T2S Based Social Media Bot Detection Using Machine Learning

Sakshi Pachori[1],Dr. Preeti rai[2]
[1, 2] Dept of CSE
[1, 2] Gyan Ganga Institute of Technology and Sciences, Jabalpur, M.P.

Abstract- *Recently, due to the rapid development of online social networks (OSNs) such as Facebook, Twitter, and Weibo, the number of calculators/social bots that imitate human users has increased. As artificial intelligence (AI) improves, social bots are getting smarter and better at manipulating people's calculation behaviors. Building a reliable and efficient search engine is crucial to keeping OSNs clean and users safe. Despite the rapid development of social bot search platforms, state-of-the-art systems still face challenges related to model generalization (and whether it can be adapted to different types of OSNs) and good networks. Bots spread misinformation and are difficult to detect based on a single piece of content, but advanced techniques can detect bots with high accuracy. Social media bot detection can use negative comments or other scripts to detect bots. Social media bots can target different audiences by creating fake models. The proposed model for Bot Operated Account Detection vs. Human Operated account detection method is based on past tweeting history of the user. Certain attributes such as friends, followers count, and favorites were considered as features for designing a classifier to detect Bots. In the proposed model, the historical behavior based on user posted tweets are the main concerned for detecting all the accounts.*

Keywords- Bot detection, social media, Word2Vec, Profile behaviors, Machine learning, Logistic regression.

I. INTRODUCTION

Recently, due to the rapid development of online social networks (OSNs) such as Facebook, Twitter, and Weibo, the number of calculators/social bots that imitate human users has increased. As artificial intelligence (AI) improves, social bots are getting smarter and better at manipulating people's calculation behaviours. Building a reliable and efficient search engine is crucial to keeping OSNs clean and users safe. Despite the rapid development of social bot search platforms, state-of-the-art systems still face challenges related to model generalization (and whether it can be adapted to different types of OSNs) and good networks [2]. Bots spread misinformation and are difficult to detect based on a single piece of content [3] [4] [5], but advanced techniques can detect bots with high accuracy [6] [7] [8]. Social media bot detection can use negative comments or other scripts to detect bots [9] [10]. Social media bots can target different audiences by creating fake models [11], [12]. In Cresci et al. [13], The authors identified different types of spam bots, including promotional bots, URL spam bots, and fake audiences. URL spam bots spread fake URL links by embedding these malicious links in referrals from legitimate users [14]. According to the research of Howard et al. [15], URL sharing bots are used to continuously copy tweets of legitimate users over a period to uncover malicious URLs. A popular bot detection program is "Bot meter", a monitoring system for identifying social bots [16]. Bot meter uses metadata (such as network properties, user properties, and time properties) associated with each Twitter account to provide a random forest classification algorithm. The properties of the network indicate how information is transmitted between groups of users. User characteristics include username, screen name, account creation time, and geographic location, as well as physical characteristics that show patterns in tweet time. A graph-based bot detection method that uses all methods related to the Twitter account to detect bots [17]. Advocacy can have negative effects when a person clings to long-held beliefs and decides not to change their old beliefs. Giving more facts leads to more resistance to accepting the truth rather than telling the truth behind the scenes. The person wants to tweet someone, retweet someone, or reply to someone's tweet. Therefore, the target of the discussion decides not to change their mind, regardless of the truth.Another important aspect of society is the "recognition bias".

On social media and Twitter, tweets that support people's ideas are more popular than tweets that oppose them. Some cases and positions that are clearly seen in the media are designed to exploit the backfire effect and false recognition. Social media bots can use the fake endorsement or "tweet effect" to create fake models, create fake money, and sell products. Such bots try to inject fake events into the user's mind.

