A Framework for Evaluating Anti Spammer Systems for Twitter

Kenny Ho[1], Veronica Liesaputra[1], Sira Yongchareon[2] and Mahsa Mohaghegh[2]

[1]Department of Computer Science, Unitec Institute of Technology, New Zealand
kenny098x@yahoo.co.nz,vliesaputra@unitec.ac.nz
[2]Department of Information Technology and Software Engineering, Auckland University of Technology, New Zealand
{sira.yongchareon, mahsa.mohaghegh}@aut.ac.nz

Abstract. Despite several benefits to modern communities and businesses, Twitter has attracted many spammers overwhelming legitimate users with unwanted and disruptive advertising and fake information. Detecting spammers is always challenging because there is a huge volume of data that needs to be analyzed while at the mean time spammers continue learning and changing their ways to avoid being detected by anti-spammer systems. Several spam classification systems are proposed using various features extracted from the content and user's information from their Tweets. Nevertheless, no comprehensive study has been done to compare and evaluate the effectiveness and efficiency of these systems. It is not known what the best anti-spammer system is and why. This paper proposes an evaluation framework that allows researchers, developers, and practitioners to access existing user-based and content-based features, implement their own features, and evaluate the performance of their systems against other systems. Our framework helps identify the most effective and efficient spammer detection features, evaluate the impact of using different numbers of recent tweets, and therefore obtaining a faster and more accurate classifier model.

Keywords: Spam detection; Evaluation workbench; Feature selection; Machine learning

1 Introduction

Spams are unwanted activities such as when marketers send members unwanted advertisements, post fake reviews, or steal user information by directing users to malicious external pages [11]. As Social Network Services (SNS) becoming an important mode of communication, it attracts spammers who overwhelms users with unwanted content. Among these sites, Twitter, which was started in 2006, has grown to be one of the most popular SNS [22]. There are 500 million number of messages (called tweets) produced by 328 million active Twitter users (called twitterers) every day. Unlike other popular SNS, tweets can be read by anyone and people can follow a user without their consent. To attract users to their target websites, spammers post a large number of coordinated messages containing specific URLs and sometimes describing them with unrelated words [26]. Because SNS helps build intrinsic trust between their users, 45% of them will click on links posted by their online friends even though they do not know those people in real life [24]. Twitterers also tend to post shortened URLs and write in abbreviated forms that rarely appear in conventional text documents or e-mails as a tweet can only contain up to 140 characters. Consequently, it is difficult for users to know the source URL and identify the content of the URL without clicking the link and loading the page. The noisy, unstructured, and informal expressions, such as “2mo is a new daaaaay!” or “TIL DC Comics stands for Detective Comics”, used in the text also made it difficult for automatic spam detection system to accurately identify the semantic meaning of the tweets. Hence, social spamming is more harmful and complex than SMS, email or Web spams. It is becoming an important problem for users and service providers.

Around 83% of users of social networks have received at least one unwanted friend request or message and over 3% of tweets are spam [7]. To make Twitter a spam-free platform, Twitter enable twitterers to report spam URLs, tweets and accounts which after being verified will be included in the Twitter’s blacklist. All URLs, tweets or accounts in the blacklist will be automatically filtered, suspended, or deleted by Twitter. However, due to time lag, 90% of users may visit a new spam link before it is included in the blacklist [26]. Furthermore, twitterers identify spammers manually based on experience that could lead to false positives. Therefore, it is important to have a tool that can automatically identify spammers. The approach must be scalable too, i.e. it can handle a large amount of data in a short amount of time with limited computation resources.

We can divide anti-social spammers systems into two types: tweet-level detection and account-level detection [24]. The tweet-level detection system check each tweet for spam text content or URLs. If an account has posted a certain number of spam tweets, it is flagged as spammers. As around 350,000 tweets are generated per minute [22], tweet-level detection consumes too much computing resources and is harder to be run in real-time. Account-level detection checks individual accounts profile and activity patterns for evidence of them sending spam tweets or is a fake account. Because there is very limited amount of imbalanced-labeled data, account-level detection system tends to have high precision and accuracy but low recall. When it predicts that an account is a spammer account, it has a high probability of it being true. However, there are many more spammer accounts out there that were not considered as spammer candidates, i.e. they are classified as legitimate. This of course is not useful to us as we are interested in detecting all spammers.

