A Performance Evaluation of Machine Learning-Based Streaming Spam Tweets Detection

Chao Chen, Jun Zhang, *Member, IEEE*, Yi Xie, Yang Xiang, *Senior Member, IEEE*, Wanlei Zhou, *Senior Member, IEEE*, Mohammad Mehedi Hassan, Abdulhameed AlElaiwi, and Majed Alrubaian

***Abstract*—The popularity of Twitter attracts more and more spammers. Spammers send unwanted tweets to Twitter users to promote websites or services, which are harmful to normal users. In order to stop spammers, researchers have proposed a number of mechanisms. The focus of recent works is on the application of machine learning techniques into Twitter spam detection. However, tweets are retrieved in a streaming way, and Twitter provides the Streaming API for developers and researchers to access public tweets in real time. There lacks a performance evaluation of existing machine learning-based streaming spam detection methods. In this paper, we bridged the gap by carrying out a performance evaluation, which was from three different aspects of data, feature, and model. A big ground-truth of over 600 million public tweets was created by using a commercial URL-based security tool. For real-time spam detection, we further extracted 12 lightweight features for tweet representation. Spam detection was then transformed to a binary classification problem in the feature space and can be solved by conventional machine learning algorithms. We evaluated the impact of different factors to the spam detection performance, which included spam to nonspam ratio, feature discretization, training data size, data sampling, time-related data, and machine learning algorithms. The results show the streaming spam tweet detection is still a big challenge and a robust detection technique should take into account the three aspects of data, feature, and model.**

I. Introduction

ONLINE social networks (OSNs), such as Twitter, Facebook, and some enterprise social network [1], have become extremely popular in the last few years. Individuals spend vast amounts of time in OSNs making friends with people who they are familiar with or interested in. Twitter, which was founded in 2006, has become one of the most popular microblogging service site. Nowadays, 200 million Twitter users generate over 400 million new tweets per day [2].

Manuscript received June 13, 2015; revised November 30, 2015; accepted January 03, 2016. This work was supported by the ARC Linkage Project LP120200266 and by the Natural Science Foundation of China under Grant NSFC61401371. The work of Y. Xie was supported by the Natural Science Foundation of both Guangdong Province and the National Sector of China under Grant 2014A030313130.

C. Chen, J. Zhang, Y. Xiang, and W. Zhou are with the School of Information Technology, Deakin University, Melbourne 3125, Vic., Australia (e-mail: chao.chen@deakin.edu.au; jun.zhang@deakin.edu.au; yang.xiang@deakin.edu.au; wanlei.zhou@deakin.edu.au).

Y. Xie is with the School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510000, China (e-mail: xieyi5@mail.sysu.edu.cn).

M. Hassan, A. AlElaiwi, and M. Alrubaian are with the College of Computer and Information Sciences, King Saud University, Riyadh 12372, Saudi Arabia (e-mail: mmhassan@ksu.edu.sa; aalelaiwi@ksu.edu.sa; malrubaian.c@ksu.edu.sa).

Digital Object Identifier 10.1109/TCSS.2016.2516039

Unfortunately, the proliferation of Twitter also contributes to the growth of spam. Twitter spam, which is referred as unsolicited tweets containing malicious links that directs victims to external sites containing malware downloads, phishing, drug sales, or scams, etc. [3], has not only affected a number of legitimate users but also polluted the whole platform. During the period of Australian Prime Minster Election (August 2013), the Australian Electoral Commission (AEC) published an alert that confirmed its Twitter account @AusElectoralCom was hacked. Many of its followers received direct spam messages which contained malicious links [4]. The ability to sort out useful information is critical for both academia and industry to discover hidden insights and predict trends on Twitter. However, spam significantly brings noise into Twitter [5].

Consequently, the research community, as well as Twitter itself, has proposed some spam detection schemes to make Twitter as a spam-free platform. For instance, Twitter has applied some "Twitter rules" to suspend accounts if they behave abnormally. Those accounts, which are frequently requesting to be friends with others, sending duplicate content, mentioning others users, or posting URL-only content, will be suspended by Twitter [6]. Twitter users can also report a spammer to the official @spam account. To automatically detect spam, machine learning algorithms have been applied by researchers to make spam detection as a classification problem [3], [7]–[23]. Most of these works classify a user is spammer or not by relying on the features which need historical information of the user or the exiting social graph. For example, the feature, "the fraction of tweets of the user containing URL" used in [3], must be retrieved from the users' tweets list; features such as, "average neighbors' tweets" in [13] and "distance" in [17] cannot be extracted without the built social graph. However, Twitter data are in the form of stream, and tweets arrive at very high speed [24]. Despite that these methods are effective in detecting Twitter spam, they are not applicable in detecting streaming spam tweets as each streaming tweet does not contain the historical information or social graph that are needed in detection.

Alternatively, classifying a streaming tweet instead of a Twitter user to spam or nonspam is more realistic in the real world [3], [25]. In this scenario, only information available in a tweet that was captured by Twitter's Streaming API can be used for classification. In order to better understand ML algorithms' power in classifying streaming spam tweets, we provided a fundamental evaluation in this work. To achieve this goal, we collected a large number of tweets. This data contained more than 600 million tweets, in which we further labeled 6.5 million

spam tweets by using Trend Micro's Web Reputation Service [26]. We also extracted some straightforward features for each tweet and examined some ML algorithms' performance on the detection of spam from various aspects. In summary, our contributions of this paper are follows.

1) We created a big ground-truth for the research on spam tweet detection. We reported the impact of the data related factors, such as spam to nonspam ratio, training data size, and data sampling, to the detection performance.
2) We extracted 12 lightweight features for streaming tweet spam detection and found feature discretization is important to spam detection performance. A new finding is that the features of spam tweets are time varying.
3) We investigated six machine learning algorithms to build up the tweet spam detection model and reported the behavior of these models under different experiment settings.

This paper is organized as follows. Section II presents a thorough review on characterizing and detecting Twitter spam. In Section III, the big ground-truth used in our work is introduced. Section II-B provides the fundamental evaluation of existing machine learning-based streaming spam detection methods from various aspects. Finally, Section V concludes this work.

II. Related Work

The severe spam problem on Twitter has already drawn researchers' attention. Some researchers have studied the characteristics of spam, after that, several significant works to detect Twitter spam have been proposed. As a result, we discuss prior related works by organizing them into two categories: 1) characterizing and 2) detecting spam on Twitter.

A. Characterizing Twitter Spam

In order to better understand Twitter spam, some in-depth analysis has been carried out. In 2010, Grier *et al.* analyzed 25 million URLs from 200 million public tweets, and found that 2 million URLs were spam, which accounts for 8% of all crawled unique URLs [27]. They further found that Twitter spam was much more harmful than email spam with a click-through rate of 0.13%, compared to a much lower rate (0.0003%–0.0006%) for email spam. Grier *et al.* also examined the performance of blacklists, and the results indicated that blacklists' delay failed to stop the spread of spam on Twitter.

In 2011, Thomas *et al.* analyzed spam characteristics on a huge dataset of 1.8 billion tweets, of which 80 million were spam [6]. They characterize the behavior of spammers and found five large campaigns. However, three of them lured victims to reputable online shopping such as Amazon, which blurs the line what constitutes spam on social networks. More interestingly, their results indicated that 77% spam accounts were suspended within one day of their first tweet and 92% spam accounts only last within three days. Under such pressure, 89% spam accounts were rarely setting up social connections with users. Instead, 52% accounts made use of unsolicited mention and 17% accounts were hijacking trending topics.

