Computer Systems Science & Engineering
DOI:10.32604/csse.2022.020836
Article

Tech Science Press

Stochastic Gradient Boosting Model for Twitter Spam Detection

K. Kiruthika Devi[1,*] and G. A. Sathish Kumar[2]

[1]Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur, 602117, India
[2]Electronics & Communication Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, 602117, India
[*]Corresponding Author: K. Kiruthika Devi. Email: kiruthika@svce.ac.in
Received: 10 June 2021; Accepted: 23 July 2021

Abstract: In today's world of connectivity there is a huge amount of data than we could imagine. The number of network users are increasing day by day and there are large number of social networks which keeps the users connected all the time. These social networks give the complete independence to the user to post the data either political, commercial or entertainment value. Some data may be sensitive and have a greater impact on the society as a result. The trustworthiness of data is important when it comes to public social networking sites like facebook and twitter. Due to the large user base and its openness there is a huge possibility to spread spam messages in this network. Spam detection is a technique to identify and mark data as a false data value. There are lot of machine learning approaches proposed to detect spam in social networks. The efficiency of any spam detection algorithm is determined by its cost factor and accuracy. Aiming to improve the detection of spam in the social networks this study proposes using statistical based features that are modelled through the supervised boosting approach called Stochastic gradient boosting to evaluate the twitter data sets in the English language. The performance of the proposed model is evaluated using simulation results.

Keywords: Twitter; spam; stochastic gradient boosting

1 Introduction

In the previous decade, the world of internet social networks has grown tremendously. Facebook and twitter, for example, have become worldwide communication platforms. With around 330 million monthly active users and 145 million daily active users, twitter is the most popular of these several social networking services. Approximately 500 million tweets are sent out every day. The chance of receiving fake spam messages increases as the size of the network expands. The length of the tweet was originally limited to 140 characters, but it has now been increased to 280 characters. Traditional spam detection and reporting techniques are difficult to use due to the tiny size of the message. The need of the hour is for reliable ways to detect and report twitter spam.

In the literature [1,2], there are various email spam filtering techniques. Because of the shorter length, use of annotations and large number of special characters, these techniques are not appropriate for twitter

messages. Furthermore, semantic classification of twitter messages is challenging, therefore the usual methods outlined in [3] cannot be used. The traditional spam detection methods focus on recognizing and extracting user base data from a twitter account, then using machine learning algorithms to detect unauthorized users or spam campaigns [4,5]. As spamming techniques change, current solutions that rely on statistical based features will be unable to detect spammers using new spamming techniques. Some solutions to combat spamming exploit social network information using ranking schemes, which can reduce spammers' influence on legitimate users [6,7]. However, relying solely on network information, these spam detection systems make it difficult to identify legitimate users from spammers. The optimization model that outperforms previous approaches uses supervised machine learning techniques that rely on only one feature which can be either text or URL based [8–10]. As described in [11] new deep learning approaches such as convolutional neural network (CNN) and long short term neural networks (LSTM) have enabled various text representation with iterative training to get better results. The study, on the other hand, ignores the randomness of the twitter messages. To address the above said problems, we propose using stochastic gradient boosting with a randomness notion.

The major contributions of the proposed work are as follows:

1. English twitter review datasets extracted from honeypot dataset was used as the public dataset.
2. A detailed study has been done to select the features for the boosting algorithm [12–15].
3. By fitting the parameterized function for spam detection, a stochastic gradient boosting technique has been modelled.
4. The accuracy of classification has been increased by injecting randomness into the training data selection process, whereas in traditional approaches, the training data is nearly consistent.
5. The results are compared using the simulation studies against the selected literature which uses Neural network and Gradient Boosting for spam detection.

The remaining sections of this article is organized as follows: A detailed literature study on traditional spam detection techniques has been done on Section 2. Section 3 describes the data collection, feature extraction and modelling of boosting approach for spam detection. The results are presented in Section 4. Finally Section 5 presents the conclusion and aspects of future enhancement for the proposed work.