Spam detection is a never-ending game of cat and mouse. Although security companies, as well as Twitter, are working on creating systems to detect spam and spammers, spammers are always trying to avoid being detected. They deploy different techniques to post unwanted messages to users on SNS for advertisement, frauds or spreading of malware through the malicious URLs [12]. For instance, spammers create many fake accounts to post spam tweets for a specific purpose (or known as spam campaign), send message with different text to convey the same meanings or pay some users to follow their accounts [15]. Thus, the statistical attributes of spammers and spam tweets vary over time. System that relies on old samples may struggle to detect the new spammers or spam tweets.

As there are many spam detection systems proposed, it is hard for users and providers to decide which the best one is. We have also found that different work uses different evaluation metrics and datasets, so it is hard to achieve a standard evaluation. This brings in a research challenge about comparing and evaluating the performance of various spam detection systems and identifying the best technique w.r.t effectiveness (i.e., accuracy, true positive rate, and precision) and efficiency (run-time execution for training and classification). This is particularly beneficial to the research community as any newly proposed techniques can be evaluated against the existing ones allowing knowing if the technique is an improvement.

In our study, we have reviewed 172 content-based and user-based features from the majority of existing literature. Based on these features, we aim to develop a workbench, namely WEST (Workbench Evaluation Spammer detection system in Twitter) to evaluate their proposed features against the ones in defined in WEST as well as to set the best number of recent tweets and find the best possible subset from all the features available. We have designed an evaluation method with a set of experiments to help select the optimal subset of features.

We organize the rest of this paper as follows. Section 2 introduce background and existing work related to spam detection for Twitter. Section 3 discusses our proposed evaluation workbench and Section 4 discusses experimental results from our study. Lastly, a conclusion and future work are given in Section 5.

2 Related Work

This section provides an overview of related work with approaches and methods for SNS spammer detection.

[25] shows that machine learning methods demonstrated by [12] can be utilized with significant success in spammer detection on Twitter. Such methods are able to extract user or context-based features from user-behavioral patterns or linguistic features in a tweet [1, 6]. [2] shows that supervised machine learning techniques such as Support Vector Machine [10, 24] are able to train features extracted from user profiles in order to find profiles linked to spam activity. Performance is evaluated based on precision (the percentage of correct positive prediction), recall (the percentage of positive instances that were predicted as positive), and accuracy (overall percentage of correct prediction). [7] demonstrates a method of extracting the user and context-based features from the dataset before running this through Meda et al.'s Random Forest classifier [25]. The output was evaluated based on precision and f-measure, the harmonic mean of precision and recall. [17] uses information gain and relief methods to determine the five best features from the dataset. [17] uses Information Gain, and Relief methods to find the best five from features. These approaches all use different features, datasets and classifiers, and as such, we are unable to evaluate and compare their performance to ours.

Context-based features are linguistic features extracted from tweet context [6]. Twitter performs no checks on the legitimacy of shortened URLs, so spammers often exploit this by using a shortened URL service in an attempt to lure in legitimate Twitter users. [19] points out that spammers often use the same URLs in multiple tweets in order to increase the chance of it being clicked on by legitimate users. A number of researchers have utilized features related to URLs. Examples of these are the number of URLs [9], the number of URLs per word [6], and the number of unique URLs [10]. In Twitter, the #hashtag is used to describe a term, event, or emotion. If multiple tweets occur with the same #hashtag, it will become a trending topic [7]. Spammers often include a trending #hashtag with their tweets (though with unrelated content) in order to lure in legitimate users [17]. #Hashtags can be manipulated in the same way as URL features, and expanded to other forms such as the number of #hashtags per work on a tweet [6]. Twitter users are able to include @username in their tweets (called a "mention"). This enables the tweet to be sent to the user in the @username, regardless of whether or not they are followers of or followed by the user who tweeted. This is a feature that spammers also exploit, enabling them to push tweets to users [17]. This feature has been explored by several authors [7, 19]. It has been noted that tweets from spammers often include a larger number of spam-related words (up to 39%) while legitimate users around 4% [1]. Because of this, some papers use a spam word feature based on spam words from sites such as Wordpress.org [1, 7]. Other methods and approaches include features such as percentage of words not contained in a dictionary in their system [8].

