Final version of the accepted paper. Cite as: *M. Abulaish and S. Y. Bhat, Classifier Ensembles using Structural Features for Spammer Detection in Online Social Networks, Foundations of Computing and Decision Sciences, Vol. 40, Issue 2, pp. 89-105, 2015.*

CLASSIFIER ENSEMBLES USING STRUCTURAL FEATURES FOR SPAMMER DETECTION IN ONLINE SOCIAL NETWORKS

MUHAMMAD ABULAISH[*] AND SAJID Y. BHAT[*]

Abstract. As the online social network technology is gaining all time high popularity and usage, the malicious behavior and attacks of spammers are getting smarter and difficult to track. The newer spamming approaches using the social engineering concepts are making traditional spam and spammer detection techniques obsolete. Especially, content-based filtering of spam messages and spammer profiles in online social networks is becoming difficult. Newer approaches for spammer detection using topological features are gaining attention. Further, the evaluation of ensemble classifiers for detection of spammers over social networking behavior-based features is still in its infancy. In this paper, we present an ensemble learning method for online social network security by evaluating the performance of some basic ensemble classifiers over novel community-based social networking features of legitimate users and spammers in online social networks. The proposed method aims to identify topological and community-based features from users' interaction network and uses popular classifier ensembles – bagging and boosting to identify spammers in online social networks. Experimental evaluation of the proposed method is done over a real-world data set with artificial spammers that follow a behavior as reported in earlier literature. The experimental results reveal that the identified features are highly discriminative to identify spammers in online social networks.

Keywords: Social network security, Spammer detection, Ensemble learning, Classifier ensembles, Feature extraction.

1. Introduction

The Online Social Network (OSN) technology has experienced an exploding popularity and usage in the recent years [20]. This popularity is mostly a result of the numerous features provided by them to their users, which mainly include joining a network (become users), defining their preferences (profile), and publishing any content which they like to share with other users in the network. An important feature of OSNs is enabling the users to define associations (friend, coworker, follower and so on) with other users in the network. These features allow creating/maintaining social relations, and share, find and endorse content and knowledge contributed by the users [26]. As the membership and activities of users in OSNs has grown, OSNs have also found their way into varied applications, including user profiling, target/viral marketing, election campaigning, e-governance, product promotions, etc. Though the OSNs have shown an enormous impact, one of the big challenges faced by OSN users is to deal with the existence of malicious users (spammers) that broadcast unsolicited information to a large number of legitimate users for spamming purpose. The motive behind spamming commonly includes promoting products, viral marketing, spreading fads, and in some cases possibly to harass legitimate users to decrease their trust in a particular service. The spamming activities also consumes large amount of network bandwidth leading to less revenue and significant financial loss to organizations.

Spamming is not a new thing and has been there since the popularity of the traditional e-mail. In traditional e-mail networks, the most common form of spamming involves the Random Link Attack (RLA) where a small number of spammers send spam to a large number of randomly selected victim nodes. Spammers tend to be senders of spam

[*] Department of Computer Science, Jamia Millia Islamia (A Central University), Delhi, India. E-mail: mAbulaish@jmi.ac.in, bhatsajid786@gmail.com

messages to a socially un-related set of receivers, unlike legitimate senders whose receivers tend to cluster or form communities [1]. Many spam/spammer detection methods have been proposed in literature, which are based on the content analysis (keywords-based filtering) of the interactions between users. However, many counter-filtering techniques based on the usage of non-dictionary words and images in spam objects are often employed by spammers [1]. Content-based spam filtering systems also demand higher computations. With the availability of popular OSN platforms, malicious minds have found a whole new world of opportunities and have devised novel and efficient techniques of taking spamming to new levels. Unlike traditional e-mail networks, filtering of spam and spammer accounts in OSNs is often faced with challenges like the existence of a thin line between the content shared by legitimate users and malicious accounts. Moreover, the deceptive accounts often tend to mimic the behavior of legitimate users, making it difficult to detect and categorize them.

Along these lines, the latest category of attacks faced by online social networks is *Social Engineering* which mainly involves extracting private information from an OSN user by means of manipulations [23]. OSNs of a user can be exploited to launch a full, highly complex social engineering attack against him/her or just can be used to take the first steps in breaching his trust/friendship nutshell. Two highly deceptive malicious activities that have often been used to launch social engineering attacks include the Sybil attack and the Cloning attack [5, 11]. A Sybil attack usually involves a single attacker to create multiple accounts interlinked with each other to form false communities. These accounts appear as legitimate nodes and disseminate spam to the legitimate parts of the network by deceiving legitimate users in creating trusted links with them, thus breaching their privacy. Similarly, a cloning attack involves copying the profile details of a legitimate user to form a fake account and then breach the trust of the friends of the legitimate user with malicious intentions.

