COMPUTERS & SECURITY 69 (2017) 35–49

Available online at www.sciencedirect.com

ScienceDirect

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

Computers & Security

Addressing the class imbalance problem in Twitter spam detection using ensemble learning

*Shigang Liu, Yu Wang *, Jun Zhang, Chao Chen, Yang Xiang*

School of Information Technology, Deakin University, Geelong, Australia

ARTICLE INFO

Article history:
Available online 13 December 2016

Keywords:
Online social networks
Twitter
Spam detection
Machine learning
Class imbalance

ABSTRACT

In recent years, microblogging sites like Twitter have become an important and popular source for real-time information and news dissemination, and they have become a prime target of spammers inevitably. A series of incidents have shown that the security threats caused by Twitter spam can reach far beyond the social media platform to impact the real world. To mitigate the threat, a lot of recent studies apply machine learning techniques to classify Twitter spam and promising results are reported. However, most of these studies overlook the class imbalance problem in real-world Twitter data. In this paper, we experimentally demonstrate that the unequal distribution between spam and non-spam classes has a great impact on spam detection rate. To address the problem, we propose FOS, a fuzzy-based oversampling method that generates synthetic data samples from limited observed samples based on the idea of fuzzy-based information decomposition. Moreover, we develop an ensemble learning approach that learns more accurate classifiers from imbalanced data in three steps. In the first step, the class distribution in the imbalanced data set is adjusted by using various strategies, including random oversampling, random undersampling and FOS. In the second step, a classification model is built upon each of the redistributed data sets. In the final step, a majority voting scheme is introduced to combine the predictions from all the classification models. We conduct experiments on real-world Twitter data for the purpose of evaluation. The results indicate that the proposed learning approach can significantly improve the spam detection rate in data sets with imbalanced class distribution.

1. Introduction

Twitter has gained significantly in popularity in recent years and become an important source for real-time information sharing and news dissemination. Inevitably, the growth of Twitter is accompanied by a significant increase of spamming activities targeting on the platform. Twitter spam is usually referred to as the unsolicited tweets that contain malicious links directing victims to external sites with malware downloads, phishing, drug sales, scams, etc. (Benevenuto et al., 2010). A series of incidents showed that Twitter spam not only affects user experience, but also poses significant threats of damages beyond the social networking platform itself. As an example, in September 2014, a nationwide Internet meltdown in New Zealand was caused by a Twitter spam campaign that spread DDoS attack malware in the guise of leaked nude photos of Hollywood celebrities (Pash, 2014).

The traditional approach to detect and filter spam is based on blacklists. For example, Trend Micro develops a blacklisting service called Web Reputation Technology system to filter spam URLs for users (Oliver et al., 2014). Twitter also

* *Corresponding author.*
E-mail addresses: shigang@deakin.edu.au (S. Liu), y.wang@deakin.edu.au (Y. Wang), jun.zhang@deakin.edu.au (J. Zhang), chao.chen@deakin.edu.au (C. Chen), yang@deakin.edu.au (Y. Xiang).
http://dx.doi.org/10.1016/j.cose.2016.12.004
0167-4048/

implements a blacklist filtering module in their anti-spam system BotMaker (Jeyaraman, 2014). Nonetheless, the blacklist-based schemes fail to protect victims from emerging spam due to the time lag (Grier et al., 2010). A previous study shows that more than 90% of victims may click through a new spam link before it is blocked by blacklists (Thomas et al., 2011).

In order to overcome the limitation of blacklisting, a lot of researchers have proposed machine learning based schemes that detect Twitter spam by mining the spammers' and spam tweets' unique patterns, without checking the embedded URLs (Gao et al., 2012; Yang et al., 2012). This kind of spam detection schemes usually involves a few steps. First of all, the features that can differentiate spam from non-spam are selected and extracted from the tweets or authors. Example features include account age, number of followers, number of following, and number of characters in a tweet. Second, a small set of training data samples are labeled (as spam or non-spam) based on some ground truth. Finally, various machine learning algorithms can be applied to develop classification models, which can then be deployed to detect spam in a real-time basis. A number of recent studies (Benevenuto et al., 2010; Stringhini et al., 2010; Yang et al., 2013; Zhang et al., 2012) have reported satisfactory results obtained using machine learning based detection schemes.

However, most previous studies overlook the issue of imbalanced class distribution that widely exists in real-world Twitter data. That is, the proportion of the spam tweets is much smaller than that of the non-spam tweets in reality. For example, a study based on a data sample back in 2009 (Twitter Study, 2009) suggested that 3.75% of tweets are spam. In the meantime, the experimental data sets used in most related works have similar amounts of spam data and non-spam data. Therefore, it is interesting to see the effectiveness of the machine learning based detection schemes on data sets with various distributions of spam and non-spam classes.

In this paper, we begin with investigating the class imbalance problem in machine learning based Twitter spam detection based on real-world data. We first examine the detection performance in data sets with varying class imbalance rates from 2 to 20 (that is, the number of non-spam tweets is twice to twenty times as the number of spam tweets). The experimental results show that the increase of class imbalance rate leads to slightly better precision in spam detection along with an over 30% decrease in detection rate. In other words, in the data sets that the non-spam tweets extremely outnumber the spam tweets, a large proportion of spam tweets can be missed by the detection schemes.

To address the problem, we firstly propose a Fuzzy-based Oversampling (FOS) method, which generates synthetic data samples based on a limited amount of observed data samples by using the idea of fuzzy-based information decomposition. In particular, FOS divides the feature space into a number of consecutive intervals in each dimension using unsupervised binning and then it adopts a fuzzy membership function to measure the information contributions from the observed data values to each interval, based on which the synthetic data values are derived. Secondly, we propose an ensemble learning approach for learning Twitter Spam classifiers from imbalanced data. The approach consists of three steps. The first step generates a number of balanced data sets from the original imbalanced training data set by adjusting the class distribution using various methods, including random oversampling, random undersampling and the proposed FOS. The second step aims to train a classification model from each of the redistributed data sets. The final step combines the predictions from all the classification models to reach a final decision using a majority voting scheme.

Extensive experiments based on real-world Twitter data sets have been conducted to evaluate the proposed scheme in comparison to existing works. In particular, we examine the spam detection accuracy of different methods in regards to the impact of different class imbalance rates in training data and different amounts of Spam Tweets in training data. The results indicate that the proposed approach is able to detect much more Spam Tweets at the cost of slightly higher false positive rate. For example, when the class imbalance rate is 20 in the data set, the proposed ensemble approach is 10% higher in true positive rate and 0.4% higher in false positive rate in comparison to the best classifier learned from imbalanced data (i.e., Random Forest classifier in our experiment).

The rest of this paper is organized as follows. Section 2 presents a review on the recent advance in Twitter spam detection and ensemble learning techniques. In Section 3, we demonstrate the class imbalance problem in Twitter spam detection and then present the proposed FOS method as well as the ensemble learning approach. Comparative experimental results obtained from real-world Twitter data are presented and analyzed in Section 4. Finally, Section 5 concludes the paper and discusses future research directions.

2. Related work

Spam mitigation is one of the key security challenges in online social networks as well as in the cyber space in general (Choo, 2011; Lai et al., 2015; Norouzi et al., 2015; Quick et al., 2014). Traditionally, blacklists are used for detecting and filtering unwanted information including spam. For example, Twitter implements a blacklist filtering module in their anti-spam system BotMaker (Jeyaraman, 2014). Trend Micro (Oliver et al., 2014) offers a blacklisting service based on the Web Reputation Technology, which is able to filter harmful spam URLs. Blacklists have a critical disadvantage. That is, it takes some time for the new malicious links to be included in the blacklists. A lot of damages can have already been caused during the time lag (Grier et al., 2010; Thomas et al., 2011).

Heuristic rule based methods for filtering Twitter spam have been developed in some earlier attempts to overcome the limitations of blacklisting. Yardi et al. (2010) proposed to detect spam in #robotpickupline (hashtag created by themselves) through three rules, which are suspicious URL search, username pattern matching and keyword detection. Kwak et al. (2010) proposes to remove all the tweets that contain more than three hashtags so as to eliminate the impact of spam for their research.

A lot of recent studies propose to apply machine learning techniques for identifying Twitter spam based on a range of features, including tweet-based, author-based, and social graph based attributes. For instance, Wang (2010) presents an approach based on Bayesian models to detect spammers on Twitter, and Benevenuto et al. (2010) propose to detect both

spammers and spam using the Support Vector Machine (SVM) algorithm. In order to avoid the class imbalance problem, they randomly select only 710 of the legitimate users for the study. Later in 2013, Mukherjee et al. (2013) employ SVM in their study of spamming behavior analysis on real-life Yelp data, and suggest that the performance of linguistic features is not very effective compared with the behavioral features. Stringhini et al. (2010) train a classifier using the Random Forest algorithm, which is then used to detect spam in three social networks, including Twitter, Facebook and MySpace. Lee et al. (2010) deploy some honeypots to derive the spammers' profiles, and they extract the statistical features for spam detection using several machine learning algorithms, such as Decorate, RandomSubSpace and J48. In Sheu et al. (2016), the authors propose an intelligent three-phase spam filtering method based on decision tree classification. Furthermore, in Dayani et al. (2015) KNN and NB are employed for spam detection and the experimental results indicate that content based features (particularly word frequencies) play a key role in rumor detection.

