International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 6, Nov-Dec-2019, ISSN (Online): 2395-566X

Social BOT Detection for Twitter Dataset Using GBC Technique

M.Tech. Scholar Roshani Singh Prof. Sumit Sharma
Dept. of Computer Science & Engg.
Vaishnavi Institute of Technology & Science
Bhopal, MP, India

Abstract - **Today, peoples are devoting large amount of time in social networks. As social media network popularity increases day by day, content writer are spamming on these platform for potential benefits of some organizations. Here graph based clustering algorithm was proposed to develop a binary cluster on the basis of serial or sequential action perform by the user. As per series of action social network graph is reduce into spanning tree where highly distance nodes are identified as abnormal behavior. So group of highly distance nodes are consider as the social bots while other are real users. Experiment was perform on real social sequential dataset of twitter. Results were compared on various evaluation parameters and it was obtained that proposed approach has improved all such parameter values as compared to previous approach adopt by researcher.**

Keywords- **Online Social Networks (OSNs), Twitter, Spammers, Legitimate users, Social bots, user behaviour, semi-supervised clustering.**

I. INTRODUCTION

During the past two decades, we have progressively turned to the Internet and social media to find news, share opinions, and entertain conversations (Morris and Ogan, 1996; Smith and Brenner, 2012). What we create and consume on the Internet impacts all aspects of our daily lives, including our political, health, financial, and entertainment decisions. This increased influence of social media has been accompanied by an increase in attempts to alter the organic nature of our online discussions and exchanges of ideas. In particular, over the past 10 years we have witnessed an explosion of social bots (Lee et al., 2011; Boshmaf et al., 2013), a presence that doesn't show signs of decline. Social bots are social media accounts controlled completely or in part by computer algorithms.

They can generate content automatically and interact with human users, often posing as, or imitating, humans (Ferrara et al., 2016a). Automated accounts can be harmless and even helpful in scenarios where they save manual labor without polluting human conversations. In this paper, however, we focus on those actors in online social networks that surreptitiously aim to manipulate public discourse and influence human opinions and behavior in an opaque fashion. While more traditional nefarious entities, like malware, attack vulnerabilities of hardware and software, social bots exploit human vulnerabilities, such as our tendencies to pay attention to what appears to be popular and to trust social contacts (Jun et al., 2017). Unlike other social engineering attacks, such as spear phishing (Jagatic et al., 2007), bots can achieve scalability through automation. For example, multiple accounts controlled by a single entity can quickly generate posts and make specific content trend or amplify misinformation. They can trick humans and engagement-based ranking algorithms alike, creating the appearance that some person or opinion is popular.

Therefore, defending from social bots raises serious research challenges (Boshmaf et al., 2012). Manipulation of public opinion is not new; it has been a common practice since the dawn of humanity. The technological tools of all eras — printed media, radio, television, and the Internet — have been abused to disseminate misinformation and propaganda. The deceptive strategies employed on all these types of channels share striking similarities (Varol and Uluturk, 2018). Nevertheless, social media are particularly vulnerable because they facilitate automatic interactions via software.

As a result, social media platforms have to combat a deluge of attacks. Facebook recently announced that 1.5 billion fake accounts were removed over six months in 2018.1 Even a very low miss rate could leave millions of accounts available to be used as bots. In this light, it is not surprising that as many as 9–15% of active Twitter accounts were estimated to be bots in 2017 (Varol et al., 2017a), and that social bots are responsible for generating two thirds of links to popular websites (Wojcik et al., 2018). Public interest in social bots has also dramatically increased during the past few years.

© 2019 IJSRET

II.RELATED WORK

Mc Cord et.al. [24] used user based features like number of friends, number of followers and content based features like number of URLs, replies/mentions, retweets, hashtags of collected database. Classifiers namely Random Forest, Support Vector Machine (SVM), Naive Bayesian and K-Nearest Neighbour have been used to identify spam profiles in Twitter. Method has been validated on 1000 users with 95.7% precision and 95.7% accuracy using the Random Forest classifier and this classifier gives the best results followed by the SMO, Naive Bayesian and K-NN classifiers. Limitation of this approach is that for considered dataset reputation feature has been showing wrong results i.e. it is not able to differentiate spammers and non-spammers, unbalanced dataset has been used so Random Forest is giving best results as this classifier is generally used in case of unbalanced dataset, and finally the approach has been validated on less dataset.