2 Related Work

The definition of spam can be formulated as follows: "Spam is an undesirable information that contain improper messages that may mislead the readers" [16]. Normally, spam communications are tough to foresee since spammers spoof authenticated users' information [17]. Several research studies have been conducted to aid in the detection of spam communications in both emails and other social networking sites. In this section, we will go through some of the most prevalent ways to spam detection that are relevant to our proposed framework.

Convolutional Neural Networks (CNN) is a type of deep learning technique that is widely utilized in natural language processing. The application of CNN to false information detection has been extended by the researchers. The study in [18] presents a CNN- based message classification approach for detecting fake news in twitter feeds. The authors in [19] combined CNN and ensemble neural networks to detect fake information on twitter. Yang et al. [20] used CNNs with text and images to identify fake content. The collected features were from the image and text. The results validate that this method is efficient to detect false information.

Researchers frequently utilize hybrid techniques for spam detection, which are created by combining any two similar deep learning architectures. In [21], the use of recurrent convolutional neural network (RCNN) to learn the contextual information has been discussed. The same CRNN model proposed in

[22] attempts to extract data from the message such as captions and keywords. The collected features were used to generate the training data set. All of these methods use a deterministic training data set that stays the same throughout the cycle. Due to unpredictable nature of twitter tweets, randomization in the data selection process may negatively impact performance. As a result, the suggested method uses a stochastic model to classify messages. The recursive neural network (RxNN) is one of the efficient models for the spam detection because of its hierarchical architecture and the use of compositional vectors for training.

In [23] the authors proposed method for extracting information from tweets that are discriminating. In general, the features vary for different kinds of rumors. This method proves to be efficient in terms of identifying random spam tweets. Many works have used multi-layer graphical model with hidden units called Deep Belief Network (DBN) to detect spam. The study in [24] employed a DBN based method to identify malicious material in personal networks, which may be extended to public domains as well. DBNs are non-supervisory in nature and has consistently outperformed restricted supervised techniques.

A deep learning model has been introduced for detecting spammers in the twitter network in [25]. To increase the performance of spammer detection in the twitter network, the techniques were applied to tweets as well as the meta-data of twitter users. The main drawbacks of using neural networks for spam detection are the high complexity and increased computational cost.

Himank introduced a method for identifying spam in the twitter network in real time in [26]. The classification of spammers is based on user and text-based features. The performance evaluation was carried out using the machine learning techniques such as Support vector machine (SVM), Neural network, Random forest and Gradient boosting. The neural network was able to reach an accuracy of 91.65%.

In our suggested model, we apply boosting algorithms with great accuracy in classification issues. In the literature, there are numerous boosting methods, however gradient boosting is the most reliable and efficient model. The suggested method employs stochastic gradient boosting [27], a variant of classical gradient boosting. This approach uses non-replacement random subsamples of training sets.

3 Proposed Model for Spam Detection

In this section we put forth a detailed modeling for spam detection based on boosting algorithm. It is a well validated observation that the majority of spam tweets contain a URL that redirect users [28,29]. In order to proceed with modeling, we extract several features from the honey pot dataset. Due to the random character of spam messages, the feature selection procedure is not easy. We make every attempt to accommodate the most popular features which appear in the majority of tweets.

3.1 Feature Selection

Various methods have been described in for extracting from linguistic datasets. The efficiency of classification is determined by the precision and number of features. Because spam attacks are unpredictable, defining features for any given data collection is not an easy operation. In our proposed approach we have identified 15 features based on the literature in [30]. The computational complexity of any classification technique can be reduced by reducing the size of the feature set with increased accuracy, making it viable to execute for a large population of tweets.

Tab. 1 shows the features that were extracted. The extracted features are classified into two categories: the first category collects information regarding the user and their features, such as account age, followers and so on and is referred to as account based features. Second, the features associated with the tweet that is being investigated for detection are collected. Hashtags, Retweets, Embedded URLS and other elements are among them and they have been categorized as content based features.

