Statistical Detection of Online Drifting Twitter Spam [Invited Paper]

Shigang Liu
School of Information Technology
Deakin University
221 Burwood Hwy, Burwood,
Vic 3125, Australia
shigang@deakin.edu.au

Jun Zhang
School of Information Technology
Deakin University
221 Burwood Hwy, Burwood,
Vic 3125, Australia
jun.zhang@deakin.edu.au

Yang Xiang
School of Information Technology
Deakin University
221 Burwood Hwy, Burwood,
Vic 3125, Australia
yang.xiang@deakin.edu.au

ABSTRACT

Spam has become a critical problem in online social networks. This paper focuses on Twitter spam detection. Recent research works focus on applying machine learning techniques for Twitter spam detection, which make use of the statistical features of tweets. We observe existing machine learning based detection methods suffer from the problem of Twitter spam drift, i.e., the statistical properties of spam tweets vary over time. To avoid this problem, an effective solution is to train one twitter spam classifier every day. However, it faces a challenge of the small number of imbalanced training data because labelling spam samples is time-consuming. This paper proposes a new method to address this challenge. The new method employs two new techniques, fuzzy-based redistribution and asymmetric sampling. We develop a fuzzy-based information decomposition technique to re-distribute the spam class and generate more spam samples. Moreover, an asymmetric sampling technique is proposed to re-balance the sizes of spam samples and non-spam samples in the training data. Finally, we apply the ensemble technique to combine the spam classifiers over two different training sets. A number of experiments are performed on a real-world 10-day ground-truth dataset to evaluate the new method. Experiments results show that the new method can significantly improve the detection performance for drifting Twitter spam.

Keywords

Twitter spam detection; social network security; security data analytics

1. INTRODUCTION

Spam detection is a curious game of cat and mouse, that is, spammers are trying to mask themselves as legitimate users while security companies want to stop spam [1]. Spam has plagued every site. Among these sites, Twitter, which was founded in 2006, is the fastest growing one. Nowadays, over 400 million new tweets are produced over 200 million Twitter users every day [2]. Twitter is used to exchange messages among friends. Unfortunately, spammers usually use Twitter as a tool to post unsolicited messages that contain malicious links, and even hijack trending topics. In this respect, the exponential growth of Twitter contributes to the increase of online spamming activities. Study show that more than 3% messages are most probably abused by spammers [1].

To deal with the increasing threats from spammers, security companies, as well as Twitter itself, are combating spammers to make Twitter a spam-free platform. For example, a spam can be reported by clicking on the 'report as spam' link in their home page on Twitter [3]. Twitter also implements blacklist filtering as a component in their detection system called BotMaker [4]. However, due to time lag, blacklist usually fails to protect victims from new spam [5]. The research shows that more than 90% victims may visit a new spam link before it is blocked by blacklists [2]. In order to address the limitations of blacklists, recently, researchers proposed machine learning based methods that regard spam detection as a binary classification problem [6]. A number of statistical features, such as account age, number of followers or friends and number of characters in a tweet, are extracted to characterise tweets. In conventional supervised detection paradigm, a set of labelled spam and non-spam sample tweets are prepared in advance for training a classification model. Afterwards, the classification model is applied to detect spam in the coming tweets. A number of machine learning methods have been investigated in the topic of Twitter spam detection [7].

However, we observe a critical problem from the real-world Twitter data, named "twitter spam drift" [2], which seriously affects the detection performance of existing machine learning-based methods. The problem is that Twitter spam is drifting over time in the statistical feature space. Thus, the classification model that is trained of using old spam samples cannot accurately recognise the drifted spam tweets. Figure 1 reports the statistics about number of characters in tweets in our experiment dataset. We can see that the number of characters in spam tweets changes quickly in the 10 days. Figure 2 shows the account age of tweets. The account ages of spam tweets have significant change in the 10 days. Although researchers are working to detect spam, spammers are also trying to avoid being detected. For example, spammers could evade current detection features through posting more tweets or even use adversarial machine learning strategy to avoid being detected [8]. To address the problem of twitter spam drift, an effective solution is to train one twit-

ASIA CCS'16, May 30–June 3, 2016, Xi'an, China.

ACM ISBN 978-1-4503-4233-9/16/05. . . $15.00

DOI: http://dx.doi.org/10.1145/2897845.2897928

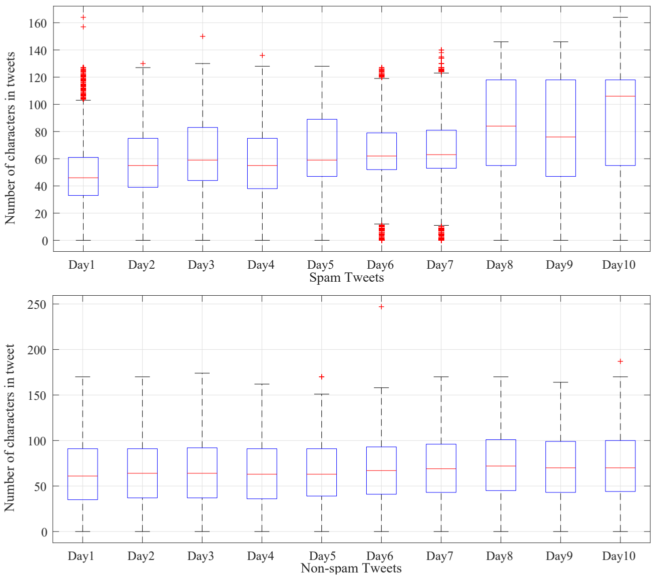

Figure 1: Number of characters in tweets

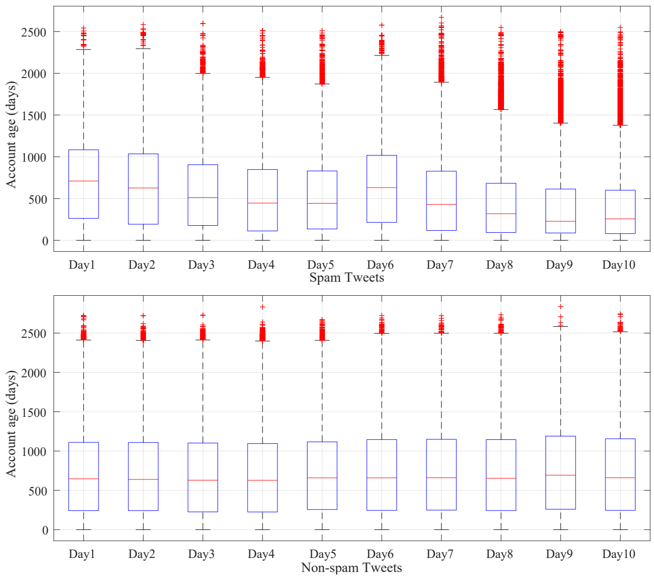

Figure 2: Account age of tweets

ter spam classifier every day. However, it faces a challenge of the small number of imbalanced training data because labelling spam samples is time-consuming. For example, if we manually label 100 tweets, we could obtain 5 spam tweets and 95 non-spam tweets. When a small number of imbalanced training data are used to train a classifier, that will cause the classifier biased toward the non-spam class. The spam detection performance will become poor.