It has been shown that some basic features used in the above studies can be easily fabricated by purchasing followers, posting more tweets, or mixing spam with normal tweets. Accordingly, researchers propose some robust features that rely on the social graph to avoid feature fabrication. For example, Song et al. (2011) propose to extract the distance and connectivity between a tweet author and its audience to determine whether it is a spam tweet. After merging the sophisticated features with the basic feature set, they show that the performance of several classifiers is improved to nearly 99% True Positive and less than 1% False Positive. Yang et al. (2013) also propose some robust spam features, which include Local Clustering Coefficient, Betweenness Centrality and Bidirectional Links Ratio. They show that their feature set can outperform the features used in the previous works (Benevenuto et al., 2010; Lee et al., 2010; Stringhini et al., 2010; Wang, 2010). In this work, we use the simple tweet-based and author-based features to study the impacts of class imbalance. We do not adopt the advanced features, because they require comprehensive knowledge of user connections.

Besides, some researchers resort to analyzing the embedded URLs in tweets to detect spam. Thomas et al. (2011) make use of several URL based features, such as the domain tokens, path tokens and query parameters, along with some features of the landing page, such as DNS information and domain information. Lee and Kim (2013) investigate into the characteristics of Correlated URL Redirect Chains, and some relevant features, such as URL redirect chain length, relative number of different initial URLs.

Ensemble learning has been widely adopted in various application domains for the improvement of the performance of individual classifiers. Kuncheva (2004) present a strategy of building classifier ensembles for non-stationary environments. Polikar (2006) explore in the area of decision making in terms of ensemble based systems. The study shows that ensemble based systems usually produce favorable results compared to an individual system with a variety of scenarios and a broad range of applications. Khreich et al. (2012) develop an adaptive ROC-based ensemble selection algorithm for the purpose of anomaly detection. An ensemble undersampling based strategy is proposed in Jin et al. (2015) to deal with the class imbalance problem on spam detection in Weibo. The proposed method splits the non-spam into R subsets randomly, and then combines each of them with the spam samples to train a classification model. The final decision making is based on the ensemble of these individual classifiers, which is similar to the idea of ensemble random undersampling. Mi et al. (2016) develop an ensemble feature construction-based algorithm for email spam detection using the term space partition method.

In Liu et al. (2016a), we present a preliminary study on the class imbalance problem in Twitter spam detection. In the current paper, we extend the work in Liu et al. (2016a) significantly with the following key points. Firstly, we present an extended explanation of the Fuzzy-based oversampling (FOS) method and provide formal analysis on the property of the synthetic data generated by FOS. Secondly, we extend the experiments for more comprehensive evaluation. In specific, we evaluate the impact of the amount of spam samples in training data, in addition to the impact of class imbalance rate. Moreover, we present an experimental comparison with traditional spam classification techniques.

3. Learning from imbalanced Twitter spam data

3.1. *Class imbalance problem in Twitter spam detection*

In order to apply machine learning techniques to Twitter spam detection, each tweet in the data set is presented as a feature vector. The feature vector consists of the observed values on a set of predetermined features of a tweet. In this work, we define the feature set using both tweet-based and author-based attributes. Specifically, the tweet-based features are number of retweets, number of hashtags, number of user mentions, number of URLs, number of characters, and number of digits in the tweet. Besides, the author-based features include number of followers, number of followings, number of published tweets, number of user favorites, and number of lists. The complete list of features is given in Table 1. We note that all the selected features can be extracted out of the tweets with

Table 1 – Twitter feature set.

No.	Feature	Description
1	account_age	The number of days since user account creation
2	no_follower	The number of followers of this twitter user
3	no_following	The number of followings/friends of this twitter user
4	no_user_favorite	The number of favorites this twitter user received
5	no_list	The number of lists this twitter user added
6	no_tweet	The number of tweets this twitter user sent
7	no_retweet	The number of retweets this tweet
8	no_hashtag	The number of hashtags included in this tweet
9	no_user_mention	The number of user mentions included in this tweet
10	no_URL	The number of URLs included in this tweet
11	no_char	The number of characters in this tweet
12	no_digit	The number of digits in this tweet

little computational overhead, such that they are suitable for real-time detection.

Now suppose we are given a set of labeled data consisting of n spam tweets and m non-spam tweets: $D = \{(\mathbf{x}_1, \omega_+), \ldots, (\mathbf{x}_n, \omega_+), (\mathbf{x}_{n+1}, \omega_-), \ldots, (\mathbf{x}_{n+m}, \omega_-)\}$, in which each $\mathbf{x}_i \in \mathbb{R}(i = 1, \ldots, n+m)$ is the feature vector of the ith tweet, while ω_+ and ω_- are the corresponding class labels for spam and non-spam respectively (here we consider spam tweets to be the positive class and non-spam tweets to be the negative class). Based on the data set, we can train a classification model using supervised learning algorithms, which can predict whether any given testing tweets belongs to ω_+ or ω_- (i.e., $F(\mathbf{x}) : \mathbb{R} \to \{\omega_+, \omega_-\}$).

As in many data mining application domains, Twitter spam detection faces the class imbalance problem. That is, in a data set randomly collected from the Twitter platform, the number of spam tweets is usually much less than the number of non-spam tweets (i.e., $n \ll m$). In this work, we define the class imbalance rate in data set D as:

$$\gamma = \frac{n}{m}.$$

In practice, the class imbalance rate can be varying depending on the activeness of spam campaigns during the observation or data collection period. For example, a study based on a data set of 2000 tweets (Twitter Study, 2009) showed that 3.75% of the collected tweets are spam. This yields a class imbalance rate over 25. However, most existing works on machine learning based spam detection are carried out without the class imbalance problem in mind, so that their evaluation results are obtained from relatively balanced data sets, which are purposely formed by including similar amounts of spam and non-spam tweet samples. Therefore, such results cannot reflect the actual performance of the spam classifiers in the real world.

To demonstrate the problem, we conduct a series of preliminary experiments using data sets with class imbalance rates varying from 2 to 20 with incremental steps of 2 (i.e., $\gamma = 2, 4, 6, \ldots, 20$). Fig. 1 shows the detection performance results, which are obtained using Random Forest (Breiman, 2001) as the base classification algorithm. We can find that as the class imbalance rate rises from 2 to 20, the true positive rate of the positive class (i.e., the spam detection rate) witnesses a significant decrease of 33% in average (among 10 independent tests with different data sets). In particular, when $\gamma = 20$, the averaged detection rate is down to 34%, which means over 66% of spam are missed by the detection. In the meantime, the false positive rate drops from around 6% to less than 1% when the class imbalance rate increases. This is because the

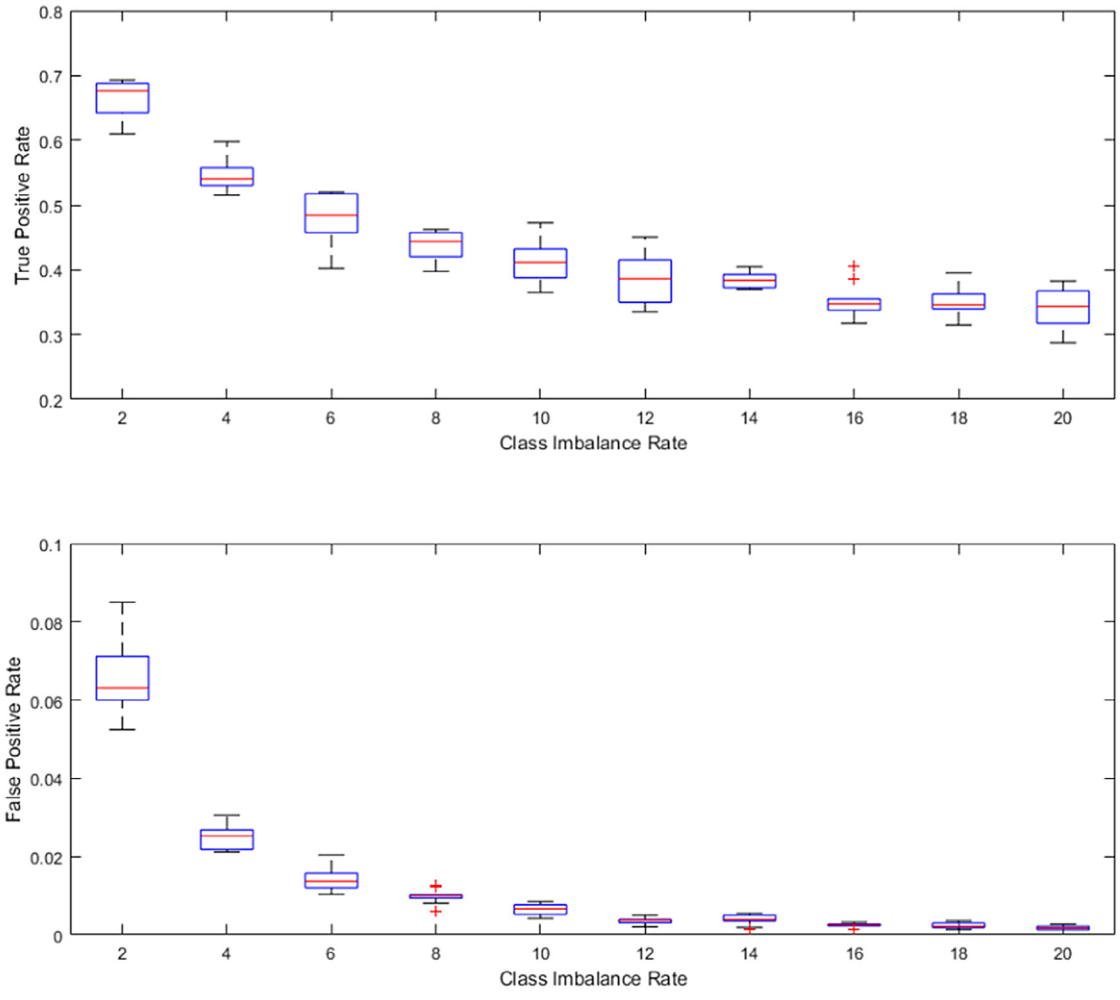

Fig. 1 – Performance degradation caused by class imbalance.

classifier is significantly biased to the negative class. In other words, the imbalance rate between spam and non-spam classes in real world data will have a significant impact on the spam detection rate.

