Final version of the accepted paper. Cite as: *"S. Y. Bhat, M. Abulaish, and A. A. Mirza, Spammer Classification using Ensemble Methods over Structural Social Network Features, In Proceedings of the IEEE/WIC/ACM Int'l. Conference on Web Intelligence (WI'14), Warsaw, Poland, pp. 454-458, Aug. 11-14, 2014."*

Spammer Classification using Ensemble Methods over Structural Social Network Features

Sajid Yousuf Bhat[1], Muhammad Abulaish[2, #]
[1,2]Department of Computer Science
Jamia Millia Islamia (A Central University)
New Delhi-110025, India
e-mail: s.yousuf.bhat@gmail.com

Abdulrahman A. Mirza[3]
[3] Information Systems Department
King Saud University
Riyadh, Kingdom of Saudi Arabia
e-mail: amirza@ksu.edu.sa

Abstract -- **The overwhelming growth and popularity of online social networks is also facing the issues of spamming, which mainly leads to uncontrolled dissemination of malware/viruses, promotional ads, phishing, and scams. It also consumes large amounts of network bandwidth leading to less revenue and significant financial losses to organizations. In literature, various machine learning techniques have been extensively used to detect spam and spammers in online social networks. Most commonly, individual classifiers are learnt over content-based features extracted from users' interactions and profiles to label them as spam/spammers or legitimate. Recently, new network structure-based features have also been proposed for spammer detection task, but their significance using ensemble learning methods has not been extensively evaluated yet. In this paper, we evaluate the performance of some ensemble learning methods using community-based structural features extracted from an interaction network for the task of spammer detection in online social networks.**

Keywords-Social network security; spam detection; machine learning; classifier ensemble

I. INTRODUCTION

One of the big challenges faced by Online Social Networks (OSNs) is to deal with undesirable users and their malicious activities like spamming, which involves malicious users (spammers) to broadcast irrelevant information to as large number of legitimate users as possible. 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. Thus, it becomes highly desirable to devise techniques and methods for identifying spammers and their behavior in online social networks. Along this direction, many spam/spammer detection methods have been proposed in literature, which are mostly based on content analysis (keywords-based filtering) of users' interaction data. However, many counter filtering techniques based on the usage of non-dictionary words and images in spam objects are often employed by spammers. Content-based spam filtering systems also demand higher computations. Alternatively, some spammer detection techniques are based on learning classification models from network-based topological features of the interacting nodes in online social networks. These features mainly include in-degree, out-degree, reciprocity, clustering coefficient, etc. Spammers are often seen to mimic some patterns of legitimate interaction behavior, making it difficult to characterize them. 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.

Recently, in [10], we have proposed some community-based topological features to learn improved classification models for identifying spammers in online social networks. However, the results only spanned over single classifiers. In this paper, we aim to evaluate the performance of the proposed features in [10] for learning ensemble classification models for the task of spammer detection in online social networks. We have used three ensemble learning methods – Bagging, Boosting, and Stacking over topological features extracted from a real-world interaction network with artificially planted spammers. Results are generated for both single and ensemble classifiers to evaluate their performance.

II. RELATED WORK

Spam/Spammer detection methods usually involve two approaches – content-based learning and topology-based learning. 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 [15]. However, such approach is often deceived by spammers using copy profiling and content obfuscation. On the other hand, topology-based learning methods aim to exploit structural social network features like clustering coefficient, community structures, reciprocity, node

Corresponding author, E-mail: mAbulaish@jmi.ac.in

degree, etc. to characterize network behavior of legitimate and spammer accounts. Shrivastava et al. [12] 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 [17]. To detect spam clusters, Gao et al. [18] 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 the *bursty* property is based on the intuition that most spam campaigns involve coordinated action by many accounts within short periods of time [19]. The methods proposed in [10] and [20] used a community detection method to split the interaction network into communities and then extract community-based features of network nodes (users) to classify them as spammers or legitimate.

One of the limitations of the approaches mentioned above is that the classification models used by them are mostly single. In literature, 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. Although, some ensemble methods have been used in content-based classification of spammers, they have not yet been evaluated for the topological features. However, the use of ensemble learning methods for improving the performance of spam detection methods has been adopted by many researchers, but the studies have been mainly oriented towards content-based classification. In [1], 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 [2] aimed to show the significance of ensemble classifiers over individual classifiers for spam detection. However, they failed to show any significant improvement in the task. In [3], the authors highlighted the high performance of ensemble classifiers involving Adaboost, Stacking, and Ensemble Decision Tree, against the best performances of single classifiers for e-mail spam detection using a content-based approach. In [4], the authors showed that the ensemble classifier proposed by Caruana et al. [6] performed better than most individual and ensemble classifiers implemented in WEKA for the task of email spam detection. In [5], 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.