User-based features are derived from properties related to user behaviour [1]. Generally, spammers follow as many users as possible to gain their attention, and increase the likelihood of success with spam attacks [7]. Common user-based features include number of followers (users following the user in question), number of followings (the users the user in question is following), and reputation (determining the user's influence on Twitter). These features are used in different combinations with varying success. [7] uses the number following and number followed features, and achieve 95.7% precision, whereas [14] uses number following, number followed, and reputation, but only achieves 91% precision. Both [12] and [2] use the followers-to-followings ratio. The reason for this feature is that while spammers attempt to follow as many accounts as possible, it is difficult to achieve "follow-backs", and the features ensures a healthy ratio of followers and followings is maintained. The approach used in [12] was able to achieve 93.6% precision. However, [15] points out followers can be purchased from certain websites, effectively reducing the reliability of the followers-to-followings ratio feature. They introduce a new feature called bi-directional links ratio. This is defined as "mutual followings" – i.e., two accounts following each other. This feature is more difficult for spammers to evade, since it results in them having to purchase more followers. The only way this is evaded is through reflexive reciprocity – when a user follows someone back out of courtesy [27]. 95% of spam tweets contain shortened malicious URLs. [16] proposes URL rate and Interaction Rate - two features to address URL-based spam attacks. Interaction rate notes the lack of normal interaction behavior in spammers, while URL rate compares the ratio of URL-based tweets to normal tweets. This is a particularly effective feature since is it almost impossible to evade.

3 Evaluation Workbench

As fundamentally formulated in the existing literature, given a set of users $U = \{u_1, u_2, u_3, \cdots, u_j\}$ in the dataset D, a spammer detection system is intended to build a classifier model, c, to predict whether a user, u_i, is a spammer based on a set of features $F = \{f_1, f_2, f_3, \cdots, f_k\}$ extracted from the user's social activities A_i, relations R_i, profiles P_i and/or textual contents T_i gathered from the user's N recent tweets. Spammers True Positive Rate, Spammers False Positive Rate, Precision, Accuracy, F$_1$-measure, and Time are then used to measure the efficiency and effectiveness of the model.

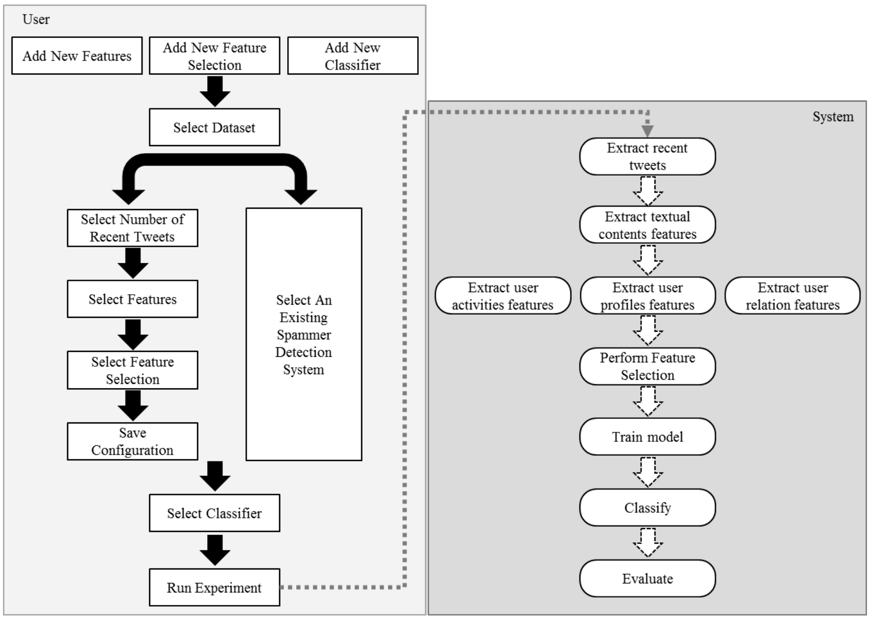
User
Add New Features
Add New Feature Selection
Add New Classifier
Select Dataset
Select Number of Recent Tweets
Select Features
Select Feature Selection
Save Configuration
Select An Existing Spammer Detection System
Select Classifier
Run Experiment
System
Extract recent tweets
Extract textual contents features
Extract user activities features
Extract user profiles features
Extract user relation features
Perform Feature Selection
Train model
Classify
Evaluate

Fig. 1. Framework for the Workbench for Evaluating Spammer in Twitter (WEST)

Unfortunately, so far none has compared the performance of their systems with the other systems. For example, both [1] and [2] use Support Vector Machine model to identify spammers. However, they build the model based on different features. [2] does not include any features from the users' tweet contents, while [1] does. Because [1] did not use the same dataset as [2] or compare their performance with [1] it is unclear which system is better. Spammers are continually changing their strategies to fool the anti-spam systems [3]. Current effective features now might not be effective in the future. Thus, it is imperative to be able to quickly find out the ineffective features and test the performance of new extracted features based on a new dataset.