It becomes highly desirable to devise advanced techniques and methods for identifying spammers and their behavior in online social networks. Along this direction, some spammer detection techniques are based on learning classification models that use network-based topological features like in-degree, out-degree, reciprocity, and clustering coefficient of the interacting nodes in online social networks to identify spammers. Incorporating additional sociological characteristics (like interaction behavior of nodes within and across network community structures) in the classification models can improve their performance for identifying spammers who incorporate sybil and cloning attacks. In [1], we have identified some novel community-based structural features, based on inter- and intra-community users interaction patterns, to learn improved classification models for spammer classification in online social networks. However, the results only spanned over single classifiers and the significance of using ensemble learning methods has not been extensively evaluated yet. In this paper, which is an extended version of our preliminary work on ensemble methods for spammer classification [3], we present details about the topological and community-based features and their significance towards spammer detection in online social networks. We also present an evaluation of individual features to establish their discriminative property for spammer detection using classifier ensembles. Two popular classifier ensembles – *bagging* and *boosting* are used over topological and community-based features extracted from a real-world social network data set with artificially planted spammers. Results are generated for both classifier ensembles using decision tree and naïve Bayes algorithms to evaluate the performance of the proposed spammer detection method.

The rest of the paper is organized as follows. Section 2 presents a brief review of the existing works on spam/spammer detection. Section 3 presents details about the topological and community-based features and their formulations. Section 4 briefly describes two classifier ensembles – bagging and boosting. Section 5 presents the experimental setup and evaluation results. Finally, section 6 concludes the paper with future directions of work.

2. Related work

Spam/Spammer detection methods usually involve two approaches – content-based learning and topology-based learning. In literature, most of the work has been done along the lines of detecting e-mail spam and web spam mainly by exploiting content-based patterns of spam emails and web pages. The main idea behind content-based learning revolves around the observation that spammers use distinguished keywords, URLs, etc. in their interactions and to define their profiles. Such content-based features are used to learn classification models to label messages and profiles as legitimate or spam [31]. However, such approach is often deceived by spammers using copy profiling and content obfuscation.

With the evolution of OSNs the spamming behavior shown and the content disseminated by spammers has changed and got similar to that of legitimate users. This makes detecting spammers in OSNs a highly challenging task. Along these lines topology-based learning methods aim to exploit structural social network features like clustering coefficient, community structures, reciprocity, node degree, etc. to characterize network behavior of legitimate and spammer accounts. Shrivastava *et al.* [30] incorporated features including clustering coefficient and neighborhood independence to deal with random link attacks from spammers. Gan and Suel [16] extracted features like in-links, out-links, cross-links, etc. from a Web graph to classify pages as spam or benign. Other methods include finding physical node clusters based on network-level features from online communication networks [29]. To detect spam clusters, Gao *et al.* [17] used two widely acknowledged distinguishing features of spam campaigns – *distributed coverage* and *bursty nature*. The *distributed* property is quantified using the number of users that send wall posts in the cluster, whereas *bursty* property is based on the intuition that most spam campaigns involve coordinated action by many accounts within short periods of time [33].

In order to, more strictly, characterize the spammers from legitimate users, the methods proposed in [1] and [13] exploit the existence of community structures in social networks [2] to extract novel structural features for the task. Communities resemble groups of nodes that are relatively densely connected to each other but sparsely connected to other dense groups in the network. Identifying community structure in social networks is important as it reveals the functional groups in a system and thus provides information about the role of individual nodes. In the context of spammer detection, a node at the boundary of a community which has out links to nodes that belong to other distinct communities may be considered suspicious, as legitimate users tend to show high interactivity within their respective communities [1]. For spammer detection, these methods split the interaction network of OSN users into communities and then extract community-based features of network nodes (users) to classify them as spammer or legitimate.

