Spam Filtering in Social Networks using Regularized Deep Neural Networks with Ensemble Learning

Aliaksandr Barushka[1], Petr Hajek[1]

[1]Institute of System Engineering and Informatics, Faculty of Economics and Administration, University of Pardubice, Studentska 84,
532 10 Pardubice, Czech Republic
aliaksandr.barushka@student.upce.cz, petr.hajek@upce.cz

Abstract. Spam filtering in social networks is increasingly important owing to the rapid growth of social network user base. Sophisticated spam filters must be developed to deal with this complex problem. Traditional machine learning approaches such as neural networks, support vector machine and Naïve Bayes classifiers are not effective enough to process and utilize complex features present in high-dimensional data on social network spam. To overcome this problem, here we propose a novel approach to social network spam filtering. The approach uses ensemble learning techniques with regularized deep neural networks as base learners. We demonstrate that this approach is effective for social network spam filtering on a benchmark dataset in terms of accuracy and area under ROC. In addition, solid performance is achieved in terms of false negative and false positive rates. We also show that the proposed approach outperforms other popular algorithms used in spam filtering, such as decision trees, Naïve Bayes, artificial immune systems, support vector machines, etc.

Keywords: neural network, social networks, regularization, meta-learning

1 Introduction

Generally, spam can be defined as unwanted and unsolicited messages sent to usually a large number of recipients [1]. Spam message can be sent over multiple communication channels, such as e-mail, SMS, social networks, etc. Statistics show that a large proportion of all messages in social networks are spam messages. For instance, the study by Proofpoint, a major company specialized in cyber security, reported that during the first half of 2013 there has been a 355% growth of social spam. For every seven new social media accounts, five new spammers are detected [2].

The growing opportunities of social networks and their popularity have attracted many users. These days the base of social network users is steadily growing and considerable amount of communication is done through social networks. However, along with legitimate and useful information, inappropriate and unwanted content is also released on these networks. Indeed, spam senders target social network users as well. Moreover, business social networks like Linkedin are also affected [3]. This has serious economic and social consequences. Spam messages decrease work productivity, increase IT support related resources (help desk) and may even result in security incidents. This is why a considerable attention is given to spam filtering.

Spam messages can be filtered either manually or automatically. Obviously, manual spam filtering by identifying spam message and removing it is a time consuming task. Moreover, spam messages may contain a security threat, such as links to phishing web sites or servers hosting malware. Therefore, over a number of decades researches and practitioners have worked on improving automatic spam filtering algorithms. Machine learning techniques are particularly known to be highly accurate in detecting spam messages. There is a number of existing machine learning algorithms applied to spam filtering, including neural networks [4], support vector machines (SVMs) [5], Naïve Bayes [6], random forest [7], etc.

Spam filtering task belongs to binary classification problem, each message should be identified either as spam or ham. Besides high accuracy algorithm should also perform well when it comes to false positive ratio (legitimate message is classified as spam) to avoid situations where legitimate message is not delivered to the intended receiver. The main concept of the machine learning algorithms is to build a word list and assign a weight to each word accordingly. However, spammers tend to include common legitimate messages into the spam message in order to decrease the probability of being detected. In social network spam filtering, additional attributes are therefore used, such as those related to sender's profile and behavior in the social network.

The state-of-the-art methods in social network spam filtering has recently been surveyed by [8]. According to the survey, ensemble learning methods, such as bagging and random forest, outperform traditional single classifiers. The ensemble methods combine the predictions of several base machine learning algorithms in order to improve accuracy and robustness over single algorithms. In previous studies, ensemble methods employed traditional classifiers like decision trees to effectively filter spam messages. However, surprisingly little attention has been paid to neural networks with ensemble learning. Recent evidence showed that neural networks equipped with regularization techniques may be highly accurate in detecting e-mail and SMS spam [4]. This can be attributed to better optimization convergence and resistance to overfitting. To take advantage of these qualities, here we integrate regularized neural networks with ensemble learning methods for social network spam filtering. Using the benchmark data from the Dutch social networking site Hyves, here we show that this approach can be more effective than state-of-the-art spam filtering methods in terms of accuracy.

