to determine the probability of a tweet whether it belongs to spam or not. Random Forest [19] can generate many classification trees after being trained with T_{ex} from Asymmetric Self-Learning. When classifying a new incoming tweet, each tree in the forest will give a class prediction. Then forest chooses the classification result which has the most votes. In our case, we set the number of trees to m, if n trees vote for the class "spam", the probability of the tweet to be classified as "spam" is $Pr = \frac{n}{m}$.

Through our empirical study, the mis-classification mostly occurred when $Pr \in [0.4, 0.7]$. So we set the threshold τ to $Pr \in [0.4, 0.7]$. After we pre-filter some candidate tweets to be labelled using the "Probability Threshold Filter Model", the number of tweets is still too many. We then randomly select a smaller number of tweets from the candidate tweets (we set it to be 100 in our experiments) to be manually labelled. As shown in Fig. 5.3, the manually labelled tweets, along with T_{ex} will be used to train a new classifier, which can tackle "Spam Drift" problem.

5.3.3 Performance Benefit Justification

We study the performance benefit of the proposed Lfun scheme by providing the theoretical analysis in this section. Fig. 5.4 illustrates the performance benefit by using simulation. We use three normal distributions (listed below) to simulate this: w_0

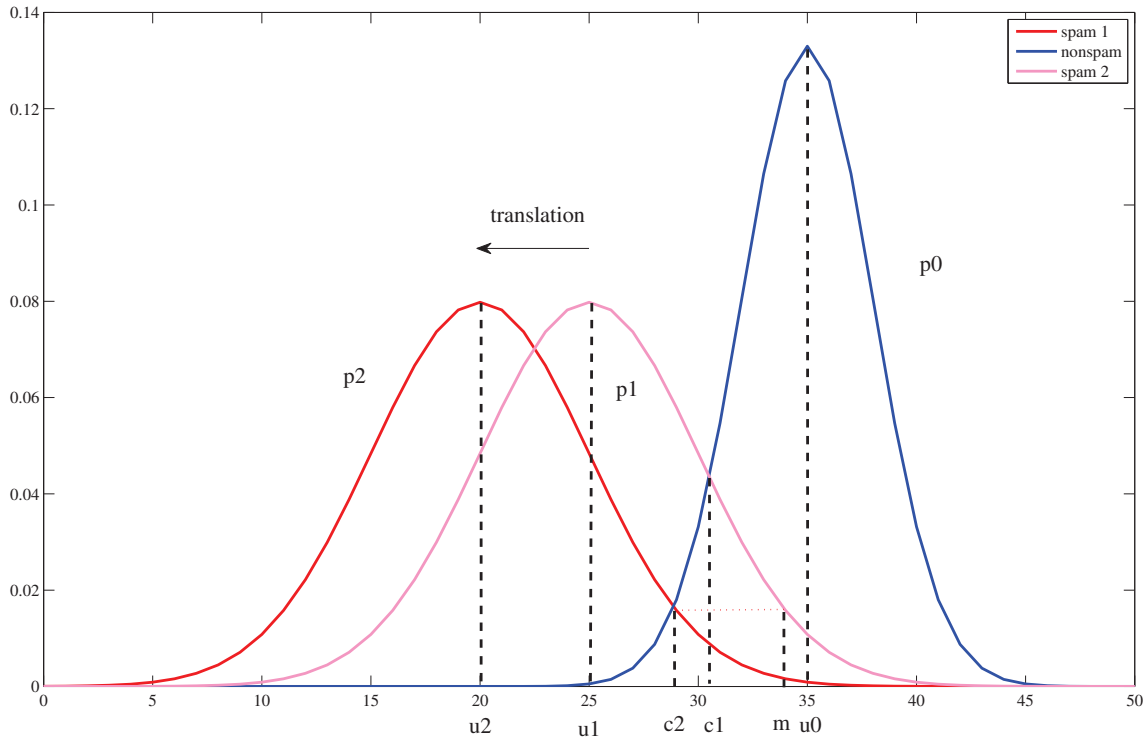
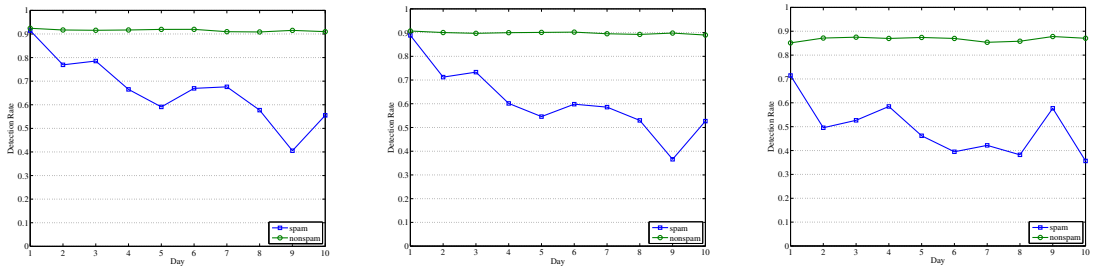


Figure 5.4: Performance Benefit Illustration

represents the distribution of non-spam, while w_1 and w_2 represents the distribution of spam before and after using our Lfun approach, respectively.

$$\begin{cases} w_0 \sim N(\mu_0, \sigma_0^2) \\ w_1 \sim N(\mu_1, \sigma_{12}^2) \\ w_2 \sim N(\mu_2, \sigma_{12}^2) \end{cases}$$

The PDFs (probability distribution functions) [38] of these three distributions, w_0, w_1 and w_2 are illustrated as p_0, p_1 and p_2 in Fig. 5.4. We assume that only the mean μ_1 of w_1 changes to μ_2, but the variance σ_{12} is not changing.



(a) Random Forest (b) C4.5 Decision Tree (c) Bayes Network

Figure 5.5: Trend of Detection Rate

As p_1 translated to p_2, we can always find m, which can make

$$m - c_2 = \mu_1 - \mu_2, \tag{5.3.1}$$

and

$$p_1(m) = p_2(c_2). \tag{5.3.2}$$

As $c_2 < c_1$, we have

$$p_0(c_2) < p_0(c_1). \tag{5.3.3}$$

We also have

$$p_0(c_1) = p_1(c_1),\ p_0(c_2) = p_2(c_2). \tag{5.3.4}$$

From Equation. 5.3.3 and Equation. 5.3.4, we get

$$p_1(c_1) > p_2(c_2). \tag{5.3.5}$$

From Equation. 5.3.2 and Equation. 5.3.5, we can have

$$p_1(c_1) > p_1(m). \tag{5.3.6}$$

As a result,

$$m > c_1. \tag{5.3.7}$$

Taking into account Equation. 5.3.7 and Equation. 5.3.1, we can have $c_1 - c_2 < \mu_1 - \mu_2$. So,

$$c_2 - \mu_2 > c_1 - \mu_1. \tag{5.3.8}$$

The error rate of classification before Lfun,

$$\begin{aligned} P_1(error) &= P(x > c_1) + P(x < c_2) \\ &= \int_{c_1}^{\infty} p_1(t)dt + \int_{-\infty}^{c_1} p_0(t)dt \\ &= 1 - \phi(\frac{c_1 - \mu_1}{\sigma_{12}}) + \phi(\frac{c_1 - \mu_0}{\sigma_0}). \end{aligned}$$

