Extreme Learning Machine for Spammer Detection and Fake User Identification from Twitter

P. Hemalatha[1], N. Ragapriya[1], V. Sanjana[1], K. Shiva Bhavani[2]

[1]UG Student, [2]Assistant Professor, [1,2]Department of Computer Science and Engineering

[1,2]Malla Reddy Engineering College for Women (UGC-Autonomous), Maisammaguda, Secunderabad, Telangana, India

ABSTRACT

Social networking sites engage millions of users around the world. The users' interactions with these social sites, such as Twitter and Facebook have a tremendous impact and occasionally undesirable repercussions for daily life. The prominent social networking sites have turned into a target platform for the spammers to disperse a huge amount of irrelevant and deleterious information. Twitter, for example, has become one of the most extravagantly used platforms of all times and therefore allows an unreasonable amount of spam. Fake users send undesired tweets to users to promote services or websites that not only affect legitimate users but also disrupt resource consumption. Moreover, the possibility of expanding invalid information to users through fake identities has increased that results in the unrolling of harmful content. Recently, the detection of spammers and identification of fake users on Twitter has become a common area of research in contemporary online social Networks (OSNs). This project proposes the detection of spammers and fake user identification on Twitter data using deep learning mechanism called extreme learning machine (ELM) and compared the obtained results with various machine learning algorithms like random forest, naevi bayes and support vector machine. Moreover, a taxonomy of the Twitter spam detection approaches is presented that classifies the techniques based on their ability to detect: (i) fake content, (ii) spam based on URL, (iii) spam in trending topics, and (iv) fake users. The presented techniques are also compared based on various features, such as user features, content features, graph features, structure features, and time features. We are hopeful that the presented study will be a useful resource for researchers to find the highlights of recent developments in Twitter spam detection on a single platform.

Keywords: Online social media, spam user, fake user identification, machine learning, extreme learning machine.

1. INTRODUCTION

In recent years, MSNs such as Twitter, Facebook and Sina Weibo have become important platforms for people to obtain information, spread information and make friends. Twitter's monthly active users (MAU) were 200 million in 2012, and the figure rose to 328 million in 2017, with 20 million tweets being posted every hour. While MSNs enrich people's lives, some security issues have emerged. Attackers spread attacks through MSNs, such as phishing, [1] drive-by download, malicious code injection and so on. According to new research, up to 15 percent of Twitter accounts are in fact bots rather than people. Malicious URLs are one of the most common methods used by attackers to initiate cyberattacks [2]. Attackers trick users into clicking malicious URLs, clicking pictures containing malicious URLs, scanning QR codes with malicious URLs, and so on by disguising themselves as well-known accounts, advertisements of discounted merchandise, or by using mutual trust between friends. In these ways, attackers lure victims to a phishing website for phishing attacks, or embed malicious software in the victim's computer to control the target host or perform an APT attack, which will cause huge losses to individuals, businesses, governments and organizations. Many MSNs use blacklist techniques to filter URLs sent by users, such as Google Safe Browsing, Phishing Tank, URIBL, and so on. However, there is often a delay in blacklist technology, and research shows that 90 percent of victims click on malicious URLs before they are blacklisted [3]. In order to provide users with a secure MSN environment, researchers have proposed many strategies to deal with online social network attacks. Existing detection methods are mainly divided into two categories. The first category includes detection algorithms based on the relation graph of social networks. Many kinds of relations exist in social networks, so researchers use relations in social networks to build a social graph. By analyzing the characteristics of the user's location in the graph, a detection algorithm can be designed based on the graph to identify suspicious messages or users. The second category includes detection methods based on machine learning algorithms. Researchers extract features from social network data such as users' personal information, social behaviors, relationships with friends, message content and so on, then use machine learning algorithms to train classifiers to identify malicious messages or users.