The rest of this paper is organized in the following way. Section 2 briefly reviews recent development in social network spam filtering. Section 3 presents the benchmark dataset. In section 4, we introduce the proposed spam filter. Section 5 shows the results of the experiments and a comparative analysis with several state-of-the-art methods used for social network spam filtering. In section 6, we conclude this paper and discuss possible future research directions.

2 Social Network Spam Filtering – A Literature Review

User base of social networks is growing over the number of years. For instance, Facebook, one of the biggest social networks in the world, grew from one billion to two billion users just in 5 years [9]. Social network spam has become a major concern of industry and academia because it may include unwanted content, such as insults, hate speech, malicious links, etc. Such messages can be seen by the recipient's followers. Moreover, they may lead to confusions and misdirection in public discussions [10]. Fighting social network spam with traditional legal methods has serious limitation, since spam messages in social networks can be sent from different countries. It is important to note that spammers may use anonymisers, making it difficult to trace them. In order to overcome this problem, several social network spam filters have recently been developed [10-22].

A Naïve Bayes classifier was proposed by [11] to detect spam in Twitter. Features related to tweet content and user behaviour were identified and used in machine learning by [12]. A hybrid approach for identifying spam profiles was proposed by [13], combining social media analytics and firefly algorithm with chaotic maps for spam detection in Twitter marketing. In addition to spam messages detection, recent studies have also considered an alternative task of social spammer detection. A large Twitter dataset was used in [14] to demonstrate that feature distributions between spammers and legitimate users are different. These feature distributions were used in a social spammer detection framework that integrated this information with a social regularization term incorporate into a classification model. In [15], the so-called "social bridges" were identified to detect spammers in Twitter. These are reported as the major supporters of malicious users and a graph-topology based classifier was used to detect such bridge linkages. Another way to tackle the issue of detecting spammers in Twitter was described in [16]. A multilayer social network was defined and the identification of spammers was based on the existence of overlapping community-based features of users represented in the form of Hypergraphs, such as structural behaviour and URL characteristics. A unified approach was proposed in [17], utilizing the fact that social spammers tend to post more spam messages. Indeed, it was shown that combining social spammer filtering with spam message filtering improves the performance of both tasks.

Although Twitter represents the most frequently used source of data, alternative social networks have also been examined. For example, data from Sina Weibo were used to study features related to message content and user behaviour in [10]. The most important features were then used in the SVM classifier for spam detection. Extreme learning machines were used by [18] on a similar dataset. A semi-supervised social media spammer filtering approach was developed in [19]. This approach outperformed traditional supervised classifiers for the spammer detection task. Similar results were obtained by for spam message detection in Hyves social network [20]. Using the same dataset, significant improvements were achieved by combining data oversampling with regularized deep multi-layer perceptron neural networks [21].

Several researchers employed feature selection and extraction methodologies to identify the most important features for social network spam filtering. The concept of rough set theory was applied by [22], concluding that the used methodology selected a smaller subset of features than those of the baseline methodologies. By considering important features of the posts and their corresponding comments, and finally applying the feature selection techniques, the method proposed in [23] selected the most effective features to detect spam using machine learning techniques. A probabilistic generative model (latent Dirichlet allocation) was proposed by [24] to detect the latent semantics from user-generated comments. Incremental learning was then used to address the issue of the changing feature space.

In summary, previous related literature attempted to overcome the problem of high-dimensional data (the curse of dimensionality) by selecting the most important features, regardless of whether content-based features or user behaviour features. This was mainly due to the risk of overfitting or poor convergence of the used classification methods. However, useful information may be hidden in higher-order features that can be extracted by using deep neural networks [21]. In fact, additional hidden layers enable the recombination of features and thus to capture higher complexity and abstraction in high-dimensional datasets [25]. Moreover, ensemble methods have become popular in social network spam detection tasks due to their capacity to reduce the risk of overfitting and variance [8].