Besides, there exist ensemble methods that can be used to improve the performance of classifiers by learning multiple models over the same training example set and then using some aggregation method to decide upon a single combined label determined by multiple classifiers. Using ensemble learning methods to improve the performance of spam detection methods have been adopted by many researchers, but their studies have been mainly oriented towards content-based classification of e-mail and web-spam [12]. It is important to note that evaluating ensemble classification methods for spammer detection over topological features of OSN user interaction graphs is still in its infancy. In [18], the authors used an ensemble under-sampling classification strategy incorporating C4.5, bagging, and adaboost. Their results using the ensemble approach showed improvement in Web spam detection performance effectively. Using a text corpus, the authors in [27] aimed to show the significance of classifier ensembles over individual classifiers for spam detection. However, they failed to show any significant improvement in the task. In [22], the authors highlighted the high performance of classifier ensembles involving Adaboost, Stacking, and Ensemble Decision Tree, against the best performance of single classifiers for e-mail spam detection using a content-based approach. In [8], the authors showed that the classifier ensemble proposed by Caruana *et al.* [9] performed better than most individual and classifier ensembles implemented in WEKA for the task of email spam detection. In [12], the authors exploited both content-based and link-based features to compile a minimal feature set that can be computed incrementally in a quick manner to allow intercepting spam. They also showed that for a selected feature set, ensemble classification technique outperforms previously published methods and the Web spam challenge 2008 best results.

3. Topological and community-based features

In this section, we present the formulation of topological and community-based features that are used to learn classifier ensembles for spammer detection in online social networks. Though topological features for each node (user) in the users' interactions network are defined using graph properties, an overlapping community structure of nodes is identified to define community-based features [4]. The community-based features include the features that express the role of a node in the community structure, i.e., whether a node is a boundary node or a core node, and the number of communities a particular node belongs to. Further details about different types of features identified from users' interactions network are given in the following sub-sections.

3.1 Topological features

In this section, various topological features are defined using the basic graph properties. For each feature, a shortened notation is assigned and given in parenthesis for reference in the remaining portion of this paper.

***Total out-degree* (*TOD*):** This is a directed graph-based feature, which captures the interaction behavior of a user with other users in the network. The *total out-degree* of a node (user) *u* represents the total number of distinct users in the social network with whom *u* has direct out-links, i.e., to whom *u* sends messages, etc. It is formally defined using Equation 1, where *V* is the set of nodes and *E* is the set of edges in social networks.

$$TOD(u) = \left|\{v \mid v \in V \wedge (u,v) \in E\}\right| \tag{1}$$

***Total reciprocity* (*TR*):** This is also a directed graph-based feature, which captures the mutual interaction pattern of a user with other users in the network. The *total reciprocity* of a node *u* represents the ratio of the number of mutual interactions of *u* to the total number of nodes with which *u* has out-links. Formally, it can be defined using Equation 2, where L_i^u is the set of links (edges) incident to node *u* and L_o^u is the set of links (edges) originating from *u*.

$$TR(u) = \frac{\left|L_i^u \cap L_o^u\right|}{\left|L_o^u\right|} \tag{2}$$

***Total in/out ratio* (*TIOR*):** This is a more generic feature than the *TR* feature which is defined for a node (user) *u* as the ratio of the number of links (edges) incident to *u* to the number of links (edges) originating from *u*. Formally, it can be defined using Equation 3, where the notations have the same interpretations as given in Equation 2.

$$TIOR(u) = \frac{\left|L_i^u\right|}{\left|L_o^u\right|} \tag{3}$$

3.2 Community-based features

In this section, various community-based features are defined which exploits the community structure of online social networks. Like topological features, community-based features are also assigned a shortened notation and given in parenthesis for their reference in the remaining portion of this paper.

***Core node* (*CN*):** This is a Boolean property which takes either 1 or 0 value. For a node (user) *u*, the *CR* feature value is set to 1 if the community detection method used to identify community structures in social network marks *u* as a core node, otherwise its value is set to 0. Formally, it can be defined using Equation 4.

$$CN(u) = \begin{cases} 1 & if\ core(u) = True \\ 0 & otherwise \end{cases} \tag{4}$$

***Community memberships* (*CM*):** This is also a community-based feature, which is defined as the number of communities to which a node (user) *u* is assigned by the overlapping community detection method. Formally, it can be defined using Equations 5 and 6, where C^u is the community set of node *u*, C_k is community identifier, and *n* is the number of communities identified by the overlapping community detection method in social network. In case the node *u* is marked as outlier by the overlapping community detection method, its *CM* value is set to 0.

$$CM(u) = \left|C^u\right| \tag{5}$$

$$C^u = \{C_k \mid u \in C_k, k = 1...n\} \tag{6}$$

In order to define the remaining community-based features, we first need to discuss the concept of *foreign node*. For a given node *u*, a node *v* is said to be a *foreign node* if both the nodes never belong to same community, i.e., the community sets of the nodes *u* and *v* are mutually exclusive. Formally, the set of foreign nodes for a node *u* (F^u) is be defined using Equation 7.