Lee et. al.[22] deployed social honeypots consisting of genuine profiles that detected suspicious users and its bot collected evidence of the spam by crawling the profile of the user sending the unwanted friend requests and hyperlinks in MySpace and Twitter. Features of profiles like their posting behavior, content and friend information to develop a machine learning classifier have been used for identifying spammers. After analysis profiles of users who sent unsolicited friend requests to these social honey pots in MySpace and Twitter have been collected. LIBSVM classifier has been used for identification of spammers. One good point in the approach is that it has been validated on two different combinations of dataset – once with 10% spammers+90% non-spammers and again with 10% non-spammers+90% spammers. Limitation of the approach is that less dataset has been used for validation.

Viswanath et al. [29] discover that dependency on community detection makes more vulnerable to Sybil attacks where honest identities conform strong communities. Because Sybils can infiltrate honest communities by carefully targeting honest accounts. That is, Sybils can be hidden as just another community on OSN by setting up a small number of the targeted links. The targeted links are the links given to the community which contains the trusted node. They make an experiment by allowing Sybils to place their links closer to the trusted node instead of random nodes, where closeness is defined by ranking used by the community detection algorithm they employ. Hence, Sybil nodes are high ranked in the defence scheme. Naturally, it leads to Sybils being less likely to be detected for that attack model because Sybils are appeared as part of the local community of the trusted node.

Boshmaf et al. [30] point out that structure-based Sybil detection algorithms should be designed to find local community structures around known honest (non-Sybil) identities, while incrementally tracking changes in the network by adding or deleting some nodes and edges dynamically in some period for better detection performance.

Chu et al. [2] make a study on profiling human, bot, and cyborgs2. They observe the difference among them in terms of tweet content, tweeting behaviour, and account properties like external URL ratio.

Benevenuto et. al. [7] detected spammers on the basis of tweet content and user based features. Tweet content attributes used are - number of hashtags per number of words in each tweet, number of URLs per word, number of words of each tweet, number of characters of each tweet, number of URLs in each tweet, number of hashtags in each tweet, number of numeric characters that appear in the text, number of users mentioned in each tweet, number of times the tweet has been retweeted. Fraction of tweets containing URLs, fraction of tweets that contains spam words, and average number of words that are hash tags on the tweets are the characteristics that differentiate spammers from non spammers.

Gee et. al. [12] utilized this feature and detected spam profiles using classification technique. Normal user profiles have been collected using Twitter API and spam profiles have been collected from "@spam" in Twitter. Collected data was represented in JSON then it was presented in matrix form using CSV format. Matrix has users as rows and features as columns. Then CSV files were trained using Naive Bayes algorithm with 27% error rate then SVM algorithm has been used with error rate of 10%. Spam profiles detection accuracy is 89.3%. Limitation of this approach is that not very technical features have been used for detection and precision is also less i.e. 89.3% so it has been suggested that aggressive deployment of any system should be done only if precision is more than 99%.

Yang et al. [33] collect Sybil accounts from Renren as ground-truth data set. Then, they analyse it by using network-based and structured-based features such as network clustering coefficient, incoming and outgoing request rate.

III. PROPOSED METHODOLOGY

1. Proposed Work

As the mining is use in various sort of information examination so for a similar all need to build the diverse strategy in the required zone. So contributing the content mining is done in this work by the proposed strategy for characterizing the social user behavior and sorts the bots in the cluster without having any earlier information of

© 2019 IJSRET

the individual tweet / comment. In the propose work no need of any configuration for the information, for example, speakers recognizable proof image or exceptional character, here all procedure is finished by using the diverse mix of content mining field.