3.2. Proposed ensemble approach

3.2.1. Data sampling techniques

In order to address the class imbalance problem in Twitter spam detection, we propose an ensemble learning approach that incorporates a majority voting scheme to combine multiple classification models. In particular, each model is independently built upon a data set that is re-balanced from the given imbalance data set using one of the following data sampling methods: random oversampling, random undersampling and fuzzy-based oversampling.

Random oversampling (ROS): This method takes as input the data of the spam class D_+ and an oversampling ratio α. It then randomly selects a number of $\alpha \times n$ samples from D_+ with replacement. The generated data samples are combined with D_+ to form the new set of training samples D'_+ for the spam class.

Random undersampling (RUS): This method takes as input the data of the non-spam class D_- and an undersampling ratio β. It then randomly selects a number of $(1-\beta)\times m$ samples from D_- and discards the rest to form a new non-spam set D'_-.

Fuzzy-based information decomposition oversampling (FOS): This method takes as input the data of the spam class D_+ as well as an oversampling ratio α. It then generates a number of $\alpha \times n$ synthetic samples based on the idea of fuzzy-based information decomposition (Liu et al., 2016b). In general terms, the algorithm derives the synthetic samples based on the observed distribution of original minority class data samples. Specifically, the algorithm first divides each dimension of the feature space into $t = \alpha \times n$ intervals, such that each interval contains approximately the same amount of observed values. Secondly, it employs a fuzzy membership function μ to estimate the weights of the observed data values with respect to each interval. Finally, these weights are used for generating a synthetic data value from each interval. The details of the algorithm are described in the following section.

3.2.2. Fuzzy-based information decomposition oversampling

The FOS algorithm is designed to generate synthetic data values based on the limited observed sample values independently in each dimension. Suppose that in the current dimension under consideration the observed data values are $\mathbf{y} = \{x_1, x_2, \ldots, x_n\}$.

In the first step, FOS adopts the unsupervised binning approach to divide the feature space into a number of consecutive intervals. In particular, let a be the minimum of the observed values, i.e.,

$$a = \min_i \{x_i\} \tag{1}$$

and b be the maximum of the observed values, i.e.,

$$b = \max_i \{x_i\}. \tag{2}$$

FOS partitions $[a, b]$ into t intervals, where each interval contains approximately n/t, where n is the total number of observed values. For each interval, we denote the width of the interval as $h_s (s = 1, 2, \cdots, t)$ and the center of the interval as $u_s (s = 1, 2, \cdots, t)$.

In the next step, FOS creates a mapping from the feature space to a discrete universe set:

$$\mu : \mathbf{y} \times \mathbf{u} \to [0, 1], \tag{3}$$

$$(x_i, u_s) \to \mu(x_i, u_s), \tag{4}$$

where $\mathbf{u}$ is a discrete universal set of $\mathbf{y}$. Specifically, we use the following fuzzy membership function:

$$\mu(x_i, u_s) = \begin{cases} 1 - \dfrac{\|x_i - u_s\|}{h_s}, & if\ \|x_i - u_s\| \le h_s \\ 0, & otherwise \end{cases} \tag{5}$$

Finally, FOS generate a synthetic data value from each interval in the following way. In the case that $\sum_i \mu(x_i, u_s) = 0$ (i.e., there is not any information contributions from the observed data values to this interval), we set the sth generated data value $\tilde{m}_s$ as the mean of all observed values. Otherwise, the generated data value is set to be the weighted mean of the data values that have information contributions to the interval. That is,

$$\tilde{m}_s = \begin{cases} \bar{y}, & if\ \sum_i \mu(x_i, u_s) = 0 \\ \dfrac{\sum_{i=1}^{n} m_{is}}{\sum_{i=1}^{n} \mu(x_i, u_s)}, & otherwise \end{cases} \tag{6}$$

where

$$m_{is} = \mu(x_i, u_s) \times x_i. \tag{7}$$

Equation (6) suggests that there are two scenarios in the proposed algorithm. Generally, if the number of synthetic data samples to be generated is less than the total number of minority class samples, we can apply the proposed algorithm directly. Otherwise, when the number of synthetic data samples to be created is more than the number of minority class samples, the mean of the minority class samples is used as the estimated values.

Theorem 1. *When $\sum_i \mu(x_i, u_s) \neq 0$ (i.e., at least some observed data contribute information to the interval), the generated data value*

$$\tilde{m}_s \in \left(a_s - \frac{h_s}{2}, b_s + \frac{h_s}{2} \right) \tag{8}$$

where $[a_s, b_s]$ denotes the boundary of sth interval.

Proof. Recall that u_s is the center of the *sth* interval and h_s is the width of the *sth* interval, thus we have $h_s = b_s - a_s$ and $\|a_s - u_s\| = \|b_s - u_s\|$.

According to the fuzzy membership function, if $x_i - u_s > h_s$ then $\mu(x_i, u_s) = 0$. This leads to the statement: if $x_i > b_s + h_s/2$ then $\mu(x_i, u_s) = 0$. Therefore, when $\sum_i \mu(x_i, u_s) \neq 0$

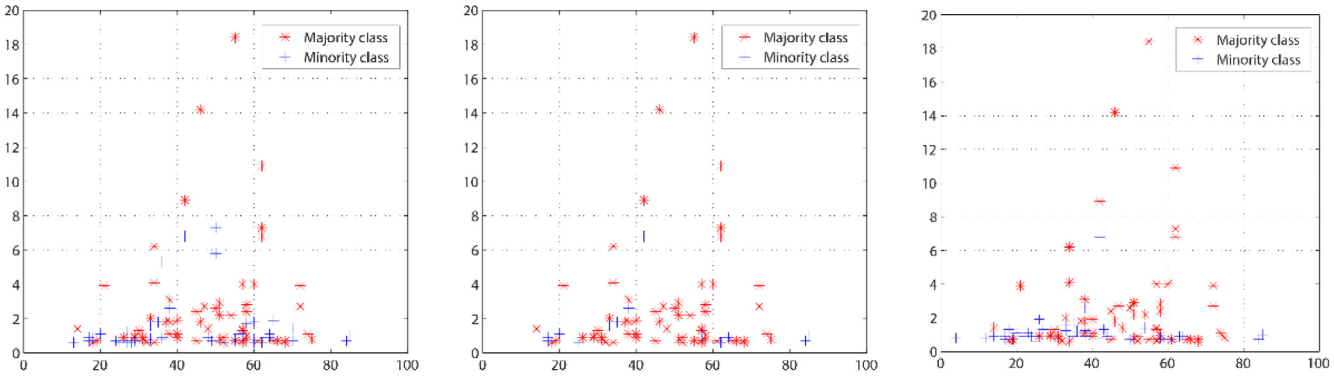

Fig. 2 – Illustration of the FOS approach based on two-dimensional simulation data (labels on x-axis and y-axis represent data values).

$$\begin{aligned}\tilde{m}_s &= \frac{\sum_{i=1}^{n} m_{is}}{\sum_{i=1}^{n}\mu(x_i,u_s)} \\ &= \frac{\sum_{i=1}^{n}\mu(x_i,u_s)\times x_i}{\sum_{i=1}^{n}\mu(x_i,u_s)} \\ &< \frac{\sum_{i=1}^{n}\mu(x_i,u_s)\times\left(b_s+\frac{h_s}{2}\right)}{\sum_{i=1}^{n}\mu(x_i,u_s)} \\ &< b_s+\frac{h_s}{2}\end{aligned} \tag{9}$$

Similarly, we have: if $x_i < a_s - h_s/2$ then $\mu(x_i,u_s)=0$. Therefore, when $\Sigma_i\mu(x_i,u_s)\neq 0$

$$\begin{aligned}\tilde{m}_s &= \frac{\sum_{i=1}^{n} m_{is}}{\sum_{i=1}^{n}\mu(x_i,u_s)} \\ &= \frac{\sum_{i=1}^{n}\mu(x_i,u_s)\times x_i}{\sum_{i=1}^{n}\mu(x_i,u_s)} \\ &> \frac{\sum_{i=1}^{n}\mu(x_i,u_s)\times\left(a_s-\frac{h_s}{2}\right)}{\sum_{i=1}^{n}\mu(x_i,u_s)} > a_s-\frac{h_s}{2}\end{aligned} \tag{10}$$

Fig. 2 gives an illustration of the FOS approach based on simulation data. In particular, the original data set shown in subgraph (a) consists of a random sample of equal size from two distributions with normal probability density functions. In the imbalanced data set shown in subgraph (b), 80% of the data points from the blue class are randomly removed so that the data set becomes class imbalanced. FOS is applied to this data set and the rebalanced data are shown in subgraph (c). We can see that the distribution of the blue class data is similar to the original data.