3 Dataset

In this study, we used the dataset from Hyves, the Dutch social networking site[1]. The original dataset contained both labelled and unlabelled messages. As we use a supervised learning approach here, we excluded the unlabelled (unannotated) messages from the dataset. Since the details on this dataset can be found in [20], we provide only a brief description in this study.

The dataset includes the following types of information: message content, spam report and user information. The messages were collected from publicly accessible profile or group pages. At least two spam reports were created for each message to obtain reliable categorization of messages into “spam” or “not spam”. The user policy of Hyves was used to define spam messages. Specifically, unsolicited and promotional messages were labelled as spam.

The social network spam dataset contained 466 spam messages and 355 legitimate messages. The messages were represented as the arrays of json objects with the following fields: the annotation of the object (either spam or legitimate), anonymized IDs of the reporters of the message, anonymized ID of the author of the message, and bag of words representation of the message (an anonymized ID was assigned to each word). To represent the bag of words, we used *tf.idf* weighting scheme. The weight v_{ij} for the i-the word in the j-th message can be calculated as follows:

$$v_{ij} = (1 + \log(tf_{ij})) \times \log(N/df_i), \qquad \mathbf{(1)}$$

[1] http://ilps.science.uva.nl/framework-unsupervised-spam-detection-social-networking-sites/

where tf_{ij} represents term frequency, df_i is document frequency, and N is the number of messages. We used top 2000 words according to their weights. This number was reported to be sufficient to perform document classification in previous studies [26].

4 Methods

Since the main interest of the paper is the proposal of a social network spam filter based on deep neural network with ensemble learning, we provide a brief description of these methods in this section.

The model of the deep neural network (DNN) used in this study is the multilayer perceptron neural network with multiple hidden layers that process complex relations between the input features and output categories. However, such a structure results in the large number of connections, leading to sampling noise. Therefore, intensive adaptation of training data may result in overfitting. To address this issue, we used dropout regularization. Indeed, increased accuracy may be achieved by dropping units from the neural network, including all their incoming and outgoing connections. The dropout regularization randomly changes the given ratio of the activations' values to zero while training is performed and therefore hidden units that produce the same result are ignored. In addition, we employed rectified linear units instead of traditional sigmoidal units in order to avoid poor local minima of training error and slow optimization convergence [27]. This is done by producing partial derivative equal to one, in case the rectified linear unit is activated. It is also worth to add that these units saturate when reaching one. This might be useful when hidden activations are selected as input features for the classifier. The mini-batch gradient descent was used as a training algorithm for the DNN. Connection weights are updated for every mini-batch of training data in the following way:

$$w_{t+1} = w_t - \eta \nabla_\theta J(w_t; x^{(i:i+n)}; y^{(i:i+n)}), \qquad \textbf{(2)}$$

where w is connection weight, t denotes time, η is learning rate, J is an objective function, x^i and y^i are the inputs and output of the i-the data sample within every mini-batch, and n is the number of data samples in the mini-batch. By using the mini-batches a stable convergence can be achieved during the DNN learning.

The goal of ensemble learning algorithms is to combine the predictions of multiple base estimators constructed with the defined learning algorithm. This approach leads to better generalizability and robustness over single estimators. There are two main classes of ensemble learning algorithms, averaging and boosting. The fundamental concept of averaging is to construct several estimators independently from each other and calculate the average of their predictions. By reducing variance, the combined estimator is more accurate than single base estimator. By contrast, boosting builds the base estimators sequentially. Thus, several sequential weak models are combined to achieve a good ensemble. Here we used three conventional ensemble learning algorithms, namely Adaboost M1 [28], bagging [29] and random subspace [30].

