Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Twitter spam detection: Survey of new approaches and comparative study

Tingmin Wu [a], Sheng Wen [a,*], Yang Xiang [a], Wanlei Zhou [b]

[a] Swinburne University of Technology, Hawthorn VIC 3122, Australia
[b] School of Information Technology, Deakin University, VIC, Australia

ARTICLE INFO

Article history:
Available online 24 November 2017

Keywords:
Twitter
Spam detection
Machine learning
Social media
Security

ABSTRACT

Twitter spam has long been a critical but difficult problem to be addressed. So far, researchers have proposed many detection and defence methods in order to protect Twitter users from spamming activities. Particularly in the last three years, many innovative methods have been developed, which have greatly improved the detection accuracy and efficiency compared to those which were proposed three years ago. Therefore, we are motivated to work out a new survey about Twitter spam detection techniques. This survey includes three parts: 1) A literature review on the state-of-art: this part provides detailed analysis (*e.g.* taxonomies and biases on feature selection) and discussion (*e.g.* pros and cons on each typical method); 2) Comparative studies: we will compare the performance of various typical methods on a universal testbed (*i.e.* same datasets and ground truths) to provide a quantitative understanding of current methods; 3) Open issues: the final part is to summarise the unsolved challenges in current Twitter spam detection techniques. Solutions to these open issues are of great significance to both academia and industries. Readers of this survey may include those who do or do not have expertise in this area and those who are looking for deep understanding of this field in order to develop new methods.

1. Introduction

Online Social Networks (OSNs) are popular collaboration and communication tools for millions of Internet users. As a major social networking platform, Twitter attracts users by providing free microblogging services for customers to broadcast or discover messages within 140 characters, follow other users and so on, through different devices such as mobile phones and desktops (Chu et al., 2012). Everyday, millions of Twitter users share their moments or post their discoveries, such as breaking news to their followers (Ghosh et al., 2012). However, the openness and convenience of Twitter platform also attract criminal accounts (spammers), so as to attack the platform for the sake of making money illegitimately. These attacks include spam, scam, phishing (Adewole et al., 2017; Zhu et al., 2012). As there is a restriction on the length of tweets, it is common for spammers to broadcast unsolicited spam tweets, which can redirect users to external malicious websites (Lee and Kim, 2013). Compared to the traditional spam which spread through emails, Twitter spam is more dangerous and sophisticated in luring Internet users to get deceived (Thomas et al., 2011). According to a recent report (Grier et al., 2010), the click-through rate of Twitter spam reaches 0.13%, while it only achieves 0.0003% ~ 0.0006% in email spam.

In order to address the problem of Twitter spam, in the recent few years, there have been many detection schemes put forward. There are three main categories among current Twitter spam detection methods: detection based on Syntax Analysis, Feature Analysis and Blacklisting Techniques. As text is the

[*] *Corresponding author.*
E-mail address: swen.works@gmail.com (S. Wen).

https://doi.org/10.1016/j.cose.2017.11.013
0167-4048/© 2017 Elsevier Ltd. All rights reserved.

only format Twitter users can use, many researchers focus on analysing tweets semantics to detect spam (Chu et al., 2012; Gao et al., 2010; Hu et al., 2013, 2014; Lee et al., 2011; Lee and Kim, 2012, 2013; Thomas et al., 2011; Wang et al., 2013; Wu et al., 2017; Yang et al., 2012; Yardi et al., 2009; Zhang et al., 2012). More work was proposed using the features from both account and message aspects and applied a statistical method to them (i.e. Feature Analysis) (Ahmed and Abulaish, 2013; Benevenuto et al., 2010; Cao et al., 2012; Castillo et al., 2011; Chen et al., 2015; Chu et al., 2012; Costa et al., 2013; Egele et al., 2013; Gao et al., 2012; Ghosh et al., 2012; Grier et al., 2010; Hu et al., 2013, 2014; Jin et al., 2011; Lee et al., 2010; Lee and Kim, 2013; Liu et al., 2017; Sabottke et al., 2015; Sala et al., 2010; Song et al., 2011; Stringhini et al., 2010; Tan et al., 2013; Thomas et al., 2011; Wang, 2010; Yang et al., 2011, 2012, 2013, 2014; Zhang et al., 2012, 2016; Zhu et al., 2012). In addition, researchers also relied on third party services such as blacklisting technique to block malicious information (Ghosh et al., 2012; Gilani et al., 2017; Grier et al., 2010; Ma et al., 2009, 2011; Zhang et al., 2012).

Why do we need this survey? Currently, many efforts have been made in developing effective Twitter spam detection methods. Especially in the last three years, there were some innovative breakthrough techniques developed (Ahmed and Abulaish, 2013; Chen et al., 2015; Costa et al., 2013; Egele et al., 2013; Ghosh et al., 2013; Hu et al., 2013, 2014; Jiang et al., 2013; Lee and Kim, 2013; Liu et al., 2016; Oliver et al., 2014; Stringhini et al., 2013; Symantec, 2015; Tan et al., 2013; Wang et al., 2013; Yang et al., 2013, 2014; Zhang et al., 2016). The improvements of these newly developed methods covered almost all research issues in Twitter spam detection field, such as selection mechanisms of spam features, sampling techniques, detection engines with better accuracy and so on. Therefore, it is necessary to provide a survey organising both past and new methods in detecting spam for future research. In this survey, we will also run comparative studies among different detection techniques.

How to use this survey? The survey contains three parts in terms of usage. Firstly, we collect and list existing related techniques for a literature review. This part will employ a taxonomy and a series of analysis to divide the state-of-art into many categories. Readers will obtain the details of each type of methods and their pros and cons in Twitter spam detection. We further select typical methods in each branch of this research field and carry out comparative studies among all kinds of methods. This part will provide readers numerical descriptions on current methods, especially on their advantages and weakness under different scenarios. Finally, we summarise related work analysis and comparative studies, and come up with open issues and potential solutions as well. The summary contributes to future research as it presents several subsequent research areas.

The rest of this survey is organised as follows. Section 2 illustrates the taxonomy of the state-of-art, followed by Section 3, 4 and 5 which show the analysis of existing Twitter spam detection methods based on syntax, feature analysis and blacklisting techniques respectively. Section 6 shows comparatives studies of above techniques. The summary and open issues are discussed in Section 7. Finally, we conclude this survey in Section 8.

2. Taxonomy

It is challenging to categorise current technologies into several parts. Because most methods may borrow the ideas from many technologies that cover different categories in Twitter spam detection, it is difficult to divide the state-of-art and make sure each work exclusively drop into one category. Therefore, to be specific, we build up the taxonomy according to extraction and classification methods. Although this taxonomy creation method cannot avoid the case of one method dropping into multiple categories, our taxonomy will well present the recent advances in Twitter spam detection in terms of feature extraction.

What does the taxonomy tell us? We categorise the state-of-art mainly by their feature selection methods. The taxonomy is shown in Fig. 1. Based on the difference from feature

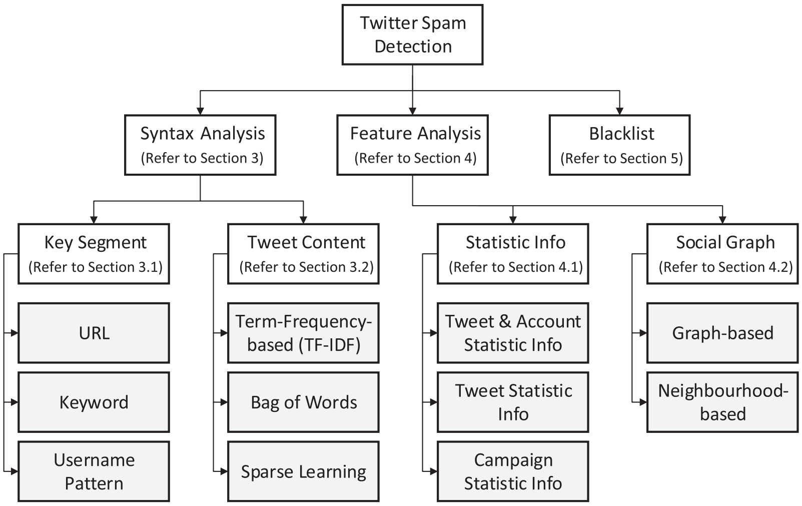
Twitter Spam Detection
Syntax Analysis (Refer to Section 3)
Feature Analysis (Refer to Section 4)
Blacklist (Refer to Section 5)
Key Segment (Refer to Section 3.1)
Tweet Content (Refer to Section 3.2)
Statistic Info (Refer to Section 4.1)
Social Graph (Refer to Section 4.2)
URL
Keyword
Username Pattern
Term-Frequency-based (TF-IDF)
Bag of Words
Sparse Learning
Tweet & Account Statistic Info
Tweet Statistic Info
Campaign Statistic Info
Graph-based
Neighbourhood-based

Fig. 1 – The taxonomy of Twitter spam detection methods. There are three main categories for current techniques. Each sub-category includes some typical methods, and we will explain them individually.

processing, we obtain three major categories: detection based on 1) syntax analysis, 2) feature analysis and 3) blacklist. Each category of detection methods relies on a specific group of features and selection methods. For the first category (*i.e.* syntax analysis), the features are mainly extracted from key segments embedded in tweets or by means which indicates the content of tweets such as bag-of-words. For the second category (*i.e.* feature analysis), the features are normally collected by statistics of tweets and/or Twitter accounts, or from the topological information of social graphs. The last category (*i.e.* blacklist) represents the simplest group of detection methods which are now commonly used in industries. This method is based

on detecting the malicious URLs from blacklist embedded in the tweets.

Why do we build up taxonomy by feature selection? Traditional taxonomies in Twitter spam detection are constructed by anti-spam strategies (Heymann et al., 2007) or types of machine learning techniques (i.e. supervised, unsupervised, or semi-supervised) (Verma and Sofat, 2014). In this survey, we build up the taxonomy based on the categories of feature selection. According to our literature review (see details in Section 3 and 4), the majority of Twitter spam detection methods rely on machine learning based techniques. However, the essential differences of those methods are mainly from the features and their selection strategies instead of machine learning algorithms. In fact, almost every paper in this field introduces a group of distinct features and then with another group of well-known machine learning based classifiers to detect spamming activities in Twitter (*e.g* (Chen et al., 2015; Hu et al., 2014; Yang et al., 2014; Zhang et al., 2016).). Our taxonomy will provide a brand-new view of angle to explore the core challenges as well as solutions in Twitter spam detection. To the best of our knowledge, this taxonomy accurately categorises various detection methods in this field.

3. Literature review part I: detection based on syntax analysis

In this section, we will analyse the detection methods based on syntax analysis and discuss their pros and cons. These methods can be categorised into two parts: 1) key segment and 2) tweet content. Before we go for the details, we briefly analyse the generic Twitter spam detection framework. As shown in Fig. 2, Twitter posts will first be collected by data collection module. Suspicious tweets will be selected by a set of basic rules and then used for feature extraction. For supervised classification engine, the resultant feature vectors will first be fed to the training module. But for unsupervised classification engine, the feature vectors will be directly fed to the engine to distinguish spam out of benign tweets. Only the benign tweets can reach the real systems. Most existing detection methods follow this framework.

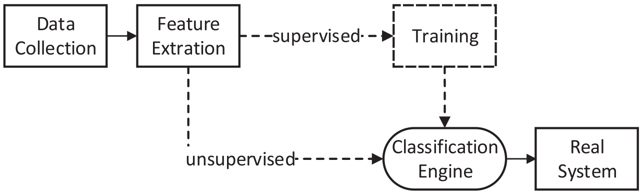

Fig. 2 – Generic Twitter spam detection framework. The data will be collected first and then the features are extracted and fed to the training and classification modules.

3.1. *Key segment*

First of all, we discuss the key segment methods. These methods collect indicative segments such as keywords, username patterns and URLs, to represent the context of tweets and posters. The motivation of these methods is that real-world spammers usually make use of sensitive information with deceptive URLs to attract Twitter users. Therefore, a tweet which contains sensitive keywords, phrases or suspicious URLs is more likely to be spam (Chen et al., 2017; Yang et al., 2012).

3.1.1. URL

The most popular key segments are about URLs. The related works that drop in this category are listed as follows: Chu et al. (2010, 2012), Gao et al. (2010), Grier et al. (2010), Klien and Strohmaier (2012), Lee and Kim (2012), Ma et al. (2009, 2011), McGrath and Gupta (2008), Stringhini et al. (2010), Thomas et al. (2011), Whittaker et al., and Zhang et al. (2012). In particular, the works Lee and Kim (2013) and Zhang et al. (2016) are published in the last three years.