The main motivation for our Workbench for Evaluating Spammers in Twitter (WEST) is to provide a comprehensive collection of social activities, relations, profiles and textual content features that researchers can quickly select and evaluate on new data sets. People can then easily determine which features or which overall system is the most effective and efficient. WEST is written in JAVA and it has extensible architecture that enables new features to be easily implemented and integrated to the workbench. It relies on WEKA [20] to perform the feature selection and classification on the dataset. Therefore, users can choose any attribute selection technique and supervised learning algorithm available in WEKA or include their own implementation to WEKA when building their anti-spammer model.

With WEST, we are able to answer the following important questions.

1. What are the most effective sets of features for identifying spammers?
2. What is the most efficient model for detecting spammers?
3. Can the number of recent tweets used significantly affect the system performance?

Figure 1 illustrates the overall framework of WEST and this section outlines the set of features included in the workbench and the way researchers or practitioners can utilize and extend WEST. To avoid confusion, from now on, we will use the term *user* to signify the users of WEST and *twitterer* to represent the users of Twitter.

3.1 Dataset

The input dataset must be comprised of two folders, *spam* and *ham*, containing XML files for spammers and ham twitterers respectively. Data about each twitterer is stored in a separate XML file as shown in Figure 2. WEST then extracts features that were selected by the users from the file.

```
<root>
<id>187758822251704321</id>
<name>water lillies</name>
<screen_name>freshlillie</screen_name>
<followers_count>1541</followers_count>
<friends_count>1996</friends_count>
<description>ownerwaterlillies bodyskincare~so pure it's edible 32 yrs. exp.</description>
<favourites_count>1</favourites_count>
<statuses_count>3328</statuses_count>
<tweet_count>139</tweet_count>
<tweets><tweet>
<text>@thepeoplescourt marilynyou're smart  i love your hair. </text>
<created_at>Tue Sep 30 05:30:32 +0000 2003</created_at>
<in_reply_to_status_id></in_reply_to_status_id>
<in_reply_to_user_id>27914673</in_reply_to_user_id>
<in_reply_to_screen_name>thepeoplescourt</in_reply_to_screen_name>
<retweet_count>0</retweet_count>
<retweeted>false</retweeted>
</tweet></tweets>
</root>
```

Fig. 2. XML file of a twitterer

Based on the collection of features that WEST currently has, Table 1 displays the XML tags read by WEST and the list of information that can be extracted from each tag. Given a twitterer ID, we utilize Twitter4J [21] to get the age of a Twitterer's account and the list of the twitterer's followees. For each followee, we find out their name and if they are followed back by the twitterer. The result is stored in a CSV file illustrated in Table 2.

XML Tags	Information Extracted
Name	Twitterer's name
Screen_name	Twitterer's username
ID	Twitterer's profile id, WEST use Twitter ID and Twitter4J to see whether a twitterer follow another twitterer and to get the age of a twitterer's account
Followers_count	Total number of followers
Friends_count	Total number of followees/friends
Description	Twitterer's profile description
Tweet_count	Total number of tweets
Retweet_count	Total number of tweets being retweeted
Statuses_count	Total number of status updates
Retweeted	Whether the tweet has been retweeted
In_reply_to_screen_name	The screen name of the Twitterers mentioned in a tweet
Created_at	Date & time a tweet is posted
Text	The content of a tweet

Table 1. List of information extracted from each twitterer XML file

Followee ID	Followee Name	Is Twitterer Follows Followee
3273499957	CloudCommerceCO	FALSE
1207668950	greatdeals2bid4	FALSE
81489175	tgparker2009	TRUE
Age of Twitterer Account In Days	2646	

Table 2. The table representation of the CSV file storing the twitterer's age of account and followee list

3.2 Number of Recent Tweets

It is computationally expensive to analyze every tweet that a twitterer send to decide if the account is a spammer account. Furthermore, one solution for tackling the drift problem of twitter spams is by re-training a spammer classifier model every day based on the new spam tweets. Consequently, anti-spammer detection system must be able to identify spammer from the least number of recent tweets possible. Different researchers extract features from different number of recent tweets. For instance, [19] used 200 tweets and achieved 81% precision while [7] extracts feature from 100 most recent tweets and obtained 95.7% precision. However, it is still unclear whether the difference in performance is due to the differing number of recent tweets or due to the set of features and dataset that they used. As shown in Section 4, with WEST, researchers can easily define the number of recent tweets to be included from the dataset and evaluate whether changing that number increases the model's performance.

3.3 Feature Extraction

In WEST, users can either select which features they want to extract from the dataset, or select the name of an existing spammer detection system. When the user selects the name of an existing anti spammer system, WEST automatically selects the set of features used by that system. WEST also allows for new features to be added by the user and presently there are 17 systems [1, 2, 4, 6–19] and 173 features. Each of those features can be grouped into four types: profile, activities, relations, and tweet contents.