Table 1: Features and their definition

Feature Type	Feature Name	Feature Definition
User Based Features	User _age	Age of account user
	Account_age	Age of account
	Num_followers	Number of followers
	Num_Following	Number followed by the user
	Num_favorites	Number of favorites received by the user
	Num_groups	Number of membership groups
	Num_liked	Number of tweets and groups liked
Content Based Features	Num_Tweets	Number of tweets by the user
	Num_retweets	Number of retweets for the tweet
	Num_tweet favorites	Number of favorites the tweet received
	Num_hashtags	Number of hashtags in the tweet
	Num_Usermentioned	Number of users mentioned the tweet
	Num_URLs	Number of URLs in the tweet
	Num_ Characters	Number of characters in this tweet
	Num_ Special Characters	Number of special characters

3.2 Stochastic Gradient Boosting

Gradient boosting generates the final conclusion by combining the predictions from multiple instances. Each subtree's nodes have thier own set of characteristics, and they aren't all the same. This boosting can be substantially improved by introducing randomness into the feature selection process, which is referred to as stochastic gradient boosting [31].

For a given input data set 'x' with 'N' Values and 'M' features there is an in -deterministic response 'Y'. The goal of the algorithm is to develop a function F*(X) that transfers the input data value (X) to the output response of spam or non-spam (Y) given a training sample of $\{y_i, x_i\}_1^N$ of known {y, x} data values. In the intended result, there is always some loss $(y,f(x)$.

The mapping function F*(X) can be calculated as follows:

$$F^*(x) = \underset{F(X)}{\operatorname{argmin}} E_{y,x} \backslash \text{phi}(y,f(x)) \tag{1}$$

The mapping function in Eq. (1) can be approximated by an additive expansion:

$$F(X) = \sum_{m=1}^{M} \beta_m h(x: a_m) \tag{2}$$

where 'm' is the set of features associated on every data set and 'a' is the parameter value of the feature 'm', $h(x: a_m)$ is the matrix of feature values for any tweet 'x' where $x \in X$ and β_m is the expansion co-efficient.

The algorithm starts with initial guess $F_0(X)$ and the expansion coefficients $\{\beta_m, a_m\}$ are fit into the initial training data sample and hence for m = 1, 2,…. M

$$(\beta_m, a_m) = \arg\min_{\beta,a} \sum_{i=1}^{N} \varphi(y_i\, F_{m-1}(x_i)) + \beta h(x_i : a) \tag{3}$$

and

$$F_m(X) = F_{m-1}(X) + \beta_m h(x : a_m) \tag{4}$$

The gradient boosting approach solves the Eq. (3) by least square approximations and hence

$$a_m = \underset{a,\rho}{\operatorname{argmin}} \sum_{i=1}^{N} \widetilde{y_{im}} - \rho h(x_i : a)]^2 \tag{5}$$

where ρ is arbitrary value and $\widetilde{y_{im}}$ is the residual data and can be formulated as a differentiable function

$$\widetilde{y_{im}} = -\left[\left[\frac{\partial \varphi(y_i, F(x_i))}{\partial F(x_i))}\right]\right] \tag{6}$$

For the given parameters $h(x_i : a)$ the optimal value of the expansion coefficient β_m is

$$\beta_m = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \varphi(y_i, F_{m-1}(x_i)) + \beta h(x_i : a_m) \tag{7}$$

The value $h(x : a)$ is the terminal node of the decision tree. At each iteration the tree partitions the input data set 'X' in to 'L' disjoint sub trees $\{R_{lm}\}_{l=1}^{L}$ and predicts a response for each iteration as follows:

$$h\big(X : \{R_{lm}\}_{l=1}^{L}\big) = \sum_{l=1}^{L} \overline{y_{lm}} 1(x \in R_{lm}) \tag{8}$$

where $\overline{y_{lm}} = \underset{x_i \in R_{lm}}{\operatorname{mean}} \widetilde{y_{im}}$ is the mean in each region.