Spam has been known as a major problem since long, but its impact on the global network infrastructure has now reached epidemic proportions. In the earliest days of spam, users could simply delete the offending messages. Later, when spam became more common, several client-side spam filtering tools became available, but they were often unreliable: users had to scan alleged spam to ensure that no important messages were deleted by mistake, with an increasing loss of time. Due to customers' complaints, governments started to contemplate anti-spam legislation, while several companies began offering spam filtering products to mail server operators and ISPs. When these countermeasures first reached the market, it seemed that simple economics could support a rapid eradication of spam: filtering out 95% of spam would suffice to increase the spammers' cost to reach the same audience by a factor of 20. It looked therefore reasonable to assume that high-accuracy filters could put a definitive end to spam, as few spammers had profit margins big enough to meet the cost increase. Unfortunately, things went quite differently: while most commercial anti-spam filters claim a much higher success rate than 95% in identifying spam, a huge amount of it still winds up in users' in-boxes, even when client-side and server-side filters are used in conjunction. It may be argued that this lack of success in the war against spam is partly due to the elusive nature of the notion, which is difficult to identify by means of a software program. Of course, many messages leave no doubt: drugs, pornography, fraud and viruses now vastly outweigh the occasional unsolicited product or service sales pitch. However, there are many borderline cases where what is spam for a user could be useful information for the next person, and it may seem unwise to curb the potential of email as a mass communication channel in the spur of indignation against spam. Recently, some approaches based on the development of a P2P network for the collaborative sharing of knowledge about spam between users have been proposed. While these approaches represent a step toward the design of a P2P collaborative spam filtering solution, they do not pay adequate attention to the aspects of message confidentiality and robustness against attacks.

2. LITERATURE SURVEY

Feng et. al [4] proposed a multistage and elastic detection framework based on deep learning, which sets up a detection system at the mobile terminal and the server, respectively. Messages are first detected on the mobile terminal, and then the detection results are forwarded to the server along with the messages. We also design a detection queue, according to which the server can detect messages elastically when computing resources are limited, and more computing resources can be used for detecting more suspicious messages. We evaluate our detection framework on a Sina Weibo dataset. The results of the experiment show that our detection framework can improve the utilization rate of computing resources and can realize real-time detection with a high detection rate at a low false positive rate. Damiani et. al [5] proposed a decentralized privacy-preserving approach to spam filtering. Our solution exploits robust digests to identify messages that are a slight variation of one another and a structured peer-to-peer architecture between mail servers to collaboratively share knowledge about spam.

Hu et. al [6] investigated whether sentiment analysis can help spammer detection in online social media. In particular, we first conduct an exploratory study to analyse the sentiment differences between spammers and normal users, and then present an optimization formulation that incorporates sentiment information into a novel social spammer detection framework. Experimental results on real-world social media datasets show the superior performance of the proposed framework by harnessing sentiment analysis for social spammer detection. Mateen et. al [7] proposed a hybrid technique which uses content-based as well as graph-based features for identification of spammers on twitter platform. We have analysed the proposed technique on real Twitter dataset with 11k uses and more than 400k tweets approximately. Our results show that the detection rate of our proposed technique is much higher than any of the existing techniques.

Wu et. al [8] proposed a hybrid semisupervised learning model titled hybrid PU-learning-based spammer detection (hPSD) for spammer detection to leverage both the users' characteristics and the user-product relations. Specifically, the hPSD model can iteratively detect multitype spammers by injecting different positive samples, and allows the construction of classifiers in a semisupervised hybrid learning framework. Comprehensive experiments on movie dataset with shilling injection confirm the superior performance of hPSD over existing baseline methods. The hPSD is then utilized to detect the hidden spammers from real-life Amazon data. A set of spammers and their underlying employers are successfully discovered and validated. These demonstrate that hPSD meets the real-world application scenarios and can thus effectively detect the potentially deceptive review writers. Thomas et. al [9] presented Monarch, a real-time system that crawls URLs as they are submitted to web services and determines whether the URLs direct to spam. We evaluate the viability of Monarch and the fundamental challenges that arise due to the diversity of web service spam. We show that Monarch can provide accurate, real-time protection, but that the underlying characteristics of spam do not generalize across web services. In particular, we find that spam targeting email qualitatively differs in significant ways from spam campaigns targeting Twitter. We explore the distinctions between email and Twitter spam, including the abuse of public web hosting and redirector services. Finally, we demonstrate Monarch's scalability, showing our system could protect a service such as Twitter -- which needs to process 15 million URLs/day -- for a bit under $800/day.