In this work, we treat Twitter spam detection as a specific machine learning problem with a small number of imbalanced training data. The major contributions of our work are summarised as follows.

- We proposes a new detection method to address the problem of twitter spam drift. The new method can learn from a small number of imbalance training data by employing two new techniques, fuzzy-based redistribution and asymmetric sampling.

- We develop a new fuzzy-based re-distribution technique that applies information decomposition to generate more spam samples in line with the spam class distribution.

- We develop a new asymmetric sampling technique to re-balance the sizes of spam samples and non-spam samples in the training data. Finally, the ensemble technique is used to combine the twitter classifiers over two different training sets.

A number of experiments are performed on a real-world 10-day ground-truth dataset to evaluate the new method. Experiments results show that the new method can significantly improve the detection performance for drifting Twitter spam. The rest of this paper is organised as follows.

Section 2 presents a review on recent works of Twitter spam detection. In Section 3, we describe the details of our new spam detection method. The experiments and results are reported in Section 4. Finally, Section 5 concludes this work.

2. RELATED WORK

Recently, many researchers have applied various machine learning techniques for Twitter spam detection [1], [11], [12]. This section gives a short review of related work from the machine learning perspective.

Many works have been carried out to have a better understanding of the nature of Twitter spam. A study based on a data sample back in 2009 [13] suggested that 3.75% of tweets were spam. In 2010, Grier et al. discussed the URLs obtained from tweets' data, and realized that 8% of all crawled unique URLs were spam, which means 2 million URLs out of 25 million were spam [5]. In 2011, Thomas et al. reported that 80 million tweets out of 1.8 billion were spam [14]. Moreover, Najada and Zhu analysed the spam detection problem with spam samples takes 20% of the whole dataset. Chao et al. collected and analysed tweets spam over 600 million tweets with URLs and found that around 1% of URLs are spam [11].

Blacklist is commonly used method for the detection and filtering of spam messages. For example, our industry partner, Trend Micro [15], offers a blacklisting service based on the Web Reputation Technology, which is able to filter harmful spam URLs. Blacklist has a critical disadvantage that it takes considerable time for the new malicious links to be included in a blacklist. In real-world scenarios, many damages should have been caused during the time lag [5].

Heuristic rule based methods are another earlier attempts for filtering Twitter spam to overcome the limitations of blacklist. Yardi et al. [16] introduced a #robotpickupline (hashtag) for spam detection through three rules, which are suspicious URL search, username pattern matching and keyword detection. Kwak et al. [17] recommended that tweets which contain more than three hashtags to be removed in order to eliminate the impact of spam for their research.

It has been reported that the 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 proposed a number of ro-

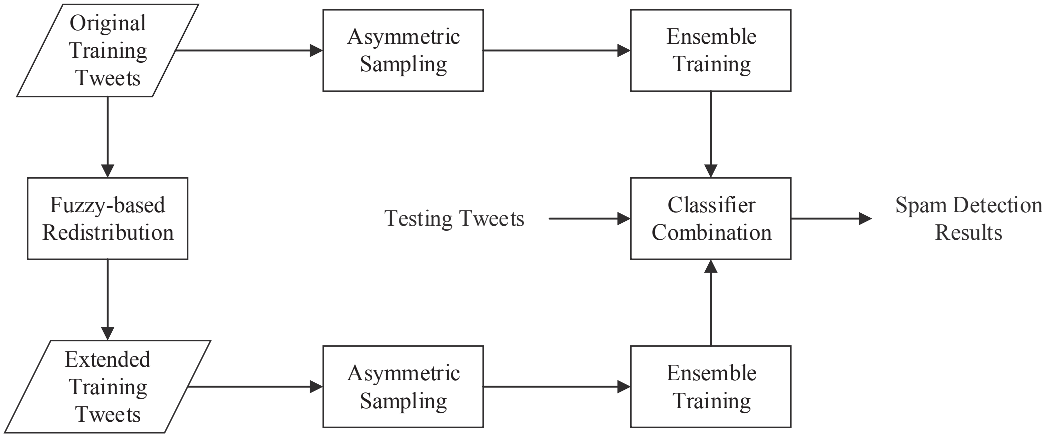

Figure 3: New detection framework

bust features that rely on the social graph to avoid feature fabrication. For example, Song et al. [22] have successfully improved the performance of several classifiers to nearly 99% True Positive and less than 1% False Positive by merging these sophisticated features with the basic feature set. Yang et al. [23] also proposed a few robust spam features, which include Local Clustering Coeffi-cient, Betweenness Centrality and Bidirectional Links Ratio. Their research shows that the new feature set can result in outstanding performance compared with four existing works [12, 18, 19, 1].

Recently, more studies proposed to apply machine learning techniques for Twitter spam detection based on a range of new features, including tweet-based, author-based, and social graph based attributes [1]. Hamzah and Xingquan [24] made use of URL based features such as Domain tokens and path tokens, along with some features from the landing page, DNS information and domain information. Chao et al. [25] collected the spam relevant features such as URL, redirect chain length, Relative number of different initial URLs etc. Wang et al. [2] introduced Bayesian model based approach to detect spammers on Twitter. Benevenuto et al. [12] proposed to detect both spammers and spam using the Support Vector Machine algorithm. Stringhini et al. [18] trained a classifier by using the Random Forest algorithm, which was then used to detect spam in three social networks, including Twitter, Face-book and MySpace. Lee et al. [19] deployed some honeypots to derive the spammers' profiles. They extracted the statistical features for spam detection using several machine learning algorithms, such as Decorate, RandomSubSpace and J48.

In our group's previous work [2, 21], it is observed that Twitter spams are drifting over time in the statistical feature space. The problem is named "twitter spam drift", which seriously affects the detection performance of existing machine learning-based methods. An effective solution for detecting drifted tweet spam is to train one twitter spam classifier every day, while it faces a challenge of the small number of imbalanced training data. In this situation, the classifiers for spam detection are most likely to be overwhelmed by the non-spam class and ignore the spam class. For example, assuming there are only 5% spam class samples and 95% non-spam samples in a given dataset. If a classifier classifies all the samples to the non-spam class, the classification accuracy would be 95%. However, this classifier is not useful in practice, because we are most interest in the spam class. This challenge becomes the motivation of our work.

3. PROPOSED METHOD

This section presents a new detection method that employs a new fuzzy-based redistribution, a new asymmetric sampling and the ensemble technique.