Profile features are information obtained from the twitterer's profile page such as screen name, description, age of the account, profile's URL, and reputation.

Relations features represent the twitterer's friendship status and activities like followers, followee, friends, bi-directional links (followers that is followed by the twitterer) and interaction (twitterer reply or mention of a follower or a non-follower's name or tweet).

Content features capture all the linguistic properties of the text in a tweet such as URLs, URLs to a social media domain, hashtags, mentions, retweets, special characters (e.g. exclamation marks, question marks, blank spaces), alphanumeric characters, capital letters, consecutive words, non-dictionary words, named entity (places, organization, people), and spam words.

Activity features are acquired from the twitterer's general activities like tweets, duplicate tweets, time a tweet has been posted and the device used to post a tweet. WEST implemented three ways of determining level of similarity between tweets: tweet cluster [18], cosine similarity [10] and minimum distance [14].

For each of those extracted information, WEST obtains the sum, minimum, maximum, median, average, and standard deviation of that feature or the unique instances of that feature appearing over a certain period of time (hours, days, weeks) or over a certain number of words or tweets. For example, total number of spam words on screen name, ratio of follower to following, median number of hashtags per word, average number of unique URLs on a tweet, maximum idle time between tweets, and total number of tweets posted between 3pm and 4pm.

3.4 Build and Evaluate Model

As WEST relies on WEKA to perform feature selection and classification, all features retrieved from every twitterer in the dataset is stored in an ARFF file format. This enables users to either use WEKA through WEST to perform their machine learning tasks or to use it directly on WEKA. Just like in WEKA, users can also add new feature selection and classification techniques.

4 Experimental Results

This section illustrates how we can use WEST to answer the three questions mentioned in Section 3. Feature selection is a step of selecting features that are more relevant to a model to improve the accuracy of a system [23]. Theoretically, we should be able to find the most effective and efficient set of features by performing feature selection. To prove this hypothesis, we need to first find the best model generated by performing feature selection and then compare it with the model generated by the existing anti-spammer model. Because we are interested in knowing the impact of the number of recent tweets on the model's performance, we will also compare the result and the attributes selected by the model generated from various number recent tweets. Section 4.1 describes the dataset, classifiers and evaluation criteria that we will use in our experiment. The results of the feature selection models obtained from varying number of recent tweets are presented in Section 4.2, and are compared with the results of the existing spammer detection systems in Section 4.3.

4.1 Experimental Setup

Dataset. The dataset collected by [18] contains the profile and 100 tweets of 7,549 twitterers separated into 315 spammers and 7234 hams. Because some of those accounts are no longer available or are missing information that we need such as Age of Account or Bi-directional links, only 1729 (206 spam and 1523 ham) are usable. We split the dataset into training and test sets. 70% of the spammers and 70% of the ham twitterers are used as training set. The rest are used for testing.

Classifiers. Five most commonly used classifiers for detecting spammers are Support Vector Machine (SVM), Decision Tree (DT), Naïve Bayes (NB), k-Nearest Neighbors (KNN) and Random Forest (RF). To find the most effective and efficient model, we will classify each selected set of features with each of those classifiers.

Evaluation Criteria. To measure the performance of a model, we will use Time, Accuracy, and Spammers' True Positive Rate (TP), False Positive Rate (FP), Precision, and F_1-measure. The time here refers to the total time required to extract features, perform feature selection, build training models and classify all test instances. As mentioned by [10] and seen in our dataset, the number of ham twitterers is much greater than that of spammers, i.e. we have a class imbalance problem. Thus, the most *effective* model is the one that can identify spammers with a high true positive rate and low false positive rate. Although many researchers have proposed various systems for detecting spammers in Twitter, none has mentioned the time it requires to achieve those. Considering that on average there are 6000 tweets sent per second [22], it is essential to have an anti-spammer system that enables legitimate twitterers quickly determine if a tweet is sent by a spammer. Hence, the most *efficient* model should be able to distinguish spammers effectively in the shortest amount of time possible. ANOVA, t-test and equivalence testing will be used to help us identify whether there are significant differences in the performance of each model, and statistically speaking which one is the best.

4.2 Best Feature Selection Model (FS)

To see whether the number of recent tweets used affect the performance of the spammer detection system, we compared the performance of the models extracted from 20, 50, 100, 150 and 200 recent tweets. For each of them, we obtained the Top 10 attributes based on their Information Gain values and classified it using the five aforementioned classifiers.