2. Pre-processing

Preprocessing is a procedure utilized for transformation of content into feature vector. Much the same as content orders the preprocessing additionally has debate about its division. This work uses tweets preprocessing which comprise of words in charge of bringing down the execution of learning models. Here dataset contain some of columns which are not fruitful for this work, hence those information is either delete or transform into different state.

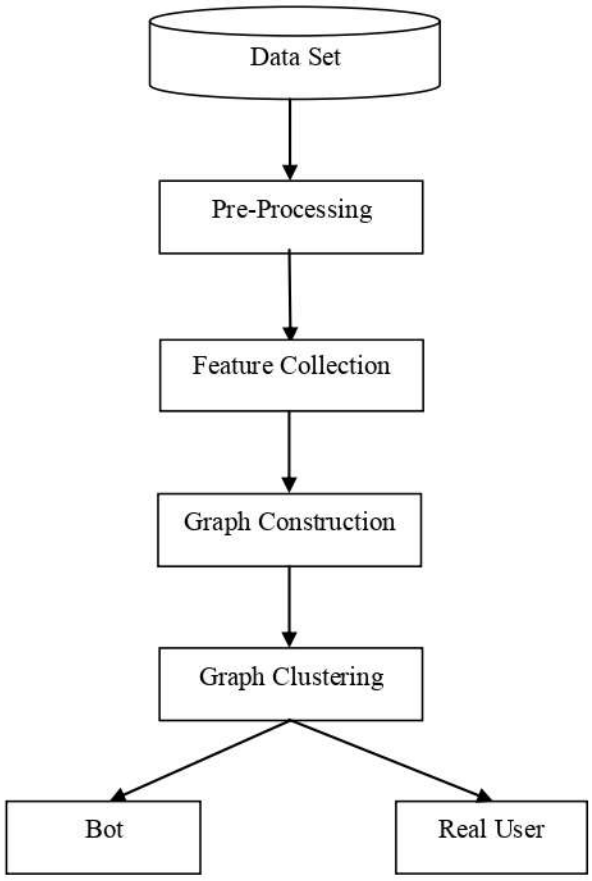

Fig. 1 Proposed work Block diagram.

3. Feature collection

This work twitter dataset was consider as the input where nine features of each user were extract. These features F represent the user behavior on the social network. Table 1 shows feature set utilize in this work.

Feature

1 Sequence of Shares
2 Sequence of Likes instance
3 Sequence of Tweets perform by user
4 Sequence of re-tweet perform by user
5. Time instance of the event

Feature from 1 to 5 can be easily extract from the dataset where as per the behavior follow by the user

Transition probability between action events steams: P(i,j) represents the probability that the click action is j at timestamp t followed by click action at timesptamp t-1. So probability finds the relation between the i and j features of the use in form of transition done in a specific set of durations.

$$P(i,j) = \frac{\sum_1^t X_i \rightarrow X_j}{\sum_1^t X_i}$$

In above equation $X_i \rightarrow X_j$ act as transition form i to j feature instance, while t as the total time instance when i feature were applied. So if work use n number of features than each user has a feature vector of nxn where cell contain a probability transition value.

$$F_x = \begin{vmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{vmatrix}$$

4. Graph Construction

In this step develop a completely connected graph where each node is connect with other node and distance between them act as weight of the edge. Estimation of the distance was done by using X and Y axis of the system. Here this can be understand as let nodes are N = {n_1, n_2, n_3,……..n_m} and distance between them are evaluate by Euclidian distance formula shown in table 4.3.

$$DD^f_{x,y} = \sqrt{(XX_{f11} - YX_{f11})^2 + (XX_{f12} - YX_{f12})^2 \ldots\ldots\ldots\ldots + (XX_{fnn} - YX_{fnn})^2}$$

Where X and Y are feature values of ending nodes, while XX_{f12} is the X user transition probability feature of action event 1and action event 2.

Now sort graph edges with Minimum Weight in a decreasing order. This can be understand as matrix S[] of three column and rows depend on number of edges present in the graph. So from table 1 total rows will be 6. Hence a shorted matrix of S=6X3 was developed in this step. Table 4.4 shows this sorted matrix.