Similarly, we have the error rate after using Lfun

$$P_2(error) = 1 - \phi(\frac{c_2 - \mu_2}{\sigma_{12}}) + \phi(\frac{c_2 - \mu_0}{\sigma_0}).$$

The difference of $P_1(error)$ and $P_2(error)$,

$$\begin{aligned} &P_1(error) - P_2(error) \\ &= \Big[\phi(\tfrac{c_2-\mu_2}{\sigma_{12}}) - \phi(\tfrac{c_1-\mu_1}{\sigma_{12}})\Big] + \Big[\phi(\tfrac{c_1-\mu_0}{\sigma_0}) - \phi(\tfrac{c_2-\mu_0}{\sigma_0})\Big], \end{aligned} \tag{5.3.9}$$

while

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-t^2/2} dt. \tag{5.3.10}$$

The differentiation of Equation 5.3.10 is $\phi'(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} > 0$. So, we can have $\phi(a) > \phi(b)$ when $a > b$. From Equation. 5.3.8, we know $\frac{c_2-\mu_2}{\sigma_{12}} > \frac{c_1-\mu_1}{\sigma_{12}}$. Consequently,

$$\phi(\frac{c_2 - \mu_2}{\sigma_{12}}) > \phi(\frac{c_1 - \mu_1}{\sigma_{12}}). \tag{5.3.11}$$

As $c_1 > c_2$, we have $\frac{c_1-\mu_0}{\sigma_0} > \frac{c_2-\mu_0}{\sigma_0}$. Then, we know

$$\phi(\frac{c_1-\mu_0}{\sigma_0}) > \phi(\frac{c_2-\mu_0}{\sigma_0}). \tag{5.3.12}$$

Substitute Equation. 5.3.11 and 5.3.12 into 5.3.9, we will have

$$P_1(error) - P_2(error) > 0. \tag{5.3.13}$$

Obviously, our proposed approach can effectively reduce the probability of error from Equation 5.3.13.

5.4 Performance Evaluation

In this section, we evaluate the performance of the proposed Lfun scheme in detecting "drifted" Twitter spam. All the experiments are carried out on our real-world 10 consecutive days' tweets with each day containing 100k spam tweets and 100k non-spam tweets.

As in existing works [135], we also use *F-measure* and *Detection Rate* to measure the performance. Despite that both of the metrics are used to evaluate all the classes' performance, we only focus on the F-measure and Detection Rate of spam class. F-measure is an evaluation metric which combines precision and recall to measure the per-class performance of classification or detection algorithms. It can be calculated by

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall}.$$

Detection Rate is defined as the ratio of those tweets correctly classified as belonging to class *spam* to the total number of tweets in class *spam*, it can be calculated by

$$DetectionRate = \frac{TP}{TP + FN}.$$

In the evaluation, we have designed three sets of experiments in order to show the *impact of spam drift* (in Section 5.4.1) firstly, then the benefit of our proposed Lfun (in Section 5.4.2) and the comparisons with other traditional machine learning algorithms (in Section 5.4.3). We repeat the experiments for 100 times with different random training samples and report the average values on all the 100 runs.

5.4.1 Impact of Spam Drift

In order to evaluate the impact of "Spam Drift" problem, we perform a number of experiments in this section. It is aiming to show that the performance of a traditional classifier, for example C4.5 Decision Tree, varies over time when "Spam Drift" exists.

During these experiments, Day 1 data is divided into two parts, half for training pool where training data can be extracted from, and another half for testing purpose. We create a classifier by using a supervised classification algorithm, and train it with 10k spam and 10k non-spam tweets which are randomly sampled from the training pool of Day 1. Then the classifier is used to classify the testing data in Day1, as well as the testing samples in Day 2 to Day 10.

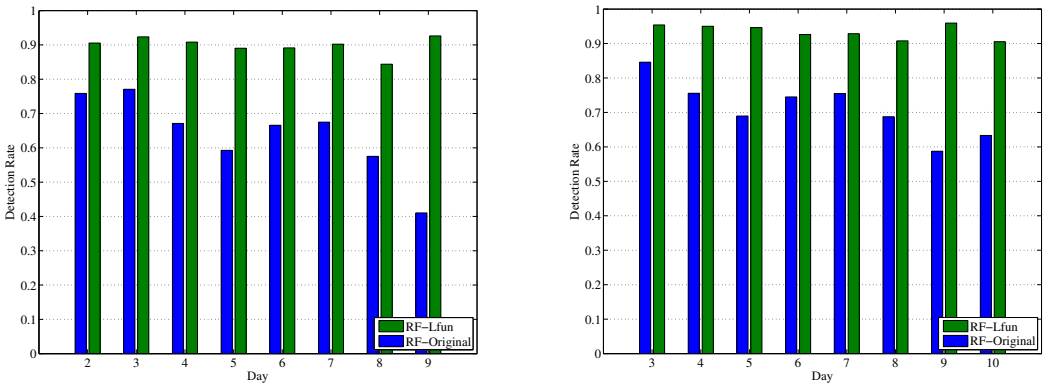
Fig. 5.5 shows the Detection Rate of both spam and non-spam tweets on three

classifiers, Random Forest, C4.5 Decision Tree and Bayes Network. We can see that, the Detection Rate of non-spam is very stable, it keeps above 90% for Random Forest and C4.5 Decision Tree, and near 90% for Bayes Network, despite the change of testing data. However, when it comes to spam tweets, the Detection Rate fluctuates dramatically, and the overall trend is decreasing. The Detection Rates for Random Forest and C4.5 Decision Tree are 90% in the first day, but they could decrease to less then 40% in the 9th day. This phenomenon also applies with Bayes Network, the Detection Rate decreases from 70% on 1st day to less than 50% for most of the other testing days.

5.4.2 Performance of Lfun

We evaluate the performance of Lfun here, by using F-measure and Detection Rate. The number labelled training samples from old day (*i.e.* Day 1 and Day 2 in this case) is 5000. The number of manually labelled samples during Lfun is set to 100.