To illustrate the relationship of spammers, Yang *et al.* first carried out an analysis on the cyber criminal ecosystem, which was composed of criminal account community and criminal supporters community on Twitter [15]. By analyzing the sampled criminal account community of 2060 accounts, they found that the inner social relationship of this community is like this: 1) criminal accounts are forming a small world and 2) criminal hubs, which sit in the centre of the social graph, are more likely to follow criminal accounts. Based on the social relations, they have proposed a criminal accounts inference algorithm, which can find unknown spammers by using a set of known spammers.

B. Detecting Twitter Spam

In response to detect Twitter spam, there have been a few works introduced. Most of these works are utilizing machine learning algorithm to separate spam and nonspam. Some preliminary works, including [3], [19], [20], [28], made use of account and content features, such as account age, number of followers or followings, URL ratio, and the length of tweet to distinguish spammers and nonspammers. These features can be extracted efficiently but also fabricated easily. Consequently, some works [13], [17] proposed robust features which rely on the social graph to avoid feature fabrication. Song *et al.* extracted the distance and connectivity between a tweet sender and a receiver to determine whether the tweet is spam or not [17]. While in [13], Yang *et al.* proposed more robust features based on the social graph, such as local clustering coefficient, betweenness centrality, and bidirectional links ratio. Such features were proved to be more discriminative than the features in previous works. However, collecting these features are very time-consuming and resource-consuming, as the Twitter social graph is extremely huge. In addition, it is unrealistic to collect those features as tweets are incoming in the form of stream.

Instead, [7] and [14] solely relied on the embedded URLs in tweets to detect spam. A number of URL-based features were used by [7], such as the domain tokens, path tokens, and query parameters of the URL, along with some features from the landing page, domain name system (DNS) information, and domain information. In [14], they studied the characteristics of correlated URL redirect chains, and further collected relevant features, such as URL redirect chain length and relative number of different initial URLs. These features also show their discriminative power when used classifying spam. However, these two works can only detect spam with URLs, as pointed out by a recent work [18]. The systems will miss the spam with only text or fabricated URLs. [18], thus, proposed a model-based spam detection scheme. They built several models, such as language model and posting time model, for each user. Once the model behaved abnormally, there would be a compromise of this account, and this account might be used for spamming activity by attackers. This method can detect whether an account was compromised or not, but cannot determine the accounts which were created by spammers fraudulently.

Although there are a few works, such as [7] and [14], which are suitable to detect streaming spam tweets, there lacks of a performance evaluation of existing machine learning-based streaming spam detection methods. In this paper, we aim to

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


Fig. 1. Tweet JSON object.

bridge the gap by carrying out a performance evaluation, which was from three different aspects of data, feature, and model.

III. A Big Dataset of Streaming Spam Tweets

A dataset with *ground-truth* (annotated instances with class labels for referencing) is needed to perform a number of challenging machine learning-based streaming spam tweets detection tasks. However, we found that no datasets are publicly available specially for our task. Although there are a few dataset published by some researchers [3], [13], the labeled instances are spammers instead of spam tweets. As a result, we decided to collect streaming tweets and generate the ground-truth. We will also make this dataset available for others researchers to use. In this section, we will describe our large dataset with over 600 million tweets, including more than 6.5 million spam tweets.

A. Collection Procedure

We used Twitter's Streaming API [29] to collect tweets with URLs. The public Streaming API provides real-time access to 1% of all the public tweets, but no access to the tweets sent by protected accounts or direct messages. A tweet is retrieved as JSON format (see Fig. 1 for an incomplete Tweet JSON example), which is very simple and easy to be parsed as each line of this format represents an object [24]. The returned tweet by the Streaming API contains many attributes of the tweets, such as the text, "the number of retweets," "contained hastags, URLs," and associated Twitter user, such as "the number of tweets," "account generated time," and "the number of friends." [30].

While it is possible to use Twitter to send spam and other messages without using URLs, the majority of spam and other malicious messages on the Twitter platform contain URLs [18]. In the thousands of spam tweets, which were inspected manually during the research, we found only a few tweets without URLs, which could be considered as spam. In addition, spammers mainly use embedded URLs to make it more convenient to direct victims to their external sites to achieve their goals, such as phishing, scams, and malware downloading [16]. Therefore, we restricted this research to tweets with URLs. During the collection period, we collected a total of over 600 million tweets with URLs [31].

B. Ground Truth

Currently researchers are using two ways to generate groundtruth, manual inspection [3], [19], and blacklists filtering, e.g. Google SafeBrowsing, [13], [15], [27], [32]. While manual inspection can label a small amount of training data, it is very time- and resource-consuming. A large group of people is needed to help during the process. Although human intelligence task (HIT) websites can help to label the tweets, it is also costly and sometimes the results are doubtful [33]. Others apply existing blacklisting service, such as Google SafeBrowsing to label spam tweets. Nevertheless, these services' API limits make it impossible to label a large amount of tweets.

We used Trend Micro's Web Reputation Service to identify which URLs were deemed malicious tweets. Trend Micros WRS maintains a large dataset of URL reputation records, which are derived from Trend Micro customer opt-in URL filtering records. WRS is dedicated to collect the latest and the most popular URLs, to analyze them, and then to provide Trend Micro customers with real-time protection while they are surfing the web. The maintaining team of WRS is using many frontier technologies to analyze and labeling URL. They will even manually visit the URL if necessary. WRS is trusted by Trend Micros large user base. According to a third party investigation carried out recently, the protection rate of define WRT is 99.8%. Thus, the results are trustworthy, as well as the analysis. Hence, through checking URLs with the WRS service, we are able to identify whether a URL is malicious and the categories a URL belongs to. We define those which contain malicious URLs as Twitter spam. In our dataset of 600 million tweets, we identified 6.5 million malicious tweets, which accounted for approximately 1% of all tweets.

C. Features

After labeling the spam tweets, we further extracted features from them. Since Twitter's Public Streaming API only returned random public tweets and they were not socially connected, we were not able to build a social graph from the data. As a result, it is not possible for us to extract social graph-based features such as local clustering coefficient, betweenness centrality [13], and distance [17]. Such expensive features are not suitable to be used in real-time detection, despite that they have more

TABLE I
EXTRACTED FEATURES

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

discriminative power in separating spam and nonspam tweets. Moreover, we are specially focusing on detecting the streaming spam tweets; features which can be straightforwardly computed from the tweet itself are preferred. We have totally extracted 12 features from our dataset as listed in Table I.

According to the object where the features were extracted, the 12 features can be divided into two categories, user-based features, and tweet-based features. *User-based features* were extracted from the JSON object "user," such as account_age, which can be calculated by using the collection date minus the account created data. Other user-based features, like no_of followers, no_of followings, no_userfavourites, no_lists, and no_tweets, can be directly parsed from the JSON structure. *Tweet-based features* includes no_retweets, no_hashtags, no_usermentions, no_urls, no_chars, and no_digits. While no_chars and no_digits needs a little computing, i.e., counting them from the tweet text, others can also be straightforwardly extracted.

D. Feature Statistics

To look into the characteristics of these features, we plotted the cumulative distribution function (cdf) of them, as shown in Fig. 2.

