# LSSL-SSD: Social Spammer Detection with Laplacian Score and Semi-supervised Learning

Wentao Li[1], Min Gao[2], Wenge Rong[3], Junhao Wen[2], Qingyu Xiong[2], and Bin Ling[4]

[1] The Centre for QCIS, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia
[2] School of Software Engineering, Chongqing University, Chongqing, China
gaomin@cqu.edu.cn
[3] School of Computer Science and Engineering, Beihang University, Beijing, China
[4] School of Engineering, University of Portsmouth, Portsmouth, England

**Abstract.** The rapid development of social networks makes it easy for people to communicate online. However, social networks always suffer from social spammers due to their openness. Spammers deliver information for economic purposes, and they pose threats to the security of social networks. To maintain the long-term running of online social networks, many detection methods are proposed. But current methods normally use high dimension features with supervised learning algorithms to find spammers, resulting in low detection performance. To solve this problem, in this paper, we first apply the Laplacian score method, which is an unsupervised feature selection method, to obtain useful features. Based on the selected features, the semi-supervised ensemble learning is then used to train the detection model. Experimental results on the Twitter dataset show the efficiency of our approach after feature selection. Moreover, the proposed method remains high detection performance in the face of limited labeled data.

**Keywords:** Social networks, Spammer detection, Laplacian score, Feature selection, Semi-supervised learning

## 1 Introduction

With the development of Web 2.0, online social networks have gained increasing attention [1]. As an open platform, social networks enable people to maintain social relationships and find common interests with each other online [2]. Though social networks bring great convenience to people, their open characteristics make them vulnerable to attacks issued by social spammers [3, 4].

Social spammers refer to those people who inject false information (i.e. advertisements, pornography) into social networks for economic purposes [5]. Because social relationships normally represent certain kind of trust, social spammers pose more threats to social networks than other types of spammers [6] For example, it indicates that advertising links in Twitter clicked by more than twice people than those in e-mails [7].

Social spammers bring economic losses to normal people and hinder the long-term development of social networks [8]. To alleviate the effect of social spammers, many notable works have been done. The purpose of these detection methods is to distinguish spammers from normal users [9]. According to the amount of needed labeled data, these methods can be classified into three categories: supervised methods that train a classifier based on features derived from relationship or content information [10], unsupervised methods that cluster users into different groups [11], and semi-supervised learning methods based on the label propagation process [12].

Among these methods, supervised methods need a large number of labeled data, which become impractical in the real-world situation because of the high cost of labeling. Unsupervised methods have low detection accuracy due to the lack of labels. In addition, they are susceptible to the interference of noise data. Existing semi-supervised methods make use of the random walk process to obtain users' credibility, but this process brings high time cost. Moreover, all these methods train models based on high dimension features because of the large scale of social networks, which reduces the detection performances.

To solve these problems, in this paper, we combine the unsupervised feature selection method and the semi-supervised ensemble learning method to get our detection method, which is called LSSL-SSD. More specifically, we first select features through their ability to maintain the local geometrical information in the original data space, or Laplacian score, without the use of label information. After selecting useful features, a semi-supervised random forest approach is used to train the detection model to make use of both labeled and unlabeled data.

Note that the feature selection process in LSSL-SSD is an unsupervised one, so it can be combined with the process of semi-supervised classification. Experiments on the Twitter data set show that the proposed method outperforms state-of-the-art methods in term of detection rates when the amount labeled data is limited. Moreover, the operation of feature extraction reduces the dimension of features, resulting in better generalization ability for spammer detection.

The next of this paper is organized as follows. In section 2, we introduce some related work about social spammer detection and feature extraction. The description of our proposed method is shown in Section 3. Section 4 introduces experimental results and discussion. Finally is the conclusion and future work.

## 2 Related Work

In this section, we first introduce current research about social spammer detection. Then, background knowledge about feature selection is described.

