Semi-Supervised Spam Detection in Twitter Stream

Surendra Sedhai and Aixin Sun

***Abstract*— Most existing techniques for spam detection on Twitter aim to identify and block users who post spam tweets. In this paper, we propose a semi-supervised spam detection (S^3D) framework for spam detection at tweet-level. The proposed framework consists of two main modules: *spam detection* module operating in real-time mode and *model update* module operating in batch mode. The spam detection module consists of four lightweight detectors: 1) blacklisted domain detector to label tweets containing blacklisted URLs; 2) near-duplicate detector to label tweets that are near-duplicates of confidently prelabeled tweets; 3) reliable ham detector to label tweets that are posted by trusted users and that do not contain spammy words; and 4) multiclassifier-based detector labels the remaining tweets. The information required by the detection module is updated in batch mode based on the tweets that are labeled in the previous time window. Experiments on a large-scale data set show that the framework adaptively learns patterns of new spam activities and maintain good accuracy for spam detection in a tweet stream.**

***Index Terms*— Semi-supervised learning, Twitter, spam.**

I. Introduction

MICRO-BLOGGING services have attracted the attention of not only legitimate users but also spammers. It is reported that 0.13% of messages advertised on Twitter are clicked, which is two orders of magnitude higher than that of email spam [11]. High click rate and effective message propagation make Twitter an attractive platform for spammers. Increasing spamming activities have adversely affected user experience as well as many tasks such as user behavior analysis and recommendation.

Most of the existing studies on Twitter spam focus on account blocking, which is to identify and block spam users, or spammers. Hu *et al.* [13] utilized the social graph and tweets of a user and formulated spammer detection as an optimization problem. Similarly, information extracted from user's tweets, demographics, shared URLs, and social connection are utilized as features in standard machine learning algorithms to detect spam users [14]. However, account blocking approach is less effective for spammers who may act as legitimate users by posting nonspam content regularly. Blocking spammers may even hurt a legitimate user who happens to grant permission to a third-party application that posts spammy tweets under her username. This legitimate account may be blocked because of such spam tweets. Furthermore, spammers change their tweet content and strategies to make their tweets and activities look like legitimate [1]. Although identifying and blocking spammer accounts remain a crucial and challenging task, tweet-level spam detection is essential to fight against spamming at a more fine-grained level, and helps to timely detect spam tweets instead of waiting for users to be detected as spammers. Similarly, Chen *et al.* [4] suggested that training data set should be continuously updated in order to deal with the changing distribution of features in tweet stream.

In this paper, we propose a semi-supervised framework for spam tweet detection. The proposed framework mainly consists of two main modules: 1) four lightweight detectors in the *spam tweet detection module* for detecting spam tweets in real time and 2) *updating module* to periodically update the detection models based on the confidently labeled tweets from the previous time window. The detectors are designed based on our observations made from a collection of 14 million tweets, and the detectors are computationally effective, suitable for real-time detection. More importantly, our detectors utilize classification techniques at two levels, tweet level and cluster level. Here, a cluster is a group of tweets with similar characteristics. With this flexible design, any features that may be effective in spam detection can be easily incorporated into the detection framework. The framework starts with a small set of labeled samples and update the detection models in a semi-supervised manner by utilizing the confidently labeled tweets from the previous time window. This semi-supervised approach helps to learn new spamming activities, making the framework more robust in identifying spam tweets.

II. Related Work

Spam is a serious problem on almost all online media, and spam detection has been studied for many years. Spammers may use different techniques on different platforms so spam detection technique developed for one platform may not be directly applicable on other platforms. Thomas *et al.* [22] reported that spam targeting email is significantly different from spam targeting Twitter. In Twitter, there are different types of spamming activities such as link farming [10], spamming trending topics [1], phishing [5], and aggressive posting using social bot [6]. These different activities pollute timeline of users as well as Twitter search results.

Many social spam detection studies focus on the identification of spam accounts. Lee *et al.* [14] analyzed and used features derived from user demographics, follower/following social graph, tweet content, and the temporal aspect of user behavior to identify content polluters. Hu *et al.* [13] exploited social graph and tweets of a user to detect spam detection on Twitter. They formulated spammer detection task as an

Manuscript received July 13, 2016; revised July 23, 2017 and October 24, 2017; accepted November 9, 2017. Date of publication December 6, 2017; date of current version February 23, 2018. This work was supported by the Singapore Ministry of Education Research Fund under Grant MOE2014-T2-2-066. *(Corresponding author: Aixin Sun.)*

The authors are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: surendra001@e.ntu.edu.sg; axsun@ntu.edu.sg).