The Adaboost M1 algorithm was developed to produce predictions with high accuracy utilizing a number of weak base learners. The algorithm keeps building the learners until there are no errors in training data predictions or the limit numbers of models is exceeded. This is done by increasing the weights of incorrectly predicted data. Finally, the predictions from all the models are combined by using a weighted majority vote to obtain the final predictions. The algorithm is defined as follows:

```
Algorithm 1: Adaboost M1
Input: The set $T$ of training data ($x^i$; $y^i$), $i$=1,2, … ,$n$; the number $B$ of base DNNs
Output: Ensemble of base DNNs {$C_b$}
For $b$=1 to $B$ {
    Construct a base DNN $C_b$ on weighted training data $T^*=(w_1T^1{}_b, w_2T^2{}_b, \dots, w_nT^n{}_b)$;
    Calculate the probability estimates of the error $\mathrm{err}_b=1/n\ \Sigma\ w_{ib}\times\xi^i{}_b$ ($\xi^i{}_b=0$ if $T^i$ classified correctly, $\xi^i{}_b=1$ otherwise);
    Set weight $c_b=0.5\times\log((1-\mathrm{err}_b)/\mathrm{err}_b)$;
    If $\mathrm{err}_b<0.5$, set $w_{ib+1}=w_{ib}\times\exp(c_b\xi^i{}_b)$;
    Otherwise, set all weights $w_{ib}=1$ and restart the algorithm;
}
Combine base DNNs $C_b$, $b$=1,2,…,$B$ into an ensemble {$C_b$} by weighted majority voting;
```

The main idea behind bagging is to construct multiple instances of black-box estimator on the random subsets of the original training data. To produce an aggregated prediction, separate predictions are then combined by using the voting procedure. Thus, the variance of base estimator is reduced by applying randomization during the process of building ensembles. The bagging algorithm employed here can be defined as follows:

```
Algorithm 2: Bagging
Input: The set $T$ of training data ($x^i$; $y^i$), $i$=1,2, … ,$n$; the number $B$ of base DNNs
Output: Ensemble of base DNNs {$C_b$}
For $b$=1 to $B$ {
    Create a bootstrapped replicate $T_b$ of the training data set $T$;
    Construct a base DNN $C_b$ on $T_b$;
}
Combine base DNNs $C_b$, $b$=1,2,…,$B$ into an ensemble {$C_b$} by simple majority voting;
```

Random subspace algorithm was proposed to handle the problem of trade-off between overfitting and achieving the highest accuracy. In fact, the random subspace algorithm is similar to bagging. The main difference is in the way they draw the random subsets of training data. In random subspace, these subsets are produced as the random subsets of the features. The random subspace algorithm applied here for social network spam filtering can be defined as follows:

```
Algorithm 3: Random subspace
Input: The set $T$ of training data ($x^i$; $y^i$), $i$=1,2, … ,$n$; the number $B$ of base DNNs
Output: Ensemble of base DNNs {$C_b$}
For $b$=1 to $B$ {
    Select an r-dimensional random subspace $T_b$ from the original training data set $T$;
    Construct a base DNN $C_b$ in $T_b$;
}
Combine base DNNs $C_b$, $b$=1,2,…,$B$ into an ensemble {$C_b$} by simple majority voting;
```

5 Experimental Results

In this section, we first describe the setting of all experiments and then we present the results in terms of four prediction measures: accuracy, area under ROC (receiver operating characteristic) curve, FN (false negative) rate and FP (false positive) rate, and F1-score. FN rate represents the percentage of spam messages incorrectly predicted as legitimate, while FP rate is the percentage of legitimate messages incorrectly predicted as spam. F1-score combines precision and recall, where precision is a fraction of messages correctly classified as spam out of all the messages the algorithm classifies as spam, whereas recall is the fraction of messages correctly classified as spam out of all the spam messages. 10-fold cross-validation was used to avoid overfitting and evaluate the prediction performance.