### 2.1 Social spammer detection methods

Due to the open nature of social networks, social spammers are able to inject false information and spread them through social networks [4]. To alleviative the harness brought by social spammers, increasing attentions have been paid to

detect them [6]. According to the way of training detection models, three kinds of methods can be summarized.

**Supervised Detection Methods** These methods mainly find features to distinguish spammers from normal users, then train classifiers based on these features. For example, Aggarwal etc. [10] detected spammers by using features from a user's registered information or content information. Lee etc. [13] did that by extracting features from behavior information crawled by honeypots. In [14], the authors proposed detection method based on social network structures. In [15], content information and structure information are used together to train models. Supervised methods performance well in detecting social spammers, but the need of labeled data makes them hard to work well in the real-world situation due to the high cost of labeling data.

**Unsupervised Detection Methods** Supervised methods need a large number of labeled samples, so some researchers put forward unsupervised detection methods. These methods mainly find spammers by using social network topology. For example, in [11], similarities of text content and URLs are used to cluster users into different groups. The intuition behind this method is that spammers have fewer similarities with normal users in terms of content information. By contrast, Tan [16] first located normal users by social relationship graph, then detect spammers through relationships between different users.

**Semi-supervised Detection Methods** Compared with supervised detection methods, unsupervised detection methods do not require manually labeled data. But the false positive rates of them are high due to the lack of labels, and their robustness is low when noisy data exist. In order to solve these problems, Li [12] proposed a semi-supervised detection framework based on trust propagation, which uses PageRank to propagate labels to find spammers. This method works well in practice, but the process of trust propagation needs high time cost.

To summarize, all these methods rely on features extracted from user behavior or relationship information, resulting in high-dimension of features. Among them, semi-supervised methods are suitable for the real-world application while the time cost is high, too. To improve the detection effectiveness and accuracy, unsupervised feature selection method is used before a novel semi-supervised learning method to form our proposed detection method.

### 2.2 Feature Selection Methods

In social networks, the size of networks is so large, which makes features derived from them so high [4]. The high dimension of features reduces the detection performances of current methods. Feature selection methods are often used to remove useless features, thus improving the detection performance [17].

There are two types of feature selection methods, i.e. wrapper and filter methods. Wrapper methods are used with a particular learning algorithm, so

these methods are limited to some specific learning tasks. Filtering methods make use of the intrinsic characteristics of data to evaluate features. Filtering methods generally require the relationship between features and labels, such as the use of Pearson correlation similarity or fisher score for feature selecting [18].

Here, we use feature selection rather than feature transformation because the latter will change the original feature space, thus reduce the diversity of features. Moreover, while little labeled data can be obtained, unsupervised feature selection method can provide suitable inputs for the next classification process.

## 3 Proposed Method

In this section, we introduce the proposed detection method (LSSL-SSD). We treat the problem of social spammer detection as a classification, that is to make a difference between normal users and spammers. Before the specific algorithm is given, the overall process of LSSL-SSD is explained in Fig. 1.

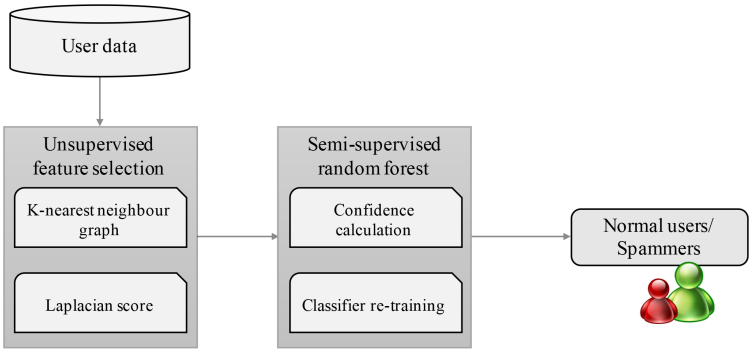


**Fig. 1.** The framework of LSSL-SSD

The whole process includes two main modules. The first one is to select features. Here we use unsupervised feature selection method, the reason is that we want to pick up useful features when the labels are lacking. The second one is to use the semi-supervised random forest learning based on the selected features to train a detection model. We will introduce the two modules in detail below.