Digital Object Identifier 10.1109/TCSS.2017.2773581

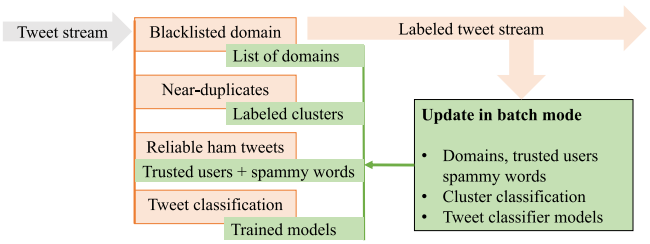

Fig. 1. System overview of the S^{3}D framework.

optimization problem. Online learning has been utilized to tackle the fast evolving nature of spammer [12]. They have utilized both content and network information and incrementally updated their spam detection model for effective social spam detection. Tan *et al.* [21] proposed an unsupervised spam detection system that exploits legitimate users in the social network. Their analysis shows the volatility of spamming patterns in social network. They have utilized nonspam patterns of legitimate users based on social graph and user-link graph to detect spam pattern. Gao *et al.* [9] identified social spam by clustering posts based on text and URL similarities and detected large clusters with bursty posting patterns. Incremental clustering-based approach has been used to detect spam campaigns on Twitter [8]. Our work is different as we focus on tweet-level spam detection.

Removing spam users cannot filter every spam message as spammer may create another account and restart spamming activity. This calls for tweet-level spam detection. Inspired from content-based techniques for emails, Santos *et al.* [18] utilized standard classifiers to detect spam tweets. Language modeling approach has been used to compute the divergence of trending topic, suspicious message, and title of the page linked in the tweet [17]. Similarly, Castillo *et al.* [3] analyzed the credibility of tweets on trending topics based on users' tweeting and retweeting behaviors, tweet content, and link present in the tweets. As spammers keep on evolving over time, semi-supervised approach is suitable for tracking such changing spamming activities. Semi-supervised spam detection approach has been utilized to identify spam on voice-over-IP call [24]. Semi-supervised approach using the Laplacian score method for feature selection has been used to detect spammer on Twitter [15]. A semi-supervised approach is reported to have better performance than supervised approach for malware detection task [19]. Similarly, trust and distrust information from social graph have also been used in semi-supervised framework and show its effectiveness [16]. Unlike these studies, our framework focuses on tweet-level spam detection. To the best of our knowledge, semi-supervised approach has not been utilized to detect individual spam tweets. Our proposed method is capable of continuously updating itself by using semi-supervised approach.

III. Semi-Supervised Spam Detection

The proposed S^{3}D contains two main modules as shown in Fig. 1. Assuming that we have all the information (e.g., a blacklist of spamming domains and trained classification models), the tweets are labeled as spam and nonspam (also known as "ham") tweets using the four detectors in real time. The required information is updated periodically based on the confidently labeled tweets from the previous time window, in a semi-supervised manner. Next, we detail the main modules.

A. Spam Tweet Detection in Real Time

For efficiency reason, the tweets are labeled by four lightweight detectors from four perspectives, in an order of the easiest to hardest in terms of difficulty in detection. Once a label is assigned by one of the detectors, the tweet need not pass to the next detector.

1) Blacklisted Domain Detector: Spammers promote their services/products by posting links in their tweets [9]. An effective way of spam detection is to detect tweets containing links from blacklisted domains. Many spammers post links in the form of shortened URLs. To translate short URLs back to full URLs, we used Java library "HttpClient,"[1] hence our data set also has full URL information of the links present in tweets.[2] The full URLs are used to extract the domain of the webpages. We utilized the domain information present in our data set to identify tweets containing links from the blacklisted domains. The list of blacklisted domains is to be updated at the end of each time window utilizing confidently labeled tweets, during the batch update.

2) Near-Duplicate Detector: Tweets that are near-duplicates of prelabeled spam/ham tweets are assigned the same labels accordingly. The near-duplicate tweets are detected by using the *MinHash* algorithm [2], which has shown effectiveness for labeling spam tweets [20]. More specifically, a signature is computed for each tweet by concatenating the three minimum hash values computed from the tweet's *unigram*, *bigram*, and *trigram* representations, respectively. If two or more tweets have the same signature, then the tweets are considered near-duplicates. If a cluster of near-duplicate tweets hashed to the same signature has been labeled as spam or ham tweets, the new tweet having the same signature receives the same label.

3) Reliable Ham Tweet Detector: Tweets posted by legitimate users can be considered as ham tweets; however, spammers may pretend as legitimate users and after gaining acceptance from other users they post spam tweets [25]. Hence, we consider a tweet to be a *reliable ham tweet* if it satisfies two conditions: 1) the tweet does not contain any spammy words and 2) the tweet is posted by a trusted user.