3.2.3. *Ensemble learning*

As shown in Fig. 3, the proposed ensemble learning approach consists of three steps.

In the first step, we adopt the data sampling methods discussed in previous sections to pre-process the imbalanced training data set. At the end of this step, three re-distributed datasets are obtained.

In the second step, a classification model is trained from each of the re-distributed data sets respectively (i.e., $D'_+ \cup D_-$, $D_+ \cup D'_-$, and $D''_+ \cup D_-$). In this work, we adopt a number of widely used supervised learning algorithms for the purpose

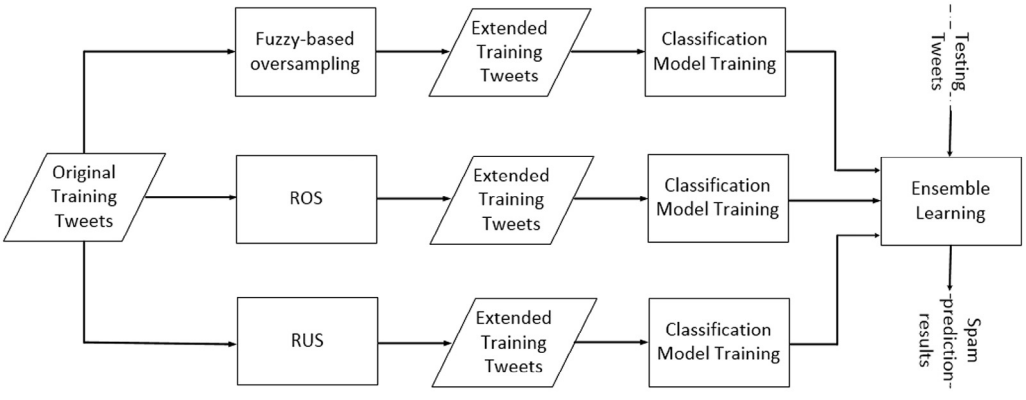

Fig. 3 – Framework of the proposed approach.

Table 2 – The proposed ensemble learning algorithm.

Ensemble learning algorithm
TRAINING
INPUT: Training data set D; Oversampling ratio α; Undersampling ratio β;
OUTPUT: Multiple base classifiers $F_1(\mathbf{x})$, $F_2(\mathbf{x})$, and $F_3(\mathbf{x})$
1: Generate $D_1 = D'_+ \cup D_-$ with random oversampling method $D'_+ = ROS(D_+, \alpha)$;
2: Generate $D_2 = D_+ \cup D'_-$ with random undersampling method $D'_- = RUS(D_-, \beta)$;
3: Generate $D_3 = D''_+ \cup D_-$ with FOS method $D''_+ = FOS(D_+, \alpha)$;
4: for i in 1, 2, and 3; do
5: Train classifier F_i using base classification algorithm: $F_i(\mathbf{x}) = BASE(D_i)$;
6: end for
7: return $F_1(\mathbf{x})$, $F_2(\mathbf{x})$, and $F_3(\mathbf{x})$
TESTING
INPUT: Test data point $\mathbf{z}$
OUTPUT: Class prediction ω for $\mathbf{z}$
8: $v_{i+} = 0$, $v_{i-} = 0$;
9: for i in 1, 2, and 3 do
10: if $F_i(\mathbf{z}) = \omega_+$ do $v_{i+} = v_{i+} + 1$;
11: else do $v_{i-} = v_{i-} + 1$;
12: end if
13: end for
14: if $v_{i+} > v_{i-}$ do return ω_+;
15: else do return ω_-;
16:.end if

of evaluation, including Naïve Bayes, Support Vector Machines (SVM), C4.5 Decision Trees and Random Forest (Weka 3: Data Mining Software in Java).

In the last step, a majority voting scheme is introduced to combine the classification models. In other words, given any given testing data, each of the classification models will derive a prediction on the class (spam or non-spam), and these predictions are combined with majority voting to derive the final decision.

The algorithm is summarized in Table 2. Lines 1–3 implement the data resampling in the first step. Lines 4–7 implement the model training on individual rebalanced data sets for the second step. Lines 8–16 implement the majority voting scheme for the purpose of testing (spam detection).

4. Evaluation

4.1. Evaluation methods

To evaluate the proposed approach, we use a real-world Twitter spam data set that we published in an earlier study (Chen et al., 2015). The data set consists of more than 600 million tweets with URLs. The ground truth of the data set is set up using the Web Reputation Service provided by Trend Micro. In particular, the service is able to identify whether any given URL is malicious as well as to which category the URL belongs. In our study, we define the tweets that contain malicious URLs as Twitter spam. The data set has been made publicly available (Twitter data sets) to fellow researchers for the purpose of validation and extension.

We construct a series of experimental data sets with ten different class imbalance rates, which are $\gamma = 2, 4, 6, \ldots, 20$. For example, given a class imbalance rate of 10, we randomly select one thousand spam tweets and ten thousand non-spam tweets to form the spam class and the non-spam class respectively. Moreover, for each of the class imbalance rates, we repeat the process for 10 times to derive 10 independent data sets. In other words, the results given in the following are obtained from 10 independent tests. In each derived data set, half of the data samples from each class are randomly selected to be training data and the rest are used for testing.

We use a number of metrics to measure the spam detection performance, including True Positive Rate (which is also called recall, or spam detection rate in our application domain), False Positive Rate, Precision, and F-Measure. Given the true positive, false negative, true negative and false positive as illustrated in Table 3, the above metrics can be calculated as follows (He and Garcia, 2009).

$$True\ Positive\ Rate = \frac{true\ positives}{true\ positives + false\ negatives} \tag{11}$$

$$False\ Positive\ Rate = \frac{false\ positives}{false\ positives + true\ negatives} \tag{12}$$

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{13}$$

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \tag{14}$$

In the evaluation, we first compare our proposed algorithm within each sampling techniques. In particular, since we employ the Random Forest algorithm (Weka 3: Data Mining Software in Java) in our study, in this respect, we report the results for Random Forest in Section 4.2. Then, to emphasize the effectivity of the proposed algorithm, we have provided a comparison with traditional developed or popular used approaches [N1], which includes support vector machine, C4.5 decision tree, K nearest neighbors (KNN), NB (Naive Bayes), RUSBoost. The experiment results are presented in Section 4.3. Both experiment results indicate that our proposed algorithm outperforms the other techniques.

For the purpose of comparison, the performance of learning from the imbalanced data directly is given as a baseline. Besides, we compare the proposed ensemble learning approach with the fundamental methods for processing imbalanced data, i.e., random oversampling (ROS), random undersampling (RUS), and the fuzzy-based oversampling (FOS). For each of the oversampling methods, we test a number of oversampling rates (i.e., α) ranging from 20% to 2000%, plus the

Table 3 – Confusion matrix.

	Classified as spam	Classified as non-spam
Spam	True positive	False negative
Non-spam	False positive	True negative

equal class distribution setting (oversampling until the two classes have the same number of data points). For the undersampling method, we test the undersampling rates (i.e., β) ranging from 20% to 90%, plus the equal distribution setting (remove majority class data until the two classes have the same number of data points). Through extensive experiments, we find that the spam detection performance is not sensitive to the selection of resampling rate. That is, different oversampling settings yield similar detection accuracy for FOS and ROS. The same happens to RUS with an undersampling rate between 20% and 90%. The equal distribution setting for RUS is an exception. It yields much poorer results as a standalone approach because it prunes too much information of the negative class (i.e., non-spam tweets), but it boosts the detection accuracy of the ensemble approach in which it is used together with FOS and ROS as a voter. In summary, the results in the following sections are based on the parameters that yield the best performance of each approach in average. In particular, we use $\alpha = 200\%$ for oversampling (that is, FOS and ROS), $\beta = 90\%$ for standalone undersampling (i.e., RUS), and equal distribution for undersampling in the ensemble approach.

4.2. *Results of comparing with sampling techniques*

We first look at the situation where the unequal distribution between classes is not so severe. Fig. 4 presents the spam detection results derived by random forest classifiers based on data sets in which the number of non-spam tweets are twice as the number of spam tweets (i.e., $\gamma = 2$).

In terms of true positive rate, directly learning from data sets with an imbalance rate of 2 yields 66% in average. While the oversampling methods FOS and ROS improve the performance to 74% and 72%, the undersampling method RUS achieves an even better result at 78%. The ensemble approach also manages to raise the true positive rate to 75%.

The ensemble approach produces an average false positive rate at 11%, which is much lower than RUS (over 16%) and slightly higher than FOS and ROS at round 10%. The classifiers trained on the original imbalanced data obtained the lowest false positive rate at around 7.5%. This indicates that sampling techniques boost the true positive rate at the cost of raising false positive rate in the meantime. The precision result is analog to false positive rate result. That is, the original imbalanced data yield the best results, followed by FOS, ROS, Ensemble approaches, while RUS produces the poorest precision.

The last subgraph of F-Measure result shows that directly learning from imbalanced data yields poorer performance. In addition, FOS, ROS, RUS and the ensemble approach can increase the performance to different extents.