$$F^u = \{v \mid v \in V \wedge (C^u \cap C^v = \varphi)\} \tag{7}$$

***Foreign out-degree* (*FOD*):** The foreign out-degree of a node *u* is defined as the number of foreign nodes to which *u* has out-links. Formally, it is defined using Equation 8, where *E* is the set of edges in the social network.

$$FOD(u) = \left|\{v \mid v \in F^u \wedge (u,v) \in E\}\right| \tag{8}$$

***Foreign in/out ratio* (*FIOR*):** The foreign in/out ratio for a node *u* is defined as the ratio of the number of foreign nodes that have out-links to *u*, to the number of foreign nodes to which the node *u* has out-links. Formally, it is defined using Equations 9-11, where F_i^u is the set of foreign nodes from which the node *u* has in-links and F_o^u is the set of foreign nodes with which u has out-links.

$$FIOR(u) = \frac{\left|F_i^u\right|}{\left|F_o^u\right|} \tag{9}$$

$$F_i^u = \left|\{v \mid v \in F^u \wedge (v,u) \in E\}\right| \tag{10}$$

$$F_o^u = \left|\{v \mid v \in F^u \wedge (u,v) \in E\}\right| \tag{11}$$

***Foreign reciprocity* (*FR*):** The *foreign reciprocity* of a node *u* represents the ratio of the number of mutual interactions of the node *u* with its foreign nodes to the total number of foreign nodes with whom *u* has out-links. Formally, it is defined using Equation 12.

$$FR(u) = \frac{\left|F_i^u \cap F_o^u\right|}{\left|F_o^u\right|} \tag{12}$$

***Foreign out-link probability* (*FOLP*):** The foreign out-link probability of a node (user) *u* represents the probability of its out-links to the foreign nodes. It is defined as a ratio of the number of foreign nodes with which *u* has out-links to the total number of nodes with which *u* has out-links, as given in Equation 13.

$$FOLP(u) = \frac{\left|F_o^u\right|}{\left|L_o^u\right|} \tag{13}$$

***Foreign out-link grouping* (*FOLG*):** The foreign out-link grouping feature of a node *u* represents the probability that the foreign nodes out-linked with *u* have a common community. If $MF_o^u \subseteq F_o^u$ is the maximal set of foreign nodes out-linked with *u* that have a common community, then the *FOLG* feature of *u* is calculated as the ratio of the number of nodes in MF_o^u to the number of nodes in F_o^u, as given in Equation 14.

$$FOLG(u) = \frac{\left|MF_o^u\right|}{\left|F_o^u\right|} \tag{14}$$

4. Classifier ensembles

Classifier ensembles combine multiple machine learning instances to improve the classification results of a system. It is based on the assumption that combination of multiple classifiers may be able to produce a better classification system, which is more stable and accurate than any of its individual components. According to Dietterich [10], the performance advantage of classifier ensembles can be attributed to three key factors: (i) Combining multiple hypotheses to form an ensemble such that the votes of individual classifiers are averaged and the risk of selecting an incorrect hypothesis is reduced, (ii) Starting a local search in different locations; ensemble can provide a better approximation of the true underlying function, (iii) Weighted sum of the hypotheses within an ensemble may extend the space of representable hypotheses to allow a more accurate representation. A brief description of the mostly used ensemble methods is given in the following sub-sections.

4.1 Bagging

This is one of the simple classifier ensemble methods. Bootstrap aggregation (or Bagging), proposed by Breiman [7], involves training multiple instances of classifiers on a sample of training examples that are taken at random with replacement (bootstrap sample). Finally, the labels of the test samples are determined by a majority vote of each internally learned classifier. However, bagging methods gives equal weightage to all classifiers.

4.2 Boosting

Also called as arcing (Adaptive Resampling and Combining) [15], boosting first involves assigning weights to the training set instances, then on each learning iteration it increases and decreases the weights of misclassified and correctly classified instances, respectively. The difficulty of the learning problem is effectively increased after each iteration, with an attempt to minimize the weighted error over the training set. It involves repeatedly learning a weak classifier on various distributed samples of the training data. The classifiers learnt at each step are then combined into a single strong classifier to achieve a higher accuracy than the individual ones. Increasing the weights of misclassified instances increases their selection probability for the next iteration, and thereby a weak learner is forced to focus on difficult examples of the training set. For a test sample, the final classification decision is based on the combination of the decisions made in all rounds, namely a weighted majority vote, where decisions with lower classification error have higher weights.

5. Experimental setup and results

In this section, we present a brief description of the experimental data set followed by some of the experimental results to establish the efficacy of the of the classifier ensembles for spammer detection in online social networks. The topological and community-based features discussed in Section 3 are extracted from a real-world social network data set with artificially planted spammers. We compare the performance of multiple classifiers, including decision tree and naïve Bayes, and their ensemble variants implemented as a part of WEKA [14], which is an implementation of machine learning algorithms for data mining tasks, ranging from data pre-processing and classification rule mining to clustering. Further details about the data set and experimental results are presented in the following sub-sections.