The sub trees can be solved independently at each region R_{lm} by the corresponding terminal node 'l' constructed for the 'm^{th}' feature. Based on the above formulations the solution to Eq. (7) reduces to a simple location based estimate which is given as follows

$$\gamma_{lm} = arg \min_{\gamma} \sum_{x_i \in R_{lm}} \varphi(y_i, F_{m-1}(x_i)) + \gamma \tag{9}$$

The mapping function $F_{m-1}(X)$ is updated separately in each region

$$F_m(X) = F_{m-1}(X) + v\gamma_{lm} 1(\, x \in R_{lm}) \tag{10}$$

where 'v' is the shrinkage parameter $0 < v < 1$ controls the learning rate of the algorithm.

In the gradient procedure modelled we incorporate randomness as part of the model. The subsample of training data for each iteration is drawn at random from the entire available data set. Let $\{y_i, x_i\}_1^N$ be the training data set and $(\pi(i))_1^N$ is the random permutation of integers {1, 2…N}. Now the random subsample $\widetilde{N} < N$ is given by $\left\{y_{\pi(i)}, x_{\pi(i)}\right\}_1^{\tilde{N}}$. The stochastic gradient boosting algorithm can now be written as follows:

4 Simulation Studies

The proposed work had aimed to detect spams in twitter messages using Stochastic gradient boosting method (SGBM). The proposed model was developed using MATLAB simulation environment. We have increased the training and testing from 100 to 10000 and evaluated the performance of the proposed model against the ground works. Tab. 2 lists the training and testing data samples with different spam ratios.

Algorithm 1: Stochastic gradient tree boosting

$$F_0(X) = \arg\min_{\gamma} \sum_{i=1}^{N} \varphi(y_i, \gamma)$$

For m = 1 to M do

$$\{\pi(i)\}_1^N = \text{rand\_perm}\{i\}_1^N$$

$$\widetilde{y_{\pi(i)m}} = -\left[\left[\frac{\partial \varphi\left(y_{\pi(i)}, F\left(x_{\pi(i)}\right)\right)}{\partial F\left(x_{\pi(i)}\right)}\right]\right] \quad i = 1 \ldots\ldots \tilde{N}$$

$$\{R_{lm}\}_1^L = L - \text{terminal node tree}\left(\left\{\widetilde{y_{\pi(i)m}}, x_{\pi(i)}\right\}\right)_1^{\tilde{N}}$$

$$\gamma_{lm} = \arg\min_{\gamma} \sum_{x_{\pi(i)} \in R_{lm}} \varphi\left(y_{\pi(i)}, F_{m-1}\left(x_{\pi(i)}\right)\right) + \gamma$$

$$F_m(X) = F_{m-1}(X) + v\gamma_{lm} 1(x \in R_{lm})$$

End for

Table 2: The training and testing dataset ratios

Training Data			Testing Data		
Data Set	**No. of spam tweets**	**No. of non spam tweets**	**Data set**	**No. of spam tweets**	**No. of non spam tweets**
1	1000	1000	1	100000	100000
2	10000	10000	2	100000	100000
3	100000	100000	3	100000	100000

4.1 Evaluation Metrics

The measure of performance is evaluated using some metrics like Accuracy, True Positive Rate (TPR), False Positive Rate (FPR) and F-measure.