We can see from Fig. 2(c) that spammers are involved in more lists than normal users, so as to be exposed more to the public. Naturally, in order to spread more spam tweets, spammers send more tweets compared to nonspammers, as shown in Fig. 2(d). In terms of "number of followings," Fig. 2(e) shows that spammers do like to follow more users than nonspammers. The aim is also to attract more attentions from victims to click their spam links.

As Fig. 2(h) shows, nonspammers use less hashtags than spammers. There are about 80% nonspam tweets do not have hashtags embedded in their sent tweets, whereas the ratio in spam tweets is only 60%. When it comes to the feature "number of characters per tweet," there is not much difference between spam tweets and nonspam tweets. The reason could be that spammers begin to imitate the posting behavior of normal users. Fig. 2(i) shows that spammers tend to use less digits than nonspammers. Due to the limit of pages, we only describe six features' characteristics here. In general, the analysis of these features has showed us their discriminative power to detect Twitter spam.

TABLE II
SAMPLED DATASETS

Dataset	Sampling method	NO. of spam Tweets	NO. of nonspam Tweets
I	Continuous	5000	5000
II	Continuous	5000	95 000
III	Noncontinuous	5000	5000
IV	Noncontinuous	5000	95 000

IV. FUNDAMENTAL EVALUATION OF ML-BASED STREAMING SPAM TWEETS DETECTION

In this section, we evaluate the spam detection performance on our dataset by using six machine learning algorithms, *random forest, C4.5 decision tree, Bayes network, Naive Bayes*, *k-nearest neighbor, and support vector machine*. We also sampled several different datasets to conduct the experiments. The datasets are listed in Table II.

In Table II, we can see that the spam to nonspam ratio is 1:1 in Datasets I and III, whereas the ratio is 1:19 in Datasets II and IV. In previous works, most of the datasets are nearly evenly distributed; the spam to nonspam ration is nearly 1:1. However, Twitter has around 5% spam tweets of all existing tweets in the real world [27]. The evenly distributed dataset cannot represent the Twitter sphere. Consequently, we sampled Datasets II and IV, which has a spam ration of 1:19 to simulate the real world scenario.

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

A. Process of ML-Based Twitter Spam Detection

This section describes the process of Twitter spam detection by using machine learning algorithms. Fig. 3 illustrates the steps involved in building a supervised classifier and detecting Twitter spam. Before classification, a classifier that contains the knowledge structure should be trained with the prelabeled tweets. After the classification model gains the knowledge structure of the training data, it can be used to predict a new incoming tweet. The whole process consists of two steps:

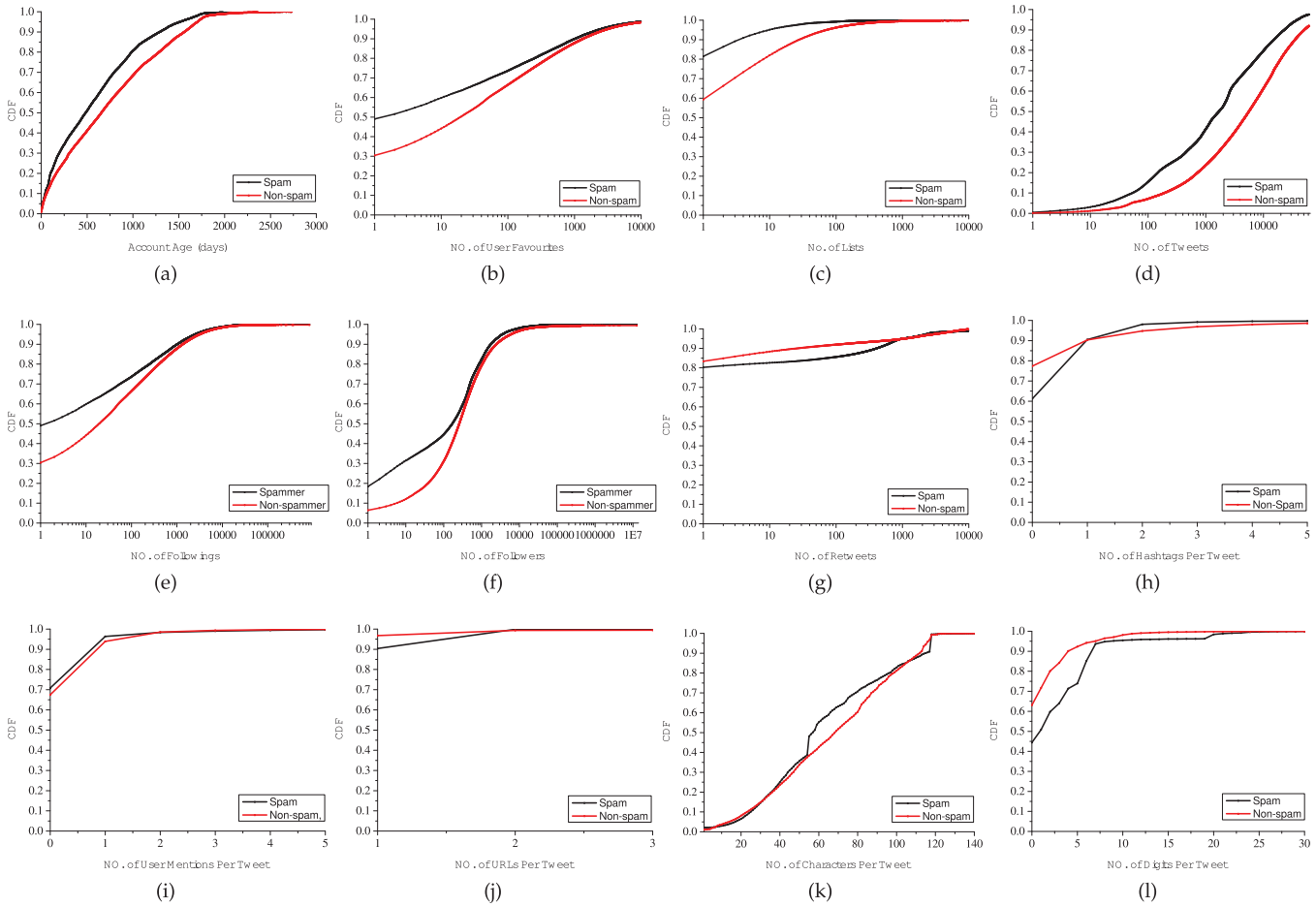


Fig. 2. Cumulative distribution functions of features. (a) Account age. (b) Number of user favorites. (c) Number of lists. (d) Number of Tweets. (e) Number of followings. (f) Number of followers. (g) Number of retweets per Tweet. (h) Number of Hashtags per Tweet. (i) Number of user mentions per Tweet. (j) Number of URLs per Tweet. (k) Number of Characters per Tweet. (l) Number of digits per Tweet.

1) learning and 2) classifying. First, features of tweets will be extracted and formatted as a vector $\vec{\boldsymbol{F}} = \{f_1, f_2, \ldots, f_n\}$. The class labels (spam or nonspam) could be get via some other approaches (like manual inspection). Features and class label will be combined as one instance for training. One training tweet can then be represented by a pair containing one feature vector, which represents a tweet, and the expected result $(\vec{\boldsymbol{F}}, \text{label})$, and the training set is the vector $\vec{\boldsymbol{TS}} = \{(\vec{\boldsymbol{F}}_1, \text{label}_1), (\vec{\boldsymbol{F}}_2, \text{label}_2), (\vec{\boldsymbol{F}}_n, \text{label}_n)\}$. The training set is the input of machine learning algorithm, the classification model will be built after training process. In the classifying process, timely captured tweets $\vec{\boldsymbol{T}} = \{f_1, f_2, \ldots, f_n\}$ will be labeled by the trained classification model.