5. Minimum Spanning Tree

In this step above S[] act as input where first row or first edge with minimum weight was select in spanning tree construction. Mark each of these node as CHK. Now for each new row of S[] check that either of vertex or node

© 2019 IJSRET

present in the CHK matrix. If both the vertex of new row are already present in the matrix than ignore this row, otherwise include new vertex and mark them as CHK.

This work use Kruskal's algorithm for tree construction where as per edge present in table 4.4 sortest weight edge was considered. Fig. 4.3 a shows first row of the tree while fig. 4.3 b shows second row insertion in tree. In similar way other set of nodes were include in the trees. Each new row insertion involve at-least one new node in the tree. Hence nodes which have higer distance from other node get easily separated without any training.

5. Graph Clustering

Once minimum spanning tree obtain than remove highly weighted edges of the tree such that whole tree get break into K cluster. This can be understand as

Removing highly weighted edge one can obtain clusters of node which are closer to each other as per distance. Here it was also obtained that proposed clustering algorithm can easily generate k clusters by simply removing the highly weighted edge only. This algorithm not required any iteration or other thing.

6. Resultant cluster

So nodes which are present with less distance edge weights are considered as the true user or real user of the social media. While nodes whose distance values are larger in oter cluster were consider as the bots. As each bot set of instance sequence were totally different from real user set of instances, so distance from other existing nodes were high. Hence cluster selection of real or bot is depends on the weight value of the partial tree present in the cluster.

7.Proposed Algorithm and Flowchart

Input: DS // DS: Sequenntial action dataset
Ouput: C_r C_b// C_r: Cluster of real user, C_b : Cluster of social Bot

1. PD←Pre-Processing(DS)
2. Loop 1:n// n: number of users
3. Loop 1:i // i, j: Sequential Features
4. Loop 1:j
5. F(i,j)←Transition_Probablity(PD,n,I,j)
6. EndLoop
7. EndLoop
8. EndLoop
9. FCG←Fully_connected_graph(F, n)// FCG: Fully Connected Graph
10. ST←Spanning_Tree(FCG)
11. Loop 1:n
12. If W(ST,n) > T
13. C_r←n
14. Otherwise
15. C_b←n
16. EndIf
17. EndLoop

IV.EXPERIMENTAL SETUP

Whole work was implement on MATLAB software. It is utilize on account of its rich library which has numerous inbuilt storage that can be specifically use in this work for various reason. Out of various storage few are crossing point, contrasting of the string, and so forth. One more essential factor is its GUI by which one who doesn't know about the code can straightforwardly runs the storage without having earlier information.

1.Dataset

In this work experiment is done on social dataset content obtained from https://botometer.iuni.iu.edu/bot-repository/datasets.html, where as per the user related twitter comments of respected user with different action and timestamp were avaialble. Description of the dataset is available in table 1

Table 1 Experimental dataset detailed description.

Features	Values
Instances	36462
Users	29
Data Year	1
Actions	4

2. Evaluation Parameter

In order to evaluate results there are many parameter such as accuracy, precision, recall, F-score, etc. Obtaining values can be put in the mention parameter formula to get better results.

$$Precision = \frac{True_Positive}{True_Positive + False_Positive}$$

$$Recall = \frac{True_Positive}{True_Positive + False_Negative}$$

$$F_Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

$$Accuracy = \frac{Correct_Classification}{Correct_Classification + Incorrect_Classification}$$

In above true positive value is obtain by the system when the specified user is real user and system also says that the user is real. While in case of false positive value it is obtain by the system when the specified that user is bot and system also says that the user is real. Similarly in case

© 2019 IJSRET

of True negative value it is obtain by the system when the algorithm specified that user is real and system also says that the user is bot. Results of the proposed graph based clustering of bots and real user were compared with existing method developed in [29].

Table 2 Precision value comparison of proposed and previous work.