1.1 Social Media Bot Identification

With the development of the Internet, the popularity of OSN has increased rapidly. OSNs such as Twitter, Facebook, Weibo and Instagram have become an important part of people's daily lives and are used to reading news, make friends and communicate with others. The volume and speed of data exchange between OSNs are very high (for example, more than 400 million tweets are created on Twitter every day, and 4.75 billion posts, photos, comments, etc. are shared on Facebook daily). However, recent research shows that most of the content/news shared on OSNs consists of spam, phishing, or misinformation generated through social networks (also known as "sybil" accounts) and targeting different accounts. Most malicious bots are the main tool for social manipulation on OSN, as seen in the Facebook-Cambridge Analytica scandal related to the 2016 US presidential election1. Social bots are designed as software/systems that can participate in social networks, online chats, etc., Like human accounts. They are used to spread misinformation or misleading messages to sway the opinions of OSN users in a certain direction. Deception and misinformation from social bots can harm human relationships and undermine social trust. To limit the spread of social bots on OSNs, many researchers and organizations have developed various methods to detect and block fake accounts/bots. One of the best ways to build a bot detection platform is to focus on learning the behaviors of accounts that flag bots. In this way, off-the-shelf machine learning algorithms are used to learn specific features extracted from large datasets before text for real bots [18, 19]. Traditional machine learning techniques for bot detection use well-known techniques such as Support Vector Machine (SVM), Logistic Regression (LR), Naive Bayes (NB), etc.
Traditional machine learning methods for social bot detection use well-known classification techniques, such as support vector machines (SVM), logistic regression (LR), Naïve Bayes (NB), etc. to train the bot detection mechanism as a binary classifier. Recently, researchers focused on using advanced deep learning architectures, such as a convolutional neural network (CNN), long-short-term memory (LSTM), and similar [20, 21] to leverage the quality of behavior-based bot detection models. Crowdsourcing methods are applied to build labeled bot datasets which are used for training bot detection platforms via machine learning algorithms. In this crowdsourcing approach the opinions of experts serve as ground-truths which are developed by humans for identifying and determining the behaviors of social bots. Another approach for social bot detection is graph-based analysis. In the graph-based social bot detection approach, the proposed models typically employ common graph analysis techniques such as the graph's property metrics (local clustering coefficient, centrality, bidirectional link ratio, random walk, etc.), graph's propagation, node clustering, etc. In fact, most of the efforts that have gone into constructing an effective bot detection system have been invested in evaluating and selecting user profile-based or graph-based features for training the classification models. Therefore, previous machine learning based approaches are considered as expensive due to the comprehensive human efforts needed for feature engineering and training set annotation. These existing challenges can be addressed by using the automatic latent feature learning of the network embedding approach.

The main idea of our work comes from the observation of the normal users/social bots'distributors in their own groups/communities in different types of OSNs (Twitter, Facebook, etc.). We recognized that bots and normal users are normally active and tend to interact (make friends/follow/share/like, etc.) with other accounts in their own communities.

By preserving both the structures of the local neighbors and intra-community of each user node, the Bot2Vec model can achieve better quality user representations in different types of social networks than recent state-of-the-art network embedding baselines, such as Deep Walk [21], LINE [22] and Node2Vec [23], in terms of bot detection.

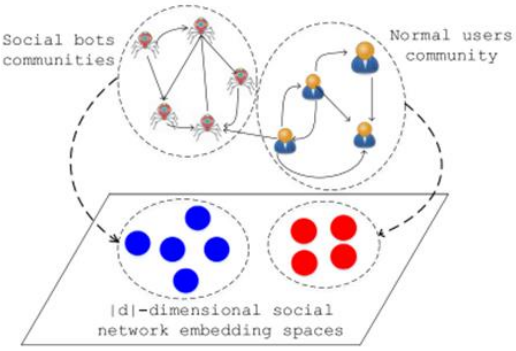

Figure 1.1: Projection with PCA.

Figure 1.1 shows the visualization of 3D projections via PCA for the Cresci-2015 dataset [24] with different network embedding techniques. By applying an intracommunity oriented random walk strategy model shows promise regarding addressing some of existing challenges of machine learning based bot detection, including model generalization and great efforts needed for feature engineering. Advertising bots are electronic accounts managed by software algorithms rather than human users. Research shows that bots are used in digital work; the widespread use of bots in information management is characterized by their role in the dissemination of information, such as information aggregators, false amplifiers [25] and disinformation in political debates. Since bots pose a threat to social media platforms by spreading and manipulating messages, there is a need for rapid detection and measurement of bots.Many social

media bot detection algorithms have been developed to identify Twitter bots. To achieve consistent bot classification, the bot scores generated by the algorithm should not be so different that agents can be classified as bots or non-bots. Bot classification is usually based on bot scores. If the account's bot score is above the threshold, the account will be classified as a bot, if the score is below the threshold, the account will be classified as a non-bot.