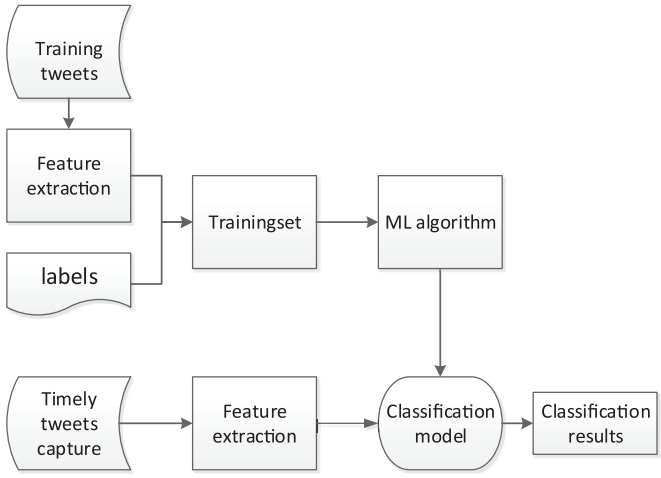


Fig. 3. ML-based spam detection process.

B. Performance Metrics

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

1) Positives and Negatives: Suppose there is a tweet t and the spam class S. The output of the classifier is whether t belongs to S or not. A common way to evaluate the classifier's performance is to use *true positives (TP), false positives (FP), true negatives (TN)*, and *false negatives (FN)* [34]. These metrics are defined as follows.

a) TP tweets of class S correctly classified as belonging to class S.

b) FP tweets not belonging to class S incorrectly classified as belonging to class S.

TABLE III
EVALUATION METRICS

		Predicted Sapm	Predicted Nonsapm
True	Spam	TP	FN
	Nonspam	FP	TN

c) TN tweets not belonging to class S correctly classified as not belonging to class S.
d) FN tweets of class S incorrectly classified as not belonging to class S.

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

In order to measure the ability to detect spam, we also import true positive rate (TPR) and false positive rate (FPR).

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

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{1}$$

b) FPR is defined as the ratio of those nonspam tweets incorrectly classified as belonging to spam class S to the total number of nonspam tweets

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{FN}}. \tag{2}$$

2) Precision, Recall, and F-measure: Literature also uses precision, recall, and F-measure to evaluate per-class performance.

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

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{3}$$

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

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{4}$$

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

$$\text{F-measure} = \frac{2 * \mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{5}$$

C. Impact of Spam to Nonspam Ratio

In this section, we evaluate the impact of spam to nonspam ratio of the above-mentioned machine learning algorithms on Datasets I and II. Each classifier in this set of experiments was trained with a dataset of 1000 spam tweets and 1000 nonspam tweets. Then, these trained classifiers were used to detect spam in the four sampled datasets. As in [13], we also used TPR, FPR, and F-measure to evaluate the performance of these classifiers.

TABLE IV
PERFORMANCE EVALUATION ON DATASETS I AND II

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

TABLE V
CONFUSION MATRIX OF RANDOM FOREST ON BOTH DATASETS

Classified as –>	Spam	Nonspam		Spam	Nonspam
Spam	4645	355		4645	355
Nonspam	282	4718		6766	88234
	Dataset I			Dataset II	

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

To figure out why F-measure drops on Dataset II, Table V outputs the confusion matrix of random forest when evaluated on both datasets. Since the classifiers were trained by the same dataset, we can see that, there was no impact on the TP and FN of spam class when the spam to nonspam ratio was changed, so Recall, which is define as the ratio of the number of tweets classified correctly as spam to the total number of real spam tweets, stayed the same. However, when more nonspam tweets were involved in the test, the number of FP increased exponentially. Thus, the precision, which is define as the ratio of the number of tweets classified correctly as spam to the total number of predicted spam tweets, decreased. As a result, F-measure, which is combination of precision and recall, decreased dramatically due the decrease of precision. Generally, we find that the F-measure of machine learning-based classifiers is quite low as there are much more nonspam tweets than spam tweets.

D. Impact of Feature Discretization

In this section, the impact of feature discretization of selected classifiers, such as Naive Bayes, kNN, and SVM when on discretized and nondiscretized Datasets I and II, is evaluated.

Figs. 4–6 show the TPR, FPR, F-measure, and classification speed of spam detection on Datasets I and II. We can see that the FPR of Naive Bayes decreases dramatically after discretization from 80% to 20% on Dataset I. Similar on Dataset II, the FPR declined from 45% to less than 5%. However, the performance of Naive Bayes also decreases in terms of TPR. The TPR of Naive Bayes drops from 94.5% to 88% and 74.5% to 58%, respectively, on Datasets I and II. When it comes to F-measure, the performance of Naive Bayes increases around 3% and over 20% on Datasets I and II. Overall, feature discretization has positive impact for Naive Bayes, especially when on Dataset II. Similarly, feature discretization can help to improve

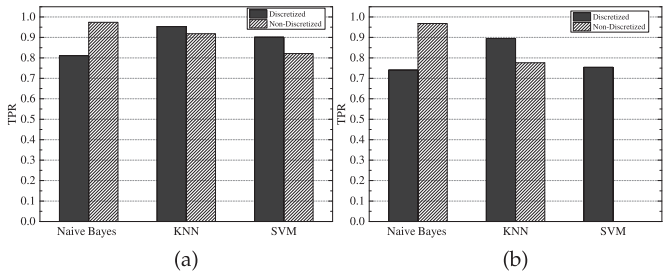


Fig. 4. TPR on spam: (a) Dataset I and (b) Dataset II.

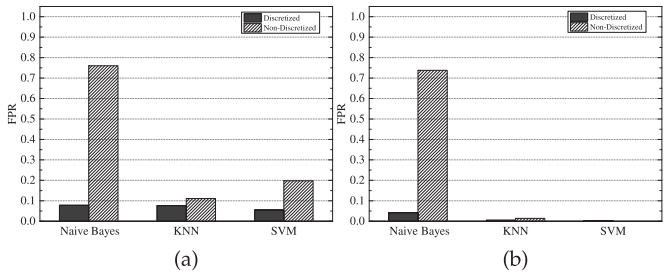


Fig. 5. FPR on spam: (a) Dataset I and (b) Dataset II.

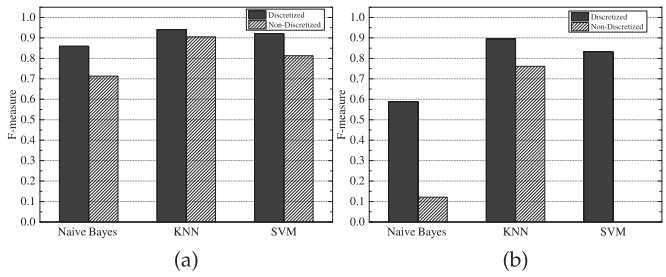


Fig. 6. F-measure on spam: (a) Dataset I and (b) Dataset II.

performance for kNN and SVM on both datasets. For example, the F-measure has been improved 5% for kNN and 10% for SVM on Dataset I. We also note that, although SVM can achieve 75% F-measure on Dataset I, it becomes useless on Dataset II with less than 5% F-measure without feature discretization. However, SVM can achieve over 80% F-measure after discretizing features. In general, feature discretization can improve performance of classifiers for Twitter spam detection.