Detailed Methods: We continue to elaborate the details of URL based detection methods as follows:

First, many URL based detection methods focus on shortened URLs. Since Twitter limits the number of characters in every tweet, URLs which are usually composed of many characters will be pre-processed by shortening services in order to accommodate more descriptive words in tweets (Chu et al., 2010; Grier et al., 2010; Klien and Strohmaier, 2012; McGrath and Gupta, 2008; Stringhini et al., 2010). Shortened URLs may obfuscate the direct lexical meanings and hide malicious links. Therefore, shortened URLs are considered as major key segments in terms of URLs. For example, to prevent the abuse of shortening services such as *bit.ly*, Thomas et al. developed a shortener identifier to detect spammers (Thomas et al., 2011). They calculated respectively the ratios of spam and non-spam tweets that have used shortening services in their descriptions. The division product of the two ratios will be regarded as the discriminant to identify spam tweets. If the product is greater than one, the usage of shortening services most probably indicates a spamming activity. Meanwhile, shorteners are ranked according to their popularity in the real world. Thomas et al. combined the reputation and frequency of each URL and domain to find the differences between spammers and benign accounts (*e.g. bit.ly* accounted for the highest share of usage as 34.86%). Because most methods extract features from shortened URLs, we will explain the usage of shortened URLs together with other measurements in the following two parts.

Second, there are also many methods that use the duplication of URLs to classify tweets or accounts. The tweets or accounts will be clustered into campaigns according to the shared URLs.

Spamming activities will be exposed based on campaigns. For example, Chu et al. and Gao et al. proposed methods to cluster correlated tweets into campaigns by both textual description and shortened URLs in each tweet (Chu et al., 2012; Gao et al., 2010). Zhang et al. developed a scheme to link accounts who post similar tweets according to their shared URLs (Zhang et al., 2012, 2016). After various Twitter campaigns are exposed by using URL based methods, traditional machine learning algorithms can further be employed to distinguish spamming campaigns from regular campaigns. These traditional algorithms usually adopt statistic features as well (refer to Section 4) (Chu et al., 2012; Zhang et al., 2012, 2016). To differentiate tweets into various campaigns, Chu et al. presented each tweet as a pair of text and URL parts (Chu et al., 2012). A tweet campaign was then represented as $c_i = \langle u_i, T_i, A_i \rangle$, where u_i denoted the *ith* shared final URL (*i.e.* the hidden URL reverse-engineered from redirection chains), T_i was the set of tweets that included u_i, and A_i denoted the accounts that were associated with T_i. Given a campaign C, for an arbitrary tweet, if the c_i calculated by its u_i belonged to C, the tweet was added into C. To cluster accounts into campaigns, Zhang et al. applied Shannon information theory and computed the entropy through URLs embedded in the tweets (Zhang et al., 2012, 2016). The similarity could be measured by the entropy shared among different accounts as well as the intervals between tweets. The method then ranked the relationships the relationships among Twitter accounts according to the values of similarities. On the basis of it, the connection graph was constructed by selecting those which had large values of similarities (*i.e.* > *threshold*). Every isolated subgraph indicated a campaign for a specific purpose.

Third, features distilled from URLs are also commonly used in spam detection. Traditionally, the features include lexical attributes, page content and domain hosting properties (Ma et al., 2009, 2011; Whittaker et al., 2010). Recent works also adopted the features which can help identify malicious URLs, such as the features related to IP addresses (Lee and Kim, 2012, 2013; Thomas et al., 2011). Their empirical analysis suggested that the newly adopted URL features were more correlated with spam activities. Generally speaking, the third type of URL based methods is used for online spam detection due to its fast classification speed (Ma et al., 2011). Note that, although we divide the URL based methods into the above three parts, most of current methods mixed the techniques in the three parts together. For example, almost all URL based methods included analysis of shortened URLs (Lee and Kim, 2012, 2013; Ma et al., 2009, 2011; Thomas et al., 2011; Whittaker et al., 2010). In this case, there are several methods that rely on features of malicious URLs themselves (Lee and Kim, 2012, 2013; Thomas et al., 2011). For example, Thomas et al. collected and extracted features from three aspects: 1) web browser (*e.g.* final URL and redirects), 2) DNS resolver (*e.g.* DNS), and 3) IP address analysis (*e.g.* geolocation) (Thomas et al., 2011). The resultant raw features will be further transformed and organised in vectors. They also introduced and combined the logistic regression method (LR) (Friedman et al., 2001) and the stochastic gradient descent method to implement to train the classifier. Lately, Lee and Kim proposed using the correlations of URL directed chains as a new URL feature (Lee and Kim, 2012, 2013). In addition to this new feature, a series of traditional URL features (Thomas et al., 2011) and tweet content features (Ahmed and Abulaish, 2013; Chen et al., 2015; Chu et al., 2012) were also used through a feature normalization process (Lee and Kim, 2013). All the features worked together and were tested on seven LIBLINEAR classifiers (Fan et al., 2008).

Performance: In this paragraph, we focus on the performance of the above URL based methods. *First*, Thomas et al. studied the likelihood ratio of shorteners (Thomas et al., 2011). According to their analysis based on shortened URLs, they found 77% of spam accounts identified by Twitter are suspended within one day of their first tweet. They also identified five spam campaigns which controlled 145 thousand accounts. Because each campaign enacted a unique spamming strategy, the spamming activities were able to persist for months at a time. Recall that URL based spam detection complies with the framework in Fig. 2. *Second*, a typical example of spamming campaign detection was proposed by Chu et al. (2012). They proposed organising features from tweets, accounts and campaigns (*e.g.* URL redirections, account reputation, and posting device makeups, respectively). Most of these features were collected from Twitter's Streaming API (Twitter Developers, 2016). According to their experiments, they found the Random Forest classifier had the best performance with an Accuracy 94.5%. *Third*, the works Lee and Kim (2012, 2013) and Thomas et al. (2011) proposed using new URL features for spam detection. Their experiment results suggested that the new features largely improved the detection accuracy. For example, Lee and Kim's work achieved 0.9027 AUC (Area Under Curve) and 91.71% Accuracy by selecting an L2-regularized L1-loss support vector classification (SVC) algorithm from seven classification algorithms (Lee and Kim, 2013).

Pros and cons: We discuss the pros and cons of URL based methods below. *We first come to the advantages*. Generally speaking, due to the length limit of tweet description, spammers are more likely to post URLs than plain text to spread malicious information. Therefore, URL based methods are implicitly tailored to examine tweets which are abundant of URLs especially on criminal accounts. Moreover, current URL based methods have overcome the inaccuracy problem caused by URL shortening services. Hidden clues of spamming activities will be exposed and replayed by reverse engineering the shortened URLs into original URLs. *There are also three weaknesses in current URL based methods*. The first weakness is about relatively high expenditure in practices. According to a recent study, the filtering cost might reach $22,751/month for 15.3 million URLs/day (Thomas et al., 2011). Second, regarding redirections in tweets, only HTTP redirection can be processed by current URL based methods. The second weakness largely narrows down the real-world usability of these methods. Finally, although URL based methods are able to run fast (*e.g.* processing 576,000 URLs/hour) (Lee and Kim, 2013), this type of methods still cannot detect malicious URLs with "conditional" behaviours (refer to Lee and Kim (2013)).

3.1.2. *Keyword/username pattern*

Keywords or username patterns can also be used to detect Twitter spam. The idea sounds very intuitive and straightforward. However, to the best of our knowledge, there are only two papers that drop in this category (Gao et al., 2010; Yardi et al., 2009).

Detailed Methods: In this paragraph, we analyse and explain the details of the method that relies on keywords and username patterns. Yardi et al. developed a spam detection algorithm by detecting keywords and matching username pattern in tweets (Yardi et al., 2009). The assumption of this method is that the username pattern "letter + number" was highly correlated with spamming accounts. Spam tweets usually referred to unsolicited messages through manipulating fake accounts automatically in conventional patterns. Meanwhile, tweets that contained sensitive keywords such as "naked" were more likely to be spam. This method has also been applied to Facebook (Gao et al., 2010). Searching function by keywords in Twitter and Facebook was employed in this method to facilitate the identification of suspicious URLs such as the keyword "click here". In practice, the keywords and username patterns will work together with shortened URLs to help distinguish regular posts from spam activities (Yardi et al., 2009).

Performance: Yardi et al. first studied #robotpickuplines tweets dataset and extracted the attributes from key segments (Yardi et al., 2009). By matching these specific segments to the sensitive keywords and username patterns, the method could achieve an Accuracy of 91%. In addition, keyword searching functions also helped detect obfuscated URLs in tweets, the performance of this method can be further improved by combining with other techniques such as removing illegal characters from the description of tweets (Gao et al., 2010).

Pros and cons: We further discuss the pros and cons of keywords and username patterns in Twitter spam detection. Because spammers usually introduce sensitive words such as "hot topics" to attract victims, the methods that rely on keywords and username patterns will efficiently capture the spam tweets (refer to Yardi et al., 2009). This type of the methods has two disadvantages. First, keywords and username patterns are actually a double-blades sword. For example, the work of Yardi et al., 2009 only investigated #robotpickuplines related datasets. Therefore, the collected properties of tweets were too less to detect spamming activities. This might significantly limit the application of this method. Second, this method cannot counter sophisticated spamming activities. By using social engineering methods, spamming activities can easily avoid the usage of popular keywords and username patterns.

3.2. *Tweet content*

We second discuss the spam detection methods that focus on tweet content. Because spammers usually compose the spam content with similar malicious topics or words, a tweet that contains those malicious topics and words are more likely to be spam. There are currently three major techniques to represent textual content of tweets: TF-IDF (Term Frequency – Inverse Document Frequency) (Aizawa, 2000), bag-of-words, and sparse learning. In the following, we will discuss the details of tweet content based methods according to the three representation techniques.

3.2.1. *TF-IDF*

TF-IDF is the most popular technique to extract the meaning of tweets. The related works that drop in this category are listed as follows: Chu et al. (2012), Lee et al. (2011), and Yang et al. (2012). In particular, the works of Hu et al. (2014) are published in the last three years.

Detailed Methods: TF-IDF has been widely used as a metric in text retrieval to obtain the representation terms and weights of contextual words (Salton and Buckley, 1988). Based on this technique, Yang et al. designed a metric to measure the tweet semantic correlation between each pair of accounts (Yang et al., 2012). Chu et al. also applied it to check the similarity among content syntaxes of tweets by the Vector Space Model (Salton et al., 1975) which could transfer tweets into vectors (Chu et al., 2012). While the content similarity of legitimate campaigns was significantly stronger than spamming campaigns, these vectors were capable of distinguishing spam from non-spam (Chu et al., 2012).

The detection based on TF-IDF technique will first identify duplicated tweets in a campaign. The Twitter campaigns will then be classified into spam campaigns and non-spam campaigns. For example, the words in the training tweets were calculated by TF-IDF technique and converted to vectors using specific techniques such as the Vector Space Model (Salton et al., 1975) (Chu et al., 2012; Yang et al., 2012). The similarity between an arbitrary pair of two tweets was obtained from the cosine of the two corresponding vectors. Along with other typical features, they were treated as the input of the classifiers for experiments.

Performance: We discuss the performance of the methods that adopted TF-IDF technique. Chu et al. compared eight conventional machine learning algorithms, and found that Random Forest outperformed other methods such as Decision Tree by using Cross Validation (Chu et al., 2012). To compare the performance of various features, the content similarity obtained by TF-IDF technique could be ranked as the 9th of the top ten features, which contributed 72.3% Accuracy to Random Forest classifier (the highest one was 85.6%) (Chu et al., 2012). Moreover, Yang et al. combined the tweet content and social relationship, and calculated the malicious scores by the content similarity between an arbitrary account and its followings (Yang et al., 2012). Their proposed method "CIA" outperformed other methods as it could infer 13 more unknown spammers than RAND giving 10 identified criminal accounts (Yang et al., 2012).

Pros and cons: For TF-IDF mehtods, we mainly focus on their weaknesses. Although the similarity was an influential feature to improve the accuracy of the spam detection, it might be replaced or modified as spammers have become more and more sophisticated deceptive.

3.2.2. *Bag-of-words*

The bag-of-words method was a typical text representation and was widely used for pre-processing tweets before training a classifier. The related works that drop in this category are listed as follows: Chu et al. (2012), and Lee et al. (2010, 2011).

Detailed Methods: The bag-of-words method employed previous TF-IDF technique as the weighted algorithm to realise the vector representation (text-based features) (Lee et al., 2010, 2011). Meanwhile, this model has also been widely adopted in Bayesian algorithm to pick up specific words from paragraphs (Chu et al., 2012). The bag-of-words method can be used to measure statistical features (*e.g.* the content similarity) or build the text classifier directly. The features extracted by the bag-of-words methods are usually text-based. By removing punctuation, lowercasing all the characters in a tweet and tokenizing each word using the bag-of-words method, the terms was weighted by TF-IDF and converted into text-based features (Lee et al., 2010). These features are also integrated with other types of features from account and tweet level and work together for Twitter spam detection. The feature "content similarity" was calculated by the standard cosine distance between two targeted bag-of-words vectors (Lee et al., 2010). In addition, for building the text classifier, when the probability of a message belonged to a class was higher than a certain threshold, then the assumption was true (Chu et al., 2012). The message was represented by a feature vector and each dimension represented one or more words as bag-of-words (Chu et al., 2012).

Performance: The bag-of-words method was effective to construct text-based features such as tweets similarity (Lee et al., 2010, 2011). Same as TF-IDF, the bag-of-words method also assumed that the content similarity of spam tweets was dramatically higher than benign tweets because spammers usually posted duplicate tweets (Lee et al., 2010). Combined with other traditional features, the Weka classifier "Decorate" achieved the best Accuracy as high as 88.98% (Lee et al., 2010). The bag-of-words method can also be used to implement a more powerful Bayesian text classifier like CRM114 (Yerazunis, 2009).