5.1 Data set

As discussed earlier, the approach followed in this paper aims to detect spammers by extracting structural features from the user interaction patterns of OSN users. In this regard the dataset required for analysis is expected to contain a weighted (weights representing the frequency of interactions) network including both legitimate and spammer nodes. However, due to the unavailability of such a dataset and the complexity of extracting the same from a OSN given the access restrictions, we are left with the option of generating an artificial network which would reflect the real world situation to a maximum extent. For the experiments conducted in this paper, we have used a real-world social network data set representing the wall post activities of about 63891 Facebook users [32]. The nodes in this network are considered to be legitimate nodes. We inject additional nodes in the network to simulate spammer behavior. In this regard, we subsequently filter out all the nodes having zero in-degree or out-degree, and any isolated nodes from the network to represent them as legitimate networks. This results in a network with 32693 legitimate nodes. Thereafter, in order to simulate spammers, we generate a set of 1000 isolated nodes for the legitimate network, which creates out-links to randomly selected nodes in the legitimate network. The out-links or the out-degree generated for the spammers are not random, rather it follows the distribution shown by spammers as reported in [19] and used in [6] and [24]. The out-degree distribution of spammer nodes (as reported in [19]) is shown in Table 1. Since the messages of the spammers are expected to be least often reciprocated, the probability of a legitimate node replying to a spammer is set to 0.05.

Table 1. Spammer out-degree distribution

Y	**P(out-degree=*y*)**
1	0.664
2	0.171
3	0.07
4	0.04
5	0.024
6	0.014
7	0.01
8	0.007

Table 2. LFR-Benchmark parameter description and values

Parameter	Description	Value
N	Number of nodes	1000
K	Average degree	15
k_{max}	Max degree	60
C_{min}	Minimum community size	15
C_{max}	Maximum community size	60
τ_1	Degree exponent	-1
τ_1	Community exponent	-1
μ	Mixing parameter	0.1

In order to make the detection task more difficult, we generate another set of 1000 spammer nodes, which try to mimic the clustering/community property of legitimate nodes. In order to do so, we have used the LFR-benchmark generator [25] to generate a directed network of 1000 nodes with embedded community structures. The LFR-benchmark parameters used to generate the network are shown in Table 2. Thereafter, for each node in the synthetic network, we rewire a set of its out-links to a set of randomly selected nodes in the legitimate network in such a way that the spamming out-degree (i.e., the rewired out-links) follows the spammer out-degree distribution given in Table 1. In this regard, a total number of 2000 spammer nodes (out of which 1000 mimic the clustering property of legitimate nodes) are added to the legitimate network, resulting in a total number of 34693 nodes in the final network. Finally, both topological and community-based features are extracted from the resultant network with injected spammers to learn various classifier ensembles. In order to extract community-based features, one of our density-based overlapping community detection algorithms proposed in [4] is applied to identify overlapping community structures in the social network.

5.2 Results

In order to evaluate the significance of the ensemble learning methods using topological and community-based features, a set of classifiers from WEKA is learned on the training data set. The performance of two well-known classifiers including J48 (decision tree) [28] and naïve Bayes [21] using bagging and boosting ensemble methods is evaluated for both *spam* and *non-spam* classes. For each classifier, 10-fold cross validation is applied to calculate the values of various performance evaluation metrics. We have considered *True*

Positive Rate (*TPR*), *False Positive Rate* (*FPR*), *Precision*, and *F-measure* to evaluate the performance of both individual classifiers and ensemble methods. These metrics are defined in terms of *True Positive* (*TP*), *False Positive* (*FP*), *True Negative* (*TN*), and *False Negative* (*FN*), where *TP* is the number of positive instances classified as positive, *FP* is the number of negative instances classified as positive, *TN* is the number of negative instances classified as negative, and *FN* is the number of positive instances classified as negative. Formally, *TPR*, *FPR*, *Precision*, *Recall*, and *F-measure* can be defined using Equations 15, 16, 17, 18, and 19, respectively.

$$TPR = \frac{TP}{TP + FN} \tag{15}$$

$$FPR = \frac{FP}{FP + TN} \tag{16}$$

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

$$F\text{-}measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{19}$$

Table 3 presents the performance (averaged for both spam and non-spam classes) of the individual classifiers on the data set with planted spammers, wherein it is clear that the decision tree based classifier J48 performs better than the naïve Bayes classifier. Tables 4 and 5 present the performance of the bagging ensemble over the base classifiers – naïve Bayes and J48, respectively, whereas Tables 6 and 7 present the performance of the boosting ensemble over the base classifiers – naïve Bayes and J48, respectively for spammer detection task. It can be observed from these tables that the performance of naïve Bayes and J48 classifiers using either bagging or boosting ensemble learning approach is better than their individual performance. However, in case of naïve Bayes classifier, the ensemble approaches show low performance than the ensemble approaches using J48 classifier.