4.1.1 True Positive Rate (TPR)

The TPR, which is also called as recall indicates the ratio of correctly identified spams to the total number of actual spams.

$$TPR = \frac{TP}{(TP + FN)}$$

4.1.2 False Positive Rate (FPR)

The FPR refers to the proportion of non-spams incorrectly classified as spams in the total number of actual non-spams.

$$\text{FPR} = \frac{\text{FP}}{(\text{FP} + \text{TN})}$$

4.1.3 Accuracy

The accuracy is the percentage of correctly identified tweets (both spams and non-spams) in the total number of examined tweets.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}$$

4.1.4 Precision

The precision is defined as the ratio of correctly classified spams to the total number of tweets that are classified as spams.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}$$

4.1.5 F-measure

The F-Measure is a measure of model accuracy of the system. It is defined as the weighted harmonic mean of precision and recall.

$$\text{F} - \text{Measure} = 2.\frac{\text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})}$$

where TP = True Positive is an outcome where the model *correctly* predicts the *positive* class.

TN = **True Negative** is an outcome where the model *correctly* predicts the *negative* class.

FP = **False Positive** is an outcome where the model *incorrectly* predicts the *positive* class.

FN = **False Negative** is an outcome where the model *incorrectly* predicts the *negative* class.

4.2 Results and Discussions

The proposed work is compared with Gradient Boosting method (GBM) and Convolutional neural network (CNN). Boosting algorithms perform well compared to the convolutional neural network. The results of the models are compared in terms of the evaluation metrics accuracy, FPR, TPR and F-measure. Three data sets were used with the spam to non-spam ratio of (1:1). The average value of the evaluation metrics for all three methods has been listed in Tab. 3.

Table 3: Comparison of evaluation metrics

Evaluation Metric (in %)	GBM	CNN	SGBM
Accuracy	83.19	78.75	89.12
TPR	78	75	83
FPR	20	25	15
Precision	83.25	78.47	94.05
F-measure	79.07	70.55	79.89

Fig. 1 shows a comparison of detection accuracy for all three techniques. As we can see, the classification accuracy of all three methods improves as the size of the training datasets grows from 1 k to 100 K. The stochastic gradient boosting approach has a greater detection accuracy than the other two techniques, as shown in the graph.

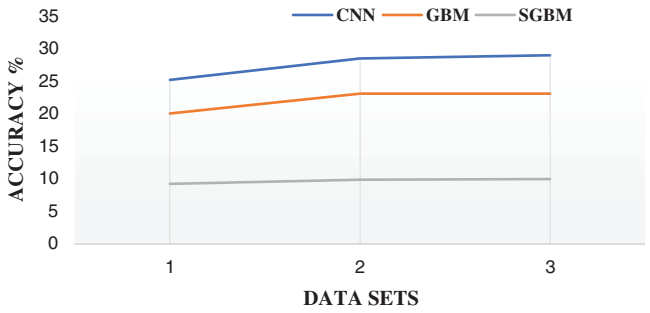

Figure 1: Detection accuracy

Fig. 2(c) shows that as the dataset value increased from 1 k, the F-measure value decreased for all of the algorithms studied. All algorithms on Dataset 3 had lower F-measure values than those on dataset1. Although, as demonstrated in Figs. 2(a) and 2(b), increasing the size of the training dataset to 100 k contributed to a minor increase of the FPR values and growth in TPR values.