Spammy words are the words whose probability of occurrence is larger in spam than in ham tweets. For example, word *followme* is likely to appear in spam tweets but the word may appear in ham tweet as well. Let the total number of tweets containing word w be $n(w)$, the number of spam tweets containing the word w be $n_s(w)$, and the number of ham tweets containing word w be $n_h(w)$. The probability of word w appearing in spam tweets $p_s(w)$ and in ham tweets $p_h(w)$

[1] https://hc.apache.org/httpcomponents-client-ga/

[2] URL translation was conducted during data set collection. To avoid possible hyperlink loop created for trapping crawlers, we followed maximum three redirections from each short URL.

TABLE I

FEATURES USED TO REPRESENT TWEETS AND CLUSTERS FOR CLASSIFICATION; FoT MEANS FRACTION OF TWEETS AND FoU MEANS FRACTION OF USERS. TOP 15 MOST EFFECTIVE FEATURES FOR TWEET CLASSIFICATION AND CLUSTER CLASSIFICATION BASED ON THE GINI IMPURITY SCORE ARE INDICATED BY THE NUMBERS (x) FOLLOWING THE FEATURES ($1 \leq x \leq 15$)

Type	Features for tweet representation	Features for cluster representation
Hashtag	Contains hashtag	FoT having hashtag (3)
	Contains more than 2 hashtags (1)	FoT having more than 2 hashtags (5)
	Contains spammy hashtag (2)	Hashtag per tweet (12)
	Contains categorical hashtag (7)	FoT having spammy hashtag (14)
	Contains capitalized hashtag (8)	FoT having categorical word as hashtag
		FoT having capitalized hashtag (4)
Content	Fraction of words that are spammy	FoT having spammy words
	Contains question mark	FoT having question mark
	Contains money sign	FoT having exclamation mark (10)
	Contains exclamation sign (13)	FoT having money sign
	Contains positive emoticons	FoT having positive emoticons (7)
	Contains negative emoticons	FoT having negative emoticons
	Contains positive words	FoT having positive words
	Contains negative words	FoT having negative words (8)
	Fraction of uppercase characters (9)	Fraction of capitalized tweets (2)
	Contains URL (11)	FoT that are retweet (6)
	Is retweet	FoT having URL
	Contains mention (14)	Mentions per tweet (15)
	Contains first person pronoun	FoT tweets having first person pronoun
	Contains second person pronoun	FoT tweets having second person pronoun
	Contains third person pronoun	FoT tweets having third person pronoun
	Normalized length of the tweet in word	Median tweet length in word/(max tweet length)
	Normalized length of the tweet in character (10)	Median tweet length in characters /140
	Contains top 10,000 uni-/bi-/tri-gram of last time window	Contains top 10,000 uni-/bi-/tri-gram of the last time window
	Day of the week in which the tweet is posted	Ratio of spam tweets in the cluster (1)
User	Has less than 5 percentile followers (3)	FoU having less than 5 percentile followers
	Has less than 5 percentile followees (6)	FoU having less than 5 percentile followees
	Has more than 50 percentile total tweets	FoU having more than 50 percentile total tweets
	Percentile followers of the user (15)	Median percentile of followers of users
	Percentile followees of the user	Median percentile of followees of users
	Percentile total tweet count of the user	Median percentile of total tweets of the users (11)
	User profile contains description (5)	FoU having description in profile
	User profile description contains spammy words	FoU having spammy words in description
	User profile has url	FoU having URL in profile
	User profile has location info	FoU having location info in profile
	User profile has time-zone info	FoU having timezone info in profile
	Followers-followees ratio (4)	FoU having followers greater than followee
	User's normalized age (12)	Median normalized age of users
		Fraction of post by the dominating user (9)
		Percentile followers of the user tweeting the most
		Percentile followees of the user tweeting the most
		Percentile total tweets of the user tweeting the most
		Tweets per user
		Standard deviation of normalized age of users (13)
		Percentile followers of the most followed user
		Percentile followees of the most followed user
		Percentile total tweets of the most followed user
Domain	URL from top 100 domains	FoT having URL from top 100 domains
	URL from top 1000 domains	FoT having URL from top 1000 domains
	URL from top 10000 domains	FoT having URL from top 10000 domains

are $p_s(w) = (n_s(w)/n(w))$ and $p_h(w) = (n_h(w)/n(w))$, respectively. Word w is a spammy word if $p_s(w) > p_h(w)$. In our implementation, words that are shorter than three characters in length are ignored.