Next, we look at the situation where the non-spam data significantly outnumber the spam data. Similarly, Fig. 5 shows the results derived by random forest classifiers based on data sets in which the number of non-spam tweets are ten times as the number of spam tweets (i.e., $\gamma = 10$). In this case, the result in terms of true positive rate is quite different to the case in Fig. 4. Firstly, the performance of directly learning from data sets with an imbalance rate of 10 drops to 41%. Secondly, RUS yields slightly higher true positive rate results at 43%. Thirdly, ROS

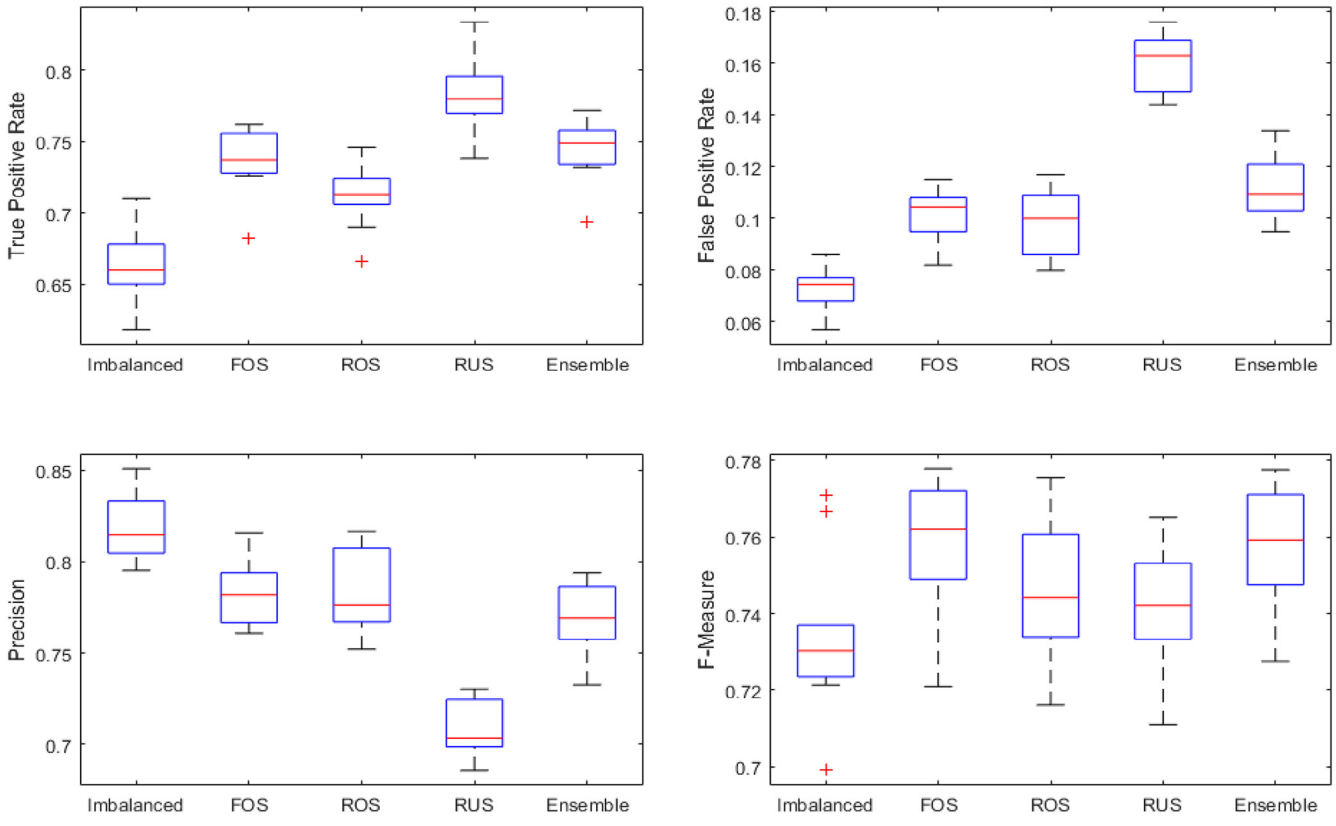

Fig. 4 – Spam detection results in data sets with a class imbalance rate of 2.

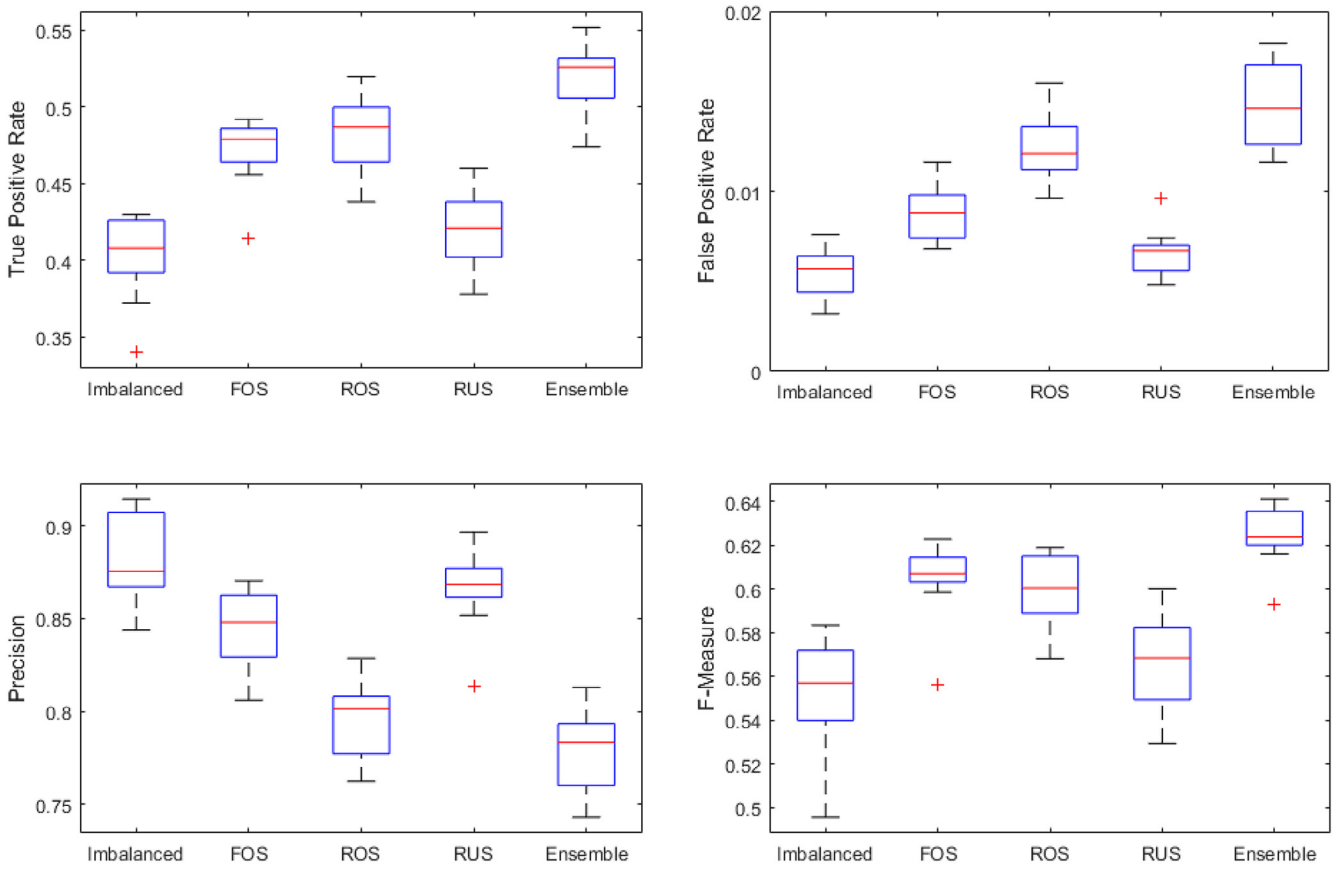

Fig. 5 – Spam detection results in data sets with a class imbalance rate of 10.

and FOS bring an improvement of roughly 7% and the ensemble approach manages to significantly raise the true positive rate with 12%. Fig. 5 also shows that the average false positive rate of all methods is lower than 2% and the average precision of all methods is higher than 78%.

In the last subgraph in Fig. 5, we can see that the imbalanced training data lead to the poorest performance at 56% in terms of average F-Measure. FOS, ROS and RUS increase the performance to 61%, 60%, and 57% respectively. Finally, the ensemble approach obtains the best F-Measure performance at 63%. Since F-measure is the harmonic mean of precision and recall. In this respect, higher F-measure values indicate the proposed algorithm can train a better classifier.

4.2.1. *Impact of class imbalance rate*

In the last section, we have seen that when the non-spam tweets significantly outnumber the spam tweets in the data, the performance of machine learning based spam detection is degraded. In the following, we continue to investigate how varying class imbalance rates affect the spam detection performance.

Fig. 6 compares the averaged true positive rate of the five different approaches in regards to the class imbalance rate γ. When $\gamma = 2$, random forest classifiers are able to identify 66% of the spam tweets by directly learning from the imbalanced data sets. By preprocessing the imbalanced data sets using FOS, ROS, and RUS, the detection rates are improved to 74%, 74% and 79% respectively. Moreover, the proposed ensemble approach successfully identifies 75% of the spam tweets in average. As γ increases, the true positive rate of all approaches exhibits a gradual decrease. After $\gamma > 4$, the true positive rate results of the ensemble approach remain the best, which are over 10% higher than directly learning from the imbalanced data.