Wang et. al [10] proposed a novel concept of a heterogeneous review graph to capture the relationships among reviewers, reviews and stores that the reviewers have reviewed. We explore how interactions between nodes in this graph can reveal the cause of spam and propose an iterative model to identify suspicious reviewers. This is the first time such intricate relationships have been identified for review spam detection. We also develop an effective computation method to quantify the trustiness of reviewers, the honesty of reviews, and the reliability of stores. Different from existing approaches, we don't use review text information. Our model is thus complementary to existing approaches and able to find more difficult and subtle spamming activities, which are agreed upon by human judges after they evaluate our results. Shehnepoor et. al [11] proposed a novel framework, named NetSpam, which utilizes spam features for modeling review data sets as heterogeneous information networks to map spam detection procedure into a classification problem in such networks. Using the importance of spam features helps us to obtain better results in terms of different metrics experimented on real-world review data sets from Yelp and Amazon Web sites. The results show that NetSpam outperforms the existing methods and among four categories of features, including review-behavioral, user-behavioral, review-linguistic, and user-linguistic, the first type of features performs better than the other categories.

Masood et. al [12] performed a review of techniques used for detecting spammers on Twitter. Moreover, a taxonomy of the Twitter spam detection approaches is presented that classifies the techniques based on their ability to detect: (i) fake content, (ii) spam based on URL, (iii) spam in trending topics, and (iv) fake users. The presented techniques are also compared based on various features, such as user features, content features, graph features, structure features, and time features. We are hopeful that the presented study will be a useful resource for researchers to find the highlights of recent developments in Twitter spam detection on a single platform.

3. EXISTING SYSTEM

Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

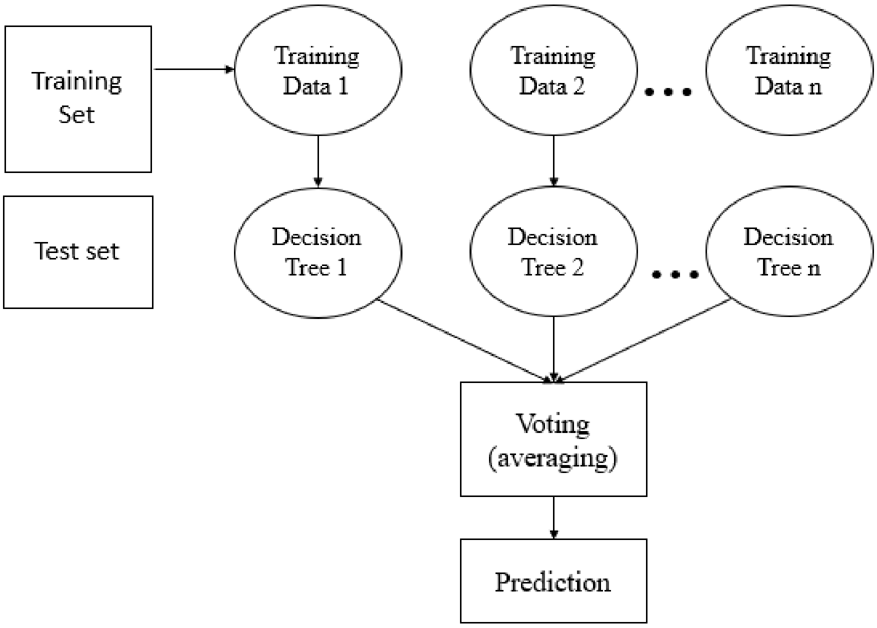


Fig. 1: Random Forest algorithm.

Random Forest algorithm

Step 1: In Random Forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

Important Features of Random Forest

- **Diversity-** Not all attributes/variables/features are considered while making an individual tree, each tree is different.
- **Immune to the curse of dimensionality-** Since each tree does not consider all the features, the feature space is reduced.
- **Parallelization-**Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.
- **Train-Test split-** In a random forest we don't have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.
- **Stability-** Stability arises because the result is based on majority voting/ averaging.

Drawbacks of existing system

- Although random forests can be an improvement on single decision trees, more sophisticated techniques are available. Prediction accuracy on complex problems is usually inferior to gradient-boosted trees.
- A forest is less interpretable than a single decision tree. Single trees may be visualized as a sequence of decisions.
- A trained forest may require significant memory for storage, due to the need for retaining the information from several hundred individual trees.