A trusted user is a user who has never posted any spam tweet and has posted at least five confident ham tweets. A tweet is a confident ham tweet if the tweet does not contain any spammy words and is predicted to be ham by all the component classifiers in the tweet classification detector. The clusters of near-duplicate tweets will also be predicted as "clusters of spam" or "clusters of ham" by multiple classifiers. The tweets in a cluster which is predicted to be ham cluster by all classifiers are also considered as confident ham tweets.

During the batch update, the list of trusted users and the list of spammy words are updated. As the list of spammy words are updated in each batch, the size of vocabulary grows over time which may help to capture spam tweets more effectively.

4) Multiclassifier-Based Detector: Tweets that are not labeled in any of the previous steps are processed and labeled in this step. Here, we develop a spam detector by using three efficient classifiers, namely, Naïve Bayes (NB), logistic regression (LR), and random forest (RF). The three classifiers use different classification techniques, i.e., generative, discriminative, and decision tree-based classification models. A full spectrum of features is extracted to represent each tweet. Listed in Table I in the column titled "Features for tweet

representation," the features include hashtag-based features, content-based features, user-based features, and domain-based features. Most features are self-explanatory and we only elaborate two features: categorical hashtag and top domains. Categorical words are the words used in one of the top-level categories in Yahoo! hierarchy, or words used to categorize content in four websites: BBC, CNN, NYTimes, and Reddit. There are 75 categories including sports, technology, business, movie, jobs, etc. The binary feature is 1 if the hashtag is one of the categorical words. The domain feature is based on the domain of the URLs contained in tweets. Domain ranking is from alexa.com. A tweet is labeled as spam if at least two of the three classifiers predict the tweet to be spam; otherwise the tweet is labeled as ham.

B. Model Update in Batch Mode

The time window for the update is set to be *one day* in our experiments. The key desideratum is to identify the confidently labeled data of the previous time window.

1) Confidently Labeled Tweets: Tweets that are labeled by the first three detectors (i.e., blacklisted domain, near-duplicate, and reliable ham tweet) are considered as confidently labeled tweets. For the classifier based detector, recall that we use three classifiers each is based on a different classification technique. Tweets that are labeled as spam by all the three classifiers are considered as confidently labeled spam tweets. Similarly, tweets that do not contain any spammy words and are labeled as ham by all the three classifiers are confidently labeled ham tweets. Excluding ham tweets containing spammy words (e.g., *followme*) will help to prevent the deviation of classifier from a burst of spammy words in ham tweets.

The identified confidently labeled spam tweets are utilized to update *blacklist domains*, and confidently labeled ham tweets are utilized to identify *trusted users*.

2) Near-Duplicate Cluster Labeling: Recall that the near-duplicate detector computes a signature for each tweet to check if the tweet is a near duplicate of a labeled cluster. If the signature of a tweet does not match any prelabeled cluster, then the tweet is passed to the next level detectors.

After each time window, all the tweets that do not match prelabeled clusters but having the same signature are grouped into a new cluster, i.e., each cluster is a collection of near-duplicate tweets. Next, we label the clusters each containing at least 10 tweets and if the labels are of high confidence, then the signatures of these newly labeled confident clusters will be used by the near-duplicate detector in the next time window. Recall that all the tweets have been labeled as spam and ham tweets (see Fig. 1), an easy approach to label these clusters is to perform a majority voting. Specifically, if there are more spams in a cluster than ham tweets, then the cluster is labeled as a spam cluster. However, the majority voting approach solely relies on the predicting power of the detectors and may not capture the new spamming patterns in the most recent time window. Moreover, because tweets in a cluster are near-duplicates, their labels assigned by the detectors are mostly the same. For this reason, we also employ a feature-based classifier.

Each cluster is represented with hashtag-based features, content-based features, user-based features, and domain-based features, as listed in Table I, the third column. Many of the features used here are adopted from the existing studies [3], [7]. Different from tweet classification (features listed in the second column), the cluster-level features represent the collective information obtained from all the tweets in the cluster. The clusters represented in feature space are classified using an LR classifier.

We consider a cluster to be a confidently labeled cluster if the labels predicted by the feature-based cluster classifier and the majority voting of the tweet labels are the same.

3) Update Detector Models: After finding the confidently labeled tweets and clusters, the models used by the detectors are updated accordingly including blacklisted domains, labeled clusters, trusted users, and tweet classification models. Blacklisted domains are updated by including domains having at least five tweets in the last time window and at least 90% of the tweets are confidently labeled as spam tweets. A user having at least five tweets and all tweets are confidently labeled ham tweets is considered as a trusted user. The classification models of the three classifiers are retrained by including the newly labeled confident tweets of the last time window.