### 3.1 Unsupervised Feature Selection Based on Laplacian Score

In social networks, each user has relationship information and content information. The large scale of social networks leads to a high dimension representation of users. Training a model on these high dimension data will result in low detection results. Therefore, a feature selection method is useful for effectively spammer detection.

At the same time, getting lots of labeled data needs high cost while traditional unsupervised and semi-supervised feature selection methods fail to remove redundant features for general tasks [4]. In order to get a better feature selection effect in the absence of labeled data, we apply the Laplacian score method that was proposed by Hu [17] for feature selection. This method is an unsupervised one, but it can achieve a fantastic effect as supervised ones.

The basic idea of the Laplacian score method is to use the feature's ability to maintain neighbor information as a selection standard. The intuition behind is that the discriminant effect of features represents in their local geometric relationships in the original data space. The key steps of this method include two steps, that is, the construction of k-nearest neighbor graph and the Laplacian score calculation.

**Construction of the *K* Nearest Neighbor Graph** To construct a neighborhood graph $G$, it needs to calculate the similarity between user data in the original data space. Here we use Euclidean similarity as the basic measure. Assuming there are $M$ users in social networks, denoted by $(x_1, x_2, \cdots, x_i, \cdots, x_M)$, where $x_i$ represents user $i$'s feature vector. For user $x_i$, we find its $k$ nearest neighbor set denoted by $n(x_i)$. Then, we add edge between $x_i$ and users in $n(x_i)$. Note that the edge is directed, that is to say, the neighbor relationship is not symmetrical.

**Laplacian Score Calculation** When the neighbor graph $G$ is obtained, the Laplacian score of each feature can be calculated. Firstly, a weighted matrix $S$ is constructed based on the graph $G$. $S$ quantifies the local geometric relationships in the original data space and it is calculated according to Formula 1.

$$S_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}}, if\ e(x_i, x_j) \in G \\ 0, otherwise \end{cases} \tag{1}$$

Here, $t$ is the adaptive constant. When $x_i$ and $x_j$ exist an edge, then $S_{ij}$ is obtained by the similarity between $x_i$ and $x_j$, otherwise the value is 0. $S$ is a weighted graph, the Laplacian matrix of $S$ is $L$=$D$-$S$, where $D = diag(S)$ is the main diagonal matrix of $S$.

Then, Assuming there are $R$ features in total, for the $r$-th feature, values of $M$ users on this features form a vector $f_r = [x_{1r}, x_{2r}, \cdots, x_{Mr}]$, and the Laplacian score of feature $r$ can be calculated by Formula 2.

$$L_r = \frac{\widetilde{f_r^T} L \widetilde{f_r}}{\widetilde{f_r^T} D \widetilde{f_r}} \tag{2}$$

Where $L_r$ is the Laplacian score of feature $r$, $L$ is the Laplacian matrix. The calculation of $F_r^T$ is shown in Formula 3.

$$\widetilde{f_r^T} = f_r - \frac{f_r^T D \mathbf{1}}{\mathbf{1}^T D \mathbf{1}} \mathbf{1} \tag{3}$$

Finally, When $L_r$ is calculated, the $R$ features can be sorted according to their scores, and the features with high scores are selected. $L_r$ does not use the label information but it has a good effect on the slection of useful features. Details of this method can be found in [17].

### 3.2 Semi-supervised Random Forest

After the feature selection process, the next step is to train a detector based on the selected features. To solve the problem of insufficient labeled data, an intuitive way is to make use of both labeled and unlabeled data. In this paper, we apply the semi-supervised random forest method. This method integrates ensemble learning and co-training to get the final detection model.

The semi-supervised random forest method [19] first learns multiple basic classifiers on labeled data, and then unlabeled data are used to improve the performance of classifiers at each iteration. The whole process consists of two parts, namely, the confidence calculation process and the classifier training process.