Pros and cons: Bag-of-words is the most typical method in email spam detection (Blanzieri and Bryl, 2008). The idea is simple and easy to be implemented. However, in practice, bag-of-words can only be one indicator of Twitter spam detection. Real-world methods will combine bag-of-words as a typical feature with other features in Twitter accounts and their social relationships (Chu et al., 2012). The methods that only rely on bag-of-words may not achieve the performance as good as other methods which employ features from other aspects in Twitter. For example, Bayesian classifier has been widely used in bag-of-words methods to detect spam (Chu et al., 2012). In this category, the input of Bayesian classifier only includes the description of tweets without account information of the posters. This considerably influences the performance of spam detection (Yerazunis, 2009).

3.2.3. Sparse learning

Sparse learning is the last one in tweet content based methods. As the traditional text presentation methods such as bag-of-words and n-gram (Pak and Paroubek, 2010) may lead to high dimension of feature vectors, Hu et al. proposed a sparse representation for the key phrases or words instead of all the sentences (Hu et al., 2013, 2014). In particular, the works of Hu et al. (2013, 2014) are published in the last three years.

Detailed Methods: This method applied a non-negative matrix factorization model (NMF) (Lee and Seung, 1999) to represent the tweets by a lower-dimension feature vector (Hu et al., 2014). An optimization algorithm was then employed to transfer the vector from the text level to the topic level. With shrunk length of vectors, the features are more representative and clustered to identify spam activities. Specifically, as there are always a few topics for an arbitrary tweet, this method can be "sparse" and more accurate.

Performance: Compared to four batch-mode leaning methods, the sparse learning method performed best in a five-fold cross-validation (Hu et al., 2014). This method performed about 8% better than LS_Content (Lawson and Hanson, 1995) (Hu et al., 2014).

Pros and cons: These methods in this category combine both network and tweet content to detect spammers. It is slightly faster than so-called batch-mode learning algorithm (*note*: as a comparison target only discussed in Hu et al. (2014)). However, if the user factors are considered, the detection accuracy will be greatly improved.

4. Literature review part II: detection based on feature analysis

In this section, we analyse feature analysis based detection methods and discuss their pros and cons. We divide this category into two sub-categories: detection approaches by statistic information and by social graph.

4.1. Statistic information

To begin with, we discuss the statistic information methods. These methods collect statistic features from user profiles and tweets. The detection methods based on statistic information comply with the framework in Fig. 2. This sub-category is further divided into three parts according to the places where we extract statistic information: 1) tweet statistic information, 2) account statistic information and 3) campaign statistic information.

4.1.1. Tweet statistic information

For the methods that are based on tweet statistic information, most works combined features from other levels such as account Ahmed and Abulaish (2013), Benevenuto et al. (2010), Chen et al. (2015), Chu et al. (2012), Lee et al. (2010), Lee and Kim (2013), Liu et al. (2016); Song et al. (2011), Stringhini et al. (2010), Yang et al. (2011, 2013) and Zhang et al. (2012, 2016). There are also some studies that only applied text-based features to construct a binary classifier (Egele et al., 2013; Gao et al., 2012). This technique is prevalent in other social networks as well (Costa et al., 2013; Jin et al., 2011). In particular, the works of Ahmed and Abulaish (2013), Chen et al. (2015), Costa et al. (2013), Egele et al. (2013), Lee and Kim (2013), Liu et al. (2016), Yang et al. (2013) and Zhang et al. (2016) are published in the last three years.

Detailed Methods: Generally, spammers employed social engineering techniques to manipulate text description. Therefore, the text-based classification methods could be very useful from the empirical perspective. Table 1 has shown a series of representative tweet-based features. These features are feasible to differentiate the spam and benign tweets. For example, it was found that spammers tried to broadcast more messages disguised as legitimate users (Yang et al., 2011). Similarly, based on cumulative distribution function (CDF) analysis, it was reported that spam tweets usually contained more hashtags, URLs and spam words than normal messages (Benevenuto et al., 2010). In addition, spammers usually included more digits in their tweets compared to benign users (Chen et al., 2015). All these features are very useful and indicative in Twitter spam detection.

Table 1 – Typical tweet-based features.

Notation	Description
no_tweet	The number of tweets a user posts.
no_retweet	The number of retweets for a tweet.
no_hashtag (#)	The number of hashtags in a tweet.
no_usermention (@)	The number of usermentions in a tweet.
no_url	The number of URLs in a tweet.
fra_url_tweet	The fraction of tweets containing URLs.
no_char	The number of characters in a tweet.
no_digit	The number of digits in a tweet.

Table 2 – Typical account-based features.

Notation	Description
account_age	The age of an account until the time point of sending the most recent messages.
no_follower	The number of follower of the Twitter user.
no_following	The number of following/friends of the Twitter user.
reputation	The ratio of following number to follower number for the Twitter user.

Performance: The usage of statistic features resulted in good performance in Twitter spam detection. Chao et al. set up the experiments on six machine learning algorithms and achieved the F-measure as high as 93.6% by Random Forest method (Chen et al., 2015). Wang developed a Naive Bayesian based method which achieved the best Precision of 89% (Wang, 2010). In addition, to explore the influence of the features, by applying χ^2 method (Chi Squared) (Yang and Pedersen, 1997), ten most significant features were ranked and it was found that the tweet related features played an important role (Benevenuto et al., 2010). For example, the fraction of tweets with URLs was ranked as the first, and the average number of URLs per tweet was ranked as the third.

Pros and cons: Many works applied content-based features to a series of machine learning classifiers, because they could represent tweets effectively. However, the contextual parts of tweets are usually manipulated by social engineering techniques, which can lead to the problem of feature fabrication (Chen et al., 2016). In addition, the differentiation ability of text-based features in different datasets varies from each other. For instance, many criminal accounts posted many duplicate tweets. But in some other datasets, there are some benign users who also post duplicate tweets (Wang, 2010). Moreover, malicious content can be changed by spammers easily. Correspondingly, the features that indicate the old spammers also need to be revised in order to catch up with the changes in Twitter spam.

4.1.2. *Account statistic information*

There are also some methods that focused on the statistic information of Twitter accounts (Ahmed and Abulaish, 2013; Benevenuto et al., 2010; Chen et al., 2015; Chu et al., 2012; Gao et al., 2012; Grier et al., 2010; Lee et al., 2010; Lee and Kim, 2013; Liu et al., 2016; Song et al., 2011; Stringhini et al., 2010; Thomas et al., 2011; Wang, 2010; Yang et al., 2011, 2013; Zhang et al., 2012, 2016). The account-based features were also utilised in other online social network platforms such as LSN (Costa et al., 2013) and Renren (Yang et al., 2014; Zhu et al., 2012). In particular, the works of Ahmed and Abulaish (2013), Chen et al. (2015), Lee and Kim (2013), Liu et al. (2016), Yang et al. (2013, 2014) and Zhang et al. (2016) are published in the last three years.

Detailed Methods: Several typical account-level features are shown in Table 2. The account-based features can also differentiate spam and non-spam effectively. For example, the number of benign accounts' followings and the number of followers were much larger than the ones in spammers, and the life cycle of spammers was also shorter compared to legitimate users (Chen et al., 2015). Specifically, the feature "reputation" had a different performance from other features. Generally, the reputation of a spammer was either 100% or very low, while the reputation of a normal user ranged from 30% to 90% (Wang, 2010). This will greatly help distinguish spam activities from normal behaviours.

Performance: Ahmed and Abulaish proposed three similar spam detection methods on Facebook and Twitter (Ahmed and Abulaish, 2013). The Twitter method achieved the best Detection Rate (DR) (98.7%) and lowest FPR (1.4%) by using the rule learner (Jrip) (Provost, 1999) (Ahmed and Abulaish, 2013). Benevenuto et al. employed a Support Vector Machine (SVM) classifier (Joachims, 1998) to detect both spam and spammers based on real-world datasets (Cha et al., 2012) (Benevenuto et al., 2010). The result showed that the Accuracy was similar: 87.2% towards spam detection and 87.6% towards spammer detection.

Pros and cons: The features based on account statistic information are more robust because they could hardly change in various spamming activities. These features are considered as the input of their classification methods. As "everything has an exception", there are some exceptions for the robust features. For instance, although criminal accounts usually do not have as many as followers of legitimate accounts, there are always some spammers who own a large number of followers (Wang, 2010). This is caused by their deceptive means in luring others to follow them. The exceptions will reduce the detection accuracy. In practice, this kind of features normally work with other features in Twitter spam detection such as tweet-based ones (see details in Section 4.1.1).

4.1.3. *Campaign statistic information*

For the methods based on campaign statistic information, we have the works of Chu et al. (2012), Gao et al. (2012), Grier et al. (2010), Stringhini et al. (2010) and Zhang et al. (2012). In particular, the work of Zhang et al. (2016) is published in the last three years.

Detailed Methods: Instead of detecting spam tweets one by one, many researchers clustered spam into a series of groups according to their similarity on tweet description or URLs (Chu et al., 2012; Gao et al., 2012; Grier et al., 2010; Zhang et al., 2012, 2016). Stringhini et al. also developed a similar detection method which targeted at spammers (Stringhini et al., 2010). Typical campaign-based features are listed in Table 3. To cluster similar spam or spammers, URL is usually considered as an important feature because similar URLs sometimes indicate same interests or topics (Chu et al., 2012; Gao et al., 2012; Grier et al., 2010; Stringhini et al., 2010; Zhang et al., 2012, 2016). Because

Table 3 – Typical campaign-based features.

Notation	Description
ratio_hashtag	The ratio of the number of hashtags in the tweets to the number of tweets in the campaign.
ratio_mention	The ratio of the number of mentions in the tweets to the number of tweets in the campaign.
score_content_similarity	The content self-similarity of the campaign.
device_posting	The ratios of manual and auto devices in the campaign.
density_following	The average number of the accounts' following in the campaign.
no_domain	The average number of distinct domains in the campaign.

some legitimate users also posted URLs, the interval between tweets can be added as a parameter to improve the similarity among spammers (Zhang et al., 2016). After the tweets or accounts are clustered, the campaign-level features are applied to classify the uncategorised groups by machine learning classifiers or manual identification. Considering the feature of content self-similarity score, as spammers may apply a text template and post similar content, the syntax similarity for criminal campaigns can be higher than benign campaigns (Chu et al., 2012).

Performance: Given a dataset as the ground-truth (McLachlan et al., 2005), Chu et al. utilised Random Forest algorithm as the classifier and achieved the Accuracy of 94.5% (Chu et al., 2012). Based on the SVM classifier, the clusters were first categorised by URLs and time interval of tweets and then classified with 87.6% F1-measure and 90.3% Precision (Zhang et al., 2016). In addition, it also ranked the campaign-based features using information gain (Kent, 1983) and χ^2 (Yang and Pedersen, 1997), and both showed the text similarity ranked the first (Zhang et al., 2016).

Pros and cons: Campaign-based methods are efficient but too error-prone. In the clustering pre-processing procedure, some tweets or accounts may be clustered into wrong groups. This recursively leads to more errors in the next classification process. Besides, some campaigns are classified manually (Thomas et al., 2011), it can be extremely time-consuming.

4.2. *Social graph*

In this subsection, we discuss the social graph based methods. These methods extract features from social graphs of Twitter users according to their following and follower relationships. We divide this sub-category into two parts: 1) graph-based methods which focus more on the macroscopic attributes of graph nodes, and 2) neighbourhood-based methods which concentrate on microscopic relationships of graph nodes.

4.2.1. *Graph-based*

There are many works that extract graph-based features to catch spammers such as social degree for a node (Gao et al., 2012; Ghosh et al., 2012; Hu et al., 2013, 2014; Song et al., 2011; Thomas et al., 2011; Wang, 2010; Yang et al., 2012, 2013). Graph-based methods are also adopted in spam detection in other social network platforms such as LSN (Cao et al., 2012; Costa et al., 2013; Sala et al., 2010). In particular, the works (Costa et al., 2013; Hu et al., 2013, 2014; Yang et al., 2013) are published in the last three years.