3.1 New Detection Framework

In this paper, we treat the detection of drifted spam tweets as a specific learning problem with a small number of imbalanced training data. The spam class is the minority class and the non-spam class is the majority class. The size of training data including labelled spam and non-spam samples is small for the binary classification task.

Figure 3 shows the new framework for detecting drifted spam tweets. In this framework, a new fuzzy-based distribution technique is applied to extend the original training dataset by creating synthetic spam samples. Then, we conduct asymmetric sampling on the two training datasets. In order to balance the size of spam and non-spam, the new asymmetric sampling technique applies the over-sampling strategy to spam training tweets and the under-sampling strategy to non-spam training tweets. Ensemble training is combined with the asymmetric sampling to construct a set of classifiers from each training dataset. Finally, two sets of classifiers are combined to detect spam from the testing tweets.

3.2 Fuzzy-Based Redistribution

To alleviate the imbalance between spam and non-spam classes in the training data, we develop a new fuzzy-based redistribution algorithm. The fuzzy-based redistribution employs information decomposition, which is a new oversampling technique proposed in our previous work [26] for class imbalance issue, to generate reliable synthetic spam samples. It takes the training spam set, S^+, and the number of synthetic spam samples to be generated, t, as input. As shown in the Algorithm 1, there are three steps, small interval partition, weights calculation and synthetic values generation.

Algorithm 1: Fuzzy-Based Redistribution

```
INPUT: Minority data $S^+$, number of synthetic samples to be generated $t$.
OUTPUT: Re-distributed minority class samples: $FID(S^+, t)$.
/*Initialization*/
for each column feature vector $\mathbf{x}_i$, do
    According to formula (1) and (2), partition the feature vector-based interval into $t$ small intervals.
    Calculate the weights using formula (3) from the observed data to each intervals.
    Calculate $\tilde{m}_{si}$ using formula (4), $\tilde{m}_{si}$ is the $s$th generated value of $\mathbf{x}_i$.
end for
```

Given a set of labelled spam and non-spam tweets, S^+ and S^-. The spam class is denoted as,

$$S^+ = (y_1, \omega^+), (y_2, \omega^+), \cdots, (y_N, \omega^+),$$

where $y_n, n = 1, 2, \cdots, N$, is a tweet sample. Let's denote

$$\mathbf{x}_i = (x_{1i}, x_{2i}, \cdots, x_{Ni})^T, i = 1, 2, \cdots, M$$

as the set of the ith feature value for all tweets, where N means the number of total spam samples, and M means the number of total feature values for each sample. Then we can obtain a value range of the i-th feature,

$$A_i = [a_i, b_i],$$

where

$$a_i = min\{x_{ji} | j = 1, 2, \cdots, m\}$$

and

$$b_i = max\{x_{ji} | j = 1, 2, \cdots, m\}.$$

To generate t synthetic spam samples, we divide the value range $[a_i, b_i]$ into t small intervals. These small intervals can be expressed by

$$A_{si} = [a_i + (s-1) * h_i, a_i + s * h_i), s = 1, 2, \cdots, t-1, \quad (1)$$

$$A_{ti} = [a_i + (t-1) * h_i, a_i + t * h_i], \quad (2)$$

where $h_i = (b_i - a_i)/t$.

The synthetic spam samples are generated according to the N labelled spam samples. The following map is used for calculating the weights from the labelled spam samples to each small interval A_{si}:

$$\mu : \mathbf{x}_i \times \mathbf{u}_i \to [0, 1],$$

$$(x_{ji}, u_{si}) \to \mu(x_{ji}, u_{si}).$$

where $\mathbf{u}_i$ is called the discrete universe set of $\mathbf{x}_i$. We choose a fuzzy membership $\mu(x_i, u_j)$ to perform the mapping.

$$\mu(x_{ji}, u_{si}) = \begin{cases} 1 - \frac{\| x_{ji} - u_{si} \|}{h_i} & if \, \| x_{ji} - u_{si} \| \le h_i \\ 0 & if \, \| x_{ji} - u_{si} \| > h_i \end{cases} \quad (3)$$

where h_i is called step length. The next equation is used to create the sth synthetic value for $\mathbf{x}_i$:

$$\tilde{m}_{si} = \begin{cases} \bar{\mathbf{x}}_i & if \sum_{j=1}^{m} \mu(x_{ji}, u_{si}) = 0 \\ \frac{\sum_{j=1}^{m} m_{jsi}}{\sum_{j=1}^{m} \mu(x_{ji}, u_{si})} & otherwise \end{cases} \quad (4)$$

where $\bar{\mathbf{x}}_i$ is the mean of the observed values of $\mathbf{x}_i$, and m_{jsi} is calculated as:

$$m_{jsi} = \mu(x_{ji}, u_{si}) * x_{ji} \quad (5)$$

Algorithm 2: Asymmetric Sampling

```
INPUT: Minority training dataset $S^+$ ; majority training dataset $S^-$; number of non-spam samples to be removed $l$ ;
OUTPUT: Re-balanced training dataset.
Randomly select $l$ samples from $S^-$, denote the new majority class as $S_n^- = R(S^-, l)$.
$S_n^+$= Bootstrap examples from $S^+$, s.t $|S_n^+| = |S_n^-|$, $S_n^+ = B(S_n^+, |S_n^+| = |S_n^-|)$;
The re-balanced training dataset is: $S_n = S_n^+ \bigcup S_n^-$.
```

Algorithm 1 summarise the spam sample generation process for fuzzy-based distribution. The synthetic spam samples are generated in the way of feature by feature, so they keep very good independence.

3.3 Ensemble with Asymmetric Sampling

In this section, a new asymmetric sampling technique is proposed to create balanced training data for training a single classifier. We apply the under-sampling strategy to the non-spam class, which randomly remove some samples from the non-spam class. We apply the over-sampling strategy to the spam class, which randomly reduplicate the spam samples. This asymmetric sampling technique can effectively combine the advantages of under-sampling and over-sampling. Algorithm 2 describes the details of asymmetric sampling.

Furthermore, we combine asymmetric sampling and bootstrap to implement an ensemble classifier. As shown in Algorithm 3, we first delete l non-spam samples and obtain

$$S\left(FID\right)_n^- = R\left(S^-, l\right).$$

Then, we make use of bootstrap method to extract the same number of spam samples as non-spam samples,

$$S\left(FID\right)_n^+ = B\left(S\left(FID\right)_n^+, \left|S\left(FID\right)_n^+\right| = \left|S\left(FID\right)_n^-\right|\right)$$

where $\left|S\left(FID\right)_n^+\right| = \left|S\left(FID\right)_n^-\right|$ means the number of spam samples equals the number of non-spam samples. Finally, with the ensemble classifiers, we apply the majority voting rule to do the decision making. Its merits lie in neither requiring any complex knowledge nor any priori knowledge [27].

This new detection method uses two different training datasets for ensemble learning.