Table 3. Classification performance (weighted avg. over both classes) of individual classifiers

Classifier	**TPR**	**FPR**	**Precision**	**F-measure**
J48	0.963	0.075	0.963	0.963
Naïve Bayes	0.914	0.175	0.917	0.915

Table 4. Classification performance using bagging with naïve Bayes classifiers

Class	**TPR**	**FPR**	**Precision**	**F-measure**	**ROC area**
Non-spam	0.957	0.175	0.989	0.973	0.959
Spam	0.825	0.043	0.542	0.654	0.959
Weighted avg.	**0.95**	**0.167**	**0.963**	**0.955**	**0.959**

Table 5. Classification performance using bagging with J48 (decision tree) classifiers

Class	**TPR**	**FPR**	**Precision**	**F-measure**	**ROC area**
Non-spam	0.999	0.022	0.999	0.999	1
Spam	0.978	0.001	0.981	0.979	1
Weighted avg.	**0.998**	**0.021**	**0.998**	**0.998**	**1**

Table 6. Classification performance using boosting with naïve Bayes classifiers

Class	TPR	FPR	Precision	F-measure	ROC area
Non-spam	0.969	0.277	0.983	0.976	0.977
Spam	0.724	0.031	0.587	0.648	0.977
Weighted avg.	**0.955**	**0.262**	**0.96**	**0.957**	**0.977**

Table 7. Classification performance using boosting with J48 (decision tree) classifiers

Class	TPR	FPR	Precision	F-measure	ROC area
Non-spam	0.999	0.016	0.999	0.999	0.999
Spam	0.984	0.001	0.987	0.986	0.999
Weighted avg.	**0.998**	**0.015**	**0.998**	**0.998**	**0.999**

Table 8. Ranking of topological and community-based features generated by Weka using bagging with J48 classifiers, based on 10-fold cross validation

Feature	Description	Category	Avg. merit	Avg. rank
TOD	Total out-degree	Topological	0.041	6
TR	Total reciprocity	Topological	0.170 ± 0.002	2
TIOR	Total in/out ratio	Topological	0.192 ± 0.001	1
CN	Core node	Community-based	0.037	7
CM	Community memberships	Community-based	0.061	5
FOD	Foreign out-degree	Community-based	0.071 ± 0.001	4
FIOR	Foreign in/out ratio	Community-based	0.018	9
FR	Foreign reciprocity	Community-based	0.035	8
FOLP	Foreign out-link probability	Community-based	0.080 ± 0.001	3
FOLG	Foreign out-link grouping	Community-based	0.016 ± 0.001	10

Table 9. Classification performance using bagging with J48 classifiers after excluding one feature at a time

Feature set	Class	TPR	FPR	Precision	F-measure
F - {TIOR}	Non-spam	0.999	0.026	0.998	0.999
	Spam	0.974	0.001	0.979	0.976
	Weighted avg.	0.997	0.025	0.997	0.997
F - {TR}	Non-spam	0.999	0.025	0.999	0.999
	Spam	0.976	0.001	0.98	0.978
	Weighted avg.	0.997	0.023	0.997	0.997
F - {FOLP}	Non-spam	0.999	0.02	0.999	0.999
	Spam	0.98	0.001	0.981	0.981
	Weighted avg.	0.998	0.019	0.998	0.998
F - {FOD}	Non-spam	0.999	0.02	0.999	0.999
	Spam	0.98	0.001	0.981	0.981
	Weighted avg.	0.998	0.019	0.998	0.998
F - {CM}	Non-spam	0.999	0.023	0.999	0.999
	Spam	0.978	0.001	0.981	0.979
	Weighted avg.	0.998	0.021	0.998	0.998
F - {TOD}	Non-spam	0.999	0.026	0.999	0.999
	Spam	0.974	0.001	0.979	0.976
	Weighted avg.	0.997	0.025	0.997	0.997
F - {CN}	Non-spam	0.999	0.022	0.999	0.999