E. Impact of Increasing Training Data

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

Fig. 7 shows the spam detection performance with increasing training samples on Dataset I. In Fig. 7(a), one can find that random forest outperforms all the other classifiers with TP rate ranging from 78% to 85%, followed by kNN. However, Navie Bayes with discretization has the lowest FP rate, whereas SVM has the highest FP rate. When it comes to F-measure, random forest still ranks as number one among all classifiers, with a range from 70% to 75%.

Fig. 8 reports the spam detection performance with increasing training samples on Dataset II. Unsurprisingly, random forest also performs the best in terms of all three metrics with more than 40% TP rate and less then 1% FP rate. In addition, the increment of F-measure from 100 training samples to 1000 training samples is more than 10% for all classifiers except for Naive Bayes. Especially for random forest, the F-measure increases from 36% to 65%, with an increment of over 30%.

One would expect that the performance of the classifiers will increase with additional training data [35]. However, we find that the performance is relatively stable even with more training data. In Fig. 7(c), we can find that F-measure of these classifiers can reach as high as 80%. However, it cannot be improved further by simply increasing the training data. Specifically, the F-measure rises slightly (less than 3%) for random forest, C4.5 decision tree, and kNN, after the training samples number of 500. There is no growth for Bayes network and SVM in terms of F-measure. Particularly, F-measure of Navie Bayes even drops with more training samples. This phenomenon also happens with Dataset II. For instance, the F-measure of Naive Bayes stays around 30% despite the growth of training samples. We conclude that there is little benefit by simply increasing the training data when the training size has reached a certain size. More preprocesses, such as developing more discriminant features or cleaning training data [36], should be done to further improve the performance.

F. Impact of Different Sampling Method

During our study, we also notice that classifiers' performance is better on the dataset where the tweets are sampled from a continuous period of time than that where the tweets are randomly selected. To further study this, Datasets III and IV are sampled. The samples in Datasets I and II are randomly selected, whereas those in Datasets III and IV are continuous. We also perform ten fold cross-validation on both datasets. The results are shown in Figs. 9 and 10.

The results in Fig. 9(a) indicate that the TP rates of all classifiers on Dataset III arise around 10% compared to the performance on Dataset I, expect Naive Bayes. For example, the TP rates of C4.5 decision tree and kNN are 12% higher on Dataset III than those on Dataset I. In addition, most of these classifiers can reach 80% TP rate; some of them, such as C4.5 decision tree, kNN, and random forest can even have over 90% TP rates when evaluated on Dataset III. Similarly, the FP rates on Dataset III drops significantly, especially for SVM, it drops from nearly 40% to less than 20%, with an decrease of 20%. Most of the classifiers have an FP rate of less than 10%. In terms of F-measure, all classifiers evaluated on Dataset III except Navie Bayes outperform those on Dataset I. Furthermore, several classifiers can have more than 90% F-measure, which is very effective in detection Twitter spam.

Fig. 10 shows the TP rates, FP rates, and F-measures of all the classifiers evaluated on Datasets II and IV. The difference of TP rates on Dataset II and IV is significantly huge, which is around 30%–40%. When it comes to the metric of F-measure, the same difference exists. For instance, the F-measure of random forest evaluated on Dataset IV can reach as high as 95%, which is 30% higher than it on Dataset II. In this set of experiments, we find that Naive Bayes and SVM work badly when

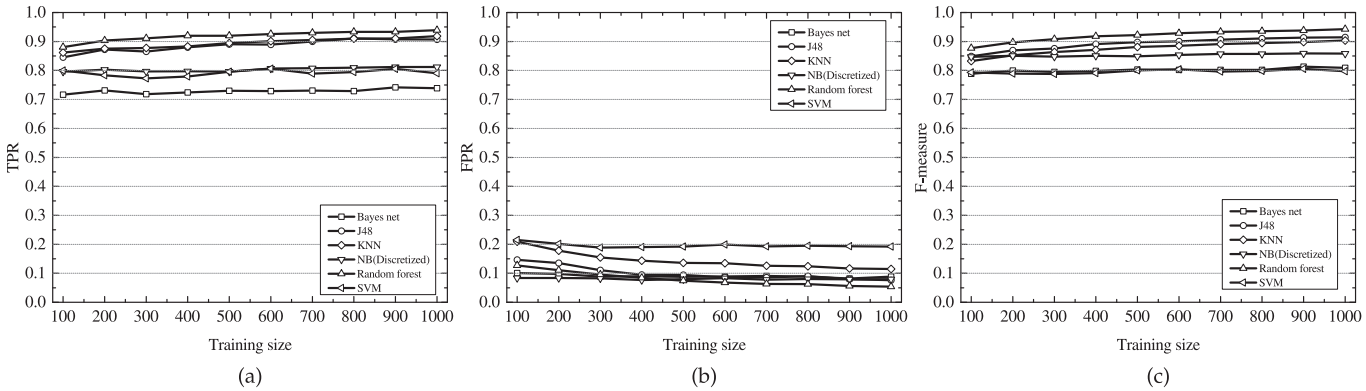


Fig. 7. Spam detection with increasing training size on Dataset I. (a) TP rate. (b) FP rate. (c) F-measure.

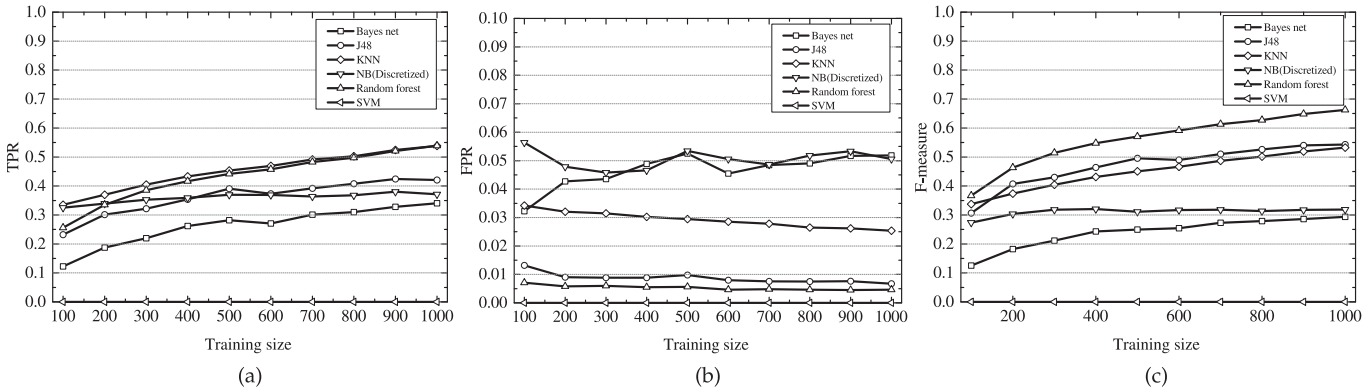


Fig. 8. Spam detection with increasing training size on Dataset II. (a) TP rate. (b) FP rate. (c) F-measure.