By updating the detection models in batch mode, the proposed semi-supervised spam detection framework is capable of capturing new vocabulary and new spamming behaviors, which makes the framework robust and adaptive to deal with the dynamic nature of spamming activities. Note that we do not consider reducing the importance of old tweets in the current framework. It is an interesting future research direction to investigate whether reducing importance of old tweets would affect the system performance.

C. Computational Efficiency

All the four spam detectors are computationally effective; hence, the proposed framework is capable of labeling tweet stream in real time. We conduct experiments on a desktop PC with octacore Intel processor of 3.70 GHz and 16-GB RAM. In our experiments, all the detectors are carried out on a single-core of the processor, except RF classifier which utilizes all the cores of the processor. Empirically, we found that on average it takes 0.495 ms to label a tweet where more than 50% of the time is used for feature extraction. Note that, our code is not optimized for real-time setting, and efficiency can be further improved by parallelizing the detectors.

IV. Experiment and Discussion

We used 15 days of data from HSpam14 data set [20] in our experiments. HSpam14 contains 14 million tweets, collected by using the trending topics on Hashtags.org for two months, May and June 2013. In this paper, we use 15 days of tweets (May 17–31, 2013), where each day has more than 35 000 tweets. The time window for batch mode update is set to be a day. Almost all tweets in HSpam14 are labeled to be a spam and ham, and the remaining small portion are labeled as unknown for not being able to determine their labels even with manual inspection. Note that more than 80% of tweets in

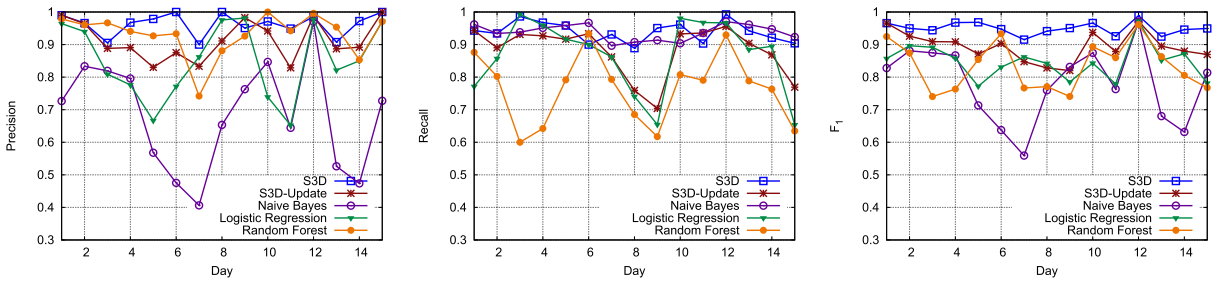

Fig. 2. Comparison of supervised and semi-supervised approach. (a) Precision. (b) Recall. (c) F_1-score.

HSpam14 are labeled automatically and the manually labeled tweets are biased to spams.

We simulate a tweet stream in our experiments. On the first day, the detectors in S^{3}D are trained by using the manually labeled tweets and the reliable ham tweets in the HSpam14 data set (the released HSpam14 data set contains the detailed labels of the tweets, i.e., on which step a tweet was labeled during data set construction). These training tweets are utilized to create the initial set of blacklisted domains, labeled clusters, trusted users, labeled tweets, and spammy words. There are 48 849 spam tweets and 22 185 ham tweets. The remaining tweets on the first day and all the tweets of the remaining 14 days are used for testing purpose. Because not all tweets in HSpam14 are manually labeled, to ensure the accuracy of the evaluation in our experiments, we manually label 300 randomly selected tweets from each time window to evaluate the performance of the system.[3] The performance is evaluated using the commonly used metrics: Precision, Recall, and F_1.

Supervised spammer detection on Twitter such as [7] and [14] focus on spammer detection, whereas our work is on tweet-level spam detection. We use some features derived and inspired from these studies in our framework. However, since our work focuses on spam detection at tweet level, these spammer detection systems cannot be used as baseline methods to compare with ours. Previous tweet-level spam detection studies used off-the-shelf classifiers, namely, NB, LR, and RF classifier, in supervised settings. We have also used these methods to compare the performance with that of our proposed system. More specifically, tweet classification using LR reported in this paper is similar to the work on information credibility [3] and also similar to the method reported in [17]. Most of the features described in the information credibility paper except propagation-related features are used in the S^{3}D as well. Propagation-related features are not available in HSpam14 data set so these features cannot be used. Similarly, NB and RF classifiers are used in [17], [18], and [23] for tweet-level spam detection. RF classifier is found to be superior among all the other methods [18], [23]. In our experiments, we compare the results of S^{3}D with the classification results using these classifiers in supervised setting.

[3]We have also evaluated the results using the manually labeled tweets in the HSpam14, similar results were obtained.