- One is the original training dataset.
- In the other training dataset, the spam class includes the original spam samples and the synthetics spam samples generated by fuzzy-based redistribution.

The ensemble with asymmetric sampling process is conduced on both of the training datasets. All twitter classifiers from the two training datasets are combined to make the final decision. Our empirical study shows this new method can effectively address the problem of a small number of imbalanced training data.

4. PERFORMANCE EVALUATION

To evaluate the new detection method, we carried out a number of experiments on a real-world twitter dataset. This section reports the experiments and results.

Algorithm 3: New Detection Method

```
TRAINING
1: INPUT: Minority training dataset $S^+$ ; majority training dataset $S^-$; number of non-spam samples to be removed $l$; size of ensemble $N$, C4.5 classifier $I$.
2: OUTPUT: Ensemble classifier$C^*$.
3: $S(FID)_n^+ = FIDoS(S^+, 2 \times |S^+|)$;
4: for $n = 1$ to $N$.
5: $S\left(FID\right)_n^+ = B\left(S\left(FID\right)_n^+, \left|S\left(FID\right)_n^+\right| = \left|S\left(FID\right)_n^-\right|\right)$,
$S\left(FID\right)_n^- = R\left(S^-, l\right)$.
6:$C_n = I\left(S\left(FID\right)_n^+, S\left(FID\right)_n^-\right)$.
7: end for
8: for $n = N + 1$ to $2N$.
9: ${S_n}^+ = B({S_n}^+, |{S_n}^+| = |{S_n}^-|)$,
$S_n^- = R(S^-, l)$.
10:$C_n = I(S_n^+, S_n^-)$.
11: end for
12: $C^* = \{C_n, 1 < n < 2N\}$.
TESTING
1: INPUT:Test data point $z$.
13: OUTPUT: Class prediction for $z$.
14: for $n = 1$to $2N$.
15: calculate $C_n(z)$.
16: end for
17: $C^*(z) = Aggregation\,\{C_n(z), 1 < n < 2N\}$.
```

4.1 Experiment Setup

We first introduce the experiment setup for the empirical study, which includes ground-truth dataset, basic classifiers and performance metrics.

4.1.1 10-day ground-truth dataset

In this work, we used Twitter's Streaming API to collect tweets with RULs in a period of 10 consecutive days [11]. Although it is possible to send spam without embedding URLs on Twitter, the majority of the spam contains RULs [28]. It is worth mentioning that we have inspected thousands of spam tweets by hand and only find a few tweets that without URLs which could be considered as spam. With the help of internal tools provided by Trend Micro [11], we totally labelled over 600 million tweets to create the 10-day ground-truth dataset for the research of spam detection.

Feature extraction is a key component in machine learning based classification tasks [11]. Some studies [1], [12], [18] have applied a few features which make use of historical information of a user, such as tweets that the user sent in a period of time. While these features may be more discriminative, it is not possible to collect them due to the restrictions of Twitter's API. Other researchers [22], [23] applied some social graph based features, which are hard to be evaded. Nevertheless, It is significantly expensive to collect those features, as they cannot be calculated until the social graph is formed. Thus, those expensive features are not suitable for real-time detection, despite that they have more discriminative power in separating spammers and legitimate users. The longer time a spam tweet exists, the more chance it can be exposure to victims. Thus, it is very important to detect spam tweets as early as possible. To reduce the loss caused by spam, real-time detection is in demand. Consequently, we only focus on extracting light-weight features which can be used for timely detection. These features can be straightforwardly extracted from the collected tweets' JSON data structure with little computation. Table 1 summarised the 12 features used in this study.

Table 1: Lightweight Statistical Features

Feature name	Description
account_age	The age (days) of an account since its creation
no_follower	The number of followers of this twitter user
no_following	The number of followings/friends of the user
no_user_favorite	The number of favorites this user received
no_list	The number of lists this twitter user added
no_tweet	The number of tweets this twitter user sent
no_retweet	The number of retweets this tweet
no_hashtag	The number of hashtags included in this tweet
no_user_mention	The number of user mentions included in tweet
no_URL	The number of URLs included in this tweet
no_char	The number of characters in this tweet
no_digits	The number of digits in this tweet

4.1.2 Base Classifiers

In order to examine the effectiveness of the new detection method, a large number of experiments have been conducted using kNN, SVM, (Support Vector Ma-chines), Naive Bayes, LDA (Linear Discriminant Analysis), C4.5 Decision Trees and Random Forest [29], [30]. We found that random forest (RF) and C4.5 achieved outstanding performance compared with other classifiers. Therefore, we only compare our method with RF and C4.5 in this paper.

We reported the average of 10 runs of each experiment in which the datasets are randomly partitioned into the training data and the testing data. In each experiment, the imbalance ratio is fixed to 10 and the original training data contains 1,000 spam tweets and 10,000 non-spam tweets. The whole dataset is divided to two part, one for generating training data and the other for generating testing data. For testing data, we used two settings. In the first case, the rate of spam to non-spam is 10. For example, the testing data have 500 spam samples and 5,000 non-spam samples. In the second case, the rate of spam to non-spam is 100. For example, the testing data includes 100 spam tweets and 100000 non-spam tweets. The different settings can help us simulate different real-world scenarios and evaluate the new method more effectively.

4.1.3 Performance metric

In the experiments, we employed Accuracy (Acc), detection rate (DR) and Area under the ROC curve (AUC) to evaluate the performance of the classifiers. AUC, which is not sensitive to the distribution between the majority and minority classes, can sort models by overall performance. AUC is often used in models assessment. We used the spam class as the positive class and the non-spam class as the negative class. The confusion matrix values are true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The following formulas are used to calculate the metrics.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$DR = \frac{TP}{TP + FN} \tag{7}$$

$$AUC = \frac{1 + \frac{TP}{TP+FN} + \frac{FP}{FP+TN}}{2} \tag{8}$$

Moreover, we used one-factor ANOVA [32] to conduct a qualitative analysis of the new detection method. The statistically significant level is set at $\alpha = 0.05$ for all performance measures. The ANOVA hypothesis is that there is no significantly difference for detection techniques in terms