	Spam	0.978	0.001	0.981	0.98
	Weighted avg.	0.998	0.021	0.998	0.998
F - {FR}	Non-spam	0.999	0.022	0.999	0.999
	Spam	0.979	0.001	0.982	0.98
	Weighted avg.	0.998	0.02	0.998	0.998
F - {FIOR}	Non-spam	0.999	0.02	0.999	0.999
	Spam	0.98	0.001	0.981	0.981
	Weighted avg.	0.998	0.019	0.998	0.998
F - {FOLG}	Non-spam	0.999	0.027	0.998	0.999
	Spam	0.974	0.001	0.978	0.976
	Weighted avg.	0.997	0.025	0.997	0.997

Since the performance of ensemble methods using J48 performs better than the ensemble methods using naïve Bayes classifier, and that the performance of bagging with J48 and boosting with J48 is comparable, we have used bagging with J48 classifier for further experiment to establish the discriminative property of both topological and community-based features. Table 8 presents the ranking of both topological and community-based features generated by Weka using bagging with J48 classifier, based on a 10-fold cross validation. To judge the discriminative property of individual features, we have repeated the classification process using bagging with J48 classifier, each time excluding one feature from the feature set and the classification performance is presented in Table 9, where *F* represents the set of all 10 features.

6. Conclusion and future work

Spammer detection in online social networks is challenging, but a highly desirable task. Numerous machine learning approaches using content-based features have been used in literature to detect email spam. However, as the spam/spammer filtering technology has evolved, so has the behavior of spammers and the techniques devised by them. The latest social engineering attacks devised by spammers mainly involve mimicking the behavior of legitimate users both content-wise and topologically. This makes it difficult for the traditional content based and topological spammer detection techniques to prove their mettle. On the other hand, ensemble learning approaches like bagging and boosting that aim to improve the performance of individual classifiers, have not been extensively evaluated for the spammer detection task in online social networks. Moreover, new structural features based on community structures of online social network users have also been proposed recently for spammer detection but still remain underexploited. In this paper, we have presented a major classification and formulation of various structure-based features and evaluated the performance of bagging and boosting ensemble learning approaches for the task of spammer detection in online social networks. Experimental results reveal that the bagging ensemble learning approach using J48 (decision tree) base classifier performs better than its individual model and also better than some other ensemble learning approaches for spammer detection using topological and community-based social network features.

As a direction of future work, structural features along with contents generated by individual users (e.g., wall posts, messages, etc.) can be considered to design more reliable spammer detection methods. Similarly, contents of spammers' posts can be analyzed using natural language processing and text mining approaches to identify different types of spam campaigns made by spammers in online social networks. Moreover, as stated earlier, evaluation of ensemble classifiers for spammer detection especially using new topological and sociological features is still in its infancy and needs more evaluation. Specifically, one of the main issues for classification models for spammer detection from OSNs is that of class imbalance i.e. more legitimate nodes and less spammer nodes. Various ensemble learning approaches have been proposed in literature to deal with overfitting and class imbalance, which can be rigorously explored for the spammer detection problem.

References

[1] Bhat S. Y., Abulaish M., Community-based features for identifying spammers in online social networks, in: *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, ACM, 2013, 100-107.

[2] Bhat S. Y., Abulaish M., Analysis and mining of online social networks: emerging trends and challenges, *WIREs: Data Mining and Knowledge Discovery*, 3, 6, 2013, 408-444.

[3] Bhat S. Y., Abulaish M., Mirza A. A., Spammer classification using ensemble methods over structural social network features, in: *Proceedings of the 14th IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Warsaw, Poland, 2014, 454-458.

[4] Bhat S. Y., Abulaish M., HOCTracker: Tracking the evolution of hierarchical and overlapping communities in dynamic social networks, *IEEE Transactions on Knowledge and Data Engineering*, 27, 4, 2014, 1019-1032.

[5] Bilge L., Strufe T., Balzarotti D., Kirda E., All your contacts are belong to us: automated identity theft attacks on social networks, in: *Proceedings of the 18th International Conference on World Wide Web (WWW)*, ACM, NY, USA, 2009, 551-560.

[6] Bouguessa M., An unsupervised approach for identifying spammers in social networks, in: *Proceedings of the IEEE 23rd International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, Washington DC, USA, 2011, 832-840.

[7] Breiman L., Bagging predictors, *Machine Learning*, 24, 2, 1996, 123-140.

[8] Carpinter J. M., *Evaluating Ensemble Classifiers for Spam Filtering*, Honours Thesis, University of Canterbury, 2005.

[9] Caruana R., Niculescu-Mizil A., Crew G., Ksikes A., Ensemble selection from libraries of models, in: *Proceedings of the 21st International Conference on Machine Learning*, 2004, 137–144.