Fig. 7 depicts the average false positive rate of the different approaches with respect to the class imbalance rate. When the imbalance among classes is not significant (i.e., $\gamma = 2$), the ensemble approach yields a false positive rate of 11% in average, while the classifiers trained from imbalanced data directly obtain the second lowest at around 6.5%. The three data preprocessing methods FOS, ROS, and RUS raise the false positive rate to 9.6%, 10.6% and 16% respectively. As the class imbalance rate rises to 4, the false positive rates of all approaches decrease to below 5%. If γ rises over 16, the false positive rates of all approaches fall below 1%. This is expected as the classifiers are biased to the negative class, so that most of the errors are type II errors (false negatives) and type I errors (false positives) are sparse.

The averaged spam detection precision performance is illustrated in Fig. 8. Learning from imbalanced data directly can achieve 82% to 90% precision across all of the class imbalance rates, and the ensemble approach shows an average precision between 76% and 78%, while the precision of other methods falls between them.

From the above discussion, we know that the proposed ensemble learning approach identifies more spam tweets (higher true positive rate) at the cost of raising a bit more false alarms

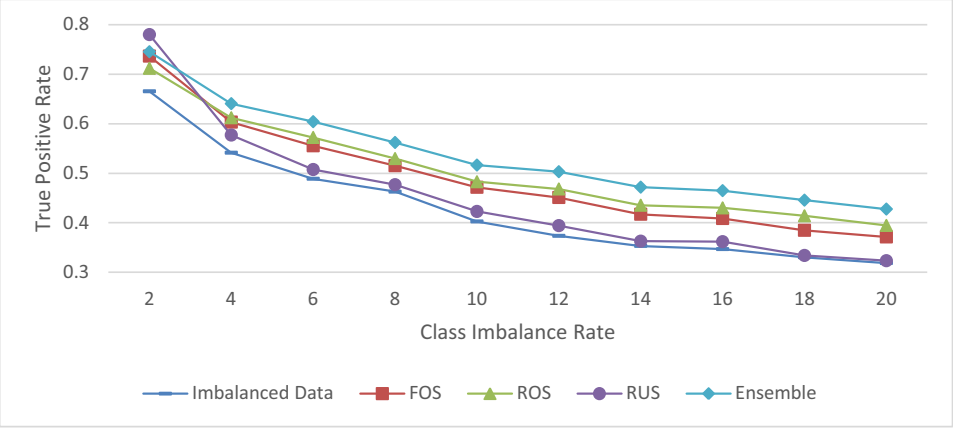

Fig. 6 – True positive rate versus varying class imbalance rate.

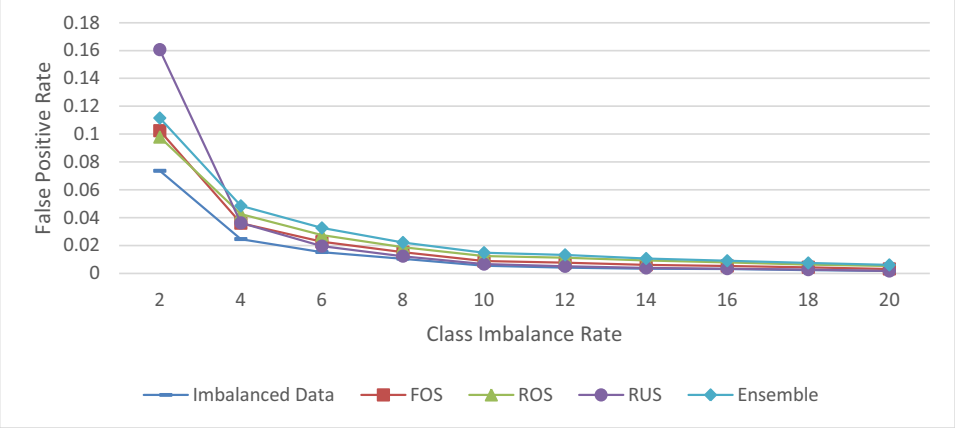

Fig. 7 – False positive rate versus varying class imbalance rate.

(higher false positive rate and lower precision) in the data sets with various rates of class imbalance. This makes the best F-measure results as given in Fig. 9, since the F-Measure is a harmonic mean of precision and recall (true positive rate). In particular, the ensemble learning approach generates an F-Measure of 76% when $\gamma = 2$, and then the value gradually decreases to around 55% as γ goes up to 20. In general, this shows an improvement up to 9% compared with learning from the imbalanced data directly, and also a lead around 3% to the other data preprocessing methods.

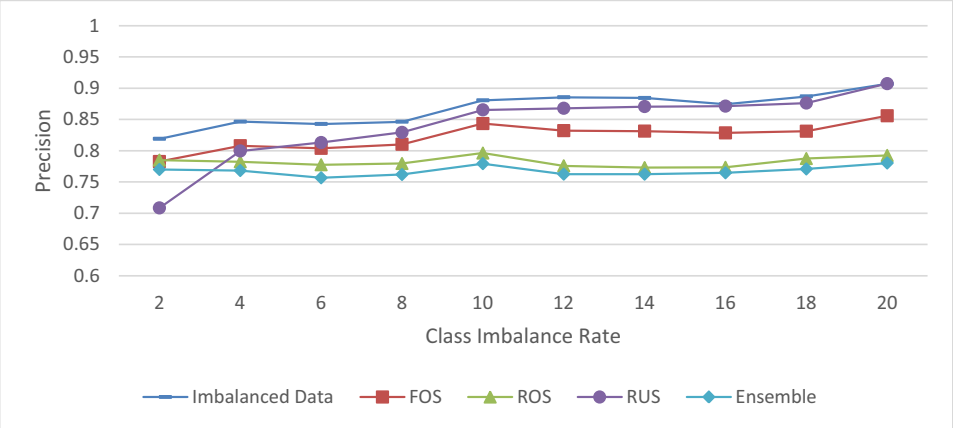

Fig. 8 – Spam detection precision versus varying class imbalance rate.

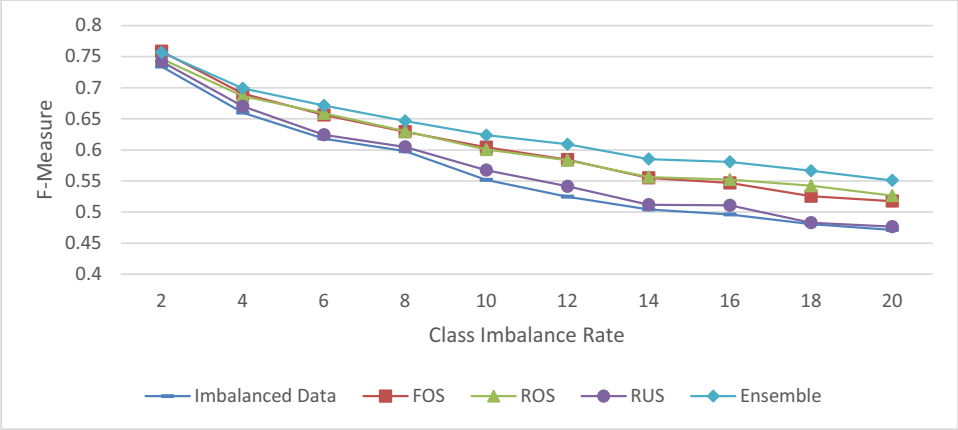

Fig. 9 – Spam detection F-measure versus varying class imbalance rate.

In general, the results show that the proposed ensemble learning scheme yields better detection performance than individual classifiers, which demonstrates the advantage of combining multiple classifiers in generating more certain and accurate results. Besides, the results also show that ROS can cause overfitting in real world problem (He and Garcia, 2009), especially when the imbalance ratio is very high. Therefore, although the accuracy on training data is good, its performance on real world unseen testing data is generally worse. The results also suggest that the performance of RUS is unstable. It is due to the fact that the undersampling scheme can miss some very important information in the random data pruning process, which leads to the poor classification performance.

4.2.2. *Impact of the amount of positive data*

In the above sections, the evaluation results are obtained by fixing the number of spam tweets in each experiment to a thousand. In the following, we explore the impact of increasing sample size. For example, given an imbalance ratio of 2, we explore how different sizes of training data sets, such as 1k spam vs 2k non-spam and 2k spam vs 4k non-spam, can impact the classification prediction results.

Fig. 10 illustrates the true positive rate results obtained with one thousand (1k) and two thousand (2k) spam data respectively. For the purpose of clarity, here we limit ourselves to the comparison of the ensemble approach and the baseline approach of learning directly from imbalanced data. We can see that by increasing the training sample size, the baseline approach obtains 3% to 5% increase given different settings of imbalance rate. Nonetheless, the ensemble approach also sees an obvious improvement from 3% to 6%. In Fig. 11, we can see increasing the number of spam samples also slightly raises the false positive rate for the ensemble learning approach when the class imbalance rate is smaller than 10. In the case where $\gamma > 12$, the false positive rate results of the ensemble approach and the baseline approach drop to below 1% and 0.4% respectively.

In Fig. 12, we can see that by increasing the spam sample size, we can improve the precision with up to 3% for both the ensemble approach and the baseline approach. Finally, Fig. 13 presents the results in terms of F-Measure. We can see that the ensemble approach with 2k spam data derives the best average F-Measure results. It starts at 78% when the imbalance rate is 2, and gradually drops to 60% when the imbalance rate increases to 20.