II. RELATED WORK

Classification models [26] are often used to detect robot attacks. These models usually use machine learning-based classes such as k-nearest neighbor (k-nn), naive bayes (nb), decision making (dt), support vector machine (svm), random forest (rf), and Bayesian necessary network (bn). This method can only analyze the operation of one robot; however, it will not be able to see the operation of many good robots. Strayer et al. [27] introduced a discovery model to focus on traffic in c&c using association and integration methods. Botnet malware groups traffic data and evaluates the similarities in activity to group each traffic. In this study, they used j48 decision tree, c4.5, nb and bn. At the end of the process, the transmitted packets are analyzed. They ranked the data size according to the packet delivery time. However, this study focuses on only one bot and is limited to internet relay chat (irc) and command and control (c&c) traffic. Also, it is not possible to capture the workers involved in the swarm operation. [28] Presented a search model to identify traffic queries to specific dns addresses. In this study, traffic classification is performed and location and ip address are periodically checked. All point-to-point traffic queries in a segment are compared with the traffic in the other segment. Here, the same data is divided into groups of bots. This research is continued by measuring the uniformity of suspicious bot traffic, removing patterns of bot behavior to show the distribution of activity over the same sites and locations.

If different segments have a similar distribution, they are set (marked by 1); otherwise, they are reset (0). This approach has shown relatively good results. However, a problem arises if bot group activities exist in the transition between segments. In this case, crucial activity information may be lost, making the method not optimal. Furthermore, their proposed model is limited to the DNS query, which does not represent the actual environment that can be varied depending on the attack purposes. Chowdhury et al. [29] introduced a graph-based clustering model to detect bot activities. A bot, represented by a node, connects to other hosts through a vertex. Some features can be extracted from this design: in- and out-degrees, in- and out-degree weights, clustering coefficients, node betweenness, and eigenvector centrality, which are the inputs to the self-organizing map (SOM)-based clustering process. It delivers the value of each fulfilled cell representing nodes, which construct a bot group activity cluster. This model can detect bot activity in the network data flow and recognize its behavior. Nevertheless, it fails to analyze the correlation between activities in the bot group.

On the other hand, bot activity can construct a group based on similarity [30] or causality between bots. In further research, Hostiadi et al. propose the B-Corr model [30], which measures the activity similarities using the intersection-probability approach. First, it detects a single bot activity using some classification algorithms and takes the best results. Next, the B-Corr model extracts bot flows into features, such as inbound, outbound, inbound degree, and outbound degree. After getting the features, this approach traces the similarity value of each feature based on network-header flows, such as the IP address, port address, protocol, and total packets. The information intersection of the network header in each feature is taken for the probability of similarities between bot activities. Finally, the similarity of the target is examined based on the intersection probability. The results show that their approach can detect bot group activities well. However, it does not show correlations such as the causality factor, considering that bot activities may influence each other.In a later study, Hostiadi et al. [30] Represent activities in the form of chains. They used time-based segmentation to identify relationships and then used sliding windows for optimization. They also track the average performance of each department; this is called multilevel analysis, and its similarity to other departments is measured using cosine similarity. Similar workplaces are organized to perform a single job from beginning to end. Experimental results show that the study can detect the chain of relationships between the warring forces from one point to another. However, it does not show how important the work done by the robot is in terms of causality. This information is important in determining how much the current robotic process will affect the future and vice versa. This complex problem is studied in more detail in [31], which gives the conditions for the attack. Create a timeline here. Research has shown that robots can interact with other robots depending on the timing of their activities. Moreover, this activity is defined as dynamic. This paper [32] proposes a new model for group detection of robots using a hybrid analysis method, which includes the use of sliding window segmentation technology to extract modeling activities, analyze similar activities of robots, and analyze their relationships. This experiment uses two demographic data to test the proposed method. The results show that it can detect the group of workers with 99.73% accuracy, which is better

than other methods, and with a false alarm rate of less than 1%. Friendship features are extracted from followers' profiles. The feature extraction formula is designed to scale as the number of followers increases. Two classifiers are developed to evaluate the efficiency and effectiveness of the proposed classes. The classifier is compared on several real-world datasets. Experimental results show that the classifier outperforms Botometer in a classification task. Scalability is evaluated by analyzing the detection efficiency of the distribution when the behavior is close to high-ego users for a certain number of accounts. Generalization ability is also verified by comparing different datasets. Finally, the performance and early discovery of social bots are discussed. These bots are usually the most active partners in terms of content distribution. Therefore, tools that identify contributors (e.g., those used to recognize and credit members' contributions) should take bots into account and not include them. While there are methods for identifying bots in software repositories, these methods are not perfect and may miss some bots or identify some human accounts as bots. In this paper [33], we investigate the accuracy of searching for bot operations on 540 code snippets from 27 GitHub projects. We show that the no bot detection system is accurate enough to detect bots of the top 20 contributors for each project. We find that combining these techniques can improve bot detection accuracy and recovery. We also emphasize the importance of considering bots when enabling human participation, as bots are among the top contributors and are responsible for most of the interaction. The research focuses on cases where the account is closed for private reasons and the bot must be identified by the friends list.