Fig. 5.6 shows the Detection Rate of Lfun, when Day 1 data (Fig. 5.6a) or Day 2 data (Fig. 5.6b) is used for training and the rest days are used for testing. We can see from Fig. 5.6a that, the Detection Rates of original Random Forest are relatively low. For example, the Detection Rate when testing on Day 9 is only around 40%. However, our RF-Lfun can reach over 90% Detection Rate on the same day. While



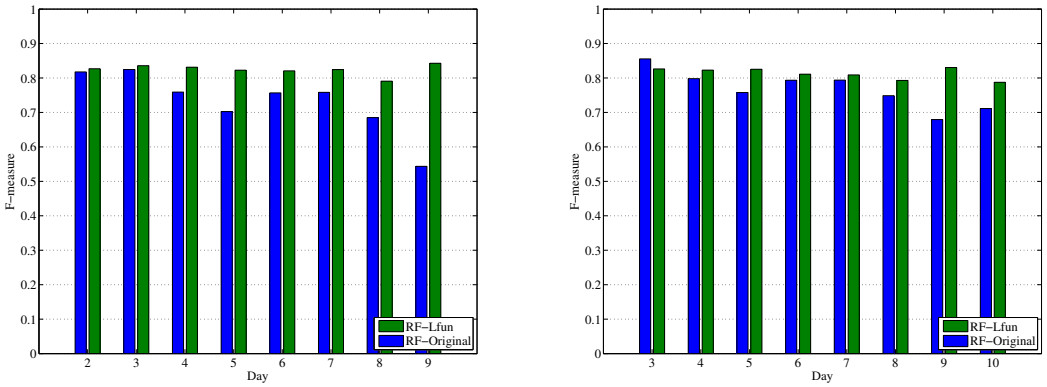
(a) Day 1 training, Day 2 to 9 testing

(b) Day 2 training, Day 3 to 10 testing

Figure 5.6: Detection Rate of Lfun

Random Forest can only achieve Detection Rate ranging from 45% to 80%, our RF-Lfun can rise as high as 90% Detection Rate. This also happens when training data is from Day 2, and testing data is from Day 3 to Day 10, as illustrated in Fig. 5.6b. The highest Detection Rate of Random Forest is around 85%, but that of RF-Lfun is over 95%. Generally, our Lfun can detect most of the spam tweets even with "Spam Drift". The reason is that, our Lfun brings more samples of "changed spam tweets" to update the training process.

Fig. 5.7 shows the F-measure of Random Forest using Lfun approach compared with it without using Lfun. We can see that, the F-measure of original Random Forest keeps decreasing from 80% to 55% as the testing data changes from Day 2 to Day 9 in Fig. 5.7a. However, once it is applied with our Lfun approach, the F-measure becomes stable, which is always greater than 80%, except on Day 8. Similarly, when



(a) Day 1 training, Day 2 to 9 testing

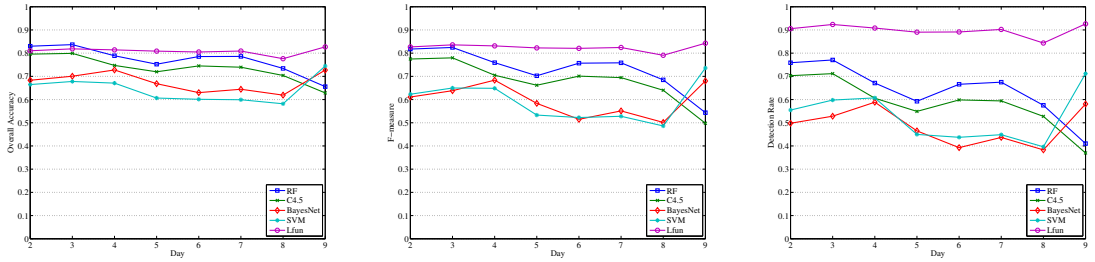
(b) Day 2 training, Day 3 to 10 testing

Figure 5.7: F-measure of Lfun

the training data is from Day 2, F-measure of Random Forest is decreasing as well. But F-measure of our Lfun-RF is not fluctuating, as shown in Fig. 5.7b. Nevertheless, the proposed Lfun can effectively improve the F-measure and the improvement is up to 25% in the best case.

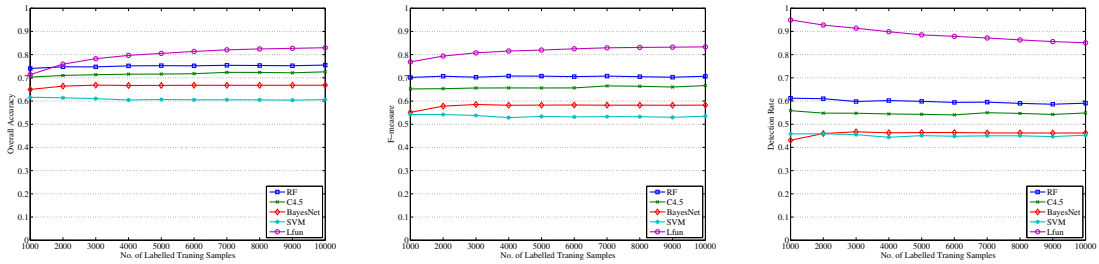
5.4.3 Comparisons with other Algorithms

In this section, we compare our Lfun approach with four traditional machine learning algorithms (*Random Forest, C4.5 Decision Tree, Bayes Network and SVM*) to detect spam tweets in the "drift" scenario. There are two sets of experiments carried out. One set is to evaluate the performance while training data is from Day 1, and testing data are varying from Day 2 to Day 9. Another set is to evaluate the performance when training and testing data are from two specified days, but the number of labelled



(a) Overall Accuracy (b) F-measure (c) Detection Rate

Figure 5.8: Comparisons with other Algorithms (changing testing days)



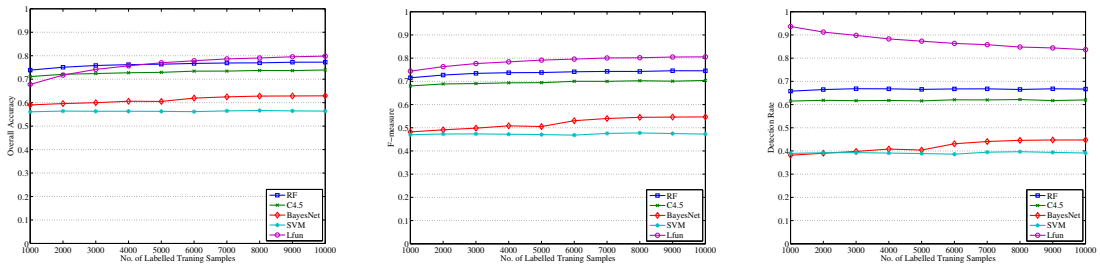
(a) Overall Accuracy (b) F-measure (c) Detection Rate

Figure 5.9: Comparisons with other Algorithms (training on Day 1 and testing on Day 5)

training data is changing from 1000 to 10000.