The DNN with ensemble learning was trained with the following setting: number of hidden layers = {1, 2, 3}, number of units in hidden layers = {10, 20, 50}, and learning rate was set to η = 0.1, size of mini-batches = 100, dropout rate for input layer = 0.2; dropout rate for hidden layers = 0.5, and number of iterations = 1000. The best setting of the DNN structure (2 hidden layers with 50 and 20 units, respectively) was obtained by using grid search. Furthermore, 10 iterations were used in the learning of Adaboost M1 ensemble, the size of each bag in bagging was 100, and the size of each subspace was 50% of all attributes in the random subspace algorithm. The learning of bagging and random subspace was also performed in 10 iterations.

To demonstrate the effectiveness of the proposed social network spam filter, we compared its performance with several methods used in previous studies for spam filtering [4-8], namely the single DNN, CNN (convolutional neural network), Naïve Bayes, *k*-NN (*k* nearest neighbour), C4.5 decision tree, MLP (multilayer perceptron), SVM (support vector machine), AIRS (artificial immune recognition system), Adaboost M1 with decision stump as base learner, and random forest. The settings of these algorithms were as follows: single DNN (the same setting as for the DNN with ensemble learning); CNN (mini-batch gradient descent algorithm with patch size 5×5 and max pool size 2×2, the remaining parameters were the same as for the DNN); *k*-NN (k = 3); C4.5 (J48 implementation with the confidence factor of 0.25 and minimum instances per leaf = 2); MLP (backpropagation with {10, 20, 50, 100} units in the hidden layer (50 units worked best), learning rate = 0.1, momentum = 0.2, and iterations = 1000); SVM (sequential minimal optimization algorithm with $C = \{2^0, 2^1, \dots, 2^6\}$ ($C = 2^2$ worked best) and polynomial kernel function); AIRS (AIRS2 parallel algorithm with affinity threshold = 0.2, clonal rate = 10, hyper-mutation rate = 2, k = 3 and stimulation threshold = 0.9); Adaboost M1 with 10 iterations and decision stump as base learner; and 100 random trees were used in random forest. All the experiments were performed in Weka 3.7.13 environment.

A brief description of the comparative methods is given as follows:

CNN uses layers together with convolving filters and filters are applied to the local features of adjacent layers. Each hidden layer consists of several feature maps (filters in a layer build a feature map sharing the same parametrization). Max-pooling is used to capture the most important features for each feature map. NB classifier utilizes information learnt from training data in order to calculate the probability of spam or legitimate class taking into consideration words found in the message. In *k*-NN, the message is classified based on the most common class in *k* neighbours. J48 algorithm generates a decision tree model, including variable classification rates built on cross-validation. MLP is a feed-forward neural network that consists of several layers of units, namely input, hidden and output layer. Each layer is directly connected to the next layer in the MLP. Backpropagation algorithm is commonly used to train this neural network. In contrast to empirical risk considered in the training of the MLP, SVM learning is based on structural risk minimization. Specifically, SVM finds the optimal separating hyperplane that represents the maximum margin between two classes and the corresponding decision boundaries are defined by the so-called support vectors. As a result, this algorithm can effectively handle high-dimensional data. AIRS is another artificial intelligence approach. This algorithm includes models resembling particular immunological processes. Finally, random forest consists of multiple tree predictors. Each of them is influenced by the values of an independently sampled random vector. Therefore, all the trees in the forest share the same distribution.

The results of the experiments are summarized in Table 1 and Table 2. The results show that the DNN with bagging performed best in terms of accuracy and area under ROC curve. Besides the DNN with ensemble learning, the single DNN, CNN and random forest also performed well. Student's paired *t*-tests were performed to compare the results statistically. The results that are statistically similar to the best performer at p=0.05 are in bold in Table 1 and Table 2.