Table 3 displays the Spammers True Positive values and the bolded value signifies the highest TP value that each subset can obtain. ANOVA result shows that there is significant difference in 95% confidence level (P-value = 0.0016) between the number of recent tweets. Although there is no significant difference between 20, 50, 100 and 150 recent tweets, t-test shows that the TP rate value from 100 recent tweets is significantly better than 200 Recent tweets (P-value = 0.02). Through equivalence testing, we found that the model obtained from 100 recent tweets is the best, followed by the model for 20, 50, 150 and 200 recent tweets.

	Number of Recent Tweets				
	20	**50**	**100**	**150**	**200**
NB	**93%**	**92%**	81%	**61%**	23%
SVM	0%	0%	0%	0%	0%
KNN	69%	62%	**100%**	13%	21%
DT	67%	61%	**100%**	37%	**26%**
RF	71%	65%	**100%**	40%	22%

Table 3. Spammers True Positive Rate results obtained by using Top 10 attributes obtained from the various number of recent tweets

From Table 4, we can see that regardless of the number of recent tweets, nine out of the Top 10 attributes are the same and the same attributes are selected when we used 20 or 50 recent tweets. Furthermore, majority of the attributes in the list are idle time related features. *Idle time* is the length of time interval between two tweets while *Mean Idle time per tweet* ($\frac{total\ idle\ time}{total\ number\ of\ tweets}$) and *Max Idle time per tweet* ($\frac{maximum\ idle\ time}{total\ number\ of\ tweets}$) represent the relationship between twitterer's idle time and the total number of posts. This supports the findings by [2] that on average spammers tend to have more posts and less idle time between posts.

	20 Tweets	50 Tweets	100 Tweets	150 Tweets	200 Tweets
1	Max. idle time per tweet	Mean idle time per tweet	Mean idle time per tweet	Mean idle time per tweet	Mean idle time per tweet
2	Mean idle time	Mean idle time	Mean idle time	Mean idle time	Mean idle time
3	Mean idle time per tweet	Max. idle time per tweet	Max. idle time per tweet	Tweet similarity - Cosine	Mean number of characters
4	Std. dev. of idle time	Std. dev. of idle time	Tweet similarity - Cosine	Max. idle time per tweet	Tweet similarity – Cosine
5	Tweet similarity - Cosine	Tweet similarity - Cosine	Std. dev. of idle time	Mean number of characters	Mean number of words
6	Max. idle time	Max. idle time	Mean number of characters	Mean number of words	Max. idle time per tweet
7	Mean number of words	Mean number of words	Mean number of words	Std. dev. of idle time	Max. idle time
8	% of followers per followees	% of followers per followees	Max. idle time	Max. idle time	Std. dev. of idle time
9	Age of account	Age of account	% of followers per followees	% of followers per followees	% of followers per followees
10	Mean number of characters	Mean number of characters	Mean number of URL per word	Fraction of tweets with spam words	Mean number of numeric characters

Table 4. Top 10 subset of features extracted from 20, 50, 100, 150 and 200 recent tweets in descending order based on their Information Gain values

As we used more number of recent tweets, the more information can be captured from the tweet content related features and so they have higher information gain values. For instance, *Mean of Number of Characters* is rank 10 in 20 and 50 recent tweets, but it moves up to rank 7 in 100 recent tweets, rank 5 in 150 recent tweets, and rank 3 in 200 recent tweets. In Table 4, we have shaded the attributes that do not appear in the list of attributes for the other number of recent tweets.

Spammer's accounts normally are banned by the legitimate twitterers and Twitter [1], so their *Age of Accounts* is smaller than normal twitterers. However, when we increased the number of recent tweets to 100, there are more tweets containing the URLs than 20RT and 50RT, so the *Mean number of URL per word* feature can captures more information with respect to the class target.

[4] says that spammers tend to post more URLs than a normal twitterers. In fact, he found that 95% of spammers' tweets contain URLs. [16] suggests using the *URL Rate* ($\frac{total\ number\ of\ URLs}{total\ number\ of\ tweets}$) to identify spammers. However, this feature was not in the best Top 10 subset of features. By looking at the tweet contents in the dataset, we found that spammers usually send tweets containing a URL with similar sort of text, for example "*#FREE PDF to Excel Converter http://t.co/XfouPlN*" and "*#FREE PDF to Word Converter http://t.co/XfouPlN*". Legitimate twitterers will have different text content and a different total number of words accompanying a URL that they post in their tweets. Therefore, *URL rates* are less effective than the *Mean Number of URLs per Word* because it ignores the contextual information surrounding the tweet and just counts the number of URLs.