Detailed Methods: Graph-based features can be classified into two categories: social and structure features. The first category is similar to account-based features (refers to Section 4.1.2). Generally speaking, each node in a social graph represents an account. The indegree of a node (*i.e.* the number of nodes adjacent to the node) denotes the number of followers, and the outdegree refers to the number of following (Gao et al., 2012; Ghosh et al., 2012; Hu et al., 2013, 2014; Thomas et al., 2011; Wang, 2010). These graph-based features are used in machine learning based classifiers to detect spamming activities. The second category focuses on the structure of social graphs. For example, Song et al. measured the Distance and Connectivity (Song et al., 2011). Distance was defined as the shortest path between an arbitrary pair of users in the graph. It was reported that spam normally came from a specific user who was at least three hops from the victim. Moreover, Connectivity measured the intensity of the connection between a pair of neighbouring users. According to the Menger's theorem (Perfect, 1968), the connection was calculated by unique paths between two nodes. Yang et al. tested three features (*i.e.* Graph Density, Reciprocity and Average Shortest Path Length) (Yang et al., 2012). Yang et al. also used three more robust features: Local Clustering Coefficient (LCC), Betweenness Centrality (BC) and Bidirectional Links Ratio (BLR) (Yang et al., 2013). LCC is the ratio of the existing links among the vertices to the maximum links (Holme and Kim, 2002). This feature is used to produce the small social graph for the targeted account. BC measures the centrality of an account in a graph (Newman, 2005). If a node stands in the middle of many shortest paths, this node will hold a high BC value. The spammers can be identified according to the BC values because they normally attach to arbitrary users in the graph (Yang et al., 2013). BLC refers to the proportion of bidirectional links (*i.e.* two accounts follow each other) to all relationships of an account. Because Twitter spammers are trying to follow a large number of users with a few account following back, their bidirectional links ratio will be significantly lower than legitimate users, which can be used to identify spammers (Yang et al., 2013).

Performance: The social graph is an effective means in spam detection, and its related features are treated as useful influence factors to distinguish spam and non-spam. For example, considering each account as a node, the social relationships in Twitter can be transferred into a directed graph according to the following direction. It was found that the average length of shortest paths among spammers was shorter than benign accounts, and those criminal accounts accordingly constructed a small social network (Yang et al., 2012). Given a criminal account, probabilistically around 0.0625 criminal accounts can be correctly inferred from the social relationships (Yang et al., 2012). Using the graph-based features, the Bagging classifier (Quinlan, 1996) could achieve the TPR as high as 95.1% (Song et al., 2011).

Pros and cons: As there are millions of Twitter users, it is impractical to collect and analyse the overall Twitter social graph.

Table 4 – Typical neighbourhood-based features.

Notation	Description
ave_nfer	The average number of neighbours' followers for an account.
ave_ntwt	The average number of neighbours' posting tweets for an account.
ratio_fing_mnfer	The ratio of followings to median neighbours' followers.

Therefore, the datasets are unbiased and the accuracy is not steady. In addition, spammers tend to act as legitimate users by attaching benign attributes such as users' followings and followers (Yang et al., 2013). These innovative features are necessary to be updated frequently in order to catch up with latest spamming activities.

4.2.2. *Neighbourhood-based*

In the recent three years, there is an innovative method that focuses on neighbourhood-based features in social graph to detect Twitter spam (Yang et al., 2013). Compared to Section 4.2.1, the neighbourhood-based method extracts features specifically from direct neighbouring relationships of graph nodes. We discuss the detail of this novel method in the following.

Detailed Methods: The features are shown in Table 4. Generally speaking, the number of followers represents the popularity of an account. The legitimate users are more likely to be neighbours of prevalent accounts. Therefore, given an arbitrary user, the average number of his/her neighbours' followers will be relatively higher if the user is a spammer. Likewise, the number of the tweets from one account can also denote the prevalence, the neighbours of criminal accounts usually broadcast fewer tweets. For the last feature in Table 4, it is calculated as: $ratio\_fing\_mnfer = \frac{N_{fing}}{M_{nfer}}$, wherein N_{fing} represents the number of following accounts, and M_{nfer} is the median number of follower numbers for an account's all following numbers. Because spammers usually follow a large number of users randomly, the value of M_{nfer} will be small and N_{fing} will be high. That will result in a higher *ratio_fing_mnfer* for spammers than the one for benign users.

Performance: Yang et al. examined four classifiers and selected the Random Forest method due to its superior performance (F1-measure: 90%). The performance improved 10.1% after the neighbourhood-based features were applied into the methods.

Pros and cons: The robustness of the features was measured in Yang et al. (2013). It was found that the robustness of the first two features is extremely low. These neighbourhood-based features utilise the number of followers and the number of tweets to represent the popularity. Since there are many popular accounts with plenty of followers or tweets, these features are easily to be evaded by spammers.

5. Literature review part III: blacklist

In this section, we analyse the detection methods which rely on the third party blacklisting techniques. We will also discuss the pros and cons of this type of methods. Blacklisting techniques are widely used in current works for Twitter spam detection or dataset labelling (Chu et al., 2012; Ghosh et al., 2012; Grier et al., 2010; Ma et al., 2009; Thomas et al., 2011; Zhang et al., 2012). In particular, the works (Lee and Kim, 2013; Zhang et al., 2016) are published in the last three years.

Detailed Methods: Blacklisting techniques usually apply the third party services to add malicious information such as URLs and accounts into a blacklist. It can detect the spam directly through scanning the list and is widely employed in the real world. For example, Twitter applied Google's Safe Browsing API (Google Developers, 2016) to prevent the unsolicited links (Grier et al., 2010). It worked with the history data from URIBL, Joewein and Google blacklists (Hayati and Potdar, 2012). Before the spam arrives at victims, the malicious links will be blocked by these blacklists. At the same time, the new suspicious links will be added into the blacklisting dataset. In fact, many blacklists check at the domain level, not the specific URL. For example, if the whole domain "bit.ly" is blacklisted, even the URL is benign, it is still blacklisted. In addition, the blacklisting related features such as URL blacklisted number are also considered as a significant input for training the classifier (Chu et al., 2012; Zhang et al., 2012, 2016).

Performance: Based on the SVM algorithm, the spam detection method which included the input of blacklisting related features achieved the F1-measure of 87.6% (Zhang et al., 2012). Besides, by applying the blacklisting method to block malicious URLs, the maximum performance could reach 98.29% in practice (Grier et al., 2010).

Pros and cons: In the real world, blacklist is the most commonly used technique for malicious information defense. However, spammers frequently update the features and appearances of spam. According to Grier et al. (2010), Li (2013), about 90% of clicks on spam URLs in Twitter happen within the first 2 days. However, averagely, it takes 4 days for the blacklist to include the new spam URLs. This leads to a big time delay for the blacklists to catch up with the updates of frequently changing spam tweets. Therefore, the performance can hardly meet the expectation of spam defenders to timely capture all the Twitter spam, particularly zero-day spam. It has been reported that the performance of blacklisting technique will decrease around 10% per day during the time lag (Grier et al., 2010). Moreover, the abuse of shortened URLs also disables the performance of blacklisting techniques because signatures of malicious Twitter spam are sometimes concealed by shortened URLs.

6. Comparative study

In this section, we compare typical Twitter spam detection methods from several angles such as features and performance. As the blacklisting techniques apply the third party services to block spam, we mainly focus on the comparison of syntax based methods and feature analysis based methods. From the experimental results, we will have a quantitative understanding of the pros and cons of the existing spam detection methods.

For the purpose of empirical study, a 10-day ground-truth dataset was collected from Twitter's Streaming API (Chen et al.,

Table 5 – Sample datasets.

Dataset No	Dataset Type	Spam : Non-spam
1	Continuous	5k : 5k
2	Continuous	5k : 95k
3	Random	5k : 5k
4	Random	5k : 95k

2015). The dataset contained more than 600 million tweets. These tweets could be processed using JSON format, where each line of them was considered as an object (Bifet and Frank, 2010). By this way, the features would be collected automatically for each record in the dataset. To effectively label the raw tweets into spam and non-spam, Trend Micro's Web Reputation Service (WRS) (Chen et al., 2011) was used to detect the malicious URLs. The tweets which contained malicious URLs would be identified as spam. Trend Micro WRS kept a large number of history URL reputations and was updated timely. When it was used for real-time protection of website browsing, the protection rate could reach 99.8% (Chen et al., 2015). By using this service, over 6.5 million spam tweets were detected from our ground-truth dataset which contained 600 million tweets in total. Although manual labeling was more accurate, it was extremely time-consuming.

We applied four sampled datasets to justify the performance evaluation as shown in Table 5. We allocated the same number of spam and benign tweets. However, in the real-world scenarios, there were only approximately 5% spam tweets in the overall Twitter social network (Grier et al., 2010). Therefore, we also utilised the datasets which have the same real-world ratio of spam (*e.g.* Dataset 2 and 4). Furthermore, we randomly and continuously sampled the datasets to explore the impact of dispersion on the performance. Basically, we ran the experiments on Windows 10 operation system at a server with Inter(R) Core(TM) i7 CPU of 12 GB. In the comparative studies, we adopted four metrics to quantify the performance of Twitter spam detection methods. We will elaborate the four performance metrics in Appendix A.

6.1. *Comparing methods in key segment category*

6.1.1. *Comparison of feature selection*

Recent key segment methods are listed in Table 6 based on different functionality categories. We can see that the majority of the methods are URL based. We further compare the methods in key segment categories according to their functionalities: "Matching", "Shortened URL", "Clustering (tweet and account)" and "URL feature". "Matching" refers to searching the objective keyfield by retrieval methods such as regular expression (Thompson, 1968). "Shortened URL" refers to inspecting shortened URLs which may be malicious links. We can see from Table 6 that almost all URL based spam detection methods apply shortened URLs. "Clustering" refers to gathering tweets or accounts into different groups according to their embedded URLs as well as the text description. The last functionality mainly focuses on detecting the features for URLs, like lexical attributes. By training the features through machine learning algorithms, the classifier can be produced to identify the unsolicited URLs. Therefore, the tweets corresponding to those URLs will be categorised as spam.

Evaluation and Results: Different key segment methods may have different requirements of datasets. They can hardly be examined in an identical dataset. Moreover, it is unfair to compare the performance of key segment methods if methods run in different datasets and with varied configurations. Therefore, in order to examine the performance of key segment methods, we focus on the changes of performance while employing different functionalities. This evaluation strategy will eliminate the comparison errors caused by different datasets and testbeds.

The works of Lee and Kim (2012, 2013) and Thomas et al. (2011) adopted "URL based features" and "Shortened URL" in Twitter spam detection. As shown in Table 6, the work of Thomas et al. (2011) outperformed the other two methods in terms of Accuracy (94.14% vs. 91.53% and 93.11%). Moreover, the Keyword/Username pattern methods (refer to 3.1.2) only applied the "Matching" feature. Compared to the URL based methods (Lee and Kim, 2012, 2013; Thomas et al., 2011), the performance decreased to 91.00% (the difference is 94.14% – 91.00% = 3.14%). In fact, when more features were involved into the detection such as in the work of Chu et al. (2012), the performance would be slightly improved (the difference was 94.50% – 94.14% = 0.36%). In terms of F-measure, the works of Zhang et al. (2012, 2016) achieved the performance of 87.60%.

6.1.2. *Studying the performance of classifiers*

The URL features and classifiers are two important factors to the performance of URL based detection methods. In this subsection, we specifically study and compare the performance from the classifier angle of view.

Table 6 – Comparison between key segment methods.

Key Segment	K/U	URL					
	(Yardi et al., 2009)	(Thomas et al., 2011)	(Chu et al., 2012)	(Zhang et al., 2012, 2016)	(Thomas et al., 2011)	(Lee and Kim, 2013)	(Lee and Kim, 2012)
Matching	√		√	√			
Shortened URL		√	√	√	√	√	√
Clustering (Tweet)			√				
Clustering (Account)				√			
URL based Features					√	√	√
Performance	91.00% ACC	N/A	94.50% ACC	87.60% F-M	94.14% ACC	91.53% ACC	93.11% ACC

Note: K/U: Keywords/Username Pattern; ACC: Accuracy; F-M: F-measure.

Table 7 – Comparison of campaign categories in URL based methods (refer to Thomas et al. (2011)), among which Source URLs feature achieves the best performance (bold entries).

Feature Type	Accuracy	FP	FN
Source URLs	**89.74%**	**1.17%**	**19.38%**
HTTP Headers	85.37%	1.23%	28.07%
HTML Content	85.32%	1.36%	28.04%
Initial URL	84.01%	1.14%	30.88%
Final URL	83.59%	2.34%	30.53%
IP (Geo/ASN)	81.52%	2.33%	34.66%
Page Links	75.72%	15.46%	37.68%
Redirects	71.93%	0.85%	55.37%
DNS	72.40%	25.77%	29.44%
Frame URLs	60.17%	0.33%	79.45%

Table 8 – Comparison of classifiers in URL based methods (refer to Lee and Kim (2013)), among which L2R L1-loss SVC (dual) achieves the best performance (bold entries).

Classifier	AUC	Accuracy	FP	FN
L2R LR (primal)	90.00%	91.90%	1.56%	6.54%
L2R L2-loss SVC (dual)	89.95%	91.79%	1.49%	6.72%
L2R L2-loss SVC (primal)	89.73%	91.76%	1.50%	6.74%
L2R L1-loss SVC (dual)	**90.28%**	**91.87%**	**1.13%**	**7.01%**
L1R L2-loss SVC (primal)	89.84%	91.78%	1.56%	6.10%
L1R LR (primal)	90.07%	91.91%	1.27%	6.52%
L2R LR (dual)	90.20%	91.96%	1.54%	6.51%

Evaluation and Results: For comparison, we select two typical URL based methods (Lee and Kim, 2013; Thomas et al., 2011), one for URL features and another one for classifiers. The results are shown in Table 7 and 8, respectively. First, Thomas et al. categorised URL features into several groups, and then identified the groups which were more influential than the left (Thomas et al., 2011). It was found that the group of source URLs was the most influential one. By feeding groups of URL features into classifiers one after another, we found that the spam detection which only relied on the group of source URL features could achieve the Accuracy of 89.74%. Comparatively, the lowest Accuracy was only 60.17% when the group of frame URL features was exclusively fed into the classifier. We then come to the comparison on the performance of different classifiers. Seven classifiers were studied in terms of AUC, Accuracy, False Positive (FP) and False Negative (FN). We could find in Table 8 that the performance of different classifiers did not differ from each other too much (*i.e.* difference < 5%). In particular, the L2-regularised L1-loss SVC classifier achieved the highest AUC and lowest FP among various classifiers.