**Confidence Calculation** The whole training set in the system can be divided into labeled data set $L$ and unlabeled data set $U$. $L$ includes $|L|$ labeled data, denoted by $(x_1, y_1), (x_2, y_2), \cdots, (x_{|L|}, y_{|L|})$, where $x_i$ represents user $i$'s feature vector, $y_i$ represents user $i$' label. $U$ includes $|U|$ unlabeled data, denoted by $x_1, x_2, \cdots, x_{|U|}$, the datum in this set is unlabeled.

The semi-supervised random forest method first uses resampling technology to get $N$ data subsets from labeled data, which can be denoted by $L_i$, $(i = 1, 2, \cdots, N)$. And then $N$ decision tree classifiers $f_i$ can be trained on each subset. In order to make use of unlabeled data, the confidence of each unlabeled datum in $U_i$ is calculated. For each base classifier, the unlabeled data with high confidence will then be moved into the corresponding labeled data subset $L_i$.

For a base classifier $f_i$, to get the confidence of unlabeled data, we use the prediction results of $N$-1 classifiers except $f_i$. If there are two groups of users, i.e. normal users whose labels are -1 and spammers whose labels are 1, then the confidence of each datum $x_i$ can be calculated by formula 4.

$$con(x_i) = \max(\sum\nolimits_{f(x_i)=1} 1, \sum\nolimits_{f(x_i)=-1} 1) \tag{4}$$

The first term $(f_i(x) = 1)$ means how many base classifiers predict $x_i$ as spammers, the second term means how many base classifiers predict $x_i$ as normal users. $con(x_i)$ reflects the consistency of classifiers to predict $x_i$. The data with top confidence values will be selected to moved from $U_i$ into $L_i$, $(i = 1, 2, \cdots, N)$.

**Re-training of Classifier** When new data are added into labeled data set, $N$ classifiers will be re-trained on the augmented labeled data. The process will continue until the output of $N$ classifiers remains the same. When the process is over, $N$ classifiers are obtained, and the label of new user data $x$ is determined by voting, as shown in Formula 5.

$$f(x) = \begin{cases} 1 \; if \; \sum_{f_i(x)=1} 1 > \sum_{f_i(x)=-1} 1 \\ -1 \; if \; \sum_{f_i(x)=1} 1 < \sum_{f_i(x)=-1} 1 \end{cases} \tag{5}$$

The label with most votes will be assigned to $x$, $i \in (1, 2, \cdots, N)$. If the votes are equal, the label can be assigned randomly. After the two stages, LSSL-SSD can be obtained. The process of the LSSL-SSD algorithm is shown in Table 1.

**Table 1.** The process of LSSL-SSD algorithm

| |
|---|
| **Input:** |
| Labeled data $L$ |
| Unlabeled data $U$ |
| $M$ users, each represented by a $R$-dimension vector, $x_i = (x_{i1}, x_{i2}, \cdots, x_{iR})$ |
| Number of nearest neighbor $k$ |
| Number of selected features $t$ |
| Number of base classifiers $N$ |
| **Output:** |
| $N$ classifiers $F = [f_1, f_2, \cdots, f_N]$ |
| **Steps:** |
| 1. Get the $k$ nearest neighbor graph $G$ of $M$ users according to Formula 1. |
| 2. Get the Laplacian scores of $R$ features according to Formula 2. |
| 3. Choose features with the top-$t$ largest Laplacian scores. |
| 4. Re-sample the original labeled data to get $N$ subsets. |
| 5. Train $N$ initial base classifiers $f_i$ based on each subset $L_i$, $i = 1, 2, \cdots, N$. |
| 6. iterate until the output of $N$ base classifiers remain the same. |
| 6.1 For each base classifier $f_i$. |
| 6.2 Calculate the confidence of each unlabeled datum according to Formula 4. |
| 6.3 Choose data with top confidence values and add them to $L_i$. |
| 6.4 Re-train $f_i$ using updated $L_i$. |
| 6. Output $N$ classifiers, and predict each new-coming datum according to Formula 5. |

## 4 Experiment Results

To analyze the performance of LSSL-SSD, in this section we conduct three groups of experiments on a real-world data set. The first is to compare LSSL-SSD with related methods. The second is to check the detection effect of the feature selection process. The third is about parametric sensitivity analysis.