TABLE II
FRACTION OF TWEETS DETECTED BY EACH DETECTOR

Step	Detector	Coverage
1	Blacklisted domain detection	5.55%
2	Near-duplicates detection	6.61%
3	Reliable ham tweets	0.64%
4	Tweet classification	87.20%

A. Result and Discussion

S^{3}D has four detectors as shown in Fig. 1. Table II reports the percentage of tweets labeled by each detector. It shows that 5.55% of the tweets are labeled by the blacklisted domain detector and 6.61% of the tweets are labeled by the near-duplicate detector. Reliable ham tweet detector has very low coverage of 0.64%. The low coverage is due to the fact that the HSpam14 data set was collected based on popular hashtags, not on user basis [20]. In other words, the data set does not contain all tweets of any user. Because a trusted user should have at least five ham tweets, only a small set of users can be identified as trusted users. Remaining 87.20% of tweets are labeled by the last detector, tweet classifier. Next, we report the spam detection performance of S^{3}D with more focus on the tweet classifier detector.

We now report the performance of the following five methods for the spam tweet detection task.

1) *NB:* This method reports the prediction results of the NB classifier that rely on the training data of the first day.
2) *LR:* This method reports the prediction results of the LR classifier that rely on the training data of the first day.
3) *RF:* This method reports the prediction results of the RF classifier that rely on the training data of the first day.
4) *S^{3}D-Update:* The results of the S^{3}D framework without batch update. That is, the detectors in the framework fully rely on the training data of the first day, the same as the three classifiers above.
5) *S^{3}D:* The results of the proposed S^{3}D framework with model update after each time window.

Fig. 2 plots the Precision, Recall, and F_1 scores of the five methods. Observed S^{3}D achieves the best F_1 scores. The significant better F_1 scores against S^{3}D-Update over all days show that semi-supervised approach is suitable for real-time spam detection in Twitter as it learns new spamming patterns continuously. Comparing S^{3}D with the F_1 scores of NB, LR, and RF shows that the proposed method is superior to the standard supervised methods. Observed F_1 scores of S^{3}D is

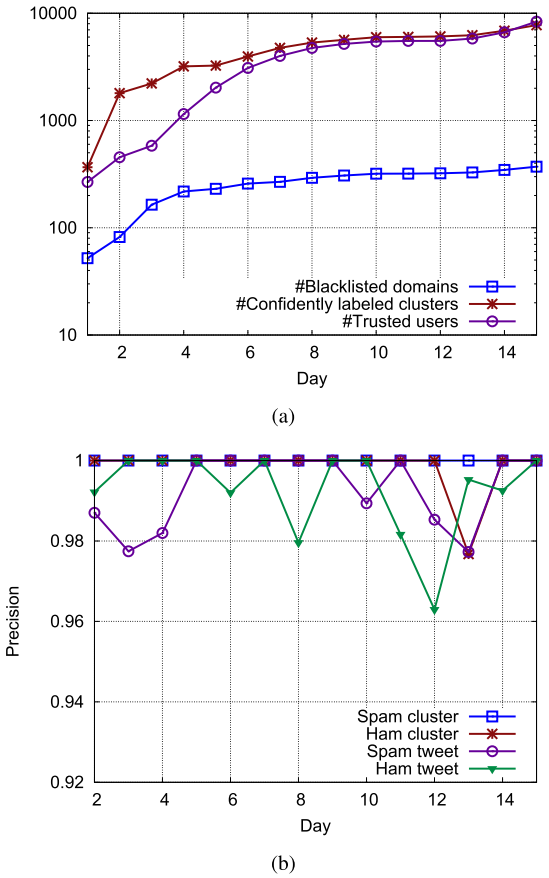

Fig. 3. Accumulated numbers of blacklisted domains, confident clusters, and trusted users, and precision of confidently labeled clusters/tweets. (a) Domains, clusters, and users. (b) Precision of the labels.

consistent over time compare to other methods. It is observed that precision score of RF is best for three days, but recall score is the lowest of all. In contrast, NB has good recall score at the expense of lower precision score. The results show that the proposed S^{3}D method is effective to capture spam tweets effectively.

The sudden rise of the F_1 scores on the 12th day is due to a large number of relatively easy to detect spam tweets in that day. In HSpam14 data set, the tweets were collected by using trending keywords of each day which leads to a change in the distribution of words of each day. The significant fluctuation in the performance of S^{3}D may be due to the changing distribution of data set in each time window. However, as S^{3}D continuously learns new patterns and vocabulary, its performance is found to be more consistent compare to the other methods. More specifically, Fig. 3 plots the number of blacklisted domains, confidently labeled near-duplicate clusters, and trusted users. It shows that S^{3}D keeps on utilizing new knowledge obtained from earlier labeled tweets and clusters to improve the capability of spam tweets detection. Furthermore, we have also used top 10 000 frequent uni/bigrams and trigrams computed at the end of each time window to update the model to deal with vocabulary change (see Table I).