Although it is good to check for the occurrence of spam words in tweets, *Fraction of tweets with spam words* is not a strong feature because it is impossible to create an exhaustive list of spam words. Spammers create new spam words all the time and it takes a long time to perform string matching on each tweet [6]. Similarly, *mean number of numeric characters* in 200 recent tweets is not a good feature as the difference between the number of numeric characters used by spammers and normal twitterers are not significant—1 numeric characters per tweets for hammers and 1.5 for spammers.

4.3 FS vs. Other Existing Spammer Detection Systems

We can compare the best models obtained via feature selection (FS) with the 17 systems we have implemented in WEST. However, due to space limitation in this paper, we choose to compare FS with the five representative systems including [1, 6, 7, 8, 13]. As shown in Table 5, we compare each of the systems against the six evaluation criteria including execution time (in minutes), accuracy, TP, FP, precision, and F_1-measure. ANOVA results show that there is significant difference of 95% confidence between the models in terms of TP (P-value = 0.0003), precision (P-value = 0.0070) and F_1-measure (P-value = 0.0001). There is no significant difference in terms of accuracy (P-value = 0.5326) and FP (P-value = 0.5581). T-test and equivalence testing show that FS is the best model and thus supporting our hypothesis that the most efficient and effective model is the model generated through feature selection.

	FS	**[1]**	**[6]**	**[7]**	**[8]**	**[13]**
Number of Recent Tweets	100	200	200	100	200	20
Execution time (min)	**16**	565	35	**30**	897	158
Accuracy	**100%**	93%	93%	**94%**	92%	91%
TP	**100%**	**65%**	61%	46%	17%	42%
FP	**0%**	2%	1%	**0%**	**0%**	**0%**
Precision	**100%**	65%	76%	72%	13%	**80%**
F_1	**100%**	**60%**	55%	51%	15%	47%

Table 5. Evaluation results for FS and the selected five existing anti spammers models

[8]'s model is worse performing ranked least efficient and effective. With this model, the system must check whether each tweet contains named entities, social media domains or non-dictionary words. It is not only time-consuming but also requires the system to keep exhaustive and up-to-date list of named entities, social media domains and dictionary words. Hence, it does not perform well on our dataset.

Although the accuracy and precision of the [7]'s and [13]'s systems are good, their TP and F_1-measure are low because they cannot handle the class imbalance problem, i.e. most instances are classified as normal twitterers. No content or idle-time related features are included and so they cannot distinguish spammers from legitimate twitterers.

Out of all the existing systems we have evaluated, [6]'s model is the most efficient. Many content related features, such as *Mean number of URL per word* and *Maximum number of words*, included in the model helped the system obtain quite good Accuracy, Spammers' TP, FP, Precision, and F_1-measure in a short amount of time. However, we can improve the performance of the model even more by including *Tweet Similarity* and time-relation features such as *Mean idle time per tweet*. This is because spammers tend to produce many tweets in a short period with duplicated contents.

Compared with [6], [1]'s system uses the same number of recent tweets and produces similar accuracy, Spammer's TP, FP and F_1-measure, nevertheless the [1]'s system takes longer time to build because it extracts many more features from the tweets than all the other systems do.

Generally, spammers will follow many accounts but almost none of them will follow them back. Nevertheless, it is not enough to calculate *number of followees*, *number of followers* and *Reputation* ($\frac{Number\ of\ follower}{Number\ of\ follower+Number\ of\ followee}$) to identify spammers because they can just buy more followers to evade these features [15]. Replacing these features in [1, 7, 13] with *% of followers per followees* or *Total Number of Bi-Directional Link* will improve the model's performance.

Furthermore, spammers tend to post many tweets containing the same URLs to increase its chance of being clicked by the legitimate twitterers and so their *Ratio Unique URL per Tweet* would be small. We can improve the performance of [1, 6–8, 13]'s models by replacing their URLs related features with *Ratio Unique URL per Tweet*.

5 Conclusion

This paper studied a number of works in spam detection for Twitter in order to build a framework for comparing and evaluating their performance, and therefore we proposed WEST as an evaluation workbench for researchers and users to measure the performance of their proposed techniques against the existing ones. We have included 172 content-based and feature-based features in our study making it easier for researchers to quickly create and evaluate their models against existing models. Our experiments found that the most effective and efficient set of features for detecting spammers are idle time related activity and tweet content features. The number of recent tweets used can significantly affect the model's performance. In the future, we will consider other types of features such as graph/network based features in the tool as well as creating a user interface to make the workbench more user friendly.