III. ENSEMBLE METHODS

Ensemble classifiers group 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 an overall classifier which is more stable and accurate than any of its individual components. According to Dietterich [8], the performance advantage of ensemble classifiers can be attributed to three key factors: (i) by combining multiple hypotheses to form an ensemble; their votes are averaged and the risk of selecting an incorrect hypothesis is reduced, (ii) by starting a local search in different locations; ensemble can provide a better approximation of the true underlying function, (iii) a weighted sum of the hypotheses within the ensemble, which may extend the space of representable hypotheses to allow a more accurate representation.

A brief description of the three mostly used ensemble methods is given in the following paragraphs.

A. Bagging

Bootstrap aggregation (or bagging), proposed by Quinlan [7], involves training multiple instances of classifiers on a sample of training examples which 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.

B. Boosting

Also called as arcing (Adaptive Resampling and Combining) [11], boosting first involves assigning weights to the training set instances, then on each learning iteration it increases and decreases the weights for misclassified and correctly classified instances, respectively. The difficulty of the learning problem is effectively increased on each iteration, with an attempt to minimize the weighted error on 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 increases the selection probability of the misclassified instances for the next iteration, thus the weak learner is forced to focus on the difficult examples of the training set. The final classification decision is a combination of the decisions made in all rounds, namely a weighted majority vote, where decisions with lower classification error have higher weight.

C. Stacking

Stacking is an ensemble learning approach which aims to determine the reliability of its constituent individual classifiers (often different models) and to

achieve the highest generalization accuracy [9]. It involves using a meta-learner, which uses the predicted classification of the constituent classifiers as input attributes, instead of using original input attributes. The test instance classification labels are first determined by each of the base classifiers which then form meta-level training set. From the training set, a meta-classifier is produced which combines different predictions into a final one. Usually, the original dataset is partitioned into two subsets, one for creating meta-dataset and the other to build base-level classifiers. The meta-classifier is used to reflect the true performance of the individual constituent classifiers.

IV. EXPERIMENTAL RESULTS

As mentioned earlier, the main aim of this paper is to evaluate the performance of the ensemble classifiers for spam/spammer detection in online social networks. In order to do so, community-based structural features proposed in [10] are extracted from a real-world social network dataset with artificially planted spammers. We compare the performance of multiple classifiers including decision trees, NaïveBayes, and k-NN and their ensemble variants implemented in WEKA [21] software.

TABLE I: 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

A. Dataset

For experimental work, we use a real-world social network dataset representing the wall post activity of about 63891 Facebook users [22]. 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 which retains 32693 legitimate nodes. Thereafter, in order to simulate spammers, we generate a set of 1000 isolated nodes for the legitimate network, which create out-links to randomly selected nodes in the legitimate network. The out-links or the out-degree generated for the spammers are not random but follow the distribution shown by spammers as reported in [13] and also used in [23] and [14] as shown in Table I. The messages of the spammers are expected to be least often reciprocated. Thus, the probability of a legitimate node replying to a spammer is set to 0.05.

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 [24] 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 II.

TABLE II: 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

Now, for each node in the synthetic network, we rewire a set of its out-links towards a set of randomly selected nodes in the legitimate network such that the spamming out-degree (i.e., the rewired out-links) follows the distribution given in Table I. In this regard, a total 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 of 34693 nodes. Thereafter, we extract community-based features proposed in [10] from the network.

B. Results

In order to evaluate the significance of the ensemble learning methods using community-based structural features, we learn a set of classifiers from WEKA on the training examples containing the community-based features from the datasets mentioned in the previous section. We evaluate the performance of three classifiers including J48 (decision-tree) [25], IBk (k-NN using k=5 nearest neighbors) [26], and NaïveBayes [27] by considering them individually and also by using bagging, boosting and stacking over them. We use 10-fold cross validation for each classifier on the dataset to evaluate the performance.

Table III presents the performance (averaged for the two classes) of the various individual classifiers on the dataset with planted spammers, wherein it is clear that the decision tree based classifier J48 performs better than the other two classifiers.

TABLE III: PERFORMANCE OF INDIVIDUAL CLASSIFIERS

Classifier (Individual)	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F-Measure**
J48	0.963	0.075	0.963	0.963	0.963
IBk (k=4)	0.938	0.159	0.937	0.938	0.937
Naïve Bayes	0.914	0.175	0.917	0.914	0.915

Tables IV and V present the performance of the bagging and boosting ensembles over the three base classifiers, respectively for the spammer detection task. It can be clearly seen from the two Tables that the performances of J48 and IBk classifiers using bagging and boosting ensemble learning approaches is better than their individual performances. However, in case of Naïve Bayes classifier, the ensemble approaches show low performance than their individual performance for the spammer detection task using structural features.