4. PROPOSED SYSTEM

Twitter Spam dataset

UtkML's Twitter Spam Detection Competition

Twitter spam is unwanted content manifesting in many ways! Including bulk messages, profanity, insults, hate speech, malicious links, and fraudulent reviews. Let's tackle this problem by building a classifier to detect when a tweet is "Quality" content or "Spam"!

What is Spam?

Spam in this competition is defined as tweets that are posted by known fake twitter accounts that are:

- Politically Motivated
- Automatically generated content
- Meaningless content
- Click Bait

Columns

- Tweet

This is the text that was tweeted

- following

The number of people the account that tweeted is following

- followers

The number of people following the account that tweeted

- actions

The total number of favorites, replies, and retweets of said tweet

- is_retweet

Binary [0,1] value: If 0 is not a retweet, if 1 it is a retweet

- location

The self-written location provided by the user on their profile, may not exist, be "Unkown", and is NOT standardized! ex. could be ("NY", "New York", "Upper East Side", Etc!)

- Type

Either Quality or Spam

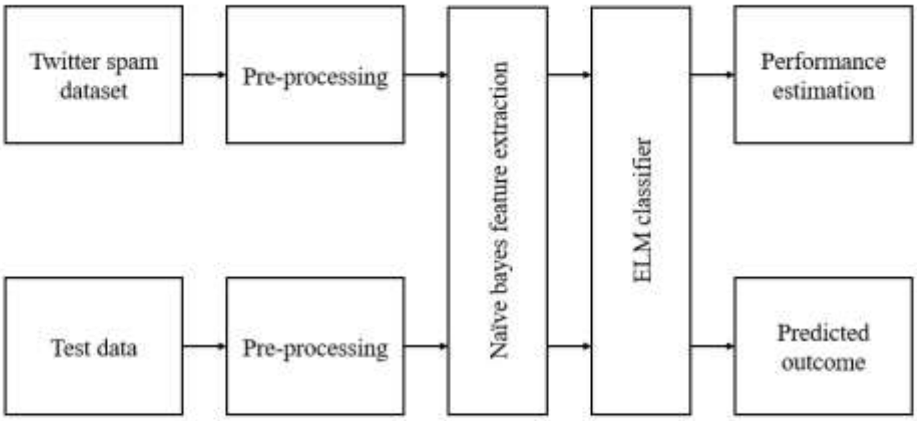


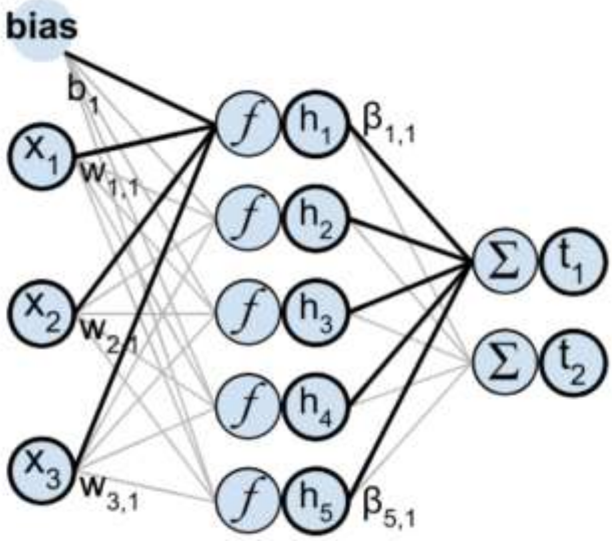
Fig. 2: Block diagram of proposed system.

Extreme learning machines

Extreme learning machines are feedforward neural networks for classification, regression, clustering, sparse approximation, compression and feature learning with a single layer or multiple layers of hidden nodes, where the parameters of hidden nodes (not just the weights connecting inputs to hidden nodes) need to be tuned. These hidden nodes can be randomly assigned and never updated (i.e., they are random projection but with nonlinear transforms), or can be inherited from their ancestors without being changed. In most cases, the output weights of hidden nodes are usually learned in a single step, which essentially amounts to learning a linear model. The name "extreme learning machine" (ELM) was given to such models by its main inventor Guang-Bin Huang. Extreme learning machines are feed-forward neural networks having a single layer or multiple layers of hidden nodes for classification, regression, clustering, sparse approximation, compression, and feature learning, where the hidden node parameters do not need to be modified. These hidden nodes might be assigned at random and never updated, or they can be inherited from their predecessors and never modified. In most cases, the weights of hidden nodes are usually learned in a single step which essentially results in a fast-learning scheme. These models, according to their inventors, are capable of producing good generalization performance and learning thousands of times quicker than backpropagation networks. These models can also outperform support vector machines in classification and regression applications, according to the research.