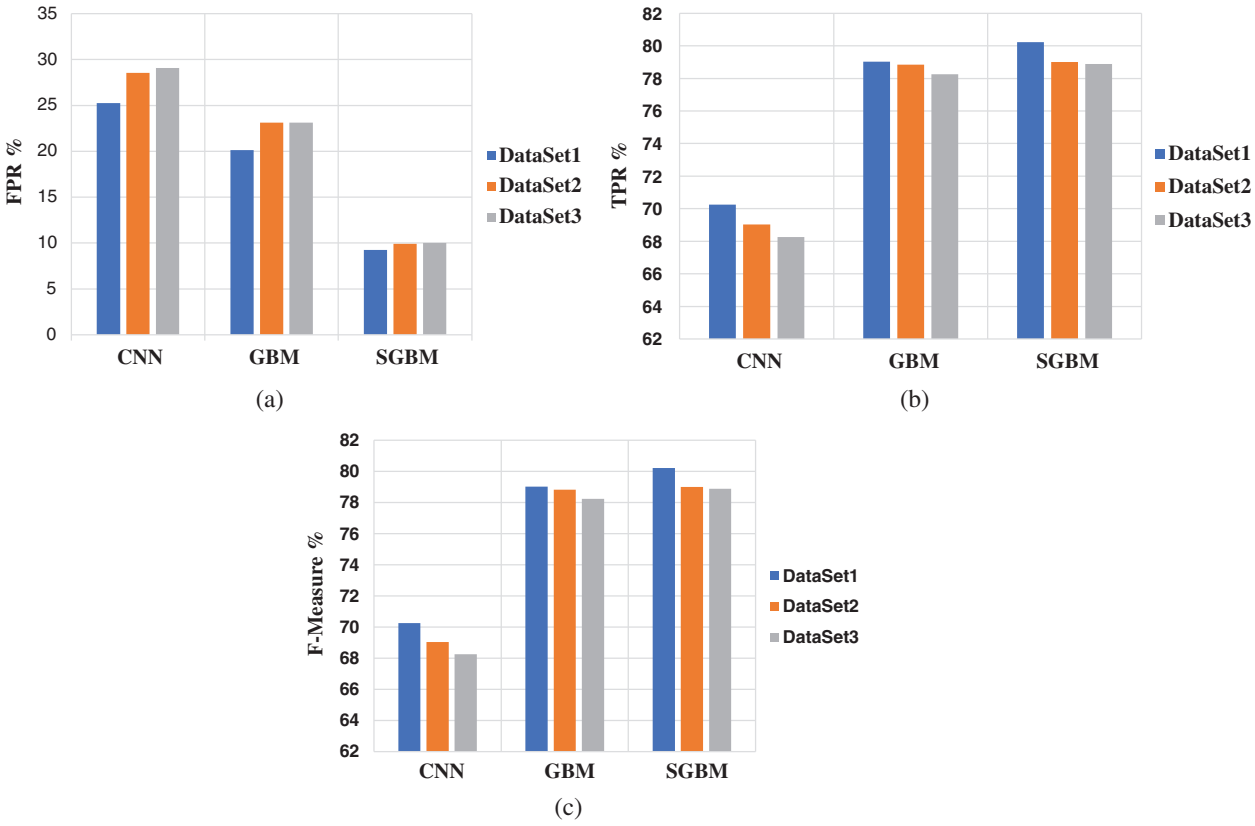

Figure 2: (a) The FPR values on dataset 1 to 3. (b) The TPR values on dataset 1 to 3 (c) The F-measure values on dataset 1 to 3

4.3 Comparative Analysis

A comparative analysis is done for the proposed method with one of the methods for detecting spammers proposed in [32]. The approach presented in [32] is compared with our proposed stochastic gradient boosting method. Fig. 3 shows the performance comparison of the proposed method in [32] in terms of accuracy. Fig. 3 reveals that the proposed method perform well in terms of accuracy.

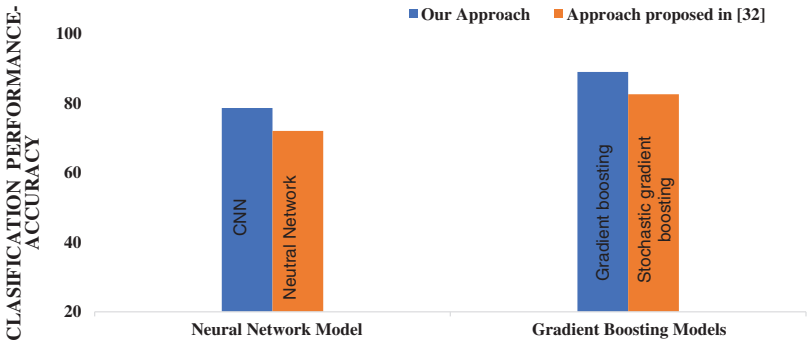

Figure 3: Performance comparison results of the proposed approach with paper [32]