References

1. Benevenuto, F., Magno, G., Rodrigues, T., Almeida V.: Detecting spammers on Twitter. In: Proceedings of the 7th Annual Collaboration, Electronic messaging, Anti-abuse and Spam Conference (2010).
2. Gee, G., Hakson, T.: Twitter Spammer Profile Detection. Available online: cs229.stanford.edu/proj2010/GeeTeh-Twitter Spammer Profile Detection.pdf, (2010).
3. Chen, C., Zhang, J., Xiang, Y., Zhou, W.: Asymmetric self-learning for tackling twitter spam drift. In Computer Communications Workshops, pp. 208–213 (2015).
4. Lee, K., Eoff, B.D., Caverlee, J.: Seven months with the devils: A long-term study of content polluters on Twitter. In: Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (2011).
5. Hu, X., Tang, J., Liu, H.: Online social spammer detection. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp. 59-65 (2014).
6. Wang, B., Zubiaga, A., Liakata, M., Procter, R.: Making the most of tweet-inherent features for social spam detection on Twitter. Available online: https://arxiv.org/pdf/1503.07405.pdf, (2015).
7. Mccord, M., Chuah, M.: Spam detection on twitter using traditional classifiers. In: Proceedings of the International Conference on Autonomic and Trusted Computing (2011).
8. Alonso, O., Carson, C., Gerster, D., Ji, X., Nabar, S. U.: Detecting uninteresting content in text streams. In: Proceedings of the SIGIR Crowdsourcing for Search Evaluation Workshop (2010).
9. Burnap, P., Javed, A., Rana, O. F., Awan, M. S.: Real-time classification of malicious URLs on Twitter using Machine Activity Data. In: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (2015).
10. 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 (2010).
11. Lupher, A., Engle, C., Xin, R.: Feature Selection and Classification of Spam on Social Networking Sites. Available online: bid.berkeley.edu/cs294-1-spring12/images/archive/6/6a/20120515031244!Spam-lupher-engle-xin.pdf, (2012)
12. Amleshwaram, A. A., Reddy, N., Yadav, S., Gu, G., Yang, C.: Cats: characterizing automation of twitter spammers. In: Proceedings of the 5th International Conference on Communication Systems and Networks (2013).
13. Wang, A. H.: Don't follow me: Spam detection in Twitter. In: Proceedings of International Conference on Security and Cryptography (2010).
14. Wang, A. H.: Detecting spam bots in online social networking sites: a machine learning approach. In: Data and Applications Security and Privacy XXIV, pp. 335-342 (2010).
15. Yang, C., Harkreader, R. C., Gu, G.: Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In: Proceedings of the International Workshop on Recent Advances in Intrusion Detection (2011).
16. Lin, P.-C., Huang, P.-M.: A study of effective features for detecting long-surviving Twitter spam accounts. In: Proceedings of the 15th International Conference on the Advanced Communication Technology (2013).
17. Song, J., Lee, S., Kim, J.: Spam Filtering in Twitter Using Sender-Receiver Relationship. In: Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection, pp. 301-317 (2011).

18. Chakraborty, A., Sundi, J., Satapathy, S.: SPAM: A Framework for Social Profile Abuse Monitoring. Available online: http://www3.cs.stonybrook.edu/~aychakrabort/courses/cse508/report.pdf, (2012).
19. Dhingra, A., Mittal, S.: Content Based Spam Classification in Twitter using Multi-Layer Perceptron Learning. In: International Journal of Latest Trends in Engineering and Technology, 5(4) (2015).
20. Piatetsky-Shapiro, G.: KDnuggets news on SIGKDD service award. Available online: www.kdnuggets.com/news/ 2005/n13/2i.html, (2005).
21. Twitter4J. Available online: twitter4j.org/en/ (2007).
22. Sayce, D.: Number of tweets per day? Available online: http://www.dsayce.com/social-media/tweets-day/ (2017).
23. Kira, K., Rendell, L. A.: A practical approach to feature selection. In: Proceedings of the 9th International Workshop on Machine learning (1992).
24. Stringhini, G., Kruegel, C., Vigna, G.: Detecting spammers on social networks. In: Proceedings of the 26th Annual Computer Security Applications Conference (2010).
25. Meda, C., Bisio, F., Gastaldo, P., Zunino, R.: Machine Learning Techniques applied to Twitter Spammers Detection. In: International Carnahan Conference on Security Technology (2014).
26. Thomas, K., Grier, C., Song, D., Paxson, V.: Suspended accounts in retrospect: an analysis of Twitter spam. In: Proceedings of ACM SIGCOMM conference on Internet Measurement Conference (2011).
27. Weng, J., Lim, E.-P., Jiang, J., He, Q.: TwitterRank: finding topic-sensitive influential twitterers. In: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (2010).