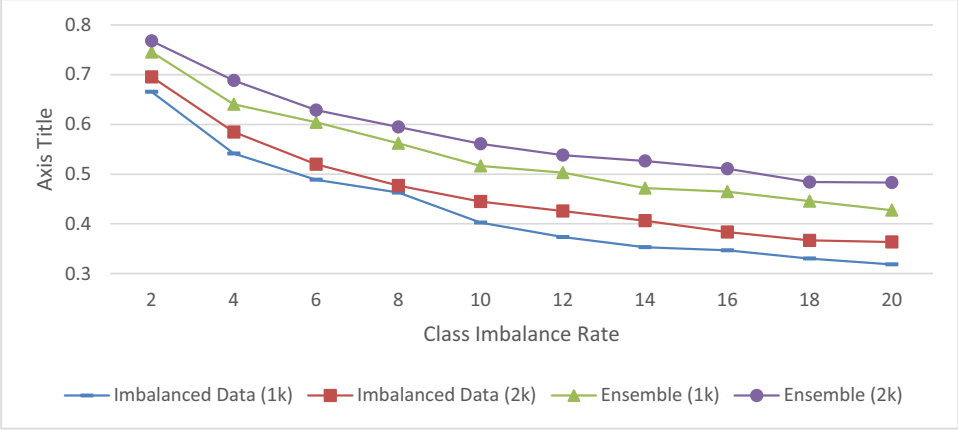

Fig. 10 – True positive rate versus varying class imbalance rate.

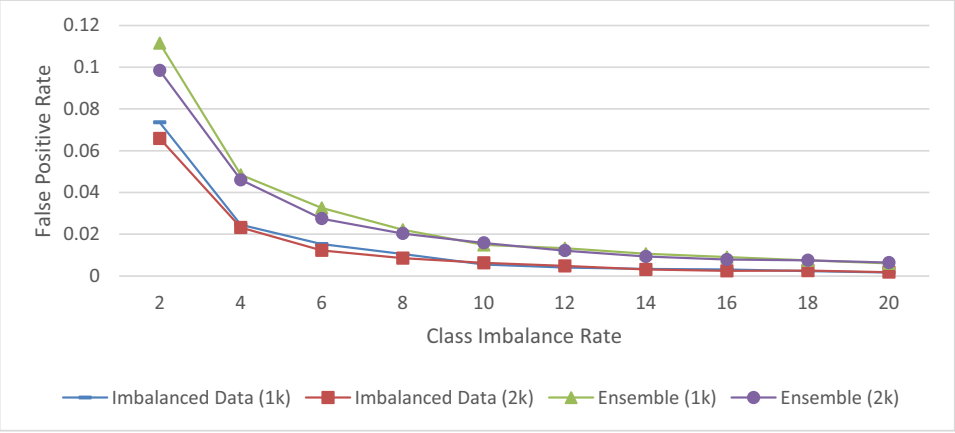

Fig. 11 – False positive rate versus varying class imbalance rate.

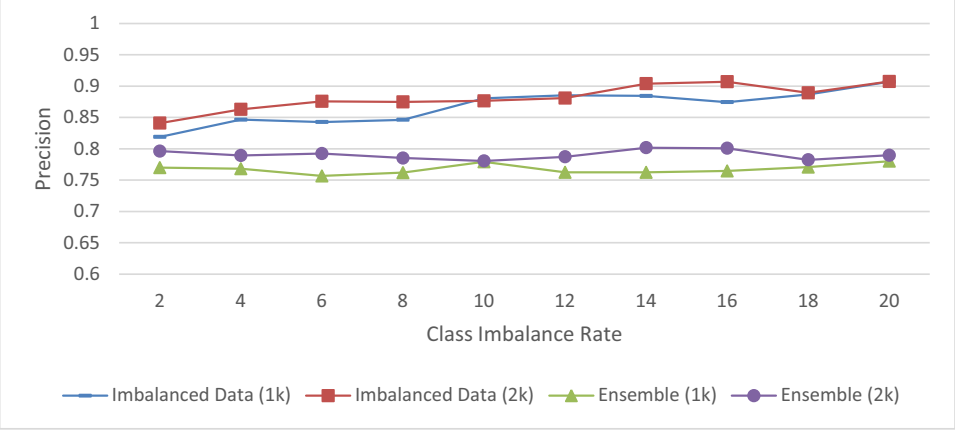

Fig. 12 – Spam detection precision versus varying class imbalance rate.

In general, we reach a conclusion that with the increase in training data size, the performance of the spam classifier is better. This result complies with intuition as with more spam training samples, the machine learning algorithms should be able to learn more knowledge from the data to build more robust classifier even though the class imbalance rate remains the same.

4.3. *Comparison with traditional classification schemes*

In this section, we present a comparative study of the proposed approach with the traditional classification schemes used related works, including SVM (Mukherjee et al., 2013), C4.5 decision tree (Sheu et al., 2016), K-Nearest Neighbor (KNN with

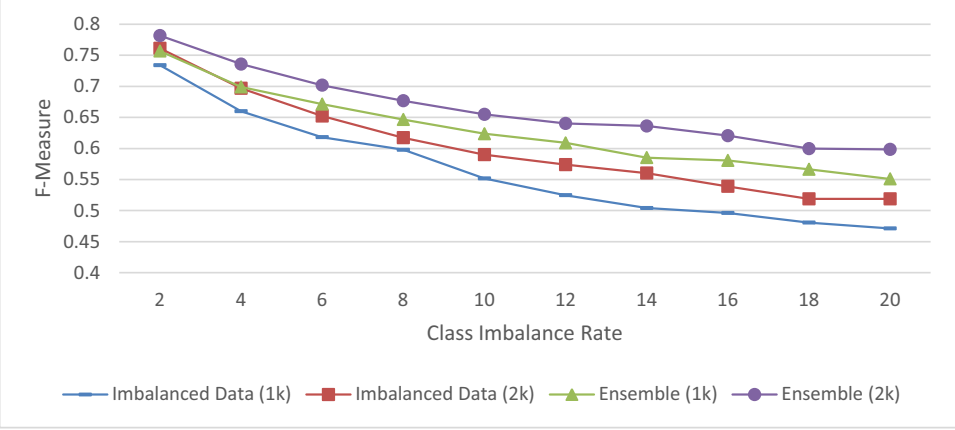

Fig. 13 – Spam detection f-measure with different training data set sizes.

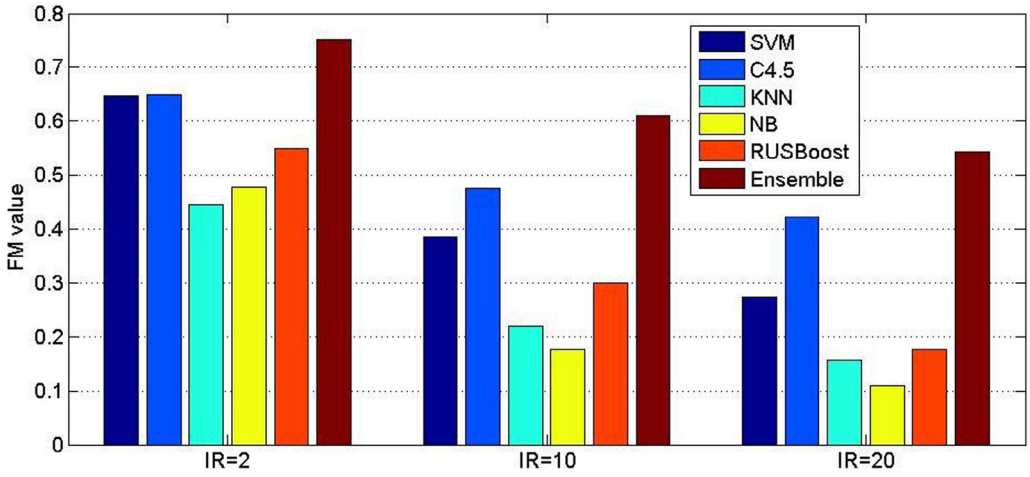

Fig. 14 – F-measure results in comparison with imbalance rate equals to 2, 10, and 20.

K = 3) (Dayani et al., 2015), Naïve Bayes (NB) (Dayani et al., 2015) and Random Undersampling Boost (RUSBoost) (Jin et al., 2015). The results are obtained based on three different settings of the class imbalance rate, i.e., IR equals to 2, 10 and 20.

Fig. 14 presents the performance of each technique in terms of F-Measure (FM). We can see that trend of the F-Measure metric drops gradually with the increase of class imbalance rate. For example, C4.5 decision tree, which performs the second best, drops from about 0.63 with IR = 2 to 0.415 with IR = 20. Naive Bayes performs the worst with the increase of imbalance ratio, which results in only about 0.105 FM value, thus the classification is unusable from a practical point of view. Although the proposed ensemble learning approach also shows a decreasing trend, it still achieves the best FM value among all the classification techniques. For example, it outperforms the second best technique C4.5 decision tree by more than 0.12 in the case of IR = 20.

Fig. 15 depicts the true positive rate (TPR), false positive rate (FPR) and Precision performance of each technique in the case of IR = 10. It can be observed that the other techniques in comparison perform quite differently in terms of different metrics. For example, Naive Bayes achieves the best true positive rate that is higher than 0.73, while it yields the worst false positive rate that is as high as 0.64. As a result, its precision performance is poorest at around 0.12. Besides, for C4.5 decision tree, it yields the second best performance in terms of F-Measure (Fig. 14) and its FPR result is the third highest one at around 0.15, but its precision result is lower than 0.5. In general, the proposed ensemble learning approach outperforms all other techniques in terms of randomly collected Twitter data.