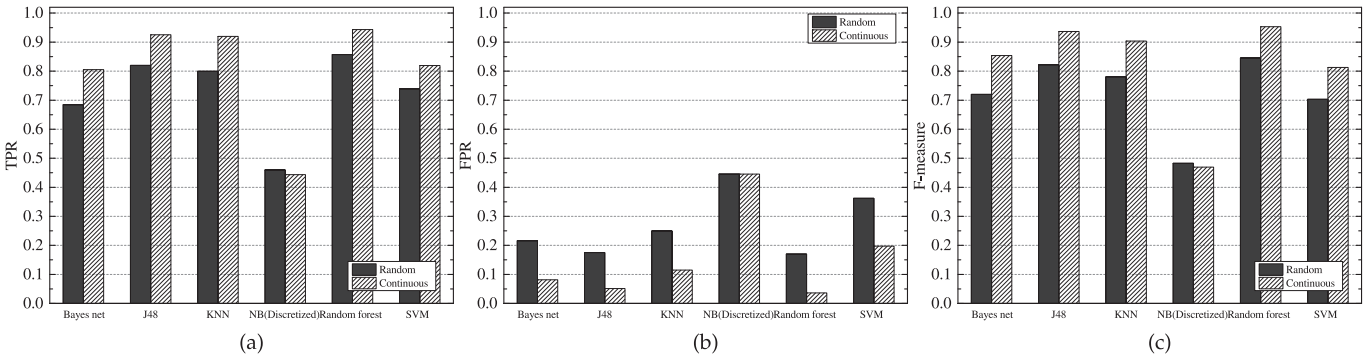


Fig. 9. Spam detection on Dataset I versus Dataset III. (a) TP rate. (b) FP rate. (c) F-measures.

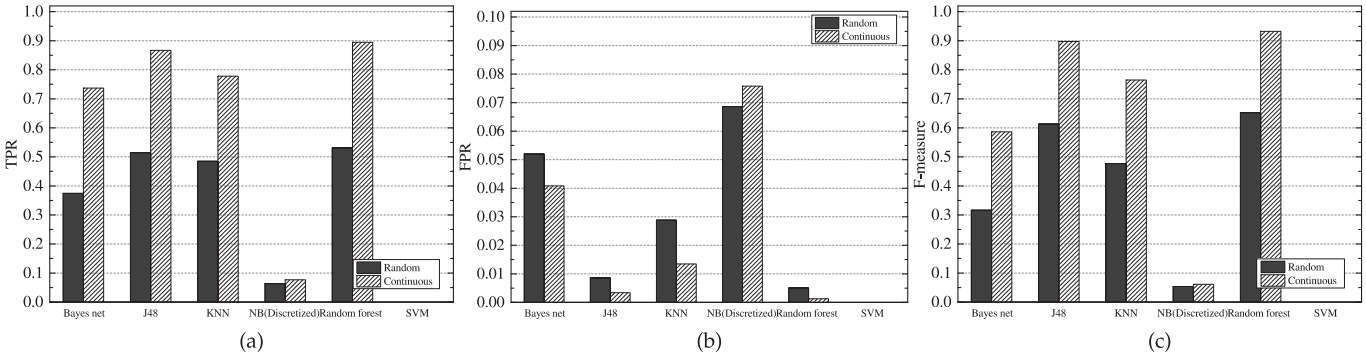


Fig. 10. Spam detection on Dataset II versus Dataset IV. (a) TP rate. (b) TP rate. (c) F-measure.

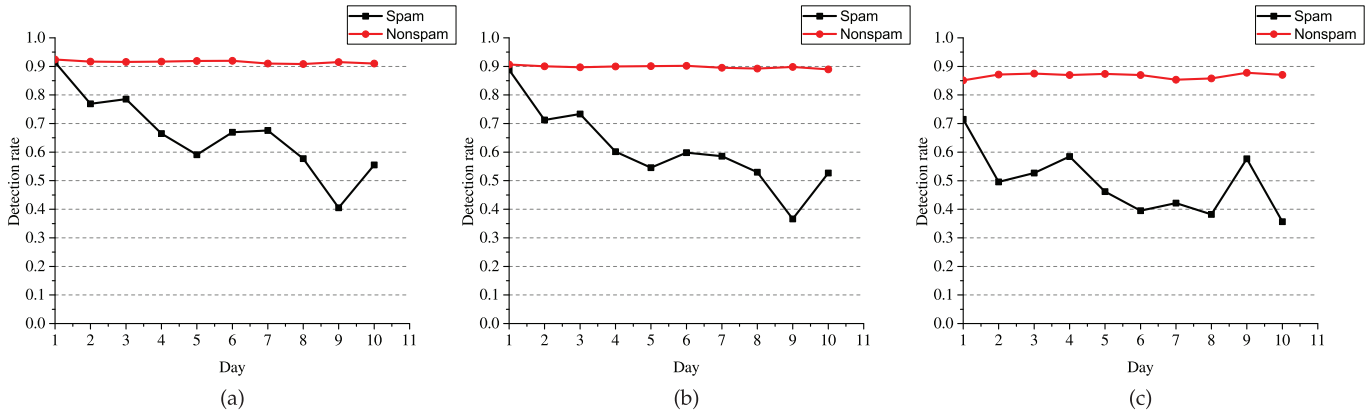


Fig. 11. Trend of detection rate. (a) Random forest. (b) C4.5 decision tree. (c) Bayes network.

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

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

G. Investigation of Time-Related Data

As discussed in the above section, the performance varies when in differently sampled datasets. We believe that "time" plays an important role in this difference. In this section, a serious of experiments are conducted from various kinds of views to investigate the "time-related" issue in detecting streaming spam. In order to perform such evaluation, we sampled a new dataset which is constituted by ten consecutive days' tweets, whereas each day contains 100k spam tweets and 100k nonspam tweets.

1) From the View of Detection Rate: We perform a serious of experiments in this section to show how detection rates of spam and nonspam changes while testing on different days. As in [13], we use detection rate to show the classifier's performance.

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

Fig. 11 shows the detection rate of both spam and nonspam tweets on three classifiers, random forest, C4.5 decision tree, and Bayes network. We can see that, the DR of nonspam is very stable, it keeps above 90% for random forest and C4.5 decision tree, and near 90% for Bayes network, despite the change of testing data. However, when it comes to spam tweets, the DR fluctuates dramatically, and the overall trend is decreasing. The DRs for random forest and C4.5 decision tree are 90% in the first day, but they could decrease to less then 40% in the 9th day. This phenomenon also applies with Bayes network, the DR decreases from 70% on the first day to less than 50% for most of the other testing days. From this, we can see that the detection rate is decreasing when training data and testing data are from different period of time.

2) From the View of Average Values of Features: To further investigate the reason why performance decreases when training and testing data are from different days, we calculate the average value of each feature in all tweets of each day, and find that the average value of features from spam tweets varies while that is more stable in terms of nonspam tweets.

Fig. 12 shows the changing trend of average value of three features for two classes in ten days. In general, the vary of average value of feature from spam tweets is greater than that of nonspam tweets. Fig. 12(a) shows that the average value of account age for spam tweets ranges from 530 to 730, and the variation is dramatic. However, it deviates from 710 to 740 for nonspam tweets. We infer that spammers are creating a large number of new accounts to send spam once their old account are blocked, which leads the decrease of average age for spammers. Naturally, spammers tend to keep following new friends as they want to be exposed to public more frequently, whereas for nonspammers, their number of followings are not changing too much once they have built their friend circle, as we can see from Fig. 12(b). Due to the page limit, we excluded the figures of other features. However, most of the other features have the same trend as expected: the average value of one feature varies for spam tweets, whereas it is stable for nonspam tweets. Consequently, the detection of one classifier become inaccurate, as the statistical features of the testing data varies.