[10] Dietterich T. G., Ensemble methods in machine learning, *Lecture Notes in Computer Science*, 1857, 2000, 1–15.

[11] Douceur J. R., The sybil attack, in: *Revised Papers from the 1st International Workshop on Peer-to-Peer Systems*, Springer-Verlag, London, UK, 2002, 251-260.

[12] Erdélyi M., Garzó A., Benczúr A. A., Web spam classification: a few features worth more, in: *Proceedings of the Joint WICOW/AIRWeb Workshop on Web Quality*, ACM, 2011, 27-34.

[13] Fire M., Katz G., Elovici Y., Strangers intrusion detection-detecting spammers and fake proles in social networks based on topology anomalies, *Human Journal*, 1, 1, 2012, 26–39.

[14] Frank E., Hall M., Holmes G., Kirkby R., Pfahringer B., Witten I., Trigg L. Weka, O. Maimon and L. Rokach (Eds.), *Data Mining and Knowledge Discovery Handbook*, Springer, 2005, 1305-1314.

[15] Freund Y., Schapire R. E., Experiments with a new boosting algorithm, in *Proceedings of the 13th International Conference on Machine Learning*, 1996, 325-332.

[16] Gan Q., Suel, T., Improving web spam classifiers using link structure, in: *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, ACM, NY, USA 2007, 17–20.

[17] Gao H., Hu J., Wilson C., Li Z., Chen Y., Zhao B. Y., Detecting and characterizing social spam campaigns, in: *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC)*, ACM, NY, USA, 2010, 35–47.

[18] Geng G. G., Wang C. H., Li Q. D., Xu L., Jin X. B., Boosting the performance of web spam detection with ensemble under-sampling classification, in: *Proceedings of the 4th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'07)*, IEEE, 2007, 583-587.

[19] Gomes L. H., Almeida R. B., Bettencourt L. M. A., Almeida V., Almeida J. M., Comparative graph theoretical characterization of networks of spam and legitimate email, in: *Proceedings of the 2nd Conference on Email and Anti-Spam (CEAS)*, 2005, 1-8.

[20] Jiang J., Wilson C., Wang X., Sha W., Huang P., Dai Y., Zhao B. Y., Understanding latent interactions in online social networks, *ACM Transactions on the Web*, 7, 4, 2013.

[21] John G. H., Langley P., Estimating continuous distributions in bayesian classifiers, in: *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI)*, San Francisco, USA, 1995, 338-345.

[22] Kiran P., Atmosukarto I., *Spam or Not Spam – that is the question*, Technical Report, University of Washington, URL: http://www.cs.washington.edu/homes/indria/research/spamfilter ravi indri.pdf, Date of access: Apr 1, 2014.

[23] Krombholz K., Hobel H., Huber M., Weippl E., Advanced social engineering attacks, *Journal of Information Security and Applications*, 2014, 1-10.

[24] Lam Ho-Y., Yeung Dit-Y., A Learning approach to spam detection based on social networks, in: *Proceedings of the 4th Conference on Email and Anti-Spam (CEAS)*, Mountain View, California, 2007.

[25] Lancichinetti A., Fortunato S., Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities, *Physical Review E*, 80, 2009.

[26] Mislove A., Marcon M., Gummadi K. P., Druschel P., Bhattacharjee B., Measurement and analysis of online social networks, in: *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ACM, 2007, 29-42.

[27] Neumayer R., Clustering based ensemble classification for spam filtering, in: *Proceedings of the 7th Workshop on Data Analysis (WDA'06)*, 2006, 11-22.

[28] Quinlan J. R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, USA, 1993.

[29] Ramachandran A., Feamster N., Vempala S., Filtering spam with behavioral blacklisting, in: *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS)*, ACM, NY, USA, 2007, 342-351.

[30] Shrivastava N., Majumder A., Rastogi R., Mining (social) network graphs to detect random link attacks, in: *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE)*, IEEE, Washington DC, 2008, 486-495.

[31] Stringhini G., Kruegel C., Vigna G., Detecting spammers on social networks, in: *Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC)*, ACM, NY, USA, ACM, 2010, 1–9.

[32] Viswanath B., Mislove A., Cha M., Gummadi K. P., On the evolution of user interaction in Facebook, in: *Proceedings of the Workshop on Online Social Networks*, 2009, 37-42.

[33] Xie Y., Yu F., Achan K., Panigrahy R., Hulten G., Osipkov I., Spamming botnets: signatures and characteristics, *SIGCOMM Computing Communication Review*, 38, 4, 2008, 171-182.