In S^{3}D, we identify the confidently labeled tweets and the confidently labeled near-duplicate clusters. The confidently labeled tweets and clusters are utilized to learn new models for the detectors. The quality of these confidently labeled tweets and clusters are therefore crucial for the performance of S^{3}D. Here, we evaluate the quality of these tweets and clusters, plotted in Fig. 3(b). The confident clusters are evaluated by manually labeled 47 randomly selected clusters on each day, which is the smallest number of confidently labeled clusters produced over the 15 days. Fig. 3(b) shows that the precision of confident clusters is almost perfect for both spam and ham clusters. The figure also shows that the precision of confidently labeled spam and ham tweets are consistently above 95%. Adding such clusters and tweets in the training process makes S^{3}D capable of capturing the emerging spamming activities as well as the new vocabulary.

TABLE III
TOP 15 MOST EFFECTIVE VOCABULARY FEATURES

Tweet	follow; follow back; back; ipad ipadgames; please follow; please; followers; retweet; tfbjp; teamfollowback; ipad; follow me; collected; followback; gameinsight
Cluster	follow; love; back; followers; follow me; follow back; retweet; openfollow; teamfollowback; want; someone; win; please; 500aday; gain

B. Feature Analysis

There are four types of features used to represent tweet and cluster for classification (see Table I). In our experiments, we observe that normalization of features gives better performance than without normalization. Because users' followers, followees, and total tweets exhibit power law distributions. The features derived from these values are normalized based on percentile. Features such as length of a tweet in characters and words show normal distribution, which are normalized by the maximum value. Based on the Gini impurity score, we identify the top 15 most effective features for tweet classification and cluster classification, respectively. These features are highlighted in Table I in "(x)" format, where x is the top ranking position.

It is observed that 10 out of the top 15 most effective features are vocabulary-based features for tweet classification, whereas in the case of cluster classification only 3 out of the top 15 features are vocabulary-based features. Metadata of a tweet contains information only about the single tweet which is comparatively less informative. In contrast, a cluster contains a number of tweets, hence metadata based features represent the collective information of tweets in the cluster and are comparatively more informative. For example, if there is a tweet from a user whose account creation date is known and has a very few followers and followees, it is hard to determine that tweet posted by these users is ham or spam. In contrast, if there is a group of users whose accounts are created around the same time and all having a very few number of followers and followees and posting near-duplicate tweets, then the tweets in this cluster are likely to be spam.

Table III lists the top 15 vocabulary-based features (unigram, bigram and trigram). Only unigram and bigram vocabularies appear in the top ranked list. One possible reason for trigram features not in the list may be due to the sparsity of

the trigram vocabulary in the data set. It is interesting to note that most of the top words based on Gini impurity score are the same as the list of hashtags having the highest *spammy-index* reported in [20].

V. Conclusion

In this paper, we propose a semi-supervised spam detection framework, named S^3D. S^3D utilizes four lightweight detectors to detect spam tweets on real-time basis and update the models periodically in batch mode. The experiment results demonstrate the effectiveness of semi-supervised approach in our spam detection framework. In our experiment, we found that confidently labeled clusters and tweets make the system effective in capturing new spamming patterns.

Tweet-level spam detection is a fine-grained approach which can be used to detect spam tweets in real time. However, for a given tweet only limited information can be obtained. In contrast, more discriminative features can be derived from user account, historical tweets of the users, and social graph. However, by the time a malicious user is detected, the user might affect many other users. We believe that tweet-level spam detection complements user-level spam detection. Due to the limited user information in our data set, we have used the simple technique to deal with user-level spam detection. Nevertheless, we argue that the user-level spam detection can be incorporated into S^3D, which is part of our future work.

References

[1] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on Twitter," in *Proc. CEAS*, 2010, p. 12.

[2] A. Broder, "On the resemblance and containment of documents," in *Proc. Compress. Complex. Sequences*, 1997, pp. 21–29.

[3] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on Twitter," in *Proc. WWW*, 2011, pp. 675–684.

[4] C. Chen *et al.*, "A performance evaluation of machine learning-based streaming spam tweets detection," *IEEE Trans. Comput. Social Syst.*, vol. 2, no. 3, pp. 65–76, Sep. 2015.

[5] S. Chhabra, A. Aggarwal, F. Benevenuto, and P. Kumaraguru, "Phi.sh/$oCiaL: The phishing landscape through short URLs," in *Proc. CEAS*, 2011, pp. 92–101.