Instances	User	Previous Work	Proposed Work
20000	11	0.5556	0.7778
25000	20	0.7143	0.8571
30000	24	0.7778	0.8889
35000	29	0.8261	0.913

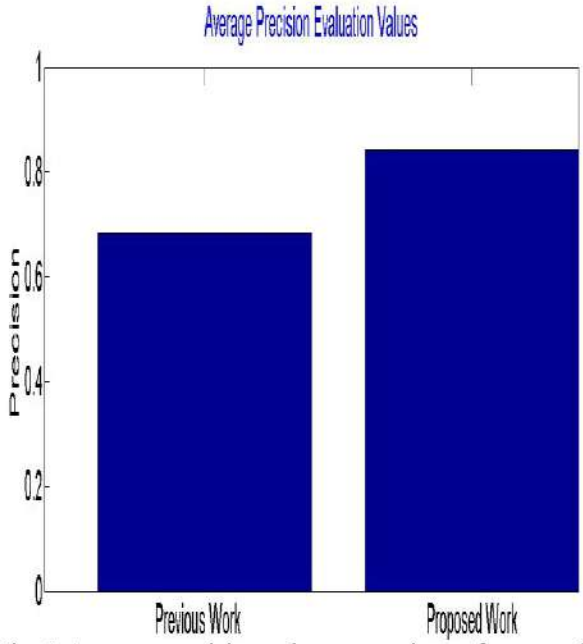

Fig. 1 Average precision value comparison of proposed and previous work.

Above table 2 and fig. 1 shows that Precision value of proposed work was high as compared to previous algorithm [29]. Here proper leaning feature with pre-processing filter increase the efficiency of the work. It has been observed that proposed work of spanning tree construction for clustering plays an important role for unsupervised classification of data into real and bot users.

Table 3 Recall value comparison of proposed and previous work.

Instances	User	Previous Work	Proposed Work
20000	11	0.7143	1
25000	20	0.625	0.75
30000	24	0.7	0.8889
35000	29	0.76	0.913

Above table 3 shows that Recall value of proposed work was high as compared to previous algorithm [29]. Here proper leaning feature with pre-processing filter increase the efficiency of the work. It has been observed that proposed work of spanning tree construction for clustering plays an important role for unsupervised classification of data into real and bot users.

Table 4 F-Measure value comparison of proposed and previous work.

Instances	User	Previous Work	Proposed Work
20000	11	0.625	0.8750
25000	20	0.6667	0.8
30000	24	0.7368	0.8889
35000	29	0.7917	0.913

Above table 4 shows that F-Measure value of proposed work was high as compared to previous algorithm [29]. Here proper leaning feature with pre-processing filter increase the efficiency of the work. It has been observed that proposed work of spanning tree construction for clustering plays an important role for unsupervised classification of data into real and bot users.

Table 5 Accuracy value comparison of proposed and previous work.

Instances	User	Previous Work	Proposed Work
20000	11	0.4545	0.8182
25000	20	0.5	0.7
30000	24	0.5833	0.8333
35000	29	0.6552	0.8621

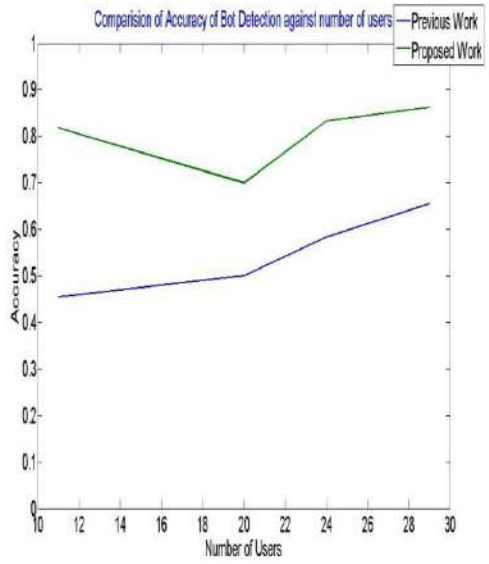

Fig. 2 Comparison of bot detection accuracy against different number of users.