6.2. *Comparing methods in tweet content category*

In this subsection, we carry out comparative studies to demonstrate the performance of tweet content based methods. We mainly focus on the differences of performance caused by content based features. Because the content based features and statistic features can be quantified for a unified comparison, we will take typical methods from these two categories for comparison. More concretely, we select n-gram (Brown et al., 1992) as the typical tweet content based method and the work in Chen et al. (2015) as the typical statistic information based method. Based on the same dataset, the text of tweets is transferred into the vector format by applying the n-gram technique. We set the character-based 3-gram tokens as parameters, and then each tweet is mapped to a highly dimensional space in which each dimension is a token with length 3. To make the features comparable, both of them are transferred into two-dimension vectors using PCA Dimensionality Reduction (Bhraguram et al., 2012). The details are explained in Appendix B.

Evaluation and Results: The resultant two dimensions are shown in Fig. 3, and 4, respectively. It can be found that the distinction of the features derived from tweet content is more apparent. The features are represented by the red (spam) or black (benign) dots. Compared to statistic features, the dots for tweet content occupy a much bigger area. While the dots for feature statistics are distributed in a narrow angle without clear boundaries between two colours (spam and non-spam). There are two distinct parts in Fig. 3. Although there is a small mixed

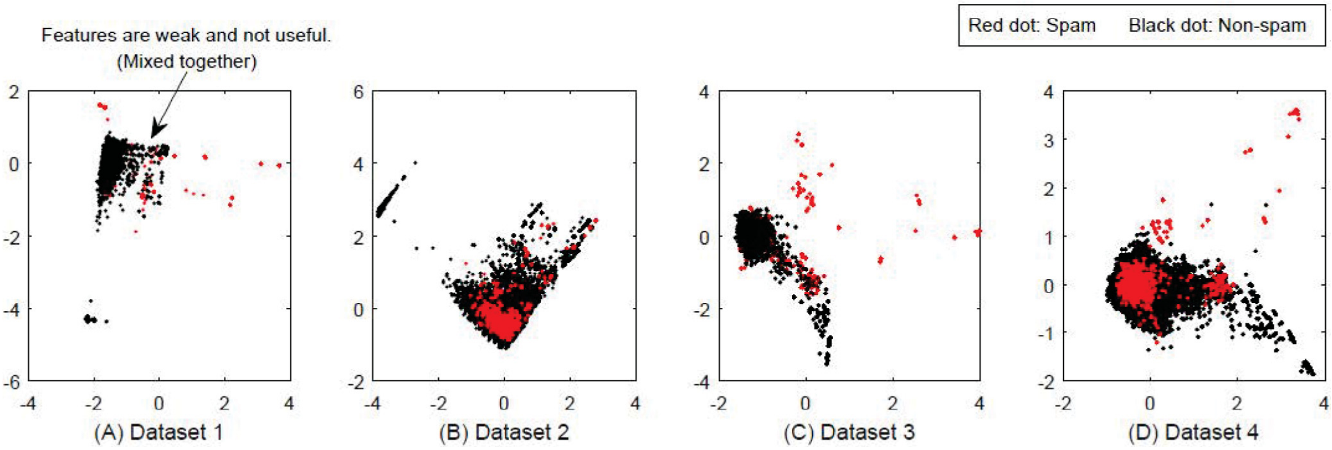

Fig. 3 – Input dimensionality reduction of tweet content methods. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

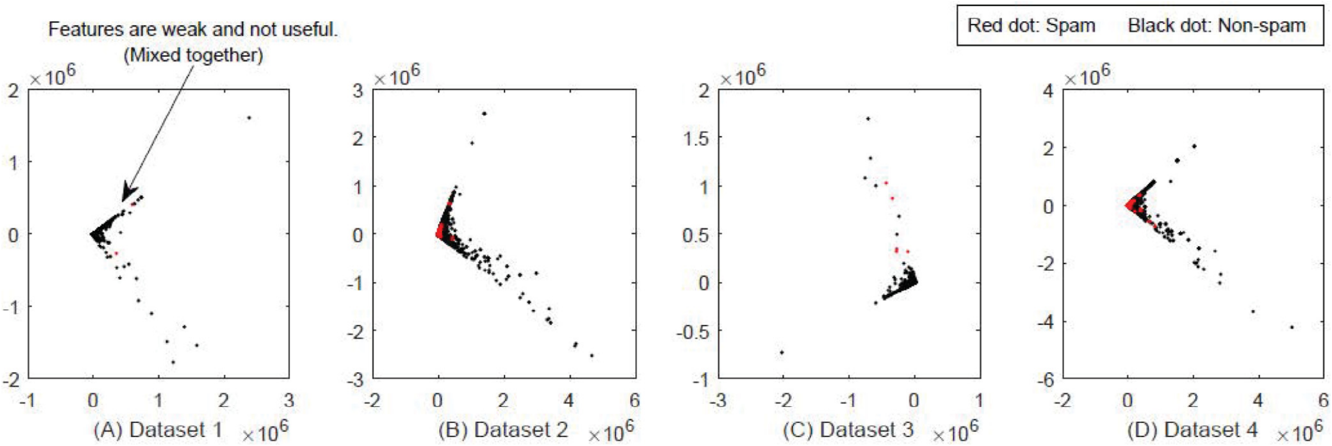

Fig. 4 – Input dimensionality reduction of feature statistics methods. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

area for Dataset 1 and 3 in Fig. 3, it has shown apparent differences on the representation of features.

6.3. *Comparing methods in statistic information category*

In this subsection, we continue to compare the statistic information based methods from two parts: classifiers and features. The first part includes feature extraction, which identifies the attributes of the spam or spammers and convert them into a vector format. The second part employs a set of machine learning classifiers with specific features to compare the performance.

6.3.1. *Studying the performance of classifiers*

Because statistic features can be extracted from our ground-truth datasets, we will carry out empirical analysis to demonstrate and compare the performance of various classifiers used in this category. Accordingly, we collect twelve popular features in this category (as shown in Table 9) for the input of classifiers (Chen et al., 2015). The classifiers have been implemented in Weka (Hall et al., 2009) integrated in the KNIME (Knime, 2016) platform. For each experiment, we will apply cross validation to obtain the results in terms of Accuracy, Recall, Precision and F-measure. There are five classifiers as well as four sampled datasets (refer to Table 5) involved in this part of analysis. The comparison results are shown in Fig. 5, 6, 7 and 8, respectively.

Evaluation and Results: The Random Forest method performed the best all the time, with all the results close to 100%. Likewise, the performance of Decorate classifier was slightly weak but still ranked the second. However, after we have applied randomly selected datasets, the performance value dropped dramatically. The Recall and F-measure only remained less than 80% in Dataset 4. On the contrary, the worst classifier was Naive Bayes. The average performance for this method could achieve around 60% according to the four evaluation criteria in 5K:5K datasets (*i.e.* the number of spammers = the number of non-spammers). While in the 5K:95K datasets (*i.e.* the number of spammers < < the number of non-spammers), the Precision and F-measure were not higher than 15%. In addition, we can see from Fig. 8 that the randomly sampled unbiased dataset provided the worst performance compared to other datasets.

6.3.2. *Studying the performance of features*

We further compare the typical statistical features by using distance metrics (Mullikin, 1992). The features and class values will first be transferred into a set of vectors, and then the correlation of each feature and classification can be quantified by the distance metrics. For the convenience of readers, we further introduce the "Distance Performance" to present the

Table 9 – Twelve statistical features used in Chen et al. (2015).

Number	Notation	Description
1	account_age	The age (days) of an account since its creation until the time of sending the most recent tweet.
2	no_follower	The number of followers of this twitter user.
3	no_following	The number of followings/friends of this twitter user.
4	no_userfavourites	The number of favourites this twitter user received.
5	no_lists	The number of lists this twitter user added.
6	no_tweets	The number of tweets this twitter user sent.
7	no_retweets	The number of retweets this tweet.
8	no_hashtag	The number of hashtags included in this tweet.
9	no_usermention	The number of user mentions included in this tweet.
10	no_urls	The number of URLs included in this tweet.
11	no_char	The number of characters in this tweet.
12	no_digits	The number of digits in this tweet.

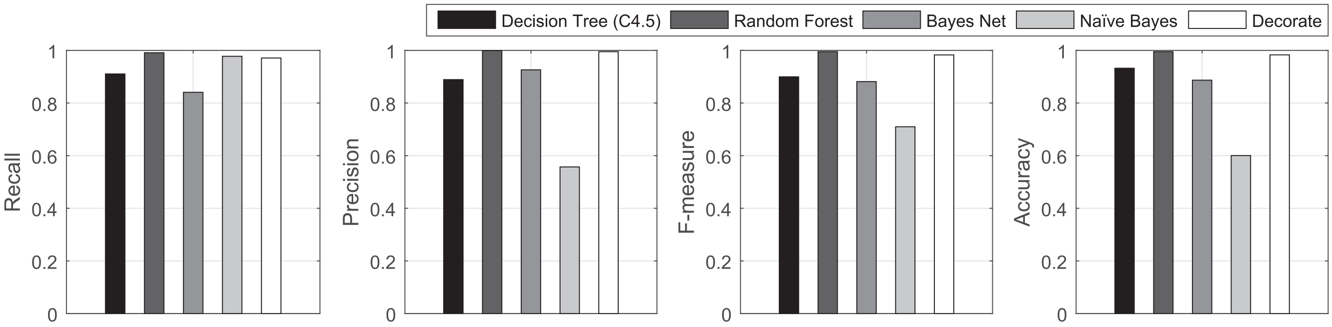

Fig. 5 – Performance Evaluation Compared on five classifiers in Dataset 1. Decision Tree is used in Ahmed and Abulaish (2013), Chen et al. (2015), Chu et al. (2012), Gao et al. (2012), Lee et al. (2010), Liu et al. (2016), Song et al. (2011) and Yang et al. (2011, 2013); Random Forest is used in Chen et al. (2015), Chu et al. (2012), Liu et al. (2016), Stringhini et al. (2010) and Yang et al. (2011, 2013); Bayes Net is used in Chen et al. (2015), Chu et al. (2012), Song et al. (2011) and Yang et al. (2011, 2013); Naive Bayes is used in Chen et al. (2015) and Wang, (2010); Decorate is used in Lee et al. (2010) and Yang et al. (2011, 2013).

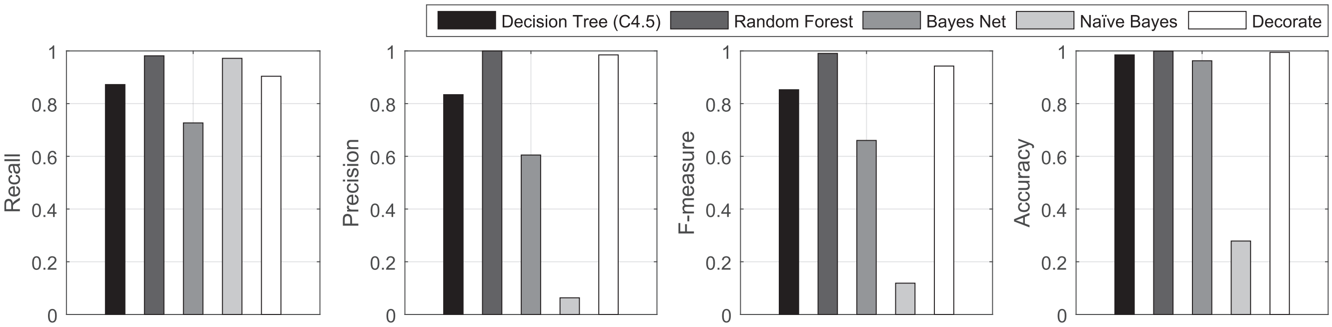

Fig. 6 – Performance Evaluation Compared on Five Classifiers in Dataset 2. Decision Tree is used in Ahmed and Abulaish (2013),; Chen et al. (2015), Chu et al. (2012), Gao et al. (2012), Lee et al. (2010), Liu et al. (2016), Song et al. (2011) and Yang et al. (2011, 2013); Random Forest is used in Chen et al. (2015), Chu et al. (2012), Liu et al. (2016), Stringhini et al. (2010) and Yang et al. (2011, 2013); Bayes Net is used in Chen et al. (2015), Chu et al. (2012), Song et al. (2011) and Yang et al. (2011, 2013); Naive Bayes is used in Chen et al. (2015) and Wang (2010); Decorate is used in Lee et al. (2010) and Yang et al. (2011, 2013).