### 4.1 Experiment Setup

**Data Set** In this paper, we use the Twitter data set provided by Benevenuto [20]. This data set is collected since August 2009, which includes eight million users. 1065 users in this data set have been labeled, including 710 normal users

and 355 spammers. Each user has 62 features, which are derived from behavior and content information. More details about this data set can be found in [20].

**Evaluation Metrics** To evaluate the accuracy of the proposed method, Precision, Recall and F1-measure are used as evaluation metrics. We denote $N_a$ as the number of spammers who are correctly detected, $N$ as the number of spammers predicted by the algorithm, and $N_t$ as the number of spammers in the systems. Precision, Recall and F1-measure are calculated by Formulas 6, 7 and 8.

$$Precision = \frac{N_a}{N} \tag{6}$$

$$Recall = \frac{N_a}{N_t} \tag{7}$$

$$F1 - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{8}$$

$Precision$ is the ratio between the number of correctly predicted spammers and those who are predicted as spammers. $Recall$ is the ratio between the number of correctly predicted spammers and the total number of spammers. $F1 - measure$ is the weighted average of $precision$ and $recall$. The range of both three metrics is 0 to 1 and the best value is 1 while the worst one is 0.

**Experimental Settings** LSSL-SSD is a semi-supervised one. We hope our method outperforms supervised methods in terms of detection accuracy. Here we use a common testing set for fair comparison. We fix the size of this public testing set to 20% of all the data. For supervised algorithms, the remaining 80% data is used for training. For the semi-supervised algorithm, we divided the training set into the labeled set $L$ and unlabeled set $U$. The experiment was conducted 100 times, and the average results are used to report results.

### 4.2 Experimental Results and Discussion

**Comparison of Detection Performances Between LSSL-SSD and Supervised Methods** To show that the proposed method has better detection performance, we compare LSSL-SSD with traditional methods. Here naive Bayes, decision tree, logistic regression, support vector machine (SVM) and random forest are used to compare. We change the size of labeled training data from 10%, 20% to 40%. Here 10% means 10% of the original data is labeled training data. Comparison of these methods is shown in Fig. 2.

From Fig. 2, it can be found that the F1-measure values of LSSL-SSD are higher than those of all supervised algorithms. In addition, with the increase of labeled training data size, F1-measure of LSSL-SSD becomes better. Even when only 10% labeled data is obtained, LSSL-SSD outperforms other methods.

In terms of precision, the supervised random forest has a better performance. When the amount of labeled data is small, LSSL-SSD does not perform well, but

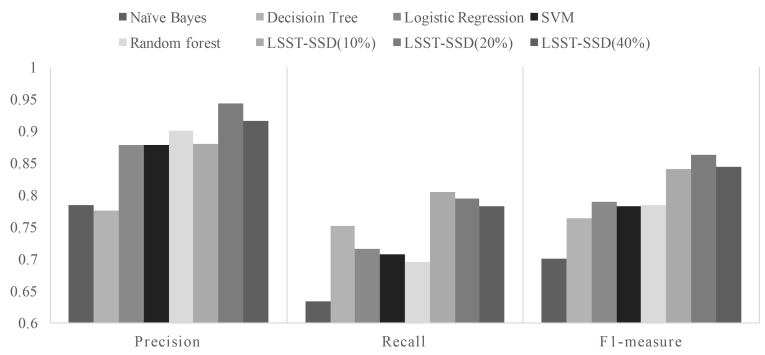


**Fig. 2.** Comparison of detection performance between LSSL-SSD and others

with the increase of labeled data size, prediction rates become good. In terms of recall, the decision tree has a better performance. But recall rates of LSSL-SSD is better than those of other algorithms.