Based on the above results, we reach the conclusion that the proposed approach achieves better Twitter spam detection performance than other techniques in imbalanced data. The reason behind is that we employ three different kinds of data sampling techniques in our approach. Each sampling technique is unique from each other, such that the diversity of resultant training data sets leads to a diversity in the resultant classification models. Then by using the ensemble technique to combine the models, the final prediction can be more robust.

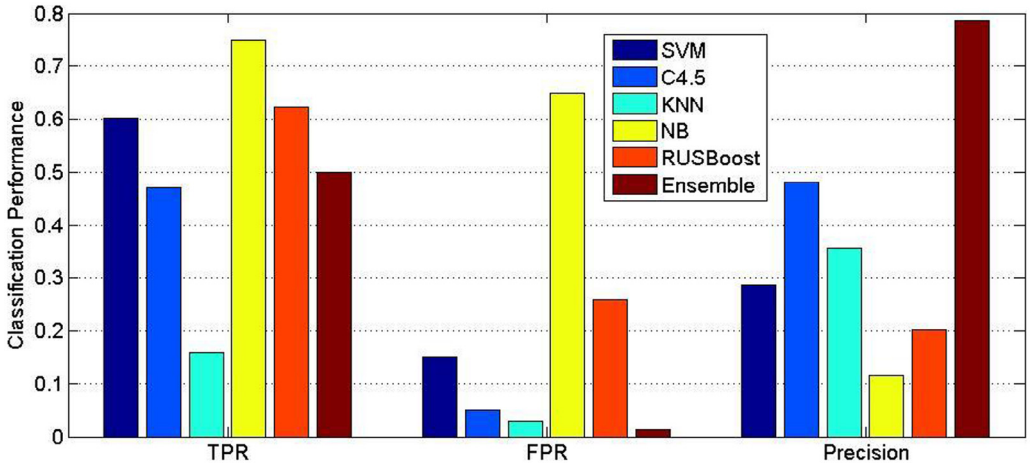

Fig. 15 – TPR, FPR, precision results in comparison with imbalance rate equals to 10.

5. Conclusions and future work

This paper investigates the class imbalance problem in machine learning based Twitter spam detection. It has been shown that the effectiveness of detection can be severely affected by the imbalanced distribution of spam tweets and non-spam tweets, which is widely seen in real-world Twitter data sets. An ensemble approach has been proposed to mitigate the impact of class imbalance. Extensive experiments have been conducted using real-world Twitter data. The results show that the proposed approach can improve the spam detection performance on imbalanced Twitter data sets with a range of imbalance degrees.

Future research directions are identified as follows. Firstly, this paper is the first investigation to the class imbalance problem in the domain of Twitter spam detection. The FOS method proposed in this work employs a fuzzy membership function as in equation (5). An interesting direction is to explore other kinds of membership functions and discover the one that yields the best result in spam detection. Secondly, we assume the Tweet features are independent to each other in the process of generating synthetic data samples using the proposed FOS method. In the future, we plan to extend the synthetic data generation scheme to incorporate correlations among features.

REFERENCES

Benevenuto F, Magno G, Rodrigues T, Almeida V. Detecting spammer on twitter. In: Seventh annual collaboration, electronic messaging, anti-abuse and spam conference. 2010 July.

Breiman L. Random forests. Mach Learn 2001;45(1):5–32.

Chen C, Zhang J, Chen X, Xiang Y, Zhou W. 6 million spam tweets: a large ground truth for timely Twitter spam detection. In: IEEE international conference on communications. ICC; 2015.

Choo K-KR. The cyber threat landscape: challenges and future research directions. Comput Secur 2011;30(8):719–31.

Dayani R, Chhabra N, Kadian T, Kaushal R. Rumor detection in twitter: an analysis in retrospect. In: 2015 IEEE international conference on advanced networks and telecommuncations systems (ANTS). IEEE; 2015. p. 1–3.

Gao H, Chen Y, Lee K, Palsetia D, Choudhary A. Towards online spam filtering in social networks. In: NDSS. 2012.

Grier C, Thomas K, Paxson V, Zhang M. Spam: the under- ground on 140 characters or less. In: Proceedings of the 17th ACM conference on computer and communications security, CCS '10. New York (NY): ACM; 2010. p. 27–37.

He H, Garcia EA. Learning from imbalanced data. IEEE Trans Knowl Data Eng 2009;21(9):1263–84.

Jeyaraman R. Fighting spam with botmaker. Twitter Engineering Blog, August 2014.

Jin Z, Li Q, Zeng D, Wang L. Filtering spam in Weibo using ensemble imbalanced classification and knowledge expansion. In: Intelligence and security informatics (ISI), 2015 IEEE international conference on. IEEE; 2015. p. 132–4.

Khreich W, Granger E, Miri A, Sabourin R. Adaptive ROC-based ensembles of HMMs applied to anomaly detection. Pattern Recognit 2012;45(1):208–30.

Kuncheva LI. Classifier ensembles for changing environments. In: International workshop on multiple classifier systems. Springer Berlin Heidelberg; 2004. p. 1–15.

Kwak H, Lee C, Park H, Moon S. What is twitter, a social network or a news media? In: Proceedings of the 19th international conference on World wide web, WWW '10. New York (NY): ACM; 2010. p. 591–600.

Lai S, Liu JK, Choo KKR, Liang K. Secret picture: an efficient tool for miti-gating deletion delay on OSN. In: Proceedings of 2015 international conference on information and communications security (ICICS 2015), Beijing, China, vol. 9543/2016. Lecture Notes in Computer Science. Springer-Verlag; 2015. p. 467–77 9–11 September.

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

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

Liu S, Wang Y, Chen C, Xiang Y. An ensemble learning approach for addressing the class imbalance problem in twitter spam detection. In: the Proceedings of the 21st Australasian conference information security and privacy (ACISP 2016). Melbourne, Australia: Springer; 2016a. p. 215–28.

Liu S, Zhang J, Wang Y, Xiang Y. Fuzzy-based feature and instance recover. In: Nguyen TN, Trawiński B, Fujita H, Hong T-P, editors. ACIIDS 2016, vol. 9621. LNCS. Heidelberg: Springer; 2016b. p. 605–15.

Mi G, Gao Y, Tan Y. Term space partition based ensemble feature construction for spam detection. In: International conference on data mining and big data. Springer International Publishing; 2016. p. 205–16.

Mukherjee A, Venkataraman V, Liu B, Glance N. Fake review detection: classification and analysis of real and pseudo reviews. UIC-CS-03-2013. Technical Report, 2013.

Norouzi F, Dehghantanha A, Eterovic-Soric B, Choo K-KR. Investigating social net-working applications on smartphones: detecting Facebook, Twitter, LinkedIn, and Google+ Artifacts ON Android and iOS platforms. Aust J Forensic Sci 2015;doi:10.1080/00450618.2015.1066854. In press.

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

Pash C. The lure of naked Hollywood star photos sent the internet into meltdown in New Zealand. Business Insider, September 2014.

Polikar R. Ensemble based systems in decision making. IEEE Circ syst Mag 2006;6(3):21–45.

Quick D, Martini B, Choo K. Cloud storage forensics. Waltham (MA): Syngress Publishing; 2014.

Sheu J-J, Chen Y-K, Chu K-T, Tang J-H, Yang W-P. An intelligent three-phase spam filtering method based on decision tree data mining. Secur Commun Netw 2016;doi:10.1002/sec.1584.

Song J, Lee S, Kim J. Spam filtering in twitter using sender-receiver relationship. In: Proceedings of the 14th international conference on recent advances in intrusion detection, RAID'11. Berlin, Heidelberg: Springer-Verlag; 2011. p. 301–17.

Stringhini G, Kruegel C, Vigna G. Detecting spammers on social networks. In: Proceedings of the 26th annual computer security applications conference, ACSAC '10. New York (NY): ACM; 2010. p. 1–9.

Thomas K, Grier C, Song D, Paxson V. Suspended accounts in retrospect: an analysis of twitter spam. In: Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, IMC '11. New York (NY): ACM; 2011. p. 243–58.

Thomas K, Grier C, Ma J, Paxson V, Song D. Design and evaluation of a real-time url spam filtering service. In: Proceedings of the 2011 IEEE symposium on security and privacy, SP '11. Washington, DC, USA: IEEE Computer Society; 2011. p. 447–62.
Twitter data sets. Available from: http://nsclab.org.
Twitter Study – August 2009. Available from: http://pearanalytics.com/wp-content/uploads/2012/12/Twitter-Study-August-2009.pdf. [Accessed 04 January 2017].
Wang AH. Don't follow me: spam detection in twitter. In: Security and cryptography (SECRYPT), proceedings of the 2010 international conference on. 2010. p. 1–10.
Weka 3: Data Mining Software in Java. Available from: http://www.cs.waikato.ac.nz/ml/weka/.
Yang C, Harkreader R, Zhang J, Shin S, Gu G. Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter. In: Proceedings of the 21st international conference on World Wide Web, WWW '12. USA. 2012. p. 71–80.
Yang C, Harkreader R, Gu G. Empirical evaluation and new design for fighting evolving twitter spammers. IEEE Trans Inf Foren Secur 2013;8(8):1280–93.
Yardi S, Romero D, Schoenebeck G, Boyd D. Detecting spam in a twitter network. First Monday 2010;15(1–4).
Zhang X, Zhu S, Liang W. Detecting spam and promoting campaigns in the twitter social network. In: Data mining. IEEE ICDM2012. 2012. p. 1194–9.