5.4.3.1 Comparisons with Changing Days

Fig. 5.8 demonstrates the experimental results in terms of overall accuracy, F-measure and detection rate of Lfun compared to other algorithms, when the testing days are varying. We can see from Fig. 5.8a that, the overall accuracy of Lfun outperforms all the other algorithms, followed by Random Forest, C4.5 Decision Tree, Bayes Network



(a) Overall Accuracy (b) F-measure (c) Detection Rate

Figure 5.10: Comparisons with other Algorithms (training on Day 4 and testing on Day 8)

and SVM. In terms of F-measure (see Fig. 5.8b), our Lfun is also the best among all the algorithms. For example, it is over 30% higher than C4.5 Decision Tree when testing data is from Day 9. Furthermore, the performance of Lfun is much better in terms of detection rate. Fig. 5.8c show that, the detection rate of Lfun is above 90% for most of the days. However, the detection rate of all the others is below 80%. Especially, Bayes Network has the lowest detection rate, which is below 50%. In general, our Lfun is the best among all the algorithms evaluated by all the three metrics.

5.4.3.2 Comparisons with Changing Labelled Training Samples

Fig. 5.9 and Fig. 5.10 report the evaluation results when the number of labelled training samples is changing. The training and testing data is from Day 1 and Day 5 in Fig. 5.9, while the training and testing data is from Day 4 and Day 8 in Fig.

5.10. We can see that the overall accuracy of Lfun increases from 70% to 80% with the increase of labelled training samples. It is better than the four algorithms in comparison, as the best of them (C4.5 Decision Tree) can only achieve less than 74% overall accuracy. When it comes to F-measure, the performance of Lfun is still the best; it is 10% higher than that of C4.5 Decision Tree and nearly 30% higher than that of SVM. In terms of detection rate, our Lfun is about 30% higher than the second best algorithm. Similarly in Fig. 5.10, Lfun outperforms all the other algorithms.

5.5 Discussions

In research community, there are also some machine learning approaches related to our proposed method. For example, online learning and incremental learning. They are both common machine learning algorithms to continuously update the prediction model with new training data for better future classification. They can generate a prediction model and put it into operation without much training data at first, but they require new training data to update the model. When it comes to online Twitter spam classification, it is very difficult to label enough training samples to update the model. The reasons are two-folds. Firstly, it is significantly time-consuming to label a large amount of tweets by human. Secondly, it is difficult to gain enough spam tweets even we have got a large number of human-labelled tweets, as the spam rate of Twitter is about 5% [29]. If there are not enough spam samples (Lfun does not

need non-spam samples as non-spam tweets are not drifting) to retrain the model, it is not able to solve the "spam drift" issue.

Our Lfun approach has the same advantage of online learning and incremental learning, *i.e.*, it can be deployed without much training data at the beginning, but to be updated when new training data comes. Different to online and incremental learning, we incorporate both automated labelling and human labelling. The LDT component learns from the detected tweets. This competent is automatically updated with detected spam tweets with no human effort. To better adjust the prediction model, we also import LHL component, which learns from human labelling. To minimize human effort, LHL only samples a very small number of tweets for labelling, for example, 100 tweets in our experiments. In addition, it does not randomly pick up tweets to label, but to be in line with selection criteria called "Probability Threshold Filter Model" which can choose the most useful tweets. Benefiting from these two components, our Lfun approach can successfully deal with "spam drift", but with the least human effort.

5.6 Summary

In this paper, we firstly identify the "Spam Drift" problem in statistical features based Twitter spam detection. In order to solve this problem, we propose a Lfun approach. In our Lfun scheme, classifiers will be re-trained by the added "changed

spam" tweets which are learnt from unlabelled samples, thus it can reduce the impact of "Spam Drift" significantly. We evaluate the performance of Lfun approach in terms of Detection Rate and F-measure. Experimental results show that both detection rate and F-measure are improved a lot when applying with our Lfun approach. We also compare Lfun to four traditional machine learning algorithms, and find that our Lfun outperforms all four algorithms in terms of overall accuracy, F-measure and Detection Rate.

There is also a limitation in our Lfun scheme. The benefit of "old" labelled spam is to eliminate the impact of "spam drift" to classify more accurate spam tweets in future days. The effectiveness of "old" spam has been proved by our experiments during a short period. However, the effectiveness will decrease as the correlation of "very old" spam becomes less with the new spam in the long term run. In the future, we will incorporate incremental adjustment to adjust the training data, such as dropping the "too old" samples after a certain time. It can not only eliminate unuseful information in the training data but also make it faster to train the model as the number of training samples decrease.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Online Social Networks have reshaped the way of communications and information retrieval for individuals' daily life. The popularity of OSNs not only attracts legitimate users, but also spammers, who mainly spread unsolicited posts (in the form of tweets or updates) containing malicious links that directs victims to external sites containing malware downloads, phishing, drug sales, scams and so on. Due to the higher successful rate of victims to click malicious links compared to email, more and more spammers leverage OSNs to distribute spam. Such spam not only pollute the platforms but also exploit users' critical information.

To tackle this challenge, we begin with a through data analysis of spam on Twitter. We demonstrate that various deceptive content of spam performs differently in luring victims to malicious sites and the regional response rate to various Twitter

spam outbreaks varies greatly. In addition, spammers are becoming "smarter" by employing more complex spamming strategies to avoid being detected. We then carry out a performance evaluation of machine learning based streaming spam detection approaches, which is from three different aspects of data, feature and model. We, therefore, identified an unseen issue in Twitter spam detection, *i.e.* "Spam Drift".

We, thus, develop Lfun approach to address "Spam Drift" problem. Our Lfun approach contains both automated labelling and human labelling. The LDT component learns from the detected tweets. This competent is automatically updated with detected spam tweets with no human effort. To better adjust the prediction model, we also import LHL component, which learns from human labelling. By combining them together, our approach can deal with "Spam Drift".

The key value of carrying out an depth analysis of Twitter spam and thorough evaluation of streaming spam detection mechanisms is to expose current flaws in Twitter spam detection area. Thus, we can overcome the shortcomings and implement an effective and long-term detection system. We summarise our most important findings and proposed solution here for the social spam research community.

- **Better Understanding of Twitter Spam:** After analysing around 600 million of spam tweets, we find various deceptive information of spam performs differently in luring victims to malicious sites. In addition, the regional distribution of victims varies due to different types of deceptive information. We

have also identified three new spamming strategies applied by spammers.

- **Evaluating Streaming Spam Detection Schemes:** We evaluate the impact of different factors to performance the streaming spam detection, which include spam to non-spam ratio, feature discretization, training data size, data sampling, time-related data, and machine learning algorithms. The results show the streaming spam tweet detection is still a big challenge and a robust detection technique should take into account the three aspects of data, feature and model.