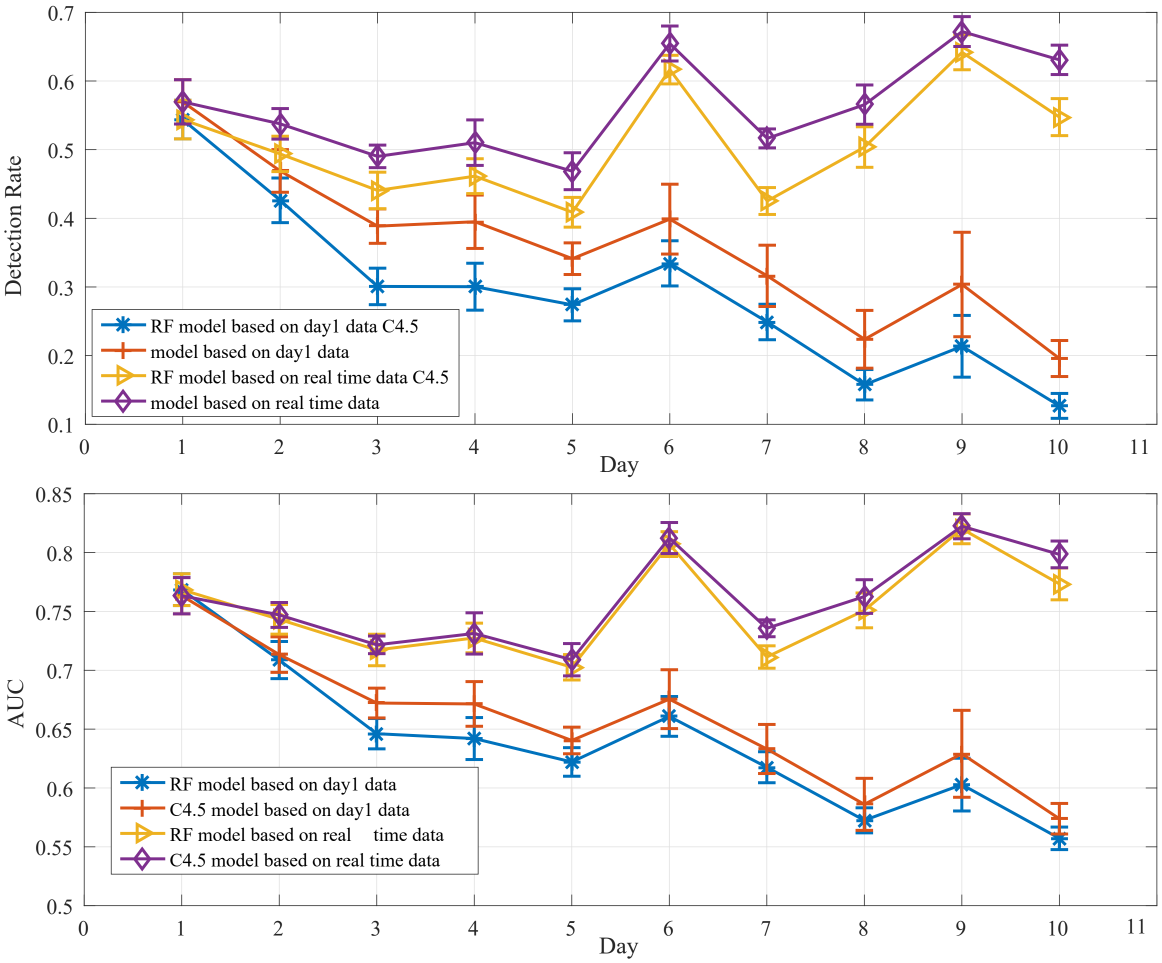

Figure 4: Impact of twitter spam drift

of AUC. The alternative hypothesis is that at least one is significant different. Once the ANOVA results were statistically significant, we performed Tukey's Honestly Significant Difference (HSD) test, which indicated the different levels of the resampling techniques' performance. For Tukey's HSD test, we use letter 'A' for the first class performance, 'B' for the second class performance and 'C' for the third class performance.

4.2 Results and Discussion

We report four sets of experimental results here.

- Section 4.2.1 reports the impact of twitter spam drift.
- Section 4.2.2 reports the overall performance through ANOVA and HSD testing.
- Sections 4.2.3 and 4.2.4 report the day-based detection performance with the different test settings.

4.2.1 *Impact of twitter spam drift*

Figure 4 illustrates the impact of Twitter spam drift problem. Precisely, 'RF model based on day1 data' means we used the tweets' data collected on the 'first day' to train a classification model and made use of it for 10 days Twitter spam detection. 'RF model based on real-time data' means we build a classification models every day. We used a part of the same day's tweets data for classifier training and used the classifiers only on that day to detect Twitter spam. The same operation was for C4.5.

In the figure, we can see that the classifiers created using the same day tweets data exhibit outstanding performance, while the performance of classifiers built of using the 'first day' tweets data decreased dramatically. For example, the detection rate of C4.5 with the first day training data is about 0.57 for the first day testing data. The detection rate decreases to only 0.4 for the 6th day testing data. If we used the 6th day training data, the detection rate achieved to 0.65. The difference is huge. For example, the AUC of RF for first day training and first day testing is about 0.77. If we used the 10th day data for testing, the AUC dramatically reduced to 0.55. However, for the same day training and testing, the AUC of RF is up to 0.82 on the 10th day.

The results show the impact of twitter spam drift to detection is very big. Twitter spam drift can affect the spam detection accuracy and the robustness of the detectors. The results also suggest the potential solution is to train a twitter spam detector for each day.

4.2.2 *Overall detection performance*

Table 2 reports the ANOVA model results for C4.5, RF

Table 2: ANOVA models for AUC

Dataset	DoF	SoS	MS	F-statistic	p-value
Day1	2	0.0442	0.0220	16.0859	<0.0001
Day2	2	0.0282	0.0141	12.2279	<0.0001
Day 3	2	0.0373	0.0187	15.7276	<0.0001
Day 4	2	0.0523	0.0261	14.5999	<0.0001
Day 5	2	0.0347	0.0174	13.0157	0.0001
Day 6	2	0.0126	0.0063	6.2288	0.006
Day 7	2	0.0290	0.0145	9.8467	0.0006
Day 8	2	0.0318	0.0159	12.803	0.0001
Day 9	2	0.0156	0.0078	7.5356	0.0025
Day 10	2	0.0118	0.0059	5.4438	0.0103

Table 3: HSD with AUC statistic

Dataset	C4.5	RF	New
Day1	B	B	A
Day 2	B	B	A
Day 3	B	B	A
Day 4	B	B	A
Day 5	B	B	A
Day 6	AB	B	A
Day 7	B	B	A
Day 8	B	B	A
Day 9	B	B	A
Day 10	AB	B	A

and the new detection method on the 10-day ground-truth dataset. In the experiments, the training data had 1,000 spam tweets and 10,000 non-spam tweets. One can see the technique selection has a significant impact to AUC at the level of $\alpha = 5\%$ because p-value < 0.05 holds for the 10 days data. Then, we used the Tukey's HSD testing to identify which technique shows a significant improvement over the others.

Table 3 reports the Tukey's HSD statistic test results on the datasets across ten days. One can see the new detection method performed significantly better than the other techniques in all cases. Precisely, the new method resulted in 'A' for all the ten small datasets, while both C4.5 and RF didn't obtain 'A' for any cases. Even though C4.5 achieved 'AB' for two cases, for most scenarios it did not exhibit any better performance compared with RF. These results confirm the new detection method is robust and outperforms existing machine-learning based twitter spam detection methods. The reason is the new method can address the problem of a small number of imbalanced training data through the combination of fuzzy-based redistribution and ensemble with asymmetric sampling.