The results show the possibility of high-accuracy bot detection on private accounts through simple off-the-shelf algorithms combined with extensive data. Random forest classification performed best when roc auc was greater than 0.9 and for was less than 0.3. In the paper, we also discuss the limitations of the experiment and plans for future research. This requires considering the search bot from the ranking analysis. However, this method alone only checks a certain number of compound words. Therefore, the nature of hybrid accounts on GitHub is unclear and the lack of appropriate data makes it difficult to study this issue. In this paper [34], we examine three review-level classification models and evaluate their classification of combined posting data. We find that the accuracy and yield of the best products according to this classification model are between 88% and 96%. However, even the most accurate classifiers cannot correctly identify hybrid accounts; we find that text content alone or text combined with patterns used by bots is useful for identifying both bots and hybrid accounts. Our research calls for a more accurate bot detection system that can identify hybrid accounts to provide more insight into software maintenance activities performed by humans and bots in social coding. Using the author requires more methods to avoid detection, and new methods are needed to distinguish legitimate accounts from bot accounts. In this paper [35], we propose to use a classifier to improve twitter bot detection.

Unlike traditional bot detection approaches that have strict requirements on data sources (e.g., private payload information, social relationships, or activity histories), this paper [36] proposes a detection method called BotFlowMon that relies only on NetFlow data as input to identify OSN bot traffic, where every NetFlow record is a summary of a traffic flow on the Internet and contains no payload content. BotFlowMon introduces several new algorithms and techniques to help use machine learning to classify the social bot traffic from the real OSN user traffic, including aggregating NetFlow records to obtain transaction data, fusing transaction data to extract features and visualize flows, as well as subdividing transactions into basic actions. Our evaluation shows that with 535GB raw NetFlow records as input, BotFlowMon can efficiently classify the traffic from social bots, including chatbots, amplification bot, post bot, crawler bot, and hybrid bot, with 92.33-93.61 % accuracy.

Twitter is a web application playing dual roles of online social networking and micro-blogging. The popularity and open structure of Twitter have attracted many automated programs, known as bots. Legitimate bots generate a large amount of benign contextual content, i.e., tweets delivering news and updating feeds, while malicious bots spread spam or malicious contents. To assist human users in identifying who they are interacting with, this paper [37] focuses on the classification of human and spambot accounts on Twitter, by employing recurrent neural networks, specifically bidirectional Long Short-term Memory (BiLSTM), to efficiently capture features across tweets. To the best of our knowledge, our work is the first that develops a recurrent neural model with word embeddings to distinguish Twitter bots from human accounts that requires no prior knowledge or assumption about users' profiles, friendship networks, or historical behavior on the target account. Moreover, our model does not require any handcrafted features. The preliminary simulation results are very encouraging. Experiments on the cresci-2017 dataset show that our approach can achieve competitive performance compared with existing state-of-the-art bot detection systems.

III. PROPOSED WORK

The proposed model for Bot Operated Account Detection vs. Human Operated account detection method is

based on past tweeting history of the user. Certain attributes such as friends, followers count, and favorites were considered as features for designing a classifier to detect Bots. In the proposed model, the historical behavior based on user posted tweets are the main concerned for detecting all the accounts. The profile page of a twitter user displays information about name, location, pictures of profile, number of tweets, retweets and replies posted by the user.