3) From the View of KL Divergence of Two Days' Feature Distribution: Previously, we simply compared the some representative statistics, such as the mean values of features to show the reason why classifiers' performance decreases while training and testing are done in different days. To further illustrate the changing of the statistical features in a dataset, a natural approach is to model the distribution of the data [37]. One of the most common measure to compute the distance of distributions

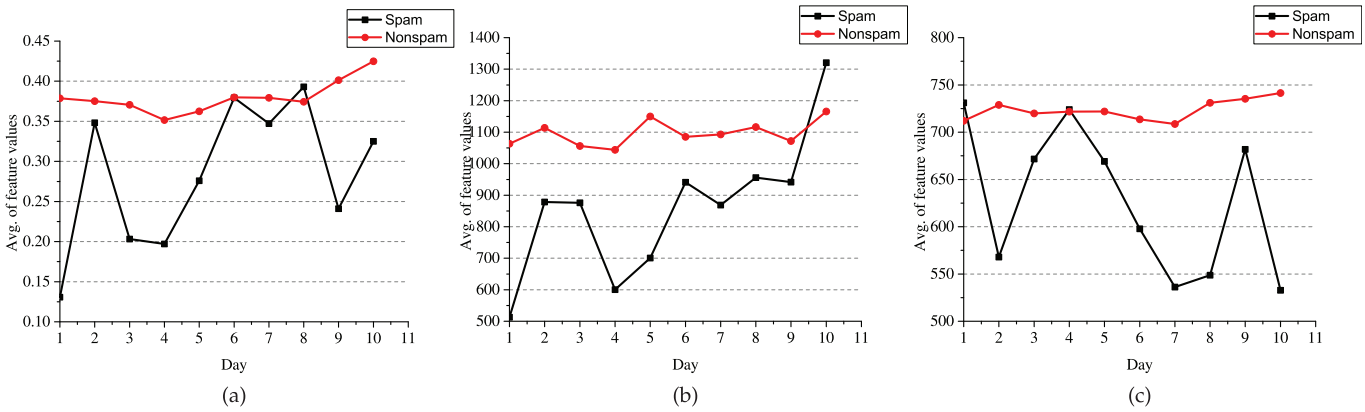


Fig. 12. Changes of average values of features. (a) Account age. (b) Number of followings. (c) Number of user mentions per Tweet.

TABLE VI
KL DIVERGENCE OF SPAM AND NONSPAM TWEETS OF TWO CONSECUTIVE DAYS

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

is Kullback–Leibler (KL) divergence [37], [38]. The suitability of KL divergence to be used in measuring distributions can be found in [37].

We compute the KL divergence of each feature of spam and nonspam tweets in consecutive two days, which is listed in Table VI. The shadowed ones are the KL divergence of features of nonspam tweets, whereas the other are the KL divergence of features of spam tweets. KL divergence indicates the dissimilarity of two distributions. The larger the value is, the more dissimilar the two distributions are. As shown in Table VI, the KL divergence of spam tweets in two consecutive days are much larger than that of the nonspam tweets for more than half the features. Taking $f1$ for example, the KL divergence of spam between Day 1 and Day 2 is 0.36, while it is only 0.04 for nonspam, which indicates that the distribution of f1 of spam in Day 1 is much different to it in D2, compared with nonspam tweets' distribution. From these KL divergence values, we can see that the distribution of spam tweets' features is changing unpredictably from day to day. Nevertheless, the distribution of training data is unchanged. So, the knowledge structure which learns from the unchanged training data is not updated while being used to classify new incoming tweets. That is why the performance of classifiers becomes inaccurate.

V. CONCLUSION AND FEATURE WORK

In this paper, we provide a fundamental evaluation of ML algorithms on the detection of streaming spam tweets. In order to perform this evaluation, we first collected a large number of 600 million public tweets. Then, we applied Trend Micro's Web Reputation System to label as many as 6.5 million spam tweets. We also extracted 12 light-weight features which are able to differentiate spam tweets and nonspam tweets from this labeled dataset. Furthermore, we used cdf figures to illustrate the characteristics of extracted features. We leveraged these features to machine learning-based spam classification later in our experiments. To investigate the ability of spam detection of different classifiers, we sampled four different datasets to simulate various scenarios. In our evaluation, we found that classifiers' ability to detect Twitter spam reduced when in a near real-world scenario since the imbalanced data brings bias. We also identified that Feature discretization was an important preprocess to ML-based spam detection. Second, increasing training data only cannot bring more benefits to detect Twitter spam after a certain number of training samples. We should try to bring more discriminative features or better model to further improve spam detection rate. Third, classifiers can detect more spam tweets when the tweets were sampled continuously rather than randomly selected tweets.

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

ACKNOWLEDGMENT

The authors would like to thank Trend Micro for providing the service to label spam tweets. They would also like to thank the reviewers' for their effort.

REFERENCES

[1] C. P.-Y. Chin, N. Evans, and K.-K. R. Choo, "Exploring factors influencing the use of enterprise social networks in multinational professional service firms," *J. Organizat. Comput. Electron. Commerce*, vol. 25, no. 3, pp. 289–315, 2015.

[2] H. Tsukayama, "Twitter turns 7: Users send over 400 million tweets per day," *Washingon Post*, Mar. 2013 [Online]. Available: http://articles.washingtonpost.com/2013-03-21/business/37889387_1_tweets-jack-dorsey-twitter

[3] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammer on Twitter," presented at the 7th Annu. Collab. Electron. Messaging Anti-Abuse Spam Conf., Redmond, WA, USA, Jul. 2010.

[4] L. Timson, "Electoral commission Twitter account hacked, voters asked not to click," *Sydney Morning Herald*, Aug. 2013 [Online]. Available: http://www.smh.com.au/it-pro/security-it/electoral-commission-twitter-account-hacked-voters-asked-not-to-click-20130807-hv1b5.html

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

[6] K. Thomas, C. Grier, D. Song, and V. Paxson, "Suspended accounts in retrospect: An analysis of Twitter spam," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, 2011, pp. 243–258.

[7] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time URL spam filtering service," in *Proc. IEEE Symp. Sec. Privacy*, 2011, pp. 447–462.

[8] X. Jin, C. X. Lin, J. Luo, and J. Han, "Socialspamguard: A data mining-based spam detection system for social media networks," *PVLDB*, vol. 4, no. 12, pp. 1458–1461, 2011.

[9] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, "Aiding the detection of fake accounts in large scale social online services," in *Proc. Symp. Netw. Syst. Des. Implement. (NSDI)*, 2012, pp. 197–210.

[10] S. Ghosh *et al.*, "Understanding and combating link farming in the Twitter social network," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 61–70.

[11] H. Costa, F. Benevenuto, and L. H. C. Merschmann, "Detecting tip spam in location-based social networks," in *Proc. 28th Annu. ACM Symp. Appl. Comput.*, 2013, pp. 724–729.

[12] E. Tan, L. Guo, X. Zhang, and Y. Zhao, "Unik: Unsupervised social network spam detection," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, San Fransisco, CA, USA, Oct. 2013, pp. 479–488.

[13] C. Yang, R. Harkreader, and G. Gu, "Empirical evaluation and new design for fighting evolving Twitter spammers," *IEEE Trans. Inf. Forensics Sec.*, vol. 8, no. 8, pp. 1280–1293, Aug. 2013.