Figure 5 shows the overall performance of C4.5, RF and the new method. Accuracy, detection rate and AUC are averaged over all experiments. In each experiment, the rate of spam testing samples to non-spam testing samples was set to 100. It simulated the realistic twitter spam rate, about 1%. In the figure, we can observe that the three methods have comparable accuracy. The new method results in outstanding performance in terms of detection rate and AUC. The detection rate of our new method is higher than the second best method, C4.5, about 10 percent. RF has the worst detection rate, which is much lower than the new method and C4.5. C4.5 and RF have comparable AUC. The AUC of the new method is higher than other methods over 5 percent.

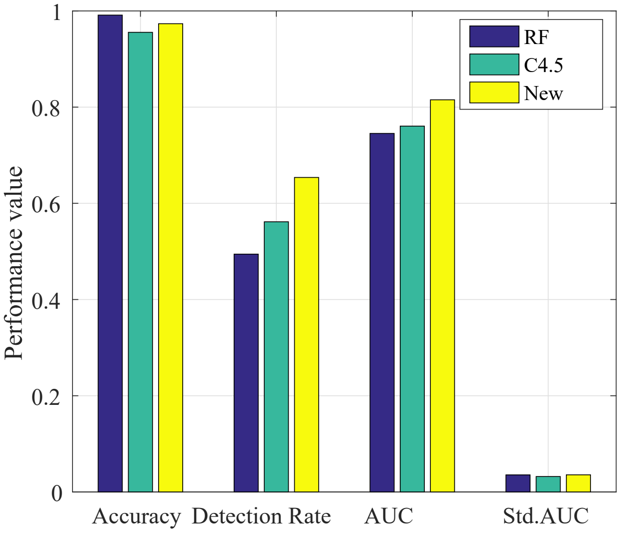

Figure 5: Average performance

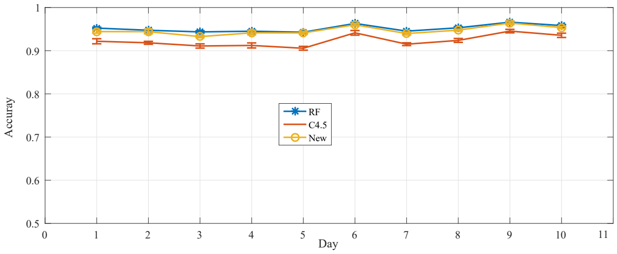

Figure 6: Accuracy for testing set-1

We can make an initial conclusion that the new method can detect more spam tweets accurately.

4.2.3 Day-based performance for testing set-1

In this work, we used two settings for testing data. In the first testing data, we set the imbalanced rate between spam and non-spam samples to 10. It simulated a very high spam rate, about 10%, in some real-world scenarios. In the second testing data, we set the imbalanced rate between spam and non-spam samples to 100. It simulated a very low spam rate, about 1%, in some real-world applications. This section reports the results on the first testing data. The results on the second testing data are reported in Section 4.2.4

Figure 6 shows the accuracy of the three methods. All the accuracy are higher than 0.9. C4.5 has the lowest accuracy, about 0.92. The new method has the similar accuracy with RF, which is higher than C4.5 up to 5 percent. In this case, accuracy is not critical. Even if we classify all testing samples to the non-spam class, the accuracy is about 0.91. However, the classifier misclassified a large portion of the spam samples as non-spam.

Figure 7 reports the AUC and detection rate of the three methods. In general, the new method shows the best AUC and the best detection rate across the 10 days. For AUC, C4.5 and RF display comparable performance. The AUC of the new method is higher than other two methods up

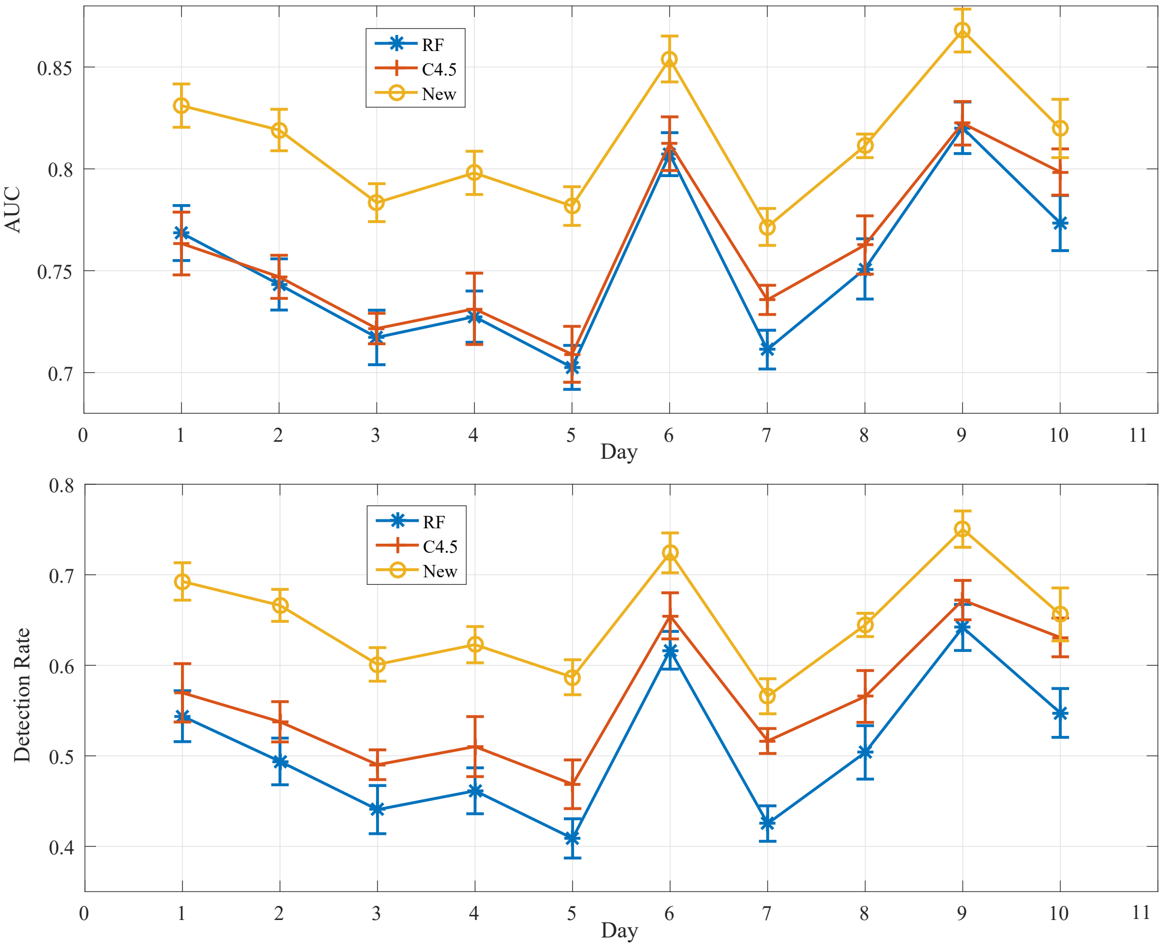

Figure 7: AUC and detection rate for testing set-1

to 8 percent. For example, on day 1, the AUC of the new method is around 0.83, while the AUC of C4.5 and RF is approximately 0.77. On day 6, all methods have very good AUC. The AUC of the new method outperforms that of C4.5 and RF about 4 percent. The worst improvement occurred on day 10. The new method has higher AUC than the second best method, C4.5, less than 3 percent.