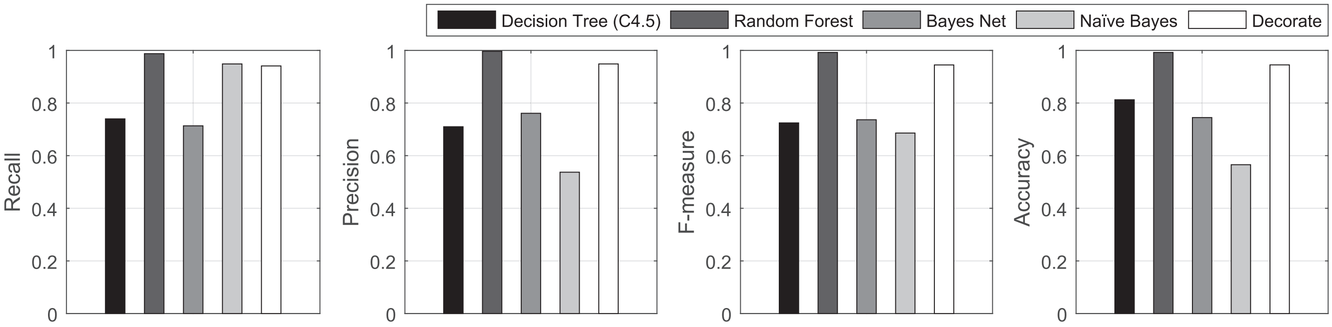

Fig. 7 – Performance Evaluation Compared on Five Classifiers in Dataset 3. Decision Tree is used in Ahmed and Abulaish (2013), Chen et al. (2015), Chu et al. (2012), Gao et al. (2012), Lee et al. (2010), Liu et al. (2016), Song et al. (2011) and Yang et al. (2011, 2013); Random Forest is used in Chen et al. (2015), Chu et al. (2012), Liu et al. (2016), Stringhini et al. (2010) and Yang et al. (2011, 2013); Bayes Net is used in Chen et al. (2015), Chu et al. (2012), Song et al. (2011) and Yang et al. (2011, 2013); Naive Bayes is used in Chen et al. (2015) and Wang (2010); Decorate is used in Lee et al. (2010) and Yang et al. (2011, 2013).

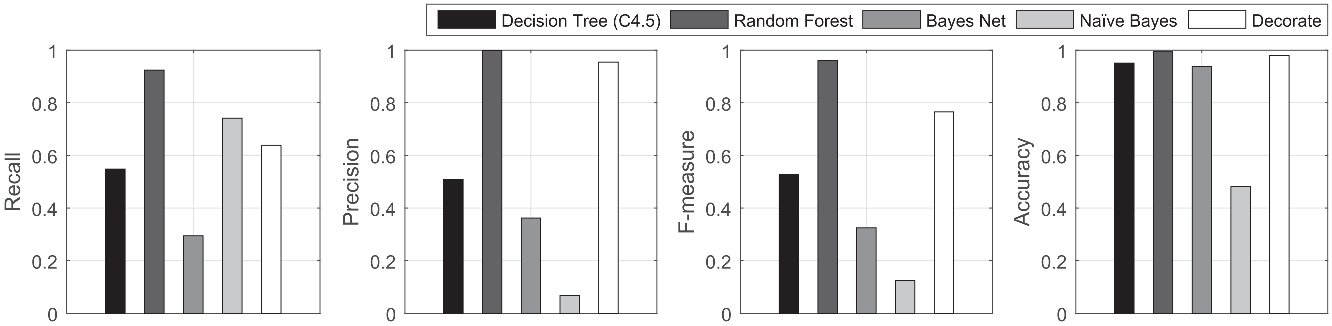

Fig. 8 – Performance Evaluation Compared on Five Classifiers in Dataset 4. Decision Tree is used in Ahmed and Abulaish (2013), Chen et al. (2015), Chu et al. (2012), Gao et al. (2012), Lee et al. (2010), Liu et al. (2016), Song et al. (2011) and Yang et al. (2011, 2013); Random Forest is used in Chen et al. (2015), Chu et al. (2012), Liu et al. (2016), Stringhini et al. (2010) and Yang et al. (2011, 2013); Bayes Net is used in Chen et al. (2015), Chu et al. (2012), Song et al. (2011) and Yang et al. (2011, 2013); Naive Bayes is used in Chen et al. (2015) and Wang (2010); Decorate is used in Lee et al. (2010) and Yang et al. (2011, 2013).

evaluation results. If the distance of two vectors (denoted by D) is shorter, the "Distance Performance" $DP = \frac{1}{D}$ will be higher (*i.e.* the corresponding features have better performance). The twelve features are presented in Table 9 for the empirical study. We will explain the details of the method that convert features into vectors and the usage of various distances in Appendix C.

Evaluation and Results: The results of the three distance methods are shown in Fig. 9, 10 and 11. We summarise three points from the results. *The first point*: Compared to the results in 5K:95K datasets, it is shown that the performance contributed by the feature "no_urls" was larger than by the feature "no_usermention" in 5K:5K datasets, in terms of Euclidean Distance. *The second point*: We can find similar phenomenon in Manhattan Distance. In addition, the performance of "no_hashtag" climbed almost twice as the performance of "no_usermention" in Dataset 1, while it was slightly lower in other datasets. The method was also influenced by the dataset type (refers to Table 5). It can be found that the performance of "no_usermention" decreased to almost the lowest among the three significant features in the continuous sampled datasets. However, the performance reached the highest in the non-continuous datasets. *The third point*: The performance in terms of Chebyshev Distance had a minor fluctuation in different datasets. The "no_urls" feature contributed the most throughout all the datasets. Overall, the features "no_hashtag", "no_usermention" and "no_urls" had larger *DP* values. The result also suggested that the tweet features played a more important role in feature statistics methods. Besides, the feature "no_digits" achieved around one third of the overall performance of tweet features (see details in Fig. 9 and 10, features {8, 9, 10} vs. feature 12).

6.4. *Comparing methods in social graph category*

We show the modus operandi of social graph based methods in Table 10. It can be seen from the table that there are mainly two kinds of social graph features (account related features and graph structure based features). For the first kind, each node in social graph represents a Twitter account. Accordingly, the contacts among Twitter users form a directed graph. For example, the indegree of a node represents the number of followers, while the outdegree represents the number of followings (friends) in the social graph. The second kind of features focuses more on the structure of the social graph. For example,

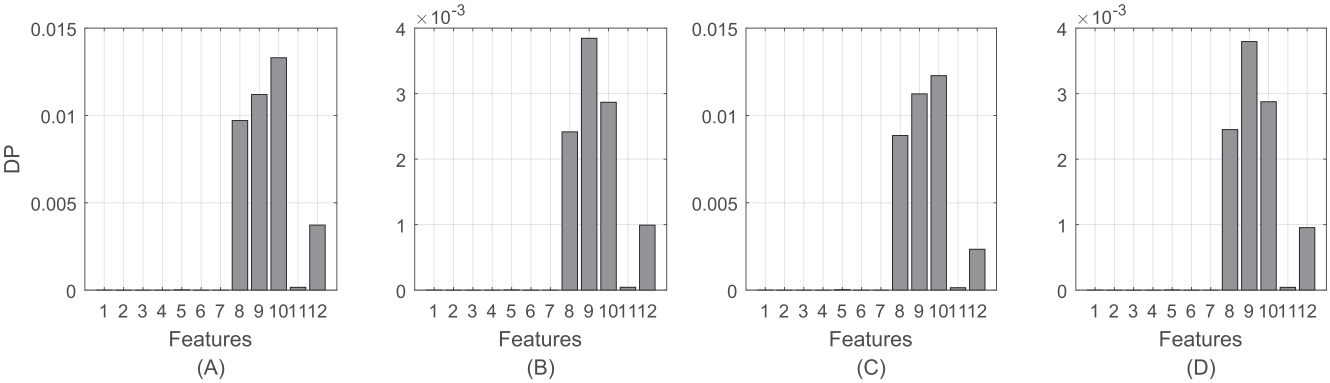

Fig. 9 – Euclidean Distance Performance (Numerical Features are Listed in Table 9 (Chen et al., 2015)). (A) Dataset 1; (B) Dataset 2; (C) Dataset 3;(D) Dataset 4.

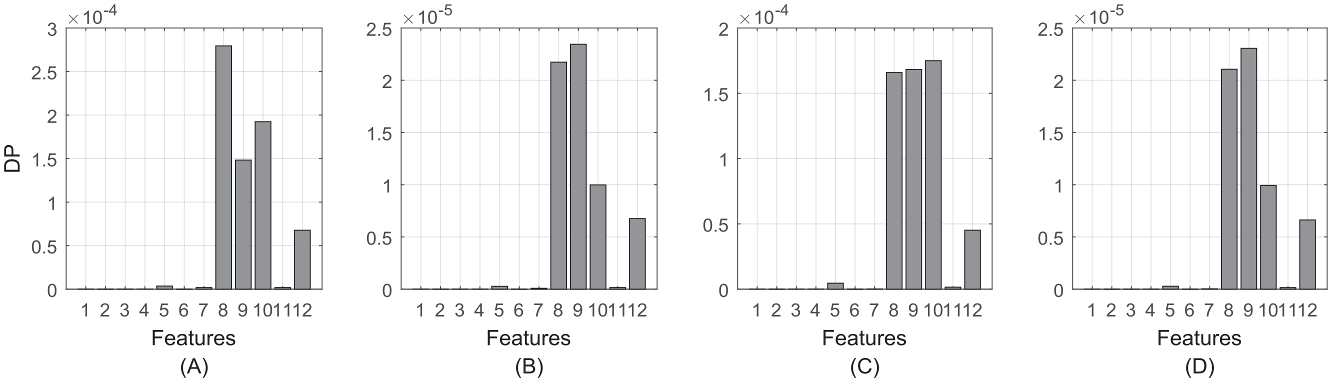

Fig. 10 – Manhattan Distance Performance (Numerical Features are Listed in Table 9 (Chen et al., 2015)). (A) Dataset 1; (B) Dataset 2; (C) Dataset 3;(D) Dataset 4.

the graph density feature denotes the ratio of the edges to the number of all possible edges existed in the graph. We summarise the features adopted by each method in Table 10.

Evaluation and Results: It is challenging to build a complete social graph with increasing number of Twitter users. Current social graph based methods used different datasets and measurements in order to evaluate their performance. Because those datasets are generally unbiased in constructing structures of social graphs (refers to Section 7.1), the direct comparison based on different datasets is unfair. Therefore, we make a basic performance comparison among the existing techniques using distinguished features. The results are also shown in Table 10. We can see that that social graph based methods have similar performance. Almost all methods achieved the performance around 90% in terms of F-measure, TPR or precision, except the work of Hu et al. (2013) (only achieved 69.8% TPR). The reason is that the work of Hu et al. (2013) only selected one social graph feature (outdegree), which was far from being enough. The social graph based methods either choose account related features or graph structure based features. However, according to the results in Table 10, the difference in feature selection did not lead to significant changes in the detection performance. In addition, there are a few studies that cannot quantify their performance. Therefore, we marked their performance by N/A in the "Performance" row (*e.g.* Yang et al.'s work (Yang et al., 2012)).

7. Summary and open issues

In this survey, we have discussed a series of Twitter spam detection methods, particularly the most recent ones which were published in the past three years. As we can see from the taxonomy and literature review parts, the majority of current methods mainly rely on machine learning based techniques (*e.g.* supervised or unsupervised) to identify Twitter spamming activities. Among these machine learning based methods, the major differences are about how and where to collect features. Therefore, our literature review is organised by the categories that are divided according to different feature selection methods. For example, readers can find out the detection methods based on syntax analysis, feature statistics and blacklists in Section 3, 4 and 5, respectively. We believe this brand-new view of angle at the state-of-art of Twitter spam detection will help readers capture the core challenges and develop promising solutions in this field.

In addition to the literature review, this survey also provides comparative studies on various Twitter spam detection methods. By extracting the features from tweets and

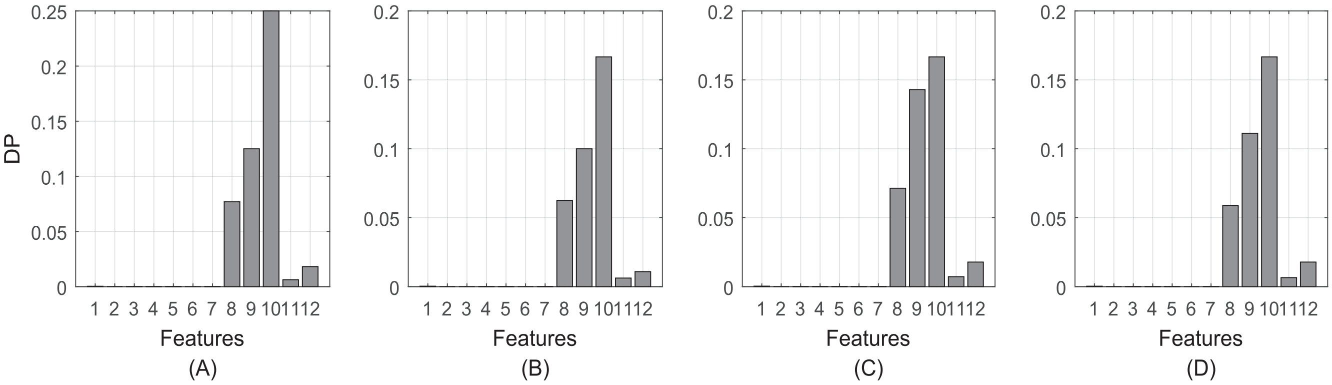

Fig. 11 – Chebyshev Distance Performance (Numerical Features are Listed in Table 9 (Chen et al., 2015)). (A) Dataset 1; (B) Dataset 2; (C) Dataset 3;(D) Dataset 4.