1) Starting with the assumption that the behaviors of a bot account is more predictable and less random than the behaviors of a genuine human-operated account. An approach based on the idea of string compression (Tweet-To-String-T2S) is designed to measure the randomness and the predictability of the behaviors of a Twitter account. A lossless compression algorithm can be used to compress the string for an account with compression ratio being used as a metric to measure the predictability of an account's behaviors.
2) The vector representation of the data would be more informative for understanding the content. In the vector representation, a value 1 represents the presence of the word at index k in the vocabulary and a value 0 represents the absence of it. However, these vectors may be very long and most of the entries may be 0. For these improvements, some word embedding techniques are used that reduce the dimensional space.
3) Word2Vec is a popular technique which is used in the proposed thesis. A pre-trained Word2Vec model contains near about 400millions tweets in English like language over a 400-dimensional feature space were used to represent the contents of the tweets. These features laterare used for training any supervised model for bot detection.
4) To get more details about a Twitter user, some subtle features need to be analyzed that capture the behavior of the account for longer time. If a Twitter account is being created for some specific purpose, then the account will Tweet and retweet much more than the human account for some specific purpose? These features would help more to distinguish between human operated accounts and bot accounts.
5) By some observations, it has been noticed that the bot operated accounts follow a greater number of accounts than human accounts for getting more followers and at the same time they have lesser number of friends. The "Follower-to- friend" ratio is a useful trend for detecting bot account shown in the literature above. Bot tends to have a higher value of this ratio.
6) For Classifying the "Bot operated account" vs. "Human operated account", the bot operated accounts will be represented by 1 (positive class) and "Human operated account" will be represented by 0 (negative class). The following classification algorithm along with proposed algorithm will be implemented and compared for the results:
7) 1. Random forest Classifier on different datasets of size.
8) 2. Logistic Regression
9) 3. Support Vector Machine.

3.1 Proposed system architecture

The Feature Extraction Process is shown below:

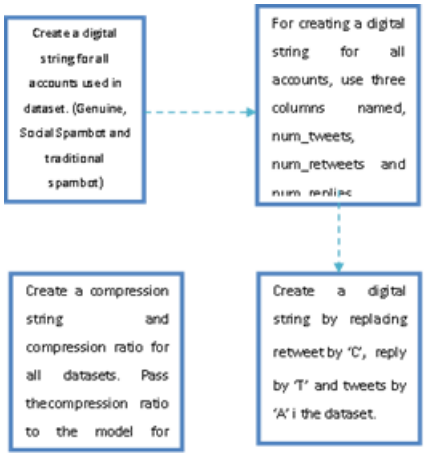

Figure 3.1: Feature Extraction steps used in proposed model.

A digital string generation and compression ratio calculation step is shown below in figure 4.3 as a sample.

The dataset consists of "User Information" and "Tweets" information. The detailed about "User Information" dataset columns is given below:

- Id- unique identification number.
- Name- It is the username of the user.
- Screen Name- It is the Screen name displayed on the account
- Statuses Count- The number of Tweets (including retweets) issued by the user.
- Follower's count- follower count is a measure of your overall reach and influence over your target audience.
- friends_count- The number of user's particular account is following.
- Favourites_count- The number of times tweets of user had been Favorited.
- Listed_count- It shows how many people have added you to a list.
- Url- Uniform Resource Locator of a Twitter User.
- Lang- language used by a user on Twitter Platform.

- time zone- Clock Time of the country the account is operated from.
- Location- Location of the user account.
- Default_profile- Is the number of twitter profile.
- Default_profile_image- Default profile image on user account.
- Geo_enabled- Location Enabled.
- Profile_image_url- URL of user profile image.
- Profile_banner_url- It allow user to customize the expensiveness of their profiles.
- Profile_use_background_image- image used in the background.
- profile_background_image_url_https
- profile_text_color
- profile_image_url_https
- profile_sidebar_border_color
- Profile_background_tile- is used for repeating background images.
- Profile_sidebar_fill_color- Profile sidebar color used for filling.
- Profile_background_image_url_https- image Url of the profile background.
- Profile_background_color- background color of the profile.
- Profile_link_color- The linking color used for profile.
- Utc_offset- It is the coordinated Universal time for world.
- Is_translator- Translators used.
- follow_request_sent
- protected
- verified
- notifications
- description
- contributors_enabled
- Test_set_1- It is the target variable used for detection.
- Test_set_2- It is target variable used. It is having values 0 or 1.

Tweet Information: The information has the data related to user tweets. It has the following columns.

Id, text, source, user-id, in-reply-to-status –id, in_reply_to_user_id, in_reply_to_screen_name, retweeted_status_id, geo, place, contributors, retweets_count, reply_count, Favorite_Count, Favorited, retweeted, Possibly sensitive, Num hashtags, Num URLs, num_mentions, Created_at, timestamp, crawled_at, updated.

IV. RESULTS AND COMPARISON

4.1 Results of Classifiers on the following Features:

- Statuses Count
- Followers Count
- Friends Count
- Favorites Count
- Listed Count

Dataset 1: Mixed set of 50% genuine accounts + 50% social spambots1.

	SVM Algorithm	**Random Forest**	**Logistic Regression**
Accuracy	74