- **Address of Unseen "Spam Drift" Problem:** We firstly identify the "Spam Drift" issue in detecting Twitter spam. We then propose a Lfun scheme which can discover "changed" spam tweets from unlabelled tweets and incorporate them into classifier's training process. Our Lfun scheme can effectively detect spam tweets even when they are drifting.

6.2 Future Work

In this thesis, we provide insights on deceptive information contained in spam tweets and emerging spamming strategies. We also firstly identify and solve the "Spam Drift" issue. However, the war between researchers and spammers is never ending. There are a couple of future works can be considered.

6.2.1 Improvement of Our Lfun Scheme

We successfully demonstrate the effectiveness of our Lfun scheme in this thesis. However, there is also a limitation in our proposed Lfun approach. The benefit of "old" labelled spam is to eliminate the impact of "Spam Drift", so as to identify more accurate spam tweets in future days. The effectiveness of "old" spam has been proved by our experiments during a short period. However, the effectiveness will decrease as the correlation of "very old" spam becomes less with the new spam in the long term run. In the future, we will incorporate incremental adjustment to adjust the training data, such as dropping the "too old" samples after a certain time. It can not only eliminate useless information in the training data but also make it faster to train the model as the number of training samples decrease.

Also, the main reason causes "Spam Drift" is that the changing of statistical features. How about we do not use statistical features? This issue will then be avoided. Deep Learning recently has shown its super power in dealing with unstructured data, such as video, image, text and audio. There has not been any works to use deep learning to solve Twitter spam issue. We can employ deep learning techniques to solve social spam problems in the future. As tweet itself contains a lot of information, Natural Language Processing techniques are also promising in this field.

6.2.2 Operational Deployment, NLP, and Deep Learning for Twitter Spam Detection

Operational deployment should be one concern of spam detection on OSNs. Real-world applications of social spam detection need to work online, reporting live information or trigger mitigations according to detection results. Online detection requires trade-offs between performance and accuracy. However, most of the current works are testing offline and using a small set of data set. As Twitter is seeing more than 500 million tweets a day, the operational deployment is a challenging area to work on.

How do human beings determine whether a tweet is spam or not? Yes, through reading and understanding the semantic meaning. Natural Language Processing (NLP) has demonstrated its success in machine translation, question answering systems, as well as filtering email spam. Due to the difference of text length among email (up to hundreds of words) and tweet (at most 140 characters), NLP's success in filtering email spam cannot be simply copied to fight with Twitter spam. However, it is very promising to use NLP to detect Twitter spam since we can also use semantic meaning to infer spam.

Deep learning is becoming extremely hot in research community. Researchers have applied deep neural nets to hand-writing recognition, voice recognition, object detection in images, as well as text understanding. How can we leverage deep learning

algorithms to detect Twitter spam? It is really a good working area.

Bibliography

[1] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, pages 41–52, New York, NY, USA, 2006. ACM.

[2] H. Achrekar, A. Gandhe, R. Lazarus, S.-H. Yu, and B. Liu. Online social networks flu trend tracker: A novel sensory approach to predict flu trends. In J. Gabriel, J. Schier, S. Huffel, E. Conchon, C. Correia, A. Fred, and H. Gamboa, editors, *Biomedical Engineering Systems and Technologies*, volume 357 of *Communications in Computer and Information Science*, pages 353–368. Springer Berlin Heidelberg, 2013.

[3] F. Ahmed and M. Abulaish. A generic statistical approach for spam detection in online social networks. *Computer Communications*, 36(101):1120 – 1129, 2013.

[4] D. Antonakaki, I. Polakis, E. Athanasopoulos, S. Ioannidis, and P. Fragopoulou.

Exploiting abused trending topics to identify spam campaigns in twitter. *Social Network Analysis and Mining*, 6(1):1–11, 2016.

[5] D. Antoniades, I. Polakis, G. Kontaxis, E. Athanasopoulos, S. Ioannidis, E. P. Markatos, and T. Karagiannis. we.b: the web of short urls. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 715–724, New York, NY, USA, 2011. ACM.

[6] L. Backstrom and J. Leskovec. Supervised random walks: Predicting and recommending links in social networks. *CoRR*, abs/1011.4071, 2010.

[7] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic. The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 519–528, New York, NY, USA, 2012. ACM.

[8] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammer on twitter. In *Seventh Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, July 2010.

[9] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, and M. Gonçalves. Detecting spammers and content promoters in online video social networks. In *Proceedings of the 32nd international ACM SIGIR conference on Research and*

development in information retrieval, SIGIR '09, pages 620–627, New York, NY, USA, 2009. ACM.

[10] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, C. Zhang, and K. Ross. Identifying video spammers in online social networks. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, AIRWeb '08, pages 45–52, New York, NY, USA, 2008. ACM.

[11] A. Bifet and E. Frank. Sentiment knowledge discovery in twitter streaming data. In *Proceedings of the 13th International Conference on Discovery Science*, DS'10, pages 1–15, Berlin, Heidelberg, 2010. Springer-Verlag.

[12] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 551–560, New York, NY, USA, 2009. ACM.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[14] J. Bollen, H. Mao, and A. Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. 2011.

[15] J. Bollen, A. Pepe, and H. Mao. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *CoRR*, abs/0911.1583, 2009.

[16] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. The socialbot network: when bots socialize for fame and money. In *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC '11, pages 93–102, New York, NY, USA, 2011. ACM.

[17] N. Bouguila, D. Ziou, and J. Vaillancourt. Unsupervised learning of a finite mixture model based on the dirichlet distribution and its application. *Trans. Img. Proc.*, 13(11):1533–1543, Nov. 2004.

[18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[19] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[20] C. Budak, D. Agrawal, and A. El Abbadi. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 665–674, New York, NY, USA, 2011. ACM.

[21] C. Cao and J. Caverlee. *Detecting Spam URLs in Social Media via Behavioral Analysis*, pages 703–714. Springer International Publishing, Cham, 2015.

[22] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Proceedings of the 9th USENIX*

Conference on Networked Systems Design and Implementation, NSDI'12, pages 15–15, Berkeley, CA, USA, 2012. USENIX Association.

[23] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 675–684, New York, NY, USA, 2011. ACM.

[24] CERT. Cert advisory ca-2000-04 love letter worm. May 2000.

[25] M. Cha, A. Mislove, and K. P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 721–730, New York, NY, USA, 2009. ACM.

[26] V. Chaoji, S. Ranu, R. Rastogi, and R. Bhatt. Recommendations to boost content spread in social networks. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 529–538, New York, NY, USA, 2012. ACM.