5 Conclusion

In the proposed methodology, we reviewed the conventional neural network design with two boosting methods and their effectiveness in terms of spam detection. In order to examine their performance in recognizing twitter spams in terms of accuracy, TPR/FPR and F-measure, the algorithms were tested in various scenarios by increasing the volume of training data while keeping the spam-to-non-spam ratio constant. The stochastic gradient boosting approach is optimal in terms of all performance metrics, according to the findings of the studies. As a future development, we can investigate the performance of these algorithms with dynamic spam to non-spam ratio and growing tweet volumes.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi *et al.,* "Machine learning for email spam filtering: Review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, pp. e01802, 2019.

[2] H. Shen and Z. Li, "Leveraging social networks for effective spam filtering," *IEEE Transactions on Computers*, vol. 63, no. 11, pp. 2743–2759, 2013.

[3] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed, M. A. Al-Garadi *et al.,* "Email classification research trends: Review and open issues," *IEEE Access*, vol. 5, pp. 9044–9064, 2017.

[4] X. Zheng, Z. Zeng, Z. Chen, Y. Yu and C. Rong, "Detecting spammers on social networks," *Neurocomputing*, vol. 159, no. 1, pp. 27–34, 2015.

[5] Z. Chu, I. Widjaja and H. Wang, "Detecting social campaigns on Twitter," *Proc. of the 10th Int. Conf. Applied Cryptography and Network Security*, Singapore, vol. 7341, 2012, pp. 455–472.

[6] S. Ghosh, B. Vishwanath, F. Kooti, N. K. Sharma, G. Korlam *et al.,* "Understanding and combating and link farming in the twitter social network," in *Proc. of the 21st Int. Conf. on World Wide Web*, New York, United States, 2012, pp. 61–70.

[7] C. Yang, R. Harkreader, J. Zhang, S. Shin and G. Gu, "Analyzing spammers social networks for fun and profit: A case study of cybercriminal ecosystem on twitter," in *Proc. of the 21st Int. Conf. on World Wide Web*, New York, United States, 2012, pp. 71–80.

[8] Y. Zhu, X. Wang, E. Zhong, N. N. Liu, H. Li *et al.,* "Discovering spammers in social networks," in *Proc. of the National Conf. on Artificial Intelligence*, Cambridge, MA, United States, 2012, pp. 171–177.

[9] X. Hu, J. Tang, Y. Zhang and H. Liu, "Social spammer detection in microblogging," in *Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence*, Beijing, China, 2013, pp. 2633–2639.

[10] X. Hu, J. Tang and H. Liu, "Online social spammer detection," in *Proc. of the 28th AAAI Conf. on Artificial Intelligence*, Quebeccity, Quebec, Canada, 2014, pp. 59–65.

[11] G. Jain, M. Sharma and B. Agarwal, "Spam detection in social media using convolutional and long short term memory neural network," in *Annals of Mathematics and Artificial Intelligence*, vol. 85, no. 1, pp. 21–44, 2019.

[12] B. Wang, A. Zubiaga, M. Liakata and R. Procter, "Making the most of tweet-inherent features for social spam detection on Twitter," 2015, arXiv preprint arXiv 2015: 1503.07405.

[13] K. Lee, J. Caverlee and S. Webb, "Uncovering social spammers: social honeypots + machine learning," in *Proc. of the 33rd Int. ACM SIGIR Conf. on Research and Development in Information*, Geneva, Switzerland, 2010, pp. 435–445.

[14] A. H. Wang, "Don't follow me: Spam detection in Twitter," in *Proc. of the Int. Conf. on Security and Cryptography (SECRYPT)*, Athens, Greece, 2010, pp. 1–10.