For detection rate, C4.5 has better performance than RF across all 10 days. The detection rate of the new method is higher than RF up to 20 percent. For example, on day 2, the AUC of the new method is about 0.68, while the AUC of C4.5 is less than 0.55. RF has the worst detection rate that is 0.5. On day 9, the detection rate of the new method is over 0.75, while C4.5's detection rate is about 6.8 and RF's detection rate is 6.5. On day 10, C4.5 has comparable detection rate with the new method. The detection rate of RF is lower than other methods about 5 percent.

4.2.4 *Day-based performance for testing set-2*

Figures 8 and 9 report the spam detection results on the second testing data. In this testing data, the imbalanced rate between spam and non-spam samples was 100. It simulated a very low spam rate, less than 1%. The results have some difference to that reported in Section 4.2.3.

Figure 8 shows very high accuracy of the three methods. RF has the best accuracy, which is close to 0.99. The difference between RF and the new method is about 2 percent. The new method is better than C4.5 in terms of accuracy. As we mentioned before, accuracy is not critical for the experiments of tweet spam detection. In this case, even if we classify all testing samples to the non-spam class, the accuracy is about 0.99. Accuracy is used here to confirm the classification method is correctly implemented. We need to pay more attention to the amount of correctly detected spam tweets.

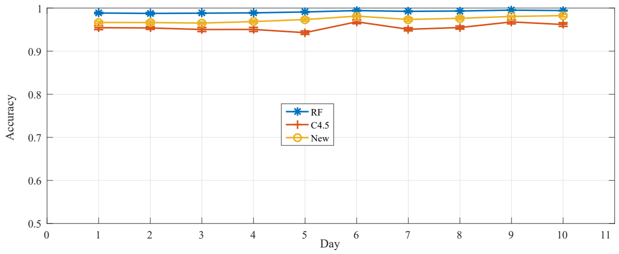

Figure 8: Accuracy for testing set-2

Figure 9 reports the AUC and detection rate of the three methods, RF, C4.5 and the new method. In line with the

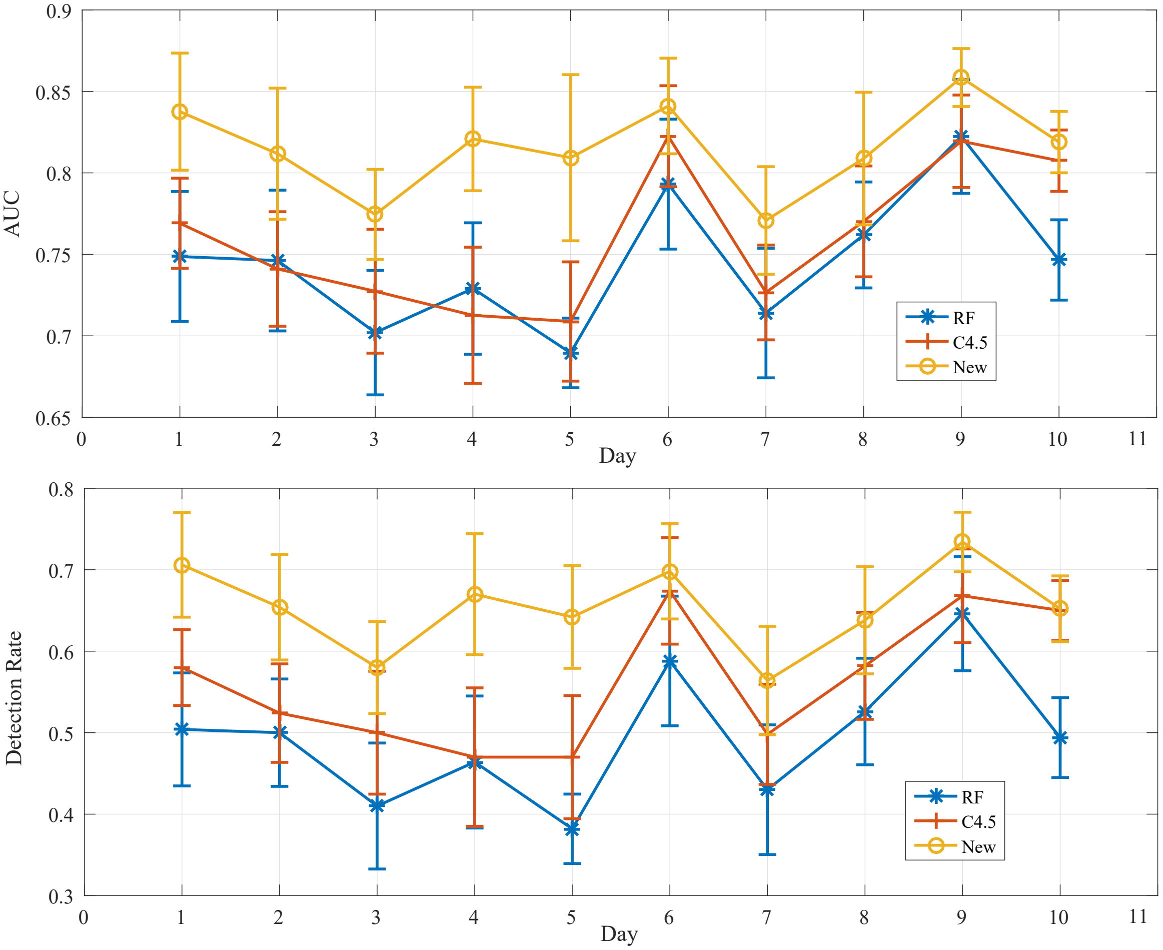

Figure 9: AUC and detection rate for testing set-2

results on the first testing data, the new method had the best AUC and the best detection rate across the 10 days. C4.5 does not always have better AUC than RF. For example, on day 1, the average AUC of the new method is around 0.84. C4.5's AUC is 0.76, which is higher than RF about 2 percent. On day 4, the second best method is RF, which has higher AUC than C4.5 about 2 percent. The AUC of the new method outperforms RF about over 10 percent on this day. The C4.5's AUC is comparable to the AUC of the new method on day 10. They are higher than RF over 5 percent.