Table 1. Results of the experiments – accuracy and area under ROC curve.

Method	Accuracy [%]	Area under ROC curve
Naïve Bayes	82.98±7.08	**0.943±0.026**
k-NN	88.37±3.71	**0.946±0.026**
C4.5	88.65±3.84	0.912±0.037
MLP	88.97±3.86	**0.942±0.027**
SVM	90.45±3.25	0.907±0.032
AIRS	80.42±7.78	0.914±0.033
Adaboost M1	89.10±3.73	0.907±0.034
Random forest	**91.94±3.09**	**0.960±0.023**
CNN	**91.11±4.52**	**0.950±0.051**
DNN	**92.27±3.04**	**0.961±0.021**
DNN with adaboost	**90.87±2.87**	**0.940±0.027**
DNN with bagging	**92.69±2.82**	**0.962±0.022**
DNN with random subspace	**92.45±3.08**	**0.961±0.023**

Table 2. Results of the experiments – FN and FP rates.

Method	FN rate [%]	FP rate [%]	F1-score
Naïve Bayes	**6.1±3.8**	30.2±19.5	0.8641±0.0461
k-NN	16.1±6.0	34.0±3.0	0.8903±0.0375
C4.5	14.0±5.5	5.7±4.3	0.8953±0.0369
MLP	11.6±17.0	6.8±6.1	0.8978±0.0364
SVM	10.8±4.9	5.1±3.8	0.9131±0.0306
AIRS	25.4±14.5	11.9±18.5	0.8415±0.0460
Adaboost M1	17.1±6.0	**2.8±2.5**	0.8952±0.0388
Random forest	10.0±4.7	5.1±3.6	**0.9264±0.0292**
CNN	**9.2±4.6**	5.2±4.2	**0.9211±0.0338**
DNN	**9.2±4.3**	3.4±3.0	**0.9292±0.0292**
DNN with adaboost	**8.8±5.4**	9.6±2.8	**0.9184±0.0273**
DNN with bagging	**9.0±3.6**	5.1±2.9	**0.9338±0.0258**
DNN with random subspace	10.1±4.5	4.2±2.4	**0.9308±0.0211**

The results in Table 2 show that the DNN with ensemble learning performed well in terms of FN rate, while relatively poorly in terms of FP rate. Overall, however, the performance was well balanced for both spam and legitimate classes. This was also confirmed with the high values of the area under ROC curve. Moreover, the DNN with bagging and DNN with random subspace performed best in terms of F1-score, indicating a balanced performance in both the precision and the recall. Notably, we obtained better results than those reported in the original study using this dataset (ROC = 0.801) [20]. However, this original study was based on a simple spam score combining messages' own and neighbour characteristics. More precisely, the best performance in terms of accuracy was achieved by using the DNN with bagging. As presented in Table 2, this can be mainly attributed to the low value of FN rate. In other words, this method performs particularly well in predicting the spam class. The DNN with adaboost performed even better in terms of this criterion but it failed to classify the legitimate messages compared with the remaining ensemble methods. Finally, the DNN with random subspace performed reasonably well with respect to both classes, resulting in high accuracy and area under ROC curve. Regarding the other comparative methods, they were significantly outperformed by the best spam filter in terms of accuracy, except random forest, CNN and DNN. Although the improvement in accuracy might not seem substantial (e.g., less than one percent over random forest), real-life spam filters achieve up to 99.9% accuracy. Every improvement is therefore highly warranted. However, some of the comparative methods were good in detecting one of the classes. Specifically, Naïve Bayes was best in detecting the social network spam class, while Adaboost M1 performed best on the legitimate class of messages. On the other hand, Adaboost M1 cannot be recommended for spam filtering in social networks due to the low value of the area under ROC curve, reflecting the poor performance on the spam class.