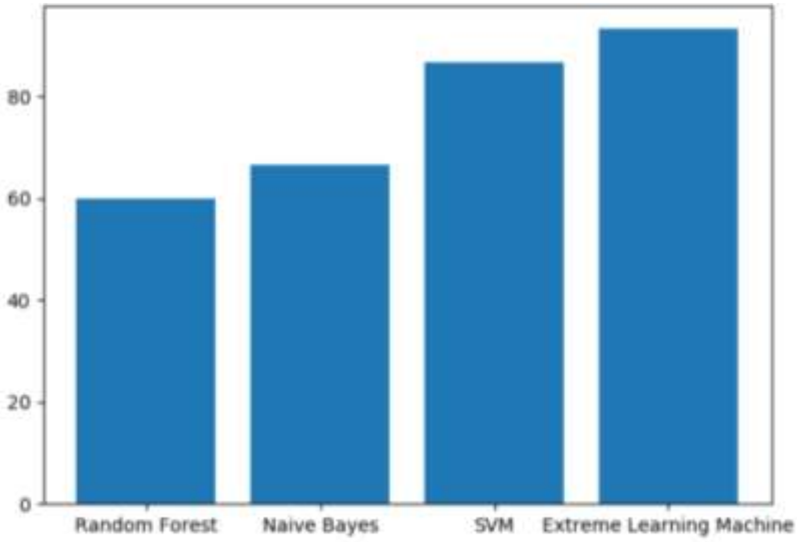
Fundamentals of ELM

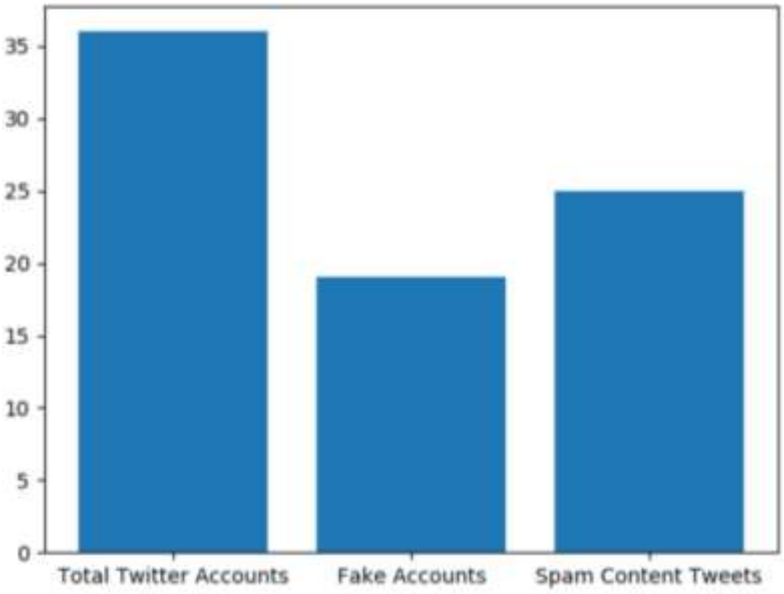
An ELM is a quick way to train SLFN networks (shown in the below figure). An SLFN comprises three layers of neurons, however, the name Single refers to the model’s one layer of non-linear neurons which is the hidden layer. The input layer offers data features but does not do any computations, whereas the output layer is linear with no transformation function and no bias. The ELM technique sets input layer weights W and biases b at random and never adjusts them. Because the input weights are fixed, the output weights ???? are independent of them (unlike in the Backpropagation training method) and have a straightforward solution that does not require iteration. Such a solution is also linear and very fast to compute for a linear output layer.