[27] C. Chen, J. Zhang, X. Chen, Y. Xiang, and W. Zhou. 6 million spam tweets: A large ground truth for timely twitter spam detection. In *IEEE ICC 2015 - Communication and Information Systems Security Symposium (ICC'15 (11) CISS)*, pages 8689–8694, London, United Kingdom, June 2015.

[28] C. Chen, J. Zhang, Y. Xiang, and W. Zhou. Asymmetric Self-Learning for tackling twitter spam drift. In *The Third International Workshop on Security and Privacy in Big Data (BigSecurity 2015)*, pages 237–242, Hong Kong, Hong Kong, Apr. 2015.

[29] C. Chen, J. Zhang, Y. Xiang, W. Zhou, and J. Oliver. Spammers are becoming smarter on twitter. *IT Professional*, 18(2):14–18, Mar.-April. 2016.

[30] F. Chen, P.-N. Tan, and A. K. Jain. A co-classification framework for detecting web spam and spammers in social media web sites. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1807–1810, New York, NY, USA, 2009. ACM.

[31] S. Chhabra, A. Aggarwal, F. Benevenuto, and P. Kumaraguru. Phi.sh/$ocial: the phishing landscape through short urls. In *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, CEAS '11, pages 92–101, New York, NY, USA, 2011. ACM.

[32] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Who is tweeting on twitter: human, bot, or cyborg? In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 21–30, New York, NY, USA, 2010. ACM.

[33] E. M. Clark, J. R. Williams, C. A. Jones, R. A. Galbraith, C. M. Danforth, and P. S. Dodds. Sifting robotic from organic text: A natural language approach for detecting automation on twitter. *Journal of Computational Science*, 16:1 – 7, 2016.

[34] A. Comparatives. Whole product dynamic real-world protection test. Technical report, AV Comparatives, Dec 2014.

[35] H. Costa, F. Benevenuto, and L. H. C. Merschmann. Detecting tip spam in location-based social networks. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 724–729, New York, NY, USA, 2013. ACM.

[36] I. Csiszar and J. Körner. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011.

[37] T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi. An information-theoretic approach to detecting changes in multi-dimensional data streams. In *In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications*, 2006.

[38] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.

[39] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna. Compa: Detecting compromised accounts on social networks. In *Annual Network and Distributed System Security Symposium*, 2013.

[40] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna. Towards detecting compromised accounts on social networks. *IEEE Transactions on Dependable and Secure Computing*, PP(99):1–1, 2015.

[41] F. Fathaliani and M. Bouguessa. A model-based approach for identifying spammers in social networks. In *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, pages 1–9, Oct 2015.

[42] J. a. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4):44:1–44:37, Mar. 2014.

[43] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. Choudhary. Towards online spam filtering in social networks. In *NDSS*, 2012.

[44] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, IMC '10, pages 35–47, New York, NY, USA, 2010. ACM.

[45] H. Gao, Y. Yang, K. Bu, Y. Chen, D. Downey, K. Lee, and A. Choudhary. Spam ain't as diverse as it seems: Throttling osn spam with templates underneath.

In *Proceedings of the 30th Annual Computer Security Applications Conference*, ACSAC '14, pages 76–85, New York, NY, USA, 2014. ACM.

[46] S. Ghosh, B. Viswanath, F. Kooti, N. K. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and K. P. Gummadi. Understanding and combating link farming in the twitter social network. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 61–70, New York, NY, USA, 2012. ACM.

[47] N. Z. Gong, A. Talwalkar, L. W. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, E. Shi, and D. Song. Predicting links and inferring attributes using a social-attribute network (san). *CoRR*, abs/1112.3265, 2011.

[48] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, pages 27–37, New York, NY, USA, 2010. ACM.

[49] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier. *TweetCred: Real-Time Credibility Assessment of Content on Twitter*, pages 228–243. Springer International Publishing, Cham, 2014.

[50] A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, Mar. 2009.

[51] N. Hamiel and S. Moyer. Satan is on my friends list. Japan, August 2008. Blackhat.

[52] K. Huang, Z. Xu, I. King, M. Lyu, and C. Campbell. Supervised self-taught learning: Actively transferring knowledge from unlabeled data. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 1272–1277, June 2009.

[53] M. Ilyas, M. Shafiq, A. Liu, and H. Radha. A distributed and privacy preserving algorithm for identifying information hubs in social networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 561–565, 2011.

[54] H. Interactive. Social network scams study. 2008.

[55] L. Invernizzi, S. Benvenuti, M. Cova, P. M. Comparetti, C. Kruegel, and G. Vigna. Evilseed: A guided approach to finding malicious web pages. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, pages 428–442, Washington, DC, USA, 2012. IEEE Computer Society.

[56] J. Jiang, C. Wilson, X. Wang, P. Huang, W. Sha, Y. Dai, and B. Y. Zhao. Understanding latent interactions in online social networks. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, IMC '10, pages 369–382, New York, NY, USA, 2010. ACM.

[57] X. Jin, C. X. Lin, J. Luo, and J. Han. Socialspamguard: A data mining-based spam detection system for social media networks. *PVLDB*, 4(12):1458–1461, 2011.

[58] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage. Spamalytics: An empirical analysis of spam marketing conversion. In *Proceedings of the 15th ACM Conference on Computer and Communications Security*, CCS '08, pages 3–14, New York, NY, USA, 2008. ACM.

[59] G. Kossinets, J. Kleinberg, and D. Watts. The structure of information pathways in a social communication network. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 435–443, New York, NY, USA, 2008. ACM.

[60] E. Kouloumpis, T. Wilson, and J. Moore. Twitter sentiment analysis: The good the bad and the omg! 2011.

[61] C. Kreibich and J. Crowcroft. Honeycomb: creating intrusion detection signatures using honeypots. *SIGCOMM Comput. Commun. Rev.*, 34(1):51–56, Jan. 2004.

[62] R. Krichevsky and V. Trofimov. The performance of universal encoding. *Information Theory, IEEE Transactions on*, 27(2):199–207, Mar 1981.

[63] B. Krishnamurthy and C. E. Wills. Privacy leakage in mobile online social networks. In *Proceedings of the 3rd conference on Online social networks*, WOSN'10, pages 4–4, Berkeley, CA, USA, 2010. USENIX Association.

[64] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.

[65] K. Lee, J. Caverlee, Z. Cheng, and D. Z. Sui. Content-driven detection of campaigns in social media. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 551–556, New York, NY, USA, 2011. ACM.

[66] K. Lee, J. Caverlee, and S. Webb. 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, pages 435–442, New York, NY, USA, 2010. ACM.

[67] S. Lee and J. Kim. Warningbird detecting suspicious urls in twitter stream. In *Annual Network & Distributed System Security Symposium*, 2012.

[68] S. Lee and J. Kim. Warningbird: A near real-time detection system for suspicious urls in twitter stream. *IEEE Transactions on Dependable and Secure Computing*, 10(3):183–195, 2013.