6 Conclusion

In this study, we demonstrated that ensemble learning algorithms with DNN as the base learner is more accurate than state-of-the-art spam filtering methods. The results show that bagging algorithm trained with DNNs achieved best results, with a high accuracy on both classes. This can be attributed to the capacity of bagging in reducing the risk of overfitting. In fact, bagging performs best with complex base learners, just like DNNs. Note that this is different from boosting where weak base learners are preferred. Moreover, reducing the number of features with random subspace does not seem to be beneficial in case of DNNs. To sum up, the combination of complex DNNs trained on random subsets of high-dimensional data seems to be an effective method for social network spam filtering. On the other hand, ensemble learning algorithms with DNN performed relatively poorly when it comes to FN rate.

Several limitations of this study need to be mentioned. Here we used the content of the messages, together with the information about the author (the number of messages authored) and reporter (the number of messages reported). However, the content of the neighbouring messages could be utilized in future studies. This would require a larger dataset to be collected. This model could also be used to predict spammers in addition to spam messages. Recent studies showed that such a combination might significantly improve the accuracy of spam and spammer filters.

The results obtained here suggest that DNNs with ensemble learning might have great potential also in other text categorization tasks, such as web-page classification, e-mail spam filtering, sentiment classification and so forth. It would also be interesting to investigate the effect of different feature selection approaches and therefore further investigation and experimentation is highly recommended. Moreover, it would be beneficial to investigate whether described methods show similar performance for spam datasets from different countries in different languages from different social networking platforms and whether localization of classification algorithm is required.

Acknowledgments. This article was supported by the by the grant No. SGS_2018_019 of the Student Grant Competition.

References

1. Cormack GV (2006) Email spam filtering: A systematic review. Foundations and Trends in Information Retrieval 1(4):335–455. doi: 10.1561/1500000006
2. Nexgate. 2013 State of Social Media Spam. http://nexgate.com/wp-content/uploads/2013/09/Nexgate-2013-State-of-Social-Media-Spam-Research-Report.pdf
3. Prieto VM, Alvarez M, Cacheda F (2013) Detecting linkedin spammers and its spam nets. International Journal of Advanced Computer Science and Applications (IJACSA) 4(9):189–199.
4. Barushka A, Hajek P (2016) Spam filtering using regularized neural networks with rectified linear units. In: Adorni G, Cagnoni S, Gori M, Maratea M (eds) Conference of the Italian

Association for Artificial Intelligence. Lecture Notes in Computer Science 10037. Springer, Cham, pp 65–75. doi: 10.1007/978-3-319-49130-1_6