Table 10 – Comparison between social graph methods.

Social Graph	Description	(Hu et al., 2014)	(Hu et al., 2013)	(Gao et al., 2012)	(Wang, 2010)	(Thomas et al., 2011)	(Ghosh et al., 2012)	(Yang et al., 2012)	(Song et al., 2011)	(Yang et al., 2013)
Account Related Features	indegree	√	√		√	√	√			
	outdegree	√	√	√	√	√				
	indegree/outdegree						√			
	reputation				√					
	neighbourhood-based features									√
Graph Structure Features	graph density							√		
	reciprocity							√		
	distance (the shortest path length)							√	√	
	connectivity								√	
	local clustering coefficient									√
	betweenness centrality									√
	bidirectional links ratio									√
Performance		91.8% F-M	90.1% F-M	69.8% TPR	89.0% PCN	N/A	N/A	N/A	95.1% TPR	90.0% F-M

Note: F-M: F-measure; TPR: True Positive Rate; PCN: Precision.

accounts in the ground truth dataset, machine learning based methods adopt a set of algorithms as binary classifiers. Accordingly, we compare the performance of different classifiers. Our comparison concludes that most classifiers have no significant differences in detection performance except some specific scenarios (*e.g.* refers to the Precision plot in Fig. 6). We also analyse and compare the impact of various features to the detection performance. We adopt PCA and a set of distance metrics to measure the impact. The results suggest different features may lead to dramatic difference in performance (*e.g.* refers to Fig. 3 and 9). Although the simplest method is blacklisting technique, we do not involve it in the comparison because the working processes of current blacklisting techniques are essentially the same with each other. The major differences refer to the speed of collecting malicious URLs by other modules or cybersecurity systems. This part of discussion is out of the scope of our survey.

There are still several open issues for existing methods. In the following subsections, we will introduce each open issue in details.

7.1. *Open issue 1: data collection*

In the real-world scenarios, it is hard to collect an intact dataset and label it as ground truth. Twitter provides APIs which can only support 1% sampling rate among all the posts from users (Twitter Developers, 2016). Therefore, they cannot be used for real-time spam detection. For this reason, many studies collect tweets and corresponding account information from online platforms such as the information groups related to the same topic by crawling techniques. But this kind of datasets is biased as it only covers some specific clusters of tweets or accounts information. It is challenging to acquire an unbiased dataset.

7.2. *Open issue 2: data labeling*

In Twitter spam detection, the tweets collected will be labeled as spam or non-spam, and the accounts will be identified as criminals or legitimate users. The most accurate method is to label the data manually. However, it is time-consuming to label millions of tweets. The widely adopted labeling method is to use blacklists which have recorded history and current malicious URLs. The key of this method is to detect suspicious URLs to detect spam. However, as discussed before, there may exist detection errors in blacklisting techniques (refers to Section 5). Therefore, even though the blacklisting technique is considered as an effective labeling method, the datasets cannot be labeled accurately. This leads to the errors in the classifier performance evaluation.

7.3. *Open issue 3: spam drift*

Spam drift is a serious problem when applying feature statistics methods (Chen et al., 2015). It is defined as a problem that Twitter spam drifts throughout its life cycle in the statistical processes. Generally speaking, the datasets are collected across several consecutive days. After the features are extracted, it can be found that the spam features fluctuate significantly while the non-spam features remained stable (Liu et al., 2016). As a result, the classifier trained by the past data cannot be utilised to detect spam in the new dataset since the spam features always change along with the time.

7.4. Open issue 4: imbalanced dataset

The class imbalance problem has been overlooked in most previous studies. This problem widely exists in real-world Twitter data. According to Grier et al. (2010), there are only approximately 5% tweets are spam. The uneven distribution between spam and non-spam classes largely decreases the classification performance. The reason is that traditional machine learning techniques usually have better performance to classify the majority class than the minority class. In fact, the datasets used by current techniques are almost all imbalanced. From the previous comparative study, we can see that the classifiers performed better in the biased datasets than unbiased, as shown in Fig. 5, 6, 7 and 8. For instance, the F-measure for Naive Bayes reached around 70% in Dataset 1 and 3, but decreased to only one seventh of it (10% F-measure) in Dataset 2 and 4. Likewise, in the 1:19 (the ratio of spam tweets to non-spam tweets) datasets, the Precision of Bayes Net was reduced to half of it in the 1:1 datasets.

7.5. Open issue 5: data fabrication

Data fabrication refers to the fact that the data used for training classifiers is easy to be manipulated. Spammers usually try to evade the detection system by pretending to be benign accounts (Liu et al., 2016). To avoid being identified, they generally imitate legitimate users' behaviours such as the number and frequency of posting tweets. Moreover, spammers can apply sophisticated social engineering techniques to manipulate information (Chen et al., 2016). For example, criminal accounts have applied finite-state machines to develop the spam template such as a simple sentence with a shortened URL (Chen et al., 2016).

8. Conclusion

In this paper, we review the state-of-art in Twitter spam detection techniques. We first categorised existing Twitter spam detection methods into three groups and discussed the pros and cons for every method. We further carried out comparative studies on typical methods and mainly focused on the performance comparison. It was found that most of current spam detection techniques were based on feature selection and machine learning classification. Finally, we made a brief summary and discussed the open issues according to our analysis on the current methods. We hope this survey can benefit both academia and industrial security vendors to defend spamming activities in Twitter or other isomorphic social network platforms such as Facebook and LinkedIn.

Appendix A. Performance Metrics

Generally, to evaluate the performance of Twitter spam detection, we use Accuracy, Recall (Sensitivity), Precision and F-measure to measure the capability (Wang et al., 2011).

Table A.11 – Confusion matrix.

	Predicted	
	Spam	Non-spam
Spam	TP	FP
Non-spam	FN	TN

Traditionally, the results of spam classification provide the amount of projected spam and non-spam. Table A.11 shows the variables TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative). TP is the number of spam tweets which are correctly classified as spam, and FP represents the amount of non-spam which are wrongly labeled as spam. On the contrary, TN refers to the quantity of non-spam which are exactly considered as non-spam, while FN denotes the number of spam messages which are treated as spam by mistake.

Accuracy means the ratio of tweets identified correctly to all tweets. It is expressed as

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}.$$

Recall (Sensitivity) is defined as the ratio of correctly classified spam in total real spam, as

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

Precision is defined as true projected spam to classified spam. It can be obtained by

$$Precision = \frac{TP}{TP+FP}.$$

F-measure is the harmonic mean of Precision and Recall, and it can be calculated as follow:

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

Appendix B. PCA Dimensionality Reduction

The PCA dimensionality reduction will map original high-dimension space into a space with lower dimension (Bhraguram et al., 2012). The maximal variance after mapping can be achieved as

$$max_W \frac{1}{m-1}\sum_{i=1}^{m}\left(W^T(X_i-\bar{X})\right)^2,$$

wherein m is the number of samples, X_i is the vector representation of the ith input, and $\bar{X}$ is the average of all input vectors. Considering W as a matrix that contains all the vectors, the optimal object function is expressed by

$$min_W tr(W^T AW)(s.t. W^T W = I),$$

where *tr* means the trace of matrix W, and *A* is the covariance matrix which can be obtained by

$$A = \frac{1}{m-1}\sum_{i=1}^{m}(X_i - \bar{X})(X_i - \bar{X})^T.$$

With an eigenvalue *k* (*k* = 2), the vector Y with two-level space achieved by dimensionality reduction is

$$Y = W^T X.$$

Appendix C. Distance of Feature Vectors

We first convert each feature into a vector format which can be expressed as follows:

$$a = (x_{11}, x_{12}, \ldots, x_{1n}),$$

where *n* represents the dimension number of the vector, which is also the number of records in the dataset, and each dimension is the feature value of each record.

To represent the classification of spam and non-spam, the spam is labeled as 1, and the benign records are labeled as 0. Similarly, it is denoted as:

$$b = (x_{21}, x_{22}, \ldots, x_{2n}),$$

wherein *b* contains values of 0 or 1.

Then, we apply three traditional distance calculation methods to achieve the distance between the classification vector and each feature representation in the twelve attributes. The methods include Euclidean Distance (Danielsson, 1980), Manhattan Distance (Black, 2006) and Chebyshev Distance (Klove et al., 2010) which will be explained below.

Euclidean Distance is the most direct length as the crow files between two vectors. It can be calculated by

$$D_E = \sqrt{\sum_{k=1}^{n}(x_{1k} - x_{2k})^2}.$$

Manhattan Distance does not calculate the direct length, but applies the idea of city block. The route between the two points is either vertical or horizontal and can be denoted as

$$D_M = \sum_{k=1}^{n}|x_{1k} - x_{2k}|.$$

Chebyshev Distance is the maximum length of the difference between the coordinate values of two points. It is expressed as

$$D_C = \lim_{i\to\infty}\left(\sum_{k=1}^{n}|x_{1k} - x_{2k}|^i\right)^{1/i}.$$

To compare the contributes of the features, we apply the Distance Performance *DP* instead of distance *D* (including D_E, D_M and D_C) to evaluate each feature.

REFERENCES

Adewole KS, Anuar NB, Kamsin A, Varathan KD, Razak SA. Malicious accounts: dark of the social networks. J Netw Comput Appl 2017;79:41–67.

Ahmed F, Abulaish M. A generic statistical approach for spam detection in online social networks. Comput Commun 2013;36(10):1120–9.

Aizawa A. The feature quantity: an information theoretic perspective of Tfidf-like measures. In: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. ACM; 2000. p. 104–11.

Benevenuto F, Magno G, Rodrigues T, Almeida V. Detecting spammers on twitter. In: Collaboration, electronic messaging, anti-abuse and spam conference (CEAS), Vol. 6. 2010. p. 12.

Bhraguram T, Chenthara S, Gopan G, Nair AR. An adaptive subspace clustering dimension reduction framework for time series indexing in Knime workflows. In: Global Trends in Information Systems and Software Applications. Springer; 2012. p. 727–39.

Bifet A, Frank E. Sentiment knowledge discovery in twitter streaming data. In: International Conference on Discovery Science. Springer; 2010. p. 1–15.

Black PE. Manhattan distance. Dictionary Algorithms Data Struct 2006;18:2012.

Blanzieri E, Bryl A. A survey of learning-based techniques of email spam filtering. Artif Intell Rev 2008;29(1):63–92.

Brown PF, Desouza PV, Mercer RL, Pietra VJD, Lai JC. Class-based n-gram models of natural language. Comput linguistic 1992;18(4):467–79.

Cao Q, Sirivianos M, Yang X, Pregueiro T. Aiding the detection of fake accounts in large scale social online services. In: Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12). 2012. p. 197–210.

Castillo C, Mendoza M, Poblete B. Information credibility on twitter. In: Proceedings of the 20th international conference on World wide web. ACM; 2011. p. 675–84.

Cha M, Haddadi H, Benevenuto F, Gummadi KP. Twitter datdata homepage; 2012. Available from: http://twitter.mpi-sws.org/. [Accessed 23 September 2016].

Chen C, Wen S, Zhang J, Xiang Y, Oliver J, Alelaiwi A, et al. Investigating the deceptive information in twitter spam. Future Gener Comput Syst 2017;72:319–26.

Chen C, Zhang J, Xiang Y, Zhou W. Asymmetric self-learning for tackling twitter spam drift, in: 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2015, pp. 208–13.

Chen C, Zhang J, Chen X, Xiang Y, Zhou W. 6 million spam tweets: A large ground truth for timely twitter spam detection, in: 2015 IEEE International Conference on Communications (ICC), IEEE, 2015, pp. 7065–70.

Chen C, Zhang J, Xie Y, Xiang Y, Zhou W, Hassan MM, et al. A performance evaluation of machine learning-based streaming spam tweets detection. IEEE Trans Comput Soc Syst 2015;2(3):65–76.

Chen C, Zhang J, Xiang Y, Zhou W, Oliver J. Spammers Are Becoming "Smarter" on Twitter. IT professional 2016;18(2):66–70.

Chen C.S., Su S.-A., Hung Y.-C. Protecting computer users from online frauds, uS Patent 7,958,555 (Jun. 7 2011).

Chu Z, Gianvecchio S, Wang H, Jajodia S. Who is tweeting on twitter: human, bot, or cyborg? In: Proceedings of the 26th annual computer security applications conference. ACM; 2010. p. 21–30.

Chu Z, Widjaja I, Wang H. Detecting social spam campaigns on twitter. In: International Conference on Applied Cryptography and Network Security. Springer; 2012. p. 455–72.