TABLE IV: PERFORMANCE OF CLASSIFIERS USING A BAGGING ENSEMBLE

Classifier (Bagging)	TP Rate	FP Rate	Precision	Recall	F-Measure
J48	**0.969**	**0.067**	**0.969**	**0.969**	**0.969**
IBk (k=4)	0.941	0.133	0.942	0.941	0.942
NaïveBayes	0.706	0.155	0.857	0.706	0.741

TABLE V: PERFORMANCE OF CLASSIFIERS USING BOOSTING ENSEMBLE

Classifier (Boosting)	TP Rate	FP Rate	Precision	Recall	F-Measure
J48	0.966	0.082	0.966	0.966	0.966
IBk (k=4)	0.938	0.159	0.937	0.938	0.937
NaïveBayes	0.705	0.155	0.857	0.705	0.74

TABLE VI: PERFORMANCE OF THE STACKING ENSEMBLE INVOLVING ALL THREE CLASSIFIERS AND J48 AS THE META CLASSIFIER

Classifier (Stacking)	TP Rate	FP Rate	Precision	Recall	F-Measure
J48 (Meta)	0.961	0.085	0.962	0.961	0.961

Table VI presents the performance of the stacking ensemble learning approach which incorporates all three classifiers (J48, IBk, and Naïve Bayes) as its base classifiers and J48 as its meta classifier. It can be observed that its performance is lower than the best case of the other two ensemble approaches and closer to the individual performance of the J48 classifier.

From these results, it can be observed that the bagging ensemble learning approach using J48 classifier performs significantly better than the individual performance of IBk and Naïve Bayes classifiers and also better than the other two ensemble approaches, i.e., boosting and stacking for the task of spammer detection in online social networks.

V. CONCLUSION

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 spams. Ensemble learning approaches like bagging and boosting that aim to improve the performance of individual classifiers exist in literature, but they have not been extensively evaluated for the spammer detection task. Moreover, new structural features based on community structures of online social network users have also been proposed recently for spammer detection. This paper evaluates the performance of some 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 structural social network features.

ACKNOWLEDGEMENT

The authors acknowledge the support provided by the King Abdulaziz City for Science and Technology (KACST), Kingdom of Saudi Arabia under the NPST project number 11-INF1594-02.

REFERENCES

[1] 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, pp. 583-587, 2007.

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

[3] Kiran, P. and Atmosukarto, I. Spam or Not Spam-That is the question. Tech. rep., University of Washington. http://www.cs.washington.edu/homes/indria/research/spamfilter ravi indri.pdf Date of access: 1 Apr, 2014.

[4] Carpinter, J. M. Evaluating ensemble classifiers for spam filtering. Honours thesis, University of Canterbury, 2005.

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

[6] 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, pp. 137–144, 2004.

[7] Quinlan, J. Bagging, boosting and C4.5. In 13th National Conference on Artificial Intelligence. AAAI/MIT Press, 1996.

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

[9] Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. & Stamatopoulos, P. Stacking classifiers for anti-spam filtering of e-mail, In Proceedings of the Empirical Methods in Natural Language Processing, pp. 44–50, 2001.

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

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

[12] Shrivastava, N., Majumder, A. and Rastogi, R. Mining (social) network graphs to detect random link attacks, In Proceedings of the IEEE 24th International Conference on Data Engineering

(ICDE'08), Washington DC, USA, IEEE Computer Society, pp. 486–495, 2008.

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

[14] Lam, H. A Learning Approach to Spam Detection Based on Social Networks. Hong Kong University of Science and Technology, 2007.

[15] Stringhini, G., Kruegel, C. and Vigna, G. Detecting spammers on social networks, In Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC'10), NY, USA, ACM, pp. 1–9, 2010.

[16] Gan, Q. and Suel, T. Improving web spam classifiers using link structure, In Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'07), NY, USA, ACM, pp. 17–20, 2007.

[17] Ramachandran, A., Feamster, N. and Vempala, S. Filtering spam with behavioral blacklisting, In Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07), NY, USA, ACM, pp. 342–351, 2007.

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

[19] Xie, Y., Yu, F., Achan, K., Panigrahy, R., Hulten, G. and Osipkov, I. Spamming botnets: signatures and characteristics, SIGCOMM Computing Communication Review, vol. 38, no. 4, pp. 171–182, 2008.

[20] Fire, M., Katz, G. and Elovici, Y. Strangers intrusion detection-detecting spammers and fake proles in social networks based on topology anomalies, Human Journal, vol. 1, no. 1, pp. 26–39, 2012.

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

[22] Viswanath, B., Mislove, A., Cha, M. and Gummadi, K. P. On the evolution of user interaction in Facebook, In Proceedings of the Workshop on Online Social Networks, pp. 37–42, 2009.

[23] 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'11), Washington DC, USA, IEEE Computer Society, pp. 832–840, 2011.

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

[25] Quinlan, J. R.. C4.5: Programs for machine learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 1993.

[26] Aha, D. W., Kibler, D. and Albert, M. K. Instance-based learning algorithms, Machine Learning, vol. 6, no. 1, pp. 37–66, 1991.

[27] John, G. H. and Langley, P. Estimating continuous distributions in bayesian classifiers, In Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI'95), San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., pp. 338–345, 1995.