For detection rate, the new method is the best one. RF is the worst method and C4.5 is in the middle. The detection rate of the new method is dramatically higher than other methods in most cases. For example, on day 1, the AUC of the new method achieved 0.7, while the AUC of C4.5 is less than 0.6. RF has the worst detection rate,which is 0.5. On day 4, the detection rate of the new method is about 0.67, while the detection rates of C4.5 and RF are less than 0.5. On day 10, C4.5 has the same detection rate with the new method. The detection rate of RF is lower than other methods about 15 percent.

We can see the results on two different testing dataset are consistent. The new method displays excellent robustness and outperforms c4.5 and RF significantly in any case.

5. CONCLUSIONS

In this paper, we addressed the critical challenge of Twitter spam drift. We treated it as a special machine learning problem with a small number of imbalance data. We proposed a new method combining two new techniques, fuzzy-based redistribution and asymmetric sampling, to solve this problem. The fuzzy-based redistribution technique applied information decomposition technique generate more sythetic spam samples. The asymmetric sampling technique performed over-sampling on spam samples and under-sampling on non-spam samples to balance the sizes in the training data. The ensemble technique was used to combine the spam classifiers over two different training sets in order to improve the robustness and accuracy of spam detection. To evaluate the new method, we carried out a number of experiments on a real-world 10-day ground-truth dataset. The new method was compared to other two methods, C4.5 and RF. Experiments results showed that the new method can significantly improve the detection performance for drifting Twitter spam. AUC of spam detection can be improved up to 10 percent. Detection rate can be improved over 20 percent.

6. REFERENCES

[1] Alex Hai Wang. Don't follow me: Spam detection in twitter. In

Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on, pages 1–10. IEEE, 2010.

[2] Chao Chen, Jun Zhang, Yang Xiang, and Wanlei Zhou. Asymmetric self-learning for tackling twitter spam drift. In *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, pages 208–213. IEEE, 2015.

[3] Michael Mccord and M Chuah. Spam detection on twitter using traditional classifiers. In *Autonomic and trusted computing*, pages 175–186. Springer, 2011.

[4] Reza Bosagh Zadeh. Twitter engineering blog: All-pairs similarity via dimsum. *Twitter Engineering Blog*, 2014.

[5] Chris Grier, Kurt Thomas, Vern Paxson, and Michael Zhang. @ spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 27–37. ACM, 2010.

[6] Kurt Thomas, Chris Grier, Justin Ma, Vern Paxson, and Dawn Song. Design and evaluation of a real-time url spam filtering service. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 447–462. IEEE, 2011.

[7] Xianchao Zhang, Shaoping Zhu, and Wenxin Liang. Detecting spam and promoting campaigns in the twitter social network. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1194–1199. IEEE, 2012.

[8] Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, Deepak Verma, et al. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108. ACM, 2004.

[9] Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12, 2010.

[10] Hossam Faris, Khalid Jaradat, Malek Al-Zewairi, Omar Adwan, et al. Improving knowledge based spam detection methods: The effect of malicious related features in imbalance data distribution. *International Journal of Communications, Network and System Sciences*, 8(5):118–129, 2015.

[11] Chao Chen, Jun Zhang, Xiao Chen, Yang Xiang, and Wanlei Zhou. 6 million spam tweets: A large ground truth for timely twitter spam detection. In *Communications (ICC), 2015 IEEE International Conference on*, pages 7065–7070. IEEE, 2015.

[12] Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12, 2010.

[13] Pear Analytics. Twitter study–august 2009. *San Antonio, TX: Pear Analytics. Available at: www. pearanalytics. com/blog/wp-content/uploads/2010/05/Twitter-Study-August-2009. pdf*, 2009.

[14] Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. Suspended accounts in retrospect: an analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 243–258. ACM, 2011.

[15] Jonathan Oliver, Paul Pajares, Christopher Ke, Chao Chen, and Yang Xiang. An in-depth analysis of abuse on twitter. *Trend Micro*, 225, 2014.

[16] Sarita Yardi, Daniel Romero, Grant Schoenebeck, et al. Detecting spam in a twitter network. *First Monday*, 15(1), 2009.

[17] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[18] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 1–9. ACM, 2010.

[19] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering social spammers: social honeypots+ machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442. ACM, 2010.

[20] Hamzah Al Najada and Xingquan Zhu. isrd: Spam review detection with imbalanced data distributions. In *Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference on*, pages 553–560. IEEE, 2014.

[21] Chao Chen, Jun Zhang, Yi Xie, Yang Xiang, Wanlei Zhou, et al. A performance evaluation of machine learning-based streaming spam tweets detection. *IEEE Transactions on Computational Social Systems*, 2(3):65–76, 2015.

[22] Jonghyuk Song, Sangho Lee, and Jong Kim. Spam filtering in twitter using sender-receiver relationship. In *Recent Advances in Intrusion Detection*, pages 301–317. Springer, 2011.

[23] Chao Yang, Robert Harkreader, and Guofei Gu. Empirical evaluation and new design for fighting evolving twitter spammers. *Information Forensics and Security, IEEE Transactions on*, 8(8):1280–1293, 2013.

[24] Kurt Thomas, Chris Grier, Justin Ma, Vern Paxson, and Dawn Song. Design and evaluation of a real-time url spam filtering service. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 447–462. IEEE, 2011.

[25] Sangho Lee and Jong Kim. Warningbird: A near real-time detection system for suspicious urls in twitter stream. *Dependable and Secure Computing, IEEE Transactions on*, 10(3):183–195, 2013.

[26] Shigang Liu, Jun Zhang, Yu Wang, and Yang Xiang. Fuzzy-based feature and instance recovery. In *Springer's LNAI Proceedings*. Springer, 2016.

[27] Haibo He and Yunqian Ma. *Imbalanced learning: foundations, algorithms, and applications*. John Wiley & Sons, 2013.

[28] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Compa: Detecting compromised accounts on social networks. In *NDSS*, 2013.

[29] Michael Mccord and M Chuah. Spam detection on twitter using traditional classifiers. In *Autonomic and trusted computing*, pages 175–186. Springer, 2011.

[30] R Kishore Kumar, G Poonkuzhali, and P Sudhakar. Comparative study on email spam classifier using data mining techniques. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, pages 14–16, 2012.

[31] Mohamed Bekkar, H Kheliouane Djemaa, and T Akrouf Alitouche. Evaluation measures for models assessment over imbalanced data sets. *Journal of Information Engineering and Applications*, 3(10):27–38, 2013.

[32] Chris Seiffert, Taghi M Khoshgoftaar, and Jason Van Hulse. Improving software-quality predictions with data sampling and boosting. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 39(6):1283–1294, 2009.