[69] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 641–650, New York, NY, USA, 2010. ACM.

[70] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, CIKM '03, pages 556–559, New York, NY, USA, 2003. ACM.

[71] W. Lin, X. Kong, P. S. Yu, Q. Wu, Y. Jia, and C. Li. Community detection in incomplete information networks. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 341–350, New York, NY, USA, 2012. ACM.

[72] L. Liu, Y. Lu, Y. Luo, R. Zhang, L. Itti, and J. Lu. Detecting "smart" spammers on social network: A topic model approach. *CoRR*, abs/1604.08504, 2016.

[73] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. Analyzing facebook privacy settings: user expectations vs. reality. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, IMC '11, pages 61–70, New York, NY, USA, 2011. ACM.

[74] C. Lumezanu, N. Feamster, and H. Klein. #bias: Measuring the tweeting behavior of propagandists. 2012.

[75] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Beyond blacklists: learning to detect malicious web sites from suspicious urls. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 1245–1254, New York, NY, USA, 2009. ACM.

[76] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Identifying suspicious urls: an application of large-scale online learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 681–688, New York, NY, USA, 2009. ACM.

[77] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Learning to detect malicious urls. *ACM Trans. Intell. Syst. Technol.*, 2(3):30:1–30:24, May 2011.

[78] Z. Ma, P. K. Rana, J. Taghia, M. Flierl, and A. Leijon. Bayesian estimation of dirichlet mixture model with variational inference. *Pattern Recognition*, 47(9):3143 – 3157, 2014.

[79] J. Martinez-Romo and L. Araujo. Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Systems with Applications*, 40(8):2992 – 3000, 2013.

[80] Z. Miller, B. Dickinson, W. Deitrick, W. Hu, and A. H. Wang. Twitter spammer detection using data stream clustering. *Inf. Sci.*, 260:64–73, Mar. 2014.

[81] F. Ming, F. Wong, and P. Marbach. Who are your friends? a simple mechanism that achieves perfect network formation. In *INFOCOM, 2011 Proceedings IEEE*, pages 566–570, 2011.

[82] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 29–42, New York, NY, USA, 2007. ACM.

[83] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: inferring user profiles in online social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 251–260, New York, NY, USA, 2010. ACM.

[84] A. MITCHELL, J. KILEY, J. GOTTFRIED, and E. GUSKIN. The role of news on facebook. October 2013.

[85] S. Nagaraja, A. Houmansadr, P. Piyawongwisal, V. Singh, P. Agarwal, and N. Borisov. Stegobot: a covert social network botnet. In *Proceedings of the 13th international conference on Information hiding*, IH'11, pages 299–313, Berlin, Heidelberg, 2011. Springer-Verlag.

[86] F. Nagle and L. Singh. Can friends be trusted? exploring privacy in online social networks. In *Social Network Analysis and Mining, 2009. ASONAM '09. International Conference on Advances in*, pages 312–315, 2009.

[87] A. Nazir, S. Raza, and C.-N. Chuah. Unveiling facebook: a measurement study of social network based applications. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, IMC '08, pages 43–56, New York, NY, USA, 2008. ACM.

[88] J. Oliver, C. Ke, P. Pajares, C. Chen, and Y. Xiang. An in-depth analysis of abuse on twitter. In *the 24th Virus Bulletin International Conference*, Seattle, WA, USA, 24-26 September 2014 2014.

[89] J. Oliver, P. Pajares, C. Ke, C. Chen, and Y. Xiang. An in-depth analysis of abuse on twitter. Technical report, Trend Micro, 225 E. John Carpenter Freeway, Suite 1500 Irving, Texas 75062 U.S.A., September 2014.

[90] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff. Overview of the trec-2011 microblog track. In *Proceeddings of the 20th Text REtrieval Conference (TREC 2011)*, 2011.

[91] P. Pajares. Does the twitter follower scam actually work? *Trend Micro*, March 2014.

[92] G. Pally, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814 –818, 2005.

[93] M. B. Prince, B. M. Dahl, L. Holloway, A. M. Keller, and E. Langheinrich. Understanding how spammers steal your e-mail address: An analysis of the first six months of data from project honey pot. In *CEAS*, 2005.

[94] M. S. Rahman, T.-K. Huang, H. V. Madhyastha, and M. Faloutsos. Efficient and scalable socware detection in online social networks. In *Proceedings of the 21st USENIX conference on Security symposium*, Security'12, pages 32–32, Berkeley, CA, USA, 2012. USENIX Association.

[95] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766. ACM, 2007.

[96] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonalves, A. Flammini, and F. Menczer. Detecting and tracking political abuse in social media. In L. A. Adamic, R. A. Baeza-Yates, and S. Counts, editors, *ICWSM*. The AAAI Press, 2011.

[97] D. M. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex

contagion on twitter. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 695–704, New York, NY, USA, 2011. ACM.

[98] X. Ruan, Z. Wu, H. Wang, and S. Jajodia. Profiling online social behaviors for compromised account detection. *IEEE Transactions on Information Forensics and Security*, 11(1):176–187, Jan 2016.

[99] N. Rubens, D. Kaplan, and M. Sugiyama. Active learning in recommender systems. In *Recommender Systems Handbook*, pages 735–767. Springer, 2011.

[100] M. Sachan, D. Contractor, T. A. Faruquie, and L. V. Subramaniam. Using content and interactions for discovering communities in social networks. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 331–340, New York, NY, USA, 2012. ACM.

[101] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM.

[102] R. Sebastião and J. a. Gama. Change detection in learning histograms from data streams. In *Proceedings of the Aritficial Intelligence 13th Portuguese Conference on Progress in Artificial Intelligence*, EPIA'07, pages 112–123, Berlin, Heidelberg, 2007. Springer-Verlag.

[103] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[104] J. Song, S. Lee, and J. Kim. Spam filtering in twitter using sender-receiver relationship. In *Proceedings of the 14th international conference on Recent Advances in Intrusion Detection*, RAID'11, pages 301–317, Berlin, Heidelberg, 2011. Springer-Verlag.

[105] J. Song, S. Lee, and J. Kim. I know the shortened urls you clicked on twitter: inference attack using public click analytics and twitter metadata. In *Proceedings of the 22nd international conference on World Wide Web*, WWW '13, pages 1191–1200, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[106] L. Spitzner. The honeynet project: trapping the hackers. *Security Privacy, IEEE*, 1(2):15–23, 2003.

[107] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 1–9, New York, NY, USA, 2010. ACM.

[108] C.-T. Su, W.-K. Tsao, W.-R. Chu, and M.-R. Liao. Mining web browsing log by using relaxed biclique enumeration algorithm in mapreduce. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM*

International Conferences on, volume 3, pages 54–58, Dec 2012.