5. Bhowmick A, Hazarika SM (2018) E-mail spam filtering: A review of techniques and trends. In: Kalam A, Das S, Sharma K (eds) Advances in Electronics, Communication and Computing. Lecture Notes in Electrical Engineering 443. Springer, Singapore, pp 583–590. doi: 10.1007/978-981-10-4765-7_61
6. Almeida TA, Almeida J, Yamakami A (2011). Spam filtering: how the dimensionality reduction affects the accuracy of Naive Bayes classifiers. Journal of Internet Services and Applications 1(3):183–200. doi: 10.1007/s13174-010-0014-7
7. Choudhary N, Jain AK (2017) Towards filtering of SMS spam messages using machine Learning Based Technique. In: Singh D, Raman B, Luhach A, Lingras P (eds) Advanced Informatics for Computing Research. Communications in Computer and Information Science 712. Springer, Singapore, pp 18–30. doi: 10.1007/978-981-10-5780-9_2
8. Kaur P, Singhal A, Kaur, J (2016) Spam detection on Twitter: A survey. In: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), (pp. 2570-2573). IEEE, New Delhi, pp 2570–2573.
9. Statista. https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/
10. Zheng X, Zeng Z, Chen Z, Yu Y, Rong C (2015) Detecting spammers on social networks. Neurocomputing 159:27–34. doi: 10.1016/j.neucom.2015.02.047
11. Wang AH (2010) Don’t follow me: Spam detection in Twitter. In: Proceedings of the 2010 International Conference on Security and Cryptography (SECRYPT), IEEE, pp 1–10.
12. Benevenuto F, Magno G, Rodrigues T, Almeida V (2010) Detecting spammers on twitter. In: 6th Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference (CEAS), pp 1–12.
13. Aswani R, Kar AK, Ilavarasan PV (2017) Detection of spammers in twitter marketing: a hybrid approach using social media analytics and bio inspired computing. Information Systems Frontiers 1–16. doi: 10.1007/s10796-017-9805-8
14. Shen H, Ma F, Zhang X, Zong L, Liu X, Liang W (2017) Discovering social spammers from multiple views. Neurocomputing 225:49–57. doi: 10.1016/j.neucom.2016.11.013
15. Gogoglou A, Theodosiou Z, Kounoudes T, Vakali A, Manolopoulos Y (2016). Early malicious activity discovery in microblogs by social bridges detection. In: 2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), IEEE, Limassol, pp 132–137. doi: 10.1109/ISSPIT.2016.7886022
16. Bindu PV, Mishra R, Thilagam PS (2018) Discovering spammer communities in twitter. Journal of Intelligent Information Systems 1–25. doi: 10.1007/s10844-017-0494-z
17. Wu F, Shu J, Huang Y, Yuan Z (2016) Co-detecting social spammers and spam messages in microblogging via exploiting social contexts. Neurocomputing 201:51–65. doi: 10.1016/j.neucom.2016.03.036
18. Zheng X, Zhang X, Yu Y, Kechadi T, Rong C (2016) ELM-based spammer detection in social networks. The Journal of Supercomputing 72(8):2991–3005. doi: 10.1007/s11227-015-1437-5
19. Yu D, Chen N, Jiang F, Fu B, Qin A (2017) Constrained NMF-based semi-supervised learning for social media spammer detection. Knowledge-Based Systems 125:64–73. doi: 10.1016/j.knosys.2017.03.025
20. Bosma M, Meij E, Weerkamp W (2012) A framework for unsupervised spam detection in social networking sites. In: Baeza-Yates R et al. (eds) European Conference on Information Retrieval. Springer, Berlin, Heidelberg, pp 364–375. doi: 1007/978-3-642-28997-2_31

21. Barushka A, Hajek P (2018) Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. Submitted to Applied Intelligence.
22. Dutta S, Ghatak S, Dey R, Das AK, Ghosh S (2018) Attribute selection for improving spam classification in online social networks: a rough set theory-based approach. Social Network Analysis and Mining 8(7):1–16. doi: 10.1007/s13278-017-0484-8
23. Sohrabi MK, Karimi F (2018) A Feature selection approach to detect spam in the Facebook social network. Arabian Journal for Science and Engineering 43(2):949–958. doi: 10.1007/s13369-017-2855-x
24. Song L, Lau RYK, Kwok RCW, Mirkovski K, Dou W (2017) Who are the spoilers in social media marketing? Incremental learning of latent semantics for social spam detection. Electronic Commerce Research 17(1):51–81. doi: 10.1007/s10660-016-9244-5
25. Hinton G, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov R (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580
26. Dhillon IS, Mallela S, Kumar R (2003) A divisive information-theoretic feature clustering algorithm for text classification. Journal of Machine Learning Research 3:1265–1287. doi: 10.1162/153244303322753661
27. Maas AL, Hannun AY, Ng AY (2013) Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the 30th International Conference on Machine Learning, pp 1–6.
28. Freund Y, Schapire RE (1996) Experiments with a new boosting algorithm. In: Thirteenth International Conference on Machine Learning, San Francisco, pp 148–156.
29. Breiman L (1996) Bagging predictors. Machine Learning 24(2):123–140. doi: 10.1007/BF00058655
30. Ho TK (1998) The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8):832–844. doi: 10.1109/34.709601