Random input layer weights improve the generalization qualities of a linear output layer solution because they provide virtually orthogonal (weakly correlated) hidden layer features. A linear system's solution is always in a range of inputs. If the solution weight range is constrained, orthogonal inputs provide a bigger solution space volume with these constrained weights. Smaller weight norms tend to make the system more stable and noise resistant since input errors are not aggravating in the output of the linear system with smaller coefficients. As a result, the random hidden layer creates weakly correlated hidden layer features, allowing for a solution with a low norm and strong generalization performance.

5. RESULTS AND DISCUSSION





6. CONCLUSION

It is estimated that 35 billion spam email messages per day were generated in 2004. These messages are nuisance to the receivers and in addition create low availability and network congestion. The problem of spam in VoIP networks has to be solved in real time compared to e-mail systems. Many of the techniques devised for e-mail spam detection rely upon content analysis and in the case of VoIP it is too late to analyse the media after picking up the receiver. So, we need to stop the spam calls before the telephone rings. The proposed algorithm is extreme learning machine (ELM). In computing, trust has traditionally been a term relating to authentication, security, or a measure of reliability. When it comes to receiving or rejecting a voice call social meaning of trust is applied and in particular reputation of the calling party is analyzed.

REFERENCES

[1] O. Varol et al., “Online Human-Bot Interactions: Detection, Estimation, and Characterization,” Proc. ICWSM, Montreal, Canada, May 2017.

[2] S. Liu, J. Zhang, and Y. Xiang. “Statistical Detection of Online Drifting Twitter Spam,” Proc. ACM ASIACCS, Xi’an, China, May 30–June 3 2016, pp. 1–10

[3] H. Zhu et al., “ShakeIn: Secure User Authentication of Smartphones with Habitual Single-handed Shakes,” IEEE Trans. Mobile Computing, vol. 16, no. 10, 2017, pp. 2901–12

[4] B. Feng, Q. Fu, M. Dong, D. Guo and Q. Li, "Multistage and Elastic Spam Detection in Mobile Social Networks through Deep Learning," in IEEE Network, vol. 32, no. 4, pp. 15-21, July/August 2018, doi: 10.1109/MNET.2018.1700406.

[5] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi and P. Samarati, "P2P-based collaborative spam detection and filtering," Proceedings. Fourth International Conference on Peer-to-Peer Computing, 2004. Proceedings., 2004, pp. 176-183, doi: 10.1109/PTP.2004.1334945.

[6] X. Hu, J. Tang, H. Gao and H. Liu, "Social Spammer Detection with Sentiment Information," 2014 IEEE International Conference on Data Mining, 2014, pp. 180-189, doi: 10.1109/ICDM.2014.141.

[7] M. Mateen, M. A. Iqbal, M. Aleem and M. A. Islam, "A hybrid approach for spam detection for Twitter," 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST), 2017, pp. 466-471, doi: 10.1109/IBCAST.2017.7868095.

[8] Z. Wu, J. Cao, Y. Wang, Y. Wang, L. Zhang and J. Wu, "hPSD: A Hybrid PU-Learning-Based Spammer Detection Model for Product Reviews," in IEEE Transactions on Cybernetics, vol. 50, no. 4, pp. 1595-1606, April 2020, doi: 10.1109/TCYB.2018.2877161.

[9] K. Thomas, C. Grier, J. Ma, V. Paxson and D. Song, "Design and Evaluation of a Real-Time URL Spam Filtering Service," 2011 IEEE Symposium on Security and Privacy, 2011, pp. 447-462, doi: 10.1109/SP.2011.25.

[10] G. Wang, S. Xie, B. Liu and P. S. Yu, "Review Graph Based Online Store Review Spammer Detection," 2011 IEEE 11th International Conference on Data Mining, 2011, pp. 1242-1247, doi: 10.1109/ICDM.2011.124.

[11] S. Shehnepoor, M. Salehi, R. Farahbakhsh and N. Crespi, "NetSpam: A Network-Based Spam Detection Framework for Reviews in Online Social Media," in IEEE Transactions on Information Forensics and Security, vol. 12, no. 7, pp. 1585-1595, July 2017, doi: 10.1109/TIFS.2017.2675361.

[12] F. Masood. "Spammer Detection and Fake User Identification on Social Networks," in IEEE Access, vol. 7, pp. 68140-68152, 2019, doi: 10.1109/ACCESS.2019.2918196.