[109] J. Sun, X. Zhu, and Y. Fang. A privacy-preserving scheme for online social networks with efficient revocation. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, 2010.

[110] E. Tan, L. Guo, X. Zhang, and Y. Zhao. Unik: Unsupervised social network spam detection. In *Proceedings of 22nd ACM International Conference on Information and Knowledge Management*, San Fransisco, USA, October 2013.

[111] K. Thomas. *The Role of the Underground Economy in Social Network Spam and Abuse*. PhD thesis, EECS Department, University of California, Berkeley, Dec 2013.

[112] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and evaluation of a real-time url spam filtering service. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 447–462, Washington, DC, USA, 2011. IEEE Computer Society.

[113] K. Thomas, C. Grier, and V. Paxson. Adapting social spam infrastructure for political censorship. In *Proceedings of the 5th USENIX conference on Large-Scale Exploits and Emergent Threats*, LEET'12, pages 13–13, Berkeley, CA, USA, 2012. USENIX Association.

[114] K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect: an analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, IMC '11, pages 243–258, New York, NY, USA, 2011. ACM.

[115] K. Thomas and D. Nicol. The koobface botnet and the rise of social malware. In *Malicious and Unwanted Software (MALWARE), 2010 5th International Conference on*, pages 63–70, 2010.

[116] Twitter. Tweet structure.

[117] J. Ugander, L. Backstrom, and J. Kleinberg. Subgraph frequencies: mapping the empirical and extremal geography of large graph collections. In *Proceedings of the 22nd international conference on World Wide Web*, WWW '13, pages 1307–1318, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[118] J. Ugander, B. Karrer, L. Backstrom, and J. Kleinberg. Graph cluster randomization: network exposure to multiple universes. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '13, pages 329–337, New York, NY, USA, 2013. ACM.

[119] B. Viswanath, M. A. Bashir, M. Crovella, S. Guha, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. Towards detecting anomalous user behavior in online

social networks. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 223–238, San Diego, CA, Aug. 2014. USENIX Association.

[120] B. Viswanath, E. Kiciman, and S. Saroiu. Keeping information safe from social networking apps. In *Proceedings of the 2012 ACM workshop on Workshop on online social networks*, WOSN '12, pages 49–54, New York, NY, USA, 2012. ACM.

[121] A. H. Wang. Don't follow me: Spam detection in twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–10, 2010.

[122] B. Wang, A. Zubiaga, M. Liakata, and R. Procter. Making the most of tweet-inherent features for social spam detection on twitter. *CoRR*, abs/1503.07405, 2015.

[123] D. Wang and C. Pu. Bean: A behavior analysis approach of url spam filtering in twitter. In *Information Reuse and Integration (IRI), 2015 IEEE International Conference on*, pages 403–410, Aug 2015.

[124] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao. You are how you click: clickstream analysis for sybil detection. In *Proceedings of the 22nd USENIX conference on Security*, SEC'13, pages 241–256, Berkeley, CA, USA, 2013. USENIX Association.

[125] G. Wang, M. Mohanlal, C. Wilson, X. Wang, M. J. Metzger, H. Zheng, and B. Y. Zhao. Social turing tests: Crowdsourcing sybil detection. *CoRR*, abs/1205.3856, 2012.

[126] G. Wang, C. Wilson, X. Zhao, Y. Zhu, M. Mohanlal, H. Zheng, and B. Y. Zhao. Serf and turf: crowdturfing for fun and profit. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 679–688, New York, NY, USA, 2012. ACM.

[127] N. Wang, S. Parthasarathy, K.-L. Tan, and A. K. H. Tung. Csv: visualizing and mining cohesive subgraphs. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 445–458, New York, NY, USA, 2008. ACM.

[128] S. Webb, J. Caverlee, and C. Pu. Introducing the webb spam corpus: Using email spam to identify web spam automatically. In *CEAS*, 2006.

[129] W. Wei, K. Joseph, H. Liu, and K. M. Carley. Exploring characteristics of suspended users and network stability on twitter. *Social Network Analysis and Mining*, 6(1):1–18, 2016.

[130] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *Proceedings of the 4th ACM*

European conference on Computer systems, EuroSys '09, pages 205–218, New York, NY, USA, 2009. ACM.

[131] F. Wu, J. Shu, Y. Huang, and Z. Yuan. Social spammer and spam message co-detection in microblogging with social context regularization. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1601–1610, New York, NY, USA, 2015. ACM.

[132] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts. Who says what to whom on twitter. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 705–714, New York, NY, USA, 2011. ACM.

[133] C. Xiao, D. M. Freeman, and T. Hwa. Detecting clusters of fake accounts in online social networks. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*, AISec '15, pages 91–101, New York, NY, USA, 2015. ACM.

[134] H. Xu, W. Sun, and A. Javaid. Efficient spam detection across online social networks. In *2016 IEEE International Conference on Big Data Analysis (ICBDA)*, pages 1–6, March 2016.

[135] C. Yang, R. Harkreader, and G. Gu. Empirical evaluation and new design for fighting evolving twitter spammers. *Information Forensics and Security, IEEE Transactions on*, 8(8):1280–1293, 2013.

[136] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu. Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 71–80, New York, NY, USA, 2012. ACM.

[137] C. Yang, R. C. Harkreader, and G. Gu. Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In *Proceedings of the 14th international conference on Recent Advances in Intrusion Detection*, RAID'11, pages 318–337, Berlin, Heidelberg, 2011. Springer-Verlag.

[138] X. Yang, Y. Guo, and Y. Liu. Bayesian-inference based recommendation in online social networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 551–555, 2011.

[139] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai. Uncovering social network sybils in the wild. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, IMC '11, pages 259–268, New York, NY, USA, 2011. ACM.

[140] S. Yardi, D. Romero, G. Schoenebeck, and D. Boyd. Detecting spam in a twitter network. *First Monday*, 15(1-4), January 2010.

[141] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and Y. Xiang. Internet traffic classification by aggregating correlated naive bayes predictions. *Information Forensics*

and Security, IEEE Transactions on, 8(1):5–15, Jan 2013.

[142] X. Zhang, Z. Li, S. Zhu, and W. Liang. Detecting spam and promoting campaigns in twitter. *ACM Trans. Web*, 10(1):4:1–4:28, Feb. 2016.

[143] X. Zhang, S. Zhu, and W. Liang. Detecting spam and promoting campaigns in the twitter social network. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1194–1199, 2012.

[144] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do we need more training data or better models for object detection?. In *BMVC*, volume 3, page 5. Citeseer, 2012.

[145] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes. Active learning with drifting streaming data. *IEEE transactions on neural networks and learning systems*, 25(1):27–39, 2014.