From the above results, it can be concluded that LSSL-SSD has better detection performances than traditional methods. Moreover, only a small fraction of labeled data is used to get our model, which means that the label cost can be reduced. This justifies the real-world value of our proposed method.

**Performance of LSSL-SSD with Different Number of Features** The first step of LSSL-SSD is to use the Laplacian score method to select features. To verify the effectiveness of this process, we change the number of selected features. We change the number of features from 10 to all (62). Supervised random forest is used as a comparison. Here 10% labeled training data is used and the number of base classifiers changes from 3 to 100. the results are shown in Fig. 3.

As seen from Fig. 3, LSSL-SSD achieves better than the supervised random forest in any setting. Also, we can observe that with the increase of the number of selected features, F1-measure of LSSL-SSD first increases then decreases. When the number is 20, the F1-measure is the best. This result shows the importance of feature selection process because many features are redundant. In practice, we can use cross-validation to find the suitable number of features.

**Performance of LSSL-SSD with Different Number of base Classifiers** Since LSSL-SSD uses decision tree as the base classifier, the number of base classifiers may have an impact on the performance. So we discuss the effect of the number of base classifiers on detection accuracy. Here we change the number from 3 to 100. We change the labeled data size of LSSL-SSD from 10%, 20% to 40% of the original data to report the results, as shown in Tables 2-4 respectively.

From Table 2 it can be found that when the size of labeled data is 10%, with the increase of the number of base classifiers, the precision rates increase gradually, recall rates remain the same and F1-measure values show some fluctuates.


**Fig. 3.** Comparison of detection performance between different number of features

**Table 2.** Performance of LSSL-SSD with different number of base classifiers (10%)

| | 3 | 5 | 10 | 15 | 20 | 25 | 50 | 100 |
|---|---|---|---|---|---|---|---|---|
| Precision | **0.9142** | 0.86884 | 0.8684 | 0.8571 | 0.88 | 0.8552 | 0.88 | 0.88 |
| Recall | 0.7804 | **0.8048** | **0.8048** | **0.8048** | **0.8048** | 0.7926 | **0.8048** | **0.8048** |
| F1-measure | **0.8421** | 0.8354 | 0.8385 | 0.8301 | 0.8407 | 0.8227 | 0.8407 | 0.8407 |

Results in Tables 3-4 show some similarities. The results mean that the number of base classifiers has little impact on the detection performance. As long as the number is in a certain range, detection results are acceptable. The possible reason is that LSSL-SSD needs the diversity of base classifiers, so when the number is larger than a certain value, detection performance remains the same.

**Table 3.** Performance of LSSL-SSD with different number of base classifiers (20%)

| | 3 | 5 | 10 | 15 | 20 | 25 | 50 | 100 |
|---|---|---|---|---|---|---|---|---|
| Precision | 0.8533 | 0.8684 | 0.8783 | **0.9027** | 0.8904 | 0.8767 | 0.8918 | 0.88 |
| recall | 0.771 | **0.7951** | 0.7831 | 0.7831 | 0.7831 | 0.771 | **0.7951** | **0.7951** |
| F1-measure | 0.8101 | 0.8301 | 0.828 | 0.8387 | 0.8333 | 0.8205 | **0.8407** | 0.8354 |

In conclusion, it can be found that LSSL-SSD has a good detection performance in detecting social spammers. Moreover, As a common method, LSSL-SSD works well for other social networks such as Facebook. This ensures the long-term running of social networks, which is of great significance in the real-world application.

**Table 4.** Performance of LSSL-SSD with different number of base classifiers (40%)

| | 3 | 5 | 10 | 15 | 20 | 25 | 50 | 100 |
|---|---|---|---|---|---|---|---|---|
| Precision | 0.8552 | 0.8918 | 0.9041 | 0.8918 | **0.9166** | 0.9041 | **0.9166** | 0.9041 |
| recall | 0.7831 | **0.7951** | **0.7951** | **0.7951** | **0.7951** | **0.7951** | **0.7951** | **0.7951** |
| F1-measure | 0.8176 | 0.8407 | 0.8461 | 0.8407 | **0.8516** | 0.8461 | **0.8516** | 0.8461 |

## 5 Conclusion and Future Work

The open characteristics of social networks makes them vulnerable to social spammers. To fight against social spammers, In this paper, we proposed a novel method, LSSL-SSD, for social spammer detection. It first calculates the Laplacian score of each feature for feature selection. Then, based on these selected features, the semi-supervised random forest method is used to get the final detection model. Experimental results show that the proposed method not only has a strong generalization ability due to the process of feature selection, but also has a good detection accuracy in the face of limited labeled data.