[6] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Who is tweeting on Twitter: Human, bot, or cyborg?" in *Proc. Annu. Comput. Secur. Appl. Conf.*, 2010, pp. 21–30.

[7] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini. (Jul. 2014). "The rise of social bots." [Online]. Available: https://arxiv.org/abs/1407.5225

[8] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. N. Choudhary, "Towards online spam filtering in social networks," in *Proc. Symp. Netw. Distrib. Syst. Secur. (NDSS)*, 2012, pp. 1–16.

[9] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao, "Detecting and characterizing social spam campaigns," in *Proc. IMC*, 2010, pp. 35–47.

[10] S. Ghosh *et al.*, "Understanding and combating link farming in the Twitter social network," in *Proc. WWW*, 2012, pp. 61–70.

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

[12] X. Hu, J. Tang, and H. Liu, "Online social spammer detection," in *Proc. AAAI*, 2014, pp. 59–65.

[13] X. Hu, J. Tang, Y. Zhang, and H. Liu, "Social spammer detection in microblogging," in *IJCAI*, 2013, pp. 2633–2639.

[14] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: Social honeypots + machine learning," in *Proc. SIGIR*, 2010, pp. 435–442.

[15] W. Li, M. Gao, W. Rong, J. Wen, Q. Xiong, and B. Ling, "LSSL-SSD: Social spammer detection with Laplacian score and semi-supervised learning," in *Proc. Knowl. Sci., Eng. Manage. (KSEM)*, 2016, pp. 439–450.

[16] Z. Li, X. Zhang, H. Shen, W. Liang, and Z. He, "A semi-supervised framework for social spammer detection," in *Proc. Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining (PAKDD)*, 2015, pp. 177–188.

[17] J. Martinez-Romo and L. Araujo, "Detecting malicious tweets in trending topics using a statistical analysis of language," *Expert Syst. Appl.*, vol. 40, no. 8, pp. 2992–3000, 2013.

[18] I. Santos, I. Miñambres-Marcos, C. Laorden, P. Galán-García, A. Santamaría-Ibirika, and P. G. Bringas, "Twitter content-based spam filtering," in *Proc. Joint Conf. (SOCO-CISIS-ICEUTE)*, 2013, pp. 449–458.

[19] I. Santos, J. Nieves, and P. G. Bringas, "Semi-supervised learning for unknown malware detection," in *Proc. Int. Symp. Distrib. Comput. Artif. Intell.*, 2011, pp. 415–422.

[20] S. Sedhai and A. Sun, "HSpam14: A collection of 14 million tweets for hashtag-oriented spam research," in *Proc. SIGIR*, 2015, pp. 223–232.

[21] E. Tan, L. Guo, S. Chen, X. Zhang, and Y. Zhao, "Unik: Unsupervised social network spam detection," in *Proc. CIKM*, 2013, pp. 479–488.

[22] 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. Secur. Privacy*, May 2011, pp. 447–462.

[23] B. Wang, A. Zubiaga, M. Liakata, and R. Procter. (Mar. 2015). "Making the most of tweet-inherent features for social spam detection on Twitter." [Online]. Available: https://arxiv.org/abs/1503.07405

[24] Y.-S. Wu, S. Bagchi, N. Singh, and R. Wita, "Spam detection in voice-over-IP calls through semi-supervised clustering," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, Jun. 2009, pp. 307–316.

[25] C. Yang, R. Harkreader, and G. Gu, "Die free or live hard? Empirical evaluation and new design for fighting evolving Twitter spammers," in *Recent Advances in Intrusion Detection* (Lecture Notes in Computer Science), vol. 6961. Berlin, Germany: Springer, 2011, pp. 318–337.

Surendra Sedhai received the bachelor's degree from the Institute of Engineering—Pulchowk Campus, Tribhuvan University, Lalitpur, Nepal, the master's degree from the Asian Institute of Technology, Khlong Nung, Thailand, and the Ph.D. degree in computer science from Nanyang Technological University, Singapore.

His current research interests include social data science, spam detection on social networks, and large-scale machine learning.

Aixin Sun received the Ph.D. degree from the School of Computer Science and Engineering, Nanyang Technological University, Singapore, in 2004.

He is currently an Associate Professor with the School of Computer Science and Engineering, Nanyang Technological University. His papers appear in major international conferences such as SIGIR, KDD, WSDM, and ACM Multimedia, and journals including *Data Mining and Knowledge Discovery*, the IEEE Transactions on Knowledge and Data Engineering, and the *Journal of the Association for Information Science and Technology*. His current research interests include information retrieval, text mining, social computing, and multimedia.