© 2019 IJSRET

Above table 5.5 and fig. 5.2 shows that Precision value of proposed work was high as compared to previous algorithm [29]. Here proper leaning feature with pre-processing filter increase the efficiency of the work. It has been observed that proposed work of spanning tree construction for clustering plays an important role for unsupervised classification of data into real and bot users.

V.CONCLUSIONS

This work presents a study of methods for detection of user profiles as real or social bot. Here a technique was proposed for classifying the social nodes into two cluster, where digital social network graph was reduce into spanning tree. Here weight of the graph was social sequential action transitional probability. So each node has its own set of transitional probability and distance between nodes of transitional probability act as weight of graph. So as per graph clustering technique spanning tree was developed and highly weighed nodes of this tree act as social bot cluster while other set of nodes are real. Results shows that proposed graph based clustering technique has increase the precision value by 19.59% as compared to previous approach used in [29]. While recall value was also increase by 26.88%, at the same time accuracy of the social bot identification was also increase by 46.53%.

REFERENCES

[1] Morris, M. and Ogan, C. (1996). The internet as mass medium. Journal of communication, 46(1):39-50.

[2] Lee, K., Eo_, B. D., and Caverlee, J. (2011). Seven Months with the Devils:A Long-Term Study of Content Polluters on Twitter. In Proc. AAAI Intl. Conf. on Web and Social Media (ICWSM).

[3] Ferrara, E., Varol, O., Davis, C., Menczer, F., and Flammini, A. (2016a). The rise of social bots. Communications of the ACM, 59(7):96{104.

[4] Jun, Y., Meng, R., and Johar, G. V. (2017). Perceived social presence re-duces fact-checking. Proceedings of the National Academy of Sciences,114(23):5976{5981.

[5] Jagatic, T. N., Johnson, N. A., Jakobsson, M., and Menczer, F. (2007). Social phishing. Communications of the ACM, 50(10):94{100.

[6] Boshmaf, Y., Muslukhov, I., Beznosov, K., and Ripeanu, M. (2012). Key chal- lenges in defending against malicious socialbots. In Proc. 5th USENIX Conference on Large-Scale Exploits and Emergent Threats (LEET).

[7] Varol, O. and Uluturk, I. (2018). Deception strategies and threats for online discussions. First Monday, 22(5).

[8] Wojcik, S., Messing, S., Smith, A., Rainie, L., and Hitlin, P. (2018). Bots in the twittersphere. Pew Research Center, Washington, D.C.

[9] M. McCord, M. Chuah, Spam Detection on Twitter Using Traditional Classifiers, ATC'11, Banff, Canada, Sept 2-4, 2011, IEEE.

[10] Kyumin Lee, James Caverlee, Steve Webb, Uncovering Social Spammers: Social Honeypots + Machine Learning, Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval, 2010, Pages 435–442, ACM, New York (2010).

[11] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove, "An analysis of social network-based sybil defenses," ACM SIGCOMM Computer Communication Review, vol. 40, pp. 363-374, 2010.

[12] Y. Boshmaf, K. Beznosov, and M. Ripeanu, "Graph-based sybil detection in social and information systems," in Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on, 2013, pp. 466-473.

[13] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Detecting automation of twitter accounts: Are you a human, bot, or cyborg?," IEEE Transactions on Dependable and Secure Computing, vol. 9, pp. 811-824, 2012.

[14] Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida, Detecting Spammers on Twitter, CEAS 2010 Seventh annual Collaboration, Electronic messaging, Anti Abuse and Spam Conference, July 2010, Washington, US

[15] Grace gee, Hakson Teh, Twitter Spammer Profile Detection, 2010.

[16] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai, "Uncovering social network sybils in the wild," ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 8, p. 2, 2014.

[17] M. A. Rajab, J. Zarfoss, F. Monrose and A. Terzis,\ A multifaceted approach to understanding the bot-Net phenomenon," in Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement (IMC'06), pp. 41-52, 2006.

© 2019 IJSRET