As further work, we will incorporate global information, such as the labels of data, to select features. This may increase the detection accuracy because we only care about local information in this paper. Moreover, for the problem of limited labeled data, active learning can be used together with semi-supervised learning to improve the detection performance.

## Acknowledgments

This work is supported by the Basic and Advanced Research Projects in Chongqing under Grant No. cstc2015jcyjA40049, the National Key Basic Research Program of China (973) under Grant No. 2013CB328903, the National Natural Science Foundation of China under Grant Nos. 61502062 and 61602070, the China Postdoctoral Science Foundations under Grant No.s 2012M521680 and 2014M560704, the Fundamental Research Fund for the Central Universities under Grant No. 106112014CDJZR095502, and the China Scholarship Council.

## References

1. Javier Borge-Holthoefer, Alejandro Rivero, and Yamir Moreno. Locating privileged spreaders on an online social network. *Physical review E*, 85(6):066123, 2012.
2. Zhi Wang, Lifeng Sun, Wenwu Zhu, Shiqiang Yang, Hongzhi Li, and Dapeng Wu. Joint social and content recommendation for user-generated videos in online social network. *IEEE Transactions on Multimedia*, 15(3):698–709, 2013.
3. Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A Zighed. Information diffusion in online social networks: A survey. *ACM SIGMOD Record*, 42(2):17–28, 2013.

4. Fangzhao Wu, Jinyun Shu, Yongfeng Huang, and Zhigang Yuan. Social spammer and spam message co-detection in microblogging with social context regularization. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1601–1610. ACM, 2015.
5. Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Social spammer detection with sentiment information. In *2014 IEEE International Conference on Data Mining*, pages 180–189. IEEE, 2014.
6. Xiang Zhu, Yuanping Nie, Songchang Jin, Aiping Li, and Yan Jia. Spammer detection on online social networks based on logistic regression. In *International Conference on Web-Age Information Management*, pages 29–40. Springer, 2015.
7. Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. *IEEE Internet Computing*, 11(6):36–45, 2007.
8. 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.
9. Yafeng Ren, Donghong Ji, Lan Yin, and Hongbin Zhang. Finding deceptive opinion spam by correcting the mislabeled instances. *Chinese Journal of Electronics*, 24(1):52–57, 2015.
10. Anupama Aggarwal, Jussara Almeida, and Ponnurangam Kumaraguru. Detection of spam tipping behaviour on foursquare. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 641–648. ACM, 2013.
11. Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 35–47. ACM, 2010.
12. Zhaoxing Li, Xianchao Zhang, Hua Shen, Wenxin Liang, and Zengyou He. A semi-supervised framework for social spammer detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 177–188. Springer, 2015.
13. 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.
14. Jonghyuk Song, Sangho Lee, and Jong Kim. Spam filtering in twitter using sender-receiver relationship. In *International Workshop on Recent Advances in Intrusion Detection*, pages 301–317. Springer, 2011.
15. Xia Hu, Jiliang Tang, and Huan Liu. Online social spammer detection. In *AAAI*, pages 59–65, 2014.
16. Enhua Tan, Lei Guo, Songqing Chen, Xiaodong Zhang, and Yihong Zhao. Unik: unsupervised social network spam detection. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 479–488. ACM, 2013.
17. Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *Advances in neural information processing systems*, pages 507–514, 2005.
18. Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.
19. Ming Li and Zhi-Hua Zhou. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(6):1088–1098, 2007.
20. 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.