[15] A. A. Amleshwaram, N. Reddy, S. Yadav, G. Guofei and Y. Chao, "CATS: Characterizing automation of twitter spammers," in *Proc. of the 5th Int. Conf. on Communication Systems and Networks (COMSNETS)*, Bangalore, India, 2013, pp. 1–10.

[16] C. M. Yilmaz and A. O. Durahim, "A semi supervised spam review detection framework," in *IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining*, Barcelona, Spain, 2018, pp. 306–313.

[17] O. Çıtlak, M. Dörterler and I. A. Doğru, "A survey on detecting spam accounts on twitter network," *Social Network Analysis and Mining*, vol. 9, no. 1, pp. 155, 2019.

[18] Y. Chen, Z. H. Liu and H. Y. Kao, "Convolutional neural networks for stance detection and rumor verification," in *Proc. of the 11th Int. Workshop on Semantic Evaluation*, Vancouver, Canada, 2017, pp. 465–469.

[19] S. Kumar, R. Asthana, S. Upadhyay, N. Upreti and M. Akbar, "Fake news detection using deep learning models: A novel approach," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 2, pp. e3767. 2019.

[20] Y. Yang, L. Zheng, J. Zhang, Q. Cui, Z. Li *et al.,* "Convolutional neural networks for fake news detection," arXiv, preprint: 1806.00749, 2018.

[21] X. Lin, X. Liao, T. Xu, W. Pian and P. F. Wong, "Rumor detection with hierarchical recurrent convolutional neural network," in *CCF Int. Conf. on Natural Language Processing and Chinese Computing*, Dunhuang, China, 2019, pp. 338–348.

[22] Y. Xu, C. Wang, Z. Dan, S. Sun, F. Dong *et al.,* "Deep recurrent neural network and data filtering for rumor detection on sina weibo," *Symmetry*, vol. 11, no. 11, 2019.

[23] J. Ma, W. Gao and K. F. Wong, "Rumor detection on twitter with tree structured recursive neural networks," *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 1980–1989, 2018.

[24] L. Wei, D. Gao and C. Luo, False data injection attacks detection with deep belief networks in smart grid. In: *Chinese Automation Congress (CAC)*. Xi'an, China, 2018, pp. 2621–2625.

[25] Z. Alom, B. Carminati and E. Ferrari, "A deep learning model for Twitter spam detection," *Online Social Networks and Media, Elsevier*, vol. 18, pp. 1–12, 2020.

[26] H. Gupta, M. S. Jamal, S. Madiseety and M. S. Desarkar, "A framework for real-time spam detection in twitter," in *Int. Conf. on Communication Systems & Networks*, Bengaluru, India, 2018.

[27] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics and Data Analysis, Elsevier*, vol. 4, no. 38, pp. 367–378, 2002.

[28] M. Egele, G. Stringhini, C. Kruegel and G. Vigna, "COMPA: Detecting compromised accounts on social networks," in *Proc. Network and Distributed System Security Sym.*, San Diego, CA United States, 2013.

[29] X. Zhang, S. Zhu and W. Liang, "Detecting spam and promoting campaigns in the Twitter social network," in *Proc. IEEE 12th Int. Conf. Data Mining*, Brussels, Belgium, 2012, pp. 1194–1199.

[30] C. Chen, J. Zhang, X. Chen, Y. Xiang and W. Zhou, "6 million spam tweets: A large ground truth for timely Twitter spam detection," in *Proc. IEEE Int. Conf. on Communications (ICC)*, London, UK, 2015, pp. 7065–7070.

[31] G. Lin, N. Sun, S. Nepal, J. Zhang and Y. Xiang, "Statistical Twitter spam detection demystified: Performance, stability and scalability," in *IEEE Access*. vol. 5, pp. 11142–11154, 2017.

[32] G. Himank, M. S. Jamal, M. Sreekanth and M. S. Desarkar, "A framework for real-time spam detection in Twitter," in *10th Int. Conf. on Communication Systems & Networks*, Bangalore, India, 2018, pp. 381–384.