[14] S. Lee and J. Kim, "Warningbird: A near real-time detection system for suspicious URLs in Twitter stream," *IEEE Trans. Dependable Secure Comput.*, vol. 10, no. 3, pp. 183–195, May/Jun. 2013.

[15] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu, "Analyzing spammers' social networks for fun and profit: A case study of cyber criminal ecosystem on Twitter," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 71–80.

[16] X. Zhang, S. Zhu, and W. Liang, "Detecting spam and promoting campaigns in the Twitter social network," in *Proc. IEEE 12th Int. Conf. Data Mining (ICDM)*, 2012, pp. 1194–1199.

[17] J. Song, S. Lee, and J. Kim, "Spam filtering in Twitter using sender-receiver relationship," in *Proc. 14th Int. Conf. Recent Adv. Intrusion Detect.*, 2011, pp. 301–317.

[18] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna, "COMPA: Detecting compromised accounts on social networks," presented at the 20th. Annu. Netw. Distrib. Syst. Sec. Symp., San Diego, CA, USA, Feb.. 24–27, 2013.

[19] A. H. Wang, "Don't follow me: Spam detection in Twitter," in *Proc. Int. Conf. Sec. Cryptogr. (SECRYPT)*, 2010, pp. 1–10.

[20] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proc. 26th Annu. Comput. Sec. Appl. Conf.*, 2010, pp. 1–9.

[21] S. Yardi, D. Romero, G. Schoenebeck, and D. Boyd, "Detecting spam in a Twitter network," *First Monday*, vol. 15, nos. 1–4, Jan. 2010.

[22] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?" in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 591–600.

[23] C. Chen, J. Zhang, Y. Xiang, and W. Zhou, "Asymmetric self-learning for tackling Twitter spam drift," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM)*, Apr. 2015, pp. 208–213.

[24] A. Bifet and E. Frank, "Sentiment knowledge discovery in Twitter streaming data," in *Proc. 13th Int. Conf. Discov. Sci.*, 2010, pp. 1–15.

[25] B. Wang, A. Zubiaga, M. Liakata, and R. Procter, "Making the most of tweet-inherent features for social spam detection on Twitter," arXiv preprint arXiv:1503.07405, 2015.

[26] J. Oliver, P. Pajares, C. Ke, C. Chen, and Y. Xiang, "An in-depth analysis of abuse on Twitter," Trend Micro, Irving, TX, USA, Tech. Rep., Sep. 2014.

[27] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@SPAM: The underground on 140 characters or less," in *Proc. 17th ACM Conf. Comput. Commun. Sec.*, 2010, pp. 27–37.

[28] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: social honeypots + machine learning," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2010, pp. 435–442.

[29] K. Lee, J. Caverlee, Z. Cheng, and D. Z. Sui, "Content-driven detection of campaigns in social media," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, 2011, pp. 551–556.

[30] Twitter. "Tweet structure" (2015). [Online]. Available: https://dev.twitter.com/docs/platform-objects/tweets

[31] C. Chen, J. Zhang, X. Chen, Y. Xiang, and W. Zhou, "6 Million spam tweets: A large ground truth for timely Twitter spam detection," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 7065–7070.

[32] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao, "Detecting and characterizing social spam campaigns," in *Proc. 10th ACM SIGCOMM Conf. Internet Meas.*, 2010, pp. 35–47.

[33] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on Twitter," in *Proc. 20th Int. Conf. World Wide Web*, 2011, pp. 675–684.

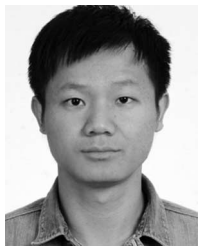
[34] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan, "Network traffic classification using correlation information," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 104–117, Jan. 2013.

[35] A. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE Intell. Syst.*, vol. 24, no. 2, pp. 8–12, Mar. 2009.

[36] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes, "Do we need more training data or better models for object detection?" in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2012, vol. 3, p. 5.

[37] T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi, "An information-theoretic approach to detecting changes in multi-dimensional data streams," in *Proc. Symp. Interface Statist. Comput. Sci. Appl.*, 2006.

[38] R. Sebastião, and J. a. Gama, "Change detection in learning histograms from data streams," in *Proc. 13th Portuguese Conf. Progr. Artif. Intell.*, 2007, pp. 112–123.



Chao Chen received the bachelor's degree in information technology (with first class Hons.) from Deakin University, Melbourne, Vic., Australia, in 2012, and is currently working toward the Ph.D degree in information technology at the same university.

His research interests include network security and social network security.

Jun Zhang (M'12) received the Ph.D. degree in computer science from University of Wollongong, Wollongong, N.S.W., Australia, in 2011.

He is currently with the School of Information Technology, Deakin University, Melbourne, Vic., Australia. He has authored more than 30 research papers in the international journals and conferences. His research interests include network and system security, pattern recognition, and multimedia processing.

Yi Xie received the B.S., M.S., and Ph.D. degrees in computer science from Sun Yat-Sen University, Guangzhou, China, in 1996, 2004 and 2008, respectively.

He was a Visiting Scholar at George Mason University, Fairfax, VA, USA, from 2007 to 2008, and Deakin University, Melbourne, Vic., Australia, from 2014 to 2015. He is currently an Associate Professor with the School of Information Science and Technology, Sun Yat-Sen University. His research interests include network security and user behavior.

Yang Xiang (SM'12) received the Ph.D. degree in computer science from Deakin University, Melbourne, Vic., Australia, in 2007.

He is currently with the School of Information Technology, Deakin University. In particular, he is leading in a research group, developing active defense systems against large-scale distributed network attacks. He is also the Chief Investigator of several projects in network and system security, funded by the Australian Research Council. His research interests include network and system security, distributed systems, and networking.

Wanlei Zhou (SM'09) received the B.Eng. and M.Eng. degrees from the Harbin Institute of Technology, Harbin, China, in 1982 and 1984, respectively, the Ph.D. degree in computer science from the Australian National University, Canberra, A.C.T, Australia, in 1991, and the D.Sc. degree from Deakin University, Melbourne, Vic., Australia, in 2002.

He is currently the Chair Professor of Information Technology with the Faculty of Science and Technology, Deakin University. His research interests include network security, distributed and parallel systems, bioinformatics, mobile computing, and eLearning.

Mohammad Mehedi Hassan received the Ph.D. degree in computer engineering from Kyung Hee University, Seoul, South Korea, in 2011.

He is currently an Assistant Professor with the Department of Information Systems, College of Computer and Information Sciences (CCIS), King Saud University (KSU), Riyadh, Saudi Arabia. He has authored over 100 research papers in journals and conferences of international repute. His research interests include cloud federation, multimedia cloud, sensor-cloud, Internet of things, big data, mobile cloud, cloud security, IPTV, sensor network, 5G network, social network, publish/subscribe system, and recommender system.

Abdulhameed AlElaiwi received the Ph.D. degree in software engineering from the College of Engineering, Florida Institute of Technology-Melbourne, Melbourne, FL, USA, in 2002.

He is an Assistant Professor with the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He has authored and coauthored many publications. His research interest includes software testing analysis and design, cloud computing, and multimedia.

Majed Alrubaian is currently working toward the Ph.D. degree in information systems at the College of Computer and Information Sciences (CCIS), King Saud University (KSU), Riyadh, Saudi Arabia.

His research interests include mining social data, cloud computing, and sensor network.

Mr. Alrubaian is a Student Member of ACM.