Chu Z, Gianvecchio S, Wang H, Jajodia S. Detecting automation of twitter accounts: are you a human, bot, or cyborg? IEEE Trans Dependable Secure Comput 2012;9(6):811–24.

Costa H, Benevenuto F, Merschmann LH. Detecting tip spam in location-based social networks. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing. ACM; 2013. p. 724–9.

Danielsson P-E. Euclidean distance mapping. Comput Graph Image Process 1980;14(3):227–48.

Egele M, Stringhini G, Kruegel C, Vigna G. Compa: Detecting compromised accounts on social networks. In: NDSS. 2013.

Fan R-E, Chang K-W, Hsieh C-J, Wang X-R, Lin C-J. Liblinear: a library for large linear classification. J Mach Learn Res 2008;9(Aug):1871–4.

Friedman J, Hastie T, Tibshirani R. The elements of statistical learning, vol. 1. Springer series in statistics. Berlin: Springer; 2001.

Gao H, Hu J, Wilson C, Li Z, Chen Y, Zhao BY. Detecting and characterizing social spam campaigns. In: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. ACM; 2010. p. 35–47.

Gao H, Chen Y, Lee K, Palsetia D, Choudhary AN. Towards online spam filtering in social networks. In: NDSS, vol. 12. 2012. p. 1–16.

Ghosh S, Viswanath B, Kooti F, Sharma NK, Korlam G, Benevenuto F, et al. Understanding and combating link farming in the twitter social network. In: Proceedings of the 21st international conference on World Wide Web. ACM; 2012. p. 61–70.

Ghosh S, Zafar MB, Bhattacharya P, Sharma N, Ganguly N, Gummadi K. On sampling the wisdom of crowds: random vs. expert sampling of the twitter stream. In: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. ACM; 2013. p. 1739–44.

Gilani Z, Farahbakhsh R, Crowcroft J. Do bots impact twitter activity? In: Proceedings of the 26th International Conference on World Wide Web Companion, International World Wide Web Conferences Steering Committee. 2017. p. 781–2.

Google Developers. Google safe browsing api; 2016. Available from: https://developers.google.com/safe-browsing/v4/. [Accessed 23 September 2016].

Grier C, Thomas K, Paxson V, Zhang M. @ spam: the underground on 140 characters or less. In: Proceedings of the 17th ACM conference on Computer and communications security. ACM; 2010. p. 27–37.

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 2009;11(1):10–18.

Hayati P, Potdar V. Spam 2.0 state of the art, 2012.

Heymann P, Koutrika G, Garcia-Molina H. Fighting spam on social web sites: a survey of approaches and future challenges. IEEE Internet Comput 2007;11(6):36–45.

Holme P, Kim BJ. Growing scale-free networks with tunable clustering. Phys Rev E 2002;65(2):026107.

Hu X, Tang J, Zhang Y, Liu H. Social spammer detection in microblogging. In: IJCAI, vol. 13. Citeseer; 2013. p. 2633–9.

Hu X, Tang J, Liu H. Online social spammer detection. In: AAAI. 2014. p. 59–65.

Jiang J, Wilson C, Wang X, Sha W, Huang P, Dai Y, et al. Understanding latent interactions in online social networks. ACM Trans Web 2013;7(4):18.

Jin X, Lin C, Luo J, Han J. A data mining-based spam detection system for social media networks. Proc VLDB Endowment 2011;4(12):1458–61.

Joachims T. Text categorization with support vector machines: learning with many relevant features. In: European conference on machine learning. Springer; 1998. p. 137–42.

Kent JT. Information gain and a general measure of correlation. Biometrika 1983;70(1):163–73.

Klien F, Strohmaier M. Short links under attack: geographical analysis of spam in a url shortener network. In: Proceedings of the 23rd ACM conference on Hypertext and social media. ACM; 2012. p. 83–8.

Klove T, Lin T-T, Tsai S-C, Tzeng W-G. Permutation arrays under the Chebyshev distance. IEEE Transactions on Information Theory 2010;56(6):2611–17.

KNIME. Knime; 2016. Available from: https://www.knime.org/. [Accessed 23 September 2016].

Lawson CL, Hanson RJ. Solving least squares problems, vol. 15. SIAM; 1995.

Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. Nature 1999;401(6755):788–91.

Lee K, Caverlee J, Webb S. Uncovering social spammers: social honeypots+ machine learning. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. ACM; 2010. p. 435–42.

Lee K, Caverlee J, Cheng Z, Sui DZ. Content-driven detection of campaigns in social media. In: Proceedings of the 20th ACM international conference on Information and knowledge management. ACM; 2011. p. 551–6.

Lee S, Kim J. Warningbird: Detecting suspicious urls in twitter stream. In: NDSS, vol. 12. 2012. p. 1–13.

Lee S, Kim J. Warningbird: a near real-time detection system for suspicious urls in twitter stream. IEEE Trans Dependable Secure Comput 2013;10(3):183–95.

Li C-T. Emerging digital forensics applications for crime detection, prevention, and security. IGI Global; 2013.

Liu S, Wang Y, Zhang J, Chen C, Xiang Y. Addressing the class imbalance problem in twitter spam detection using ensemble learning. Comput Secur 2017;69:35–49.

Liu S, Zhang J, Xiang Y. Statistical detection of online drifting twitter spam: Invited paper. In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. ACM; 2016. p. 1–10.

Liu S, Zhang J, Wang Y, Xiang Y. Fuzzy-based feature and instance recovery. In: Asian Conference on Intelligent Information and Database Systems. Springer; 2016. p. 605–15.

Ma J, Saul LK, Savage S, Voelker GM. Identifying suspicious urls: an application of large-scale online learning. In: Proceedings of the 26th annual international conference on machine learning. ACM; 2009. p. 681–8.

Ma J, Saul LK, Savage S, Voelker GM. Learning to detect malicious urls. ACM Trans Intell Syst Technol 2011;2(3):30.

McGrath DK, Gupta M. Behind phishing: An examination of phisher modi operandi., LEET 8, 2008. 4.

McLachlan G, Do K-A, Ambroise C. Analyzing microarray gene expression data, vol. 422. John Wiley & Sons; 2005.

Mullikin JC. The vector distance transform in two and three dimensions. CVGIP 1992;54(6):526–35.

Newman ME. A measure of betweenness centrality based on random walks. Soc Networks 2005;27(1):39–54.

Oliver J, Pajares P, Ke C, Chen C, Xiang Y An in-depth analysis of abuse on twitter, Trend Micro, 2014, 225.

Pak A, Paroubek P. Twitter as a corpus for sentiment analysis and opinion mining. In: LREc, vol. 10. 2010. p. 1320–6.

Perfect H. Applications of Menger's graph theorem. J Math Anal Appl 1968;22(1):96–111.

Provost J Naïve-Bayes vs. rule-learning in classification of email, University of Texas at Austin, 1999.

Quinlan JR. Bagging, boosting, and c4. 5. In: AAAI/IAAI, vol. 1. 1996. p. 725–30.

Sabottke C, Suciu O, Dumitras T. Vulnerability disclosure in the age of social media: exploiting twitter for predicting real-world exploits. In: USENIX Security Symposium. 2015. p. 1041–56.

Sala A, Cao L, Wilson C, Zablit R, Zheng H, Zhao BY. Measurement-calibrated graph models for social network experiments. In: Proceedings of the 19th international conference on World wide web. ACM; 2010. p. 861–70.

Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. Inf Process Manage 1988;24(5):513–23.

Salton G, Wong A, Yang C-S. A vector space model for automatic indexing. Commun ACM 1975;18(11):613–20.

Song J, Lee S, Kim J. Spam filtering in twitter using sender-receiver relationship. In: International Workshop on recent advances in intrusion detection. Springer; 2011. p. 301–17.

Stringhini G, Kruegel C, Vigna G. Detecting spammers on social networks. In: Proceedings of the 26th Annual Computer Security Applications Conference. ACM; 2010. p. 1–9.

Stringhini G, Wang G, Egele M, Kruegel C, Vigna G, Zheng H, et al. Follow the green: growth and dynamics in twitter follower markets. In: Proceedings of the 2013 conference on Internet measurement conference. ACM; 2013. p. 163–76.

Symantec. Internet security threat report, Tech. rep., Symantec Co. Ltd. 2015.

Tan E, Guo L, Chen S, Zhang X, Zhao Y. Unik: unsupervised social network spam detection. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. ACM; 2013. p. 479–88.

Thomas K, Grier C, Song D, Paxson V. Suspended accounts in retrospect: an analysis of twitter spam. In: Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, ACM. 2011. p. 243–58.

Thomas K, Grier C, Ma J, Paxson V, Song D. Design and evaluation of a real-time url spam filtering service, in: 2011 IEEE Symposium on Security and Privacy, IEEE, 2011, pp. 447–62.

Thompson K. Programming techniques: regular expression search algorithm. Commun ACM 1968;11(6):419–22.

Twitter Developers. Twitter's streaming api documentation; 2016. Available from: https://dev.twitter.com/streaming. [Accessed 23 September 2016].

Verma M, Sofat S. Techniques to detect spammers in twitter-a survey. Int J Comput Appl 2014;85(10).

Wang AH. Don't follow me: Spam detection in twitter, in: Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on, IEEE, 2010, pp. 1–10.

Wang D, Irani D, Pu C. A social-spam detection framework. In: Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference. ACM; 2011. p. 46–54.

Wang D, Navathe SB, Liu L, Irani D, Tamersoy A, Pu C. Click traffic analysis of short url spam on twitter, in: Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference on, IEEE, 2013, pp. 250–9.

Whittaker C, Ryner B, Nazif M. Large-scale automatic classification of phishing pages, In: NDSS, vol. 10, 2010, 2010.

Wu T, Wen S, Liu S, Zhang J, Xiang Y, Alrubaian M, et al. Detecting spamming activities in twitter based on deep-learning technique. Concurrency Comput Pract Exp 2017;29(19).

Wu T, Liu S, Zhang J, Xiang Y. Twitter spam detection based on deep learning. In: Proceedings of the Australasian Computer Science Week Multiconference. ACM; 2017. p. 3.

Yang C, Harkreader RC, Gu G. Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In: International Workshop on recent advances in intrusion detection. Springer; 2011. p. 318–37.

Yang C, Harkreader R, Zhang J, Shin S, Gu G. 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. ACM; 2012. p. 71–80.

Yang C, Harkreader R, Gu G. Empirical evaluation and new design for fighting evolving twitter spammers. IEEE Trans Inf Forensics Secur 2013;8(8):1280–93.

Yang Y, Pedersen JO. A comparative study on feature selection in text categorization. In: ICML, vol. 97. 1997. p. 412–20.

Yang Z, Wilson C, Wang X, Gao T, Zhao BY, Dai Y. Uncovering social network sybils in the wild. ACM Trans Knowl Discov Data 2014;8(1):2.

Yardi S, Romero D, Schoenebeck G, et al. Detecting spam in a twitter network. First Monday 2009;15(1).

Yerazunis B. The crm114 discriminator–the controllable regex mutilator. 2009.

Zhang X, Zhu S, Liang W. Detecting spam and promoting campaigns in the twitter social network, in: 2012 IEEE 12th International Conference on Data Mining, IEEE, 2012, pp. 1194–9.

Zhang X, Li Z, Zhu S, Liang W. Detecting spam and promoting campaigns in twitter. ACM Trans Web 2016;10(1):4.

Zhu Y, Wang X, Zhong E, Liu NN, Li H, Yang Q. Discovering spammers in social networks. In: AAAI. 2012.

Tingmin Wu received the Bachelor of Information Technology degree (with first class Hons.) from Deakin University Australia in 2016. Currently, she is a PhD student with Swinburne University of Technology and CSIRO Data61, Australia. Her research interests include cyber security, especially in social spam detection.

Sheng Wen received the degree in computer science from the Central South University of China in 2012, and the Ph.D. degree from the School of Information Technology, Deakin University, Australia, in 2015. Currently, he is a Senior Lecturer with Swinburne University of Technology. His focus is on modeling of virus spread, information dissemination, and defense strategies for the Internet threats. He is also interested in the techniques of identifying information sources in networks.

Yang Xiang received his PhD in Computer Science from Deakin University, Australia. He is the Dean of Digital Research & Innovation Capability Platform, Swinburne University of Technology, Australia. His research interests include cyber security, which covers network and system security, data analytics, distributed systems, and networking. In particular, he is currently leading his team developing active defense systems against large-scale distributed network attacks. He is the Chief Investigator of several projects in network and system security, funded by the Australian Research Council (ARC).

Wanlei Zhou received the B.Eng and M.Eng degrees from Harbin Institute of Technology, Harbin, China in 1982 and 1984, respectively, and the PhD degree from The Australian National University, Canberra, Australia, in 1991, all in Computer Science and Engineering. He also received a DSc degree (a higher Doctorate degree) from Deakin University in 2002. He is currently the Alfred Deakin Professor (the highest honour the University can bestow on a member of academic staff), Chair of Information Technology, and Associate Dean (International Research Engagement) of Faculty of Science, Engineering and Built Environment, Deakin University.