DETECTION OF SOCIAL BOTS IN SOCIAL NETWORK "VKONTAKTE"

Filipp Beldiushkin
Yuriy Shichalin's Classical Lyceum
Moscow, Russia
beldiushkin.f@protonmail.com, fkb2610@gmail.com

ABSTRACT

This report demonstrates how to identify bots in social media. The study focuses on a Russian social network called "VKontakte". This research was conducted using classical machine learning methods and graph-based approaches. All the information about the network of friends of a particular user was taken from the user's profile page. The project code is available.[1]

Keywords Detection of Social bots · VKontakte · Social and Information Networks · Machine Learning

1 Introduction

Social bots are used to spread spam messages and fake news. Fake news is often introduced in the period of political elections to alternate the course of elections or influence financial markets. The reason why social media is the best source for spreading fake news is that a user of a social platform will be more likely to believe information that comes from a trusted "circle of friends" instead of "government-controlled", less familiar media [1, 2].

Detecting bot profiles is one of the pressing tasks of modern IT technologies. It helps to reduce the theft of users' personal data, halt the spread of fake news, increase the objectivity of social research, voting, etc. [1]. At the moment, there are two main methods of identifying fake accounts: the study of personal data and the detection of abnormal activity [3]. Both of these methods are based on a numerical statistical analysis of the user's profile and activity. However, both of these methods do not take into account the user's social connections.

This report discusses one of the ways to detect bots in the social network VKontakte [2]. More specifically, the research was done by analyzing connections a user has, which are generally called "friends"[3]. Analyzing a user's friend network is a commonplace practice in research which is based on social media in general and VKontakte specifically ranging from community detection and formation among university students or software developers to more specific applications like psychological depression studies [5, 6, 7, 8]. This research's central hypothesis is that social bots choose friends from random users, while real users choose friends from their social circle (cf. [9]).

2 Data Collection

The open API of VKontakte social network[4] was used to collect data from a user's profile.

1,797 IDs of social bots were downloaded from the website gosvon.net. 9,969 IDs of real users were selected randomly from VKontakte. Data of two types was collected for each account:

1. Data from a user's profile (listed in Appendix);

[1] https://github.com/filipp134/vk_bot_detection
[2] https://vk.com/
[3] Note that this type of networks is sometimes referred as ego networks [4]
[4] https://vk.com/dev/manuals

2. The graph features extracted from the network of friends.

The primary attention in this report is paid to the graph of friends' connections created from a user's profile information. The graph was built based on the following assumptions: for each of the 11,766 accounts (denoted as "Set V_1"), their friends were taken (denoted as "Set V_2"). Then, for each of the accounts of the $V2$ set, their friends were taken (denoted as "Set V_3"). After that, for each user account, i.e. v from the set V_1, a graph G_v was constructed, which includes the node v. The adjacent nodes of v were taken from the set V_2. Node v is connected to all the nodes from the set $V_2 \cap G_v$, and all the nodes from $V_2 \cap G_v$ are also connected to their counterparts in V_2 and V_3. The edges of this graph represent friends' connections. Due to the observed symmetry of these friends' connections, the edges of the graph are undirected, and the graph itself is undirected.

Then, the graph features were calculated for each of the G_vs. The features listed in the Appendix quantitatively describe the structure of a user's friend network.

There are two examples shown in Figures 1 and 2.

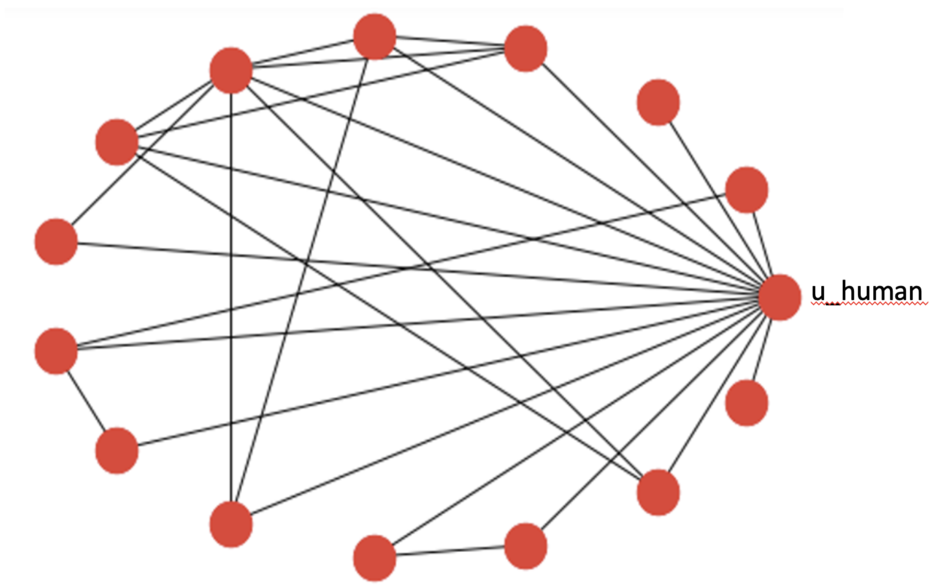

Figure 1: Graph G_v represents connections based on the profile of a real user

3 Approach

The central hypothesis of this research is that one should examine the ego network of this particular user to determine whether or not a user of the VKontakte network is a social bot. The efficiency of using graph features has to be compared to the efficiency of profile features.

The algorithm for this comparison is as follows:

1. The classification model is built;
2. The importance of each feature is measured using the Permutation Importance algorithm;
3. The rank of the importance of listed graph-based features is recorded.

Four methods were used to build classification models:

1. Logistic Regression;
2. Random Forest;
3. Gradient Boosting;
4. XGBoost.

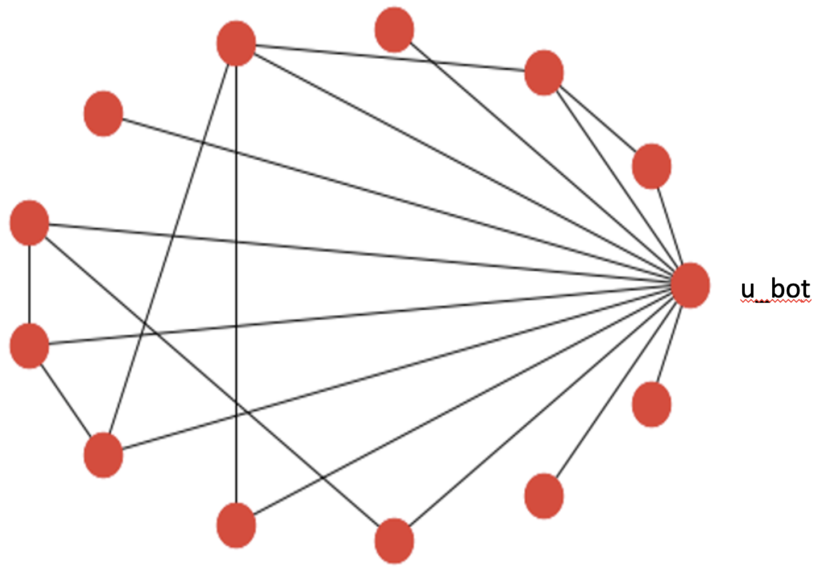

Figure 2: Graph G_v visualizes connections based on the profile of a social bot

Permutation Importance algorithm includes the following steps :

1. A validation dataset is fixed;
2. The quality of the model is calculated on the validation dataset;
3. One feature is removed from the dataset forming a revised dataset;
4. The quality of the model is calculated on the revised dataset;
5. The difference between the quality values is calculated.

Thus, the Permutation Importance algorithm demonstrates that the feature is important if its removal leads to a significant drop in quality.

4 Experimental Data and Results

The Random Forest algorithm showed the best results in terms of the area under ROC learning curve (see Table 1).

Table 1: Classifiers’ performance in terms of ROC-AUC score

Method of modeling	ROC-AUC score
Logistic Regression	.876
Random Forest	.981
Gradient Boosting	.975
XGBoost	.978

Top five features according to Permutation Importance in Logistic Regression are shown in Table 2.

Table 2: Top five features according to Permutation Importance in Logistic Regression

Feature	Weight
Average neighbor degree	0.0701 ± 0.0053
Closeness centrality	0.0289 ± 0.0017
Subscriptions	0.0419 ± 0.003
Relation	0.0178 ± 0.0013
Transitivity	0.0166 ± 0.0012

Top five features according to Permutation Importance in Random Forest are given in Table 3.

Table 3: Top five features according to Permutation Importance in Random Forest

Feature	Weight
Average neighbor degree	0.2950 ± 0.0062
Pages	0.0747 ± 0.0025
Has photo	0.0419 ± 0.0034
Subscriptions	0.0101 ± 0.0009
Friends	0.0088 ± 0.0003

Top five features according to Permutation Importance in Gradient Boosting are shown in Table 4.

Table 4: Top five features according to to Permutation Importance in Gradient Boosting

Feature	Weight
Average neighbor degree	0.2849 ± 0.0055
Has photo	0.1399 ± 0.0033
Pages	0.1147 ± 0.0012
Closeness centrality	00.0871 ± 0.0039
Friends	0.0158 ± 0.0012

Top five features according to Permutation Importance in XGBoost are shown in Table 5.

Table 5: Top five features according to to Permutation Importance in XGBoost

Feature	Weight
Average neighbor degree	0.2843 ± 0.0057
Has photo	0.1409 ± 0.0033
Closeness centrality	0.1289 ± 0.0034
Pages	0.0953 ± 0.0022
Friends	0.0144 ± 0.0011

5 Conclusion

The experiment results show that the graph analysis (namely average neighbor degree feature) gives the best indication on how to distinguish a social bot from a real user. This confirms the original hypothesis that social bots choose friends from random users, while real users choose friends from their social circle, as the average neighbor degree feature indicates how many friends of the studied users are actually friends with each other.

Finally, it is essential to emphasize that social bots can be successfully detected with the help of the built models. As demonstrated in this experiment, fabricating a plausible network of friends is much more complicated than fabricating other fake profile data.

One of the future studies may also use lessons drawn from the research of tendencies to become friends by like-minded users of social networks [9].

References

[1] AS Alymov, VV Baranjuk, and OS Smirnova. Detection of bot programs that mimic the behavior of people in the social network" vkontakte". *International Journal of Open Information Technologies*, 4(8):55–60, 2016.

[2] Samantha Bradshaw and Philip N Howard. The global disinformation order: 2019 global inventory of organised social media manipulation. *Working Paper 2019.3. Oxford, UK: Project on Computational Propaganda. comprop.oii.ox.ac.uk.*, pages 1–23, 2019.

[3] Onur Varol, Emilio Ferrara, Clayton A. Davis, Filippo Menczer, and Alessandro Flammini. Online human-bot interactions: Detection, estimation, and characterization. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017*, pages 280–289. AAAI Press, 2017.

[4] Maksim Tsvetovat and Alexander Kouznetsov. *Social Network Analysis for Startups: Finding connections on the social web*. " O'Reilly Media, Inc.", 2011.

[5] Dmitry Gnatyshak, Dmitry I. Ignatov, Alexander Semenov, and Jonas Poelmans. Gaining insight in social networks with biclustering and triclustering. In Natalia Aseeva, Eduard Babkin, and Oleg Kozyrev, editors, *Perspectives in Business Informatics Research - 11th International Conference, BIR 2012, Nizhny Novgorod, Russia, September 24-26, 2012. Proceedings*, volume 128 of *Lecture Notes in Business Information Processing*, pages 162–171. Springer, 2012.

[6] Sofia Dokuka, Diliara Valeeva, and Maria Yudkevich. Formation and evolution mechanisms in online network of students: The vkontakte case. In Mikhail Yu. Khachay, Natalia Konstantinova, Alexander Panchenko, Dmitry I. Ignatov, and Valeri G. Labunets, editors, *Analysis of Images, Social Networks and Texts - 4th International Conference, AIST 2015, Yekaterinburg, Russia, April 9-11, 2015, Revised Selected Papers*, volume 542 of *Communications in Computer and Information Science*, pages 263–274. Springer, 2015.

[7] Olessia Koltsova, Sergei Koltsov, and Yadviga Sinyavskaya. When internet really connects across space: Communities of software developers in vkontakte social networking site. In Giovanni Luca Ciampaglia, Afra J. Mashhadi, and Taha Yasseri, editors, *Social Informatics - 9th International Conference, SocInfo 2017, Oxford, UK, September 13-15, 2017, Proceedings, Part II*, volume 10540 of *Lecture Notes in Computer Science*, pages 431–442. Springer, 2017.

[8] Aleksandr M. Semenov, Alexey Natekin, Sergey I. Nikolenko, Philipp Upravitelev, Mikhail Trofimov, and Maxim Kharchenko. Discerning depression propensity among participants of suicide and depression-related groups of vk.com. In Mikhail Yu. Khachay, Natalia Konstantinova, Alexander Panchenko, Dmitry I. Ignatov, and Valeri G. Labunets, editors, *Analysis of Images, Social Networks and Texts - 4th International Conference, AIST 2015, Yekaterinburg, Russia, April 9-11, 2015, Revised Selected Papers*, volume 542 of *Communications in Computer and Information Science*, pages 24–35. Springer, 2015.

[9] Hady Wirawan Lauw, John C. Shafer, Rakesh Agrawal, and Alexandros Ntoulas. Homophily in the digital world: A livejournal case study. *IEEE Internet Comput.*, 14(2):15–23, 2010.

Appendix

The list of the attributes used in modeling and their description is presented below.

Profile features:

1. has photo – information about whether the user has a profile photo. Return values: 1 – has, 0 – does not have.
2. sex – user's gender . Return values: 1 – female, 2 – male, 0 – not indicated.
3. has mobile – information whether a user has a mobile app associated with this social network. Return values: 1 – has, 0 – does not have.
4. followers count – number of followers a user has.
5. contacts – indication of the phone number on a profile page. Return values: 1 – indicated, 0 – not indicated.
6. relatives – indication of the list of relatives on a profile page. Return values: 1 – indicated, 0 – not indicated.
7. relation – indication of whether a user is in a relationship, and if his/her status is indicated on a profile page. Return values: 1 – indicated, 0 – not indicated.
8. personal – indication of a relationship status on a profile page. Return values: 1 – indicated, 0 – not indicated.
9. activities – indication of any activities on a profile page. Return values: 1 – indicated, 0 – not indicated.
10. music – indication of favorite music on a profile page. Return values: 1 – indicated, 0 – not indicated.
11. movies – indication of any favorite movies on a profile page. Return values: 1 – indicated, 0 – not indicated.
12. tv – indication of any favorite tv shows on a profile page. Return values: 1 – indicated, 0 – not indicated.
13. books – indication of any favorite books on a profile page. Return values: 1 – indicated, 0 – not indicated.
14. about – indication whether 'about myself' field on profile was filled. Return values: 1 – filled, 0 – not filled.
15. quotes – indication of any favorite quotes on a profile page. Return values: 1 – indicated, 0 – not indicated.
16. albums – number of photo albums.
17. audios – number of audios.
18. followers – number of followers.
19. friends – number of friends.
20. pages – number of pages a user flagged as "interesting".
21. photos – number of photos.
22. subscriptions – number of subscriptions.
23. videos – number of videos.

24. age – age.
25. city – geographical location of a user (city).
26. country – geographical location of a user (country).

Graph-based features:

1. avg cl – average clustering of the graph, based on a user's network of friends. It is the average of the sum of the results of division of all existing links between the neighbors of an edge and all possible links between them for all nodes in the social graph of a user.
2. trans – transitivity of the graph. It is the relation of the tripled number of triangles to the number of triples of nodes in the social graph of the user.
3. deg centr – average of the degree centrality, which is the relation of the degree of the node to the number of nodes in the graph minus 1, for every node in the social graph of a user.
4. average neighbor degree – average of the average neighbor degree for every node in the social graph of a user.
5. average degree connectivity – average of the average nearest neighbor degree, which is the relation of the sum of the degrees of the neighbors of a node to the degree of it, for all nodes in the social graph of a user.
6. k nearest neighbors – average of the average neighbor degree for every node in the social graph of a user.
7. degree centrality – average of the degree centrality, which is the relation of the degree of the node to the number of nodes in the graph minus 1, for every node in the social graph of a user.
8. closeness centrality – the average of the closeness centrality, which is the reciprocal of the sum of the length of the shortest paths between the node and all other nodes, for every node in the social graph of a user.
9. betweenness centrality – average of the betweenness centrality, which is the sum of the relations of the number of shortest path length for all combinations of 2 nodes that pass through the given node v to the number of shortest path length for all combinations of 2 nodes, for every in the social graph of a user.
10. diameter – the longest path between any points in the social graph of a user.
11. average shortest path length – average of the lengths of the shortest path lengths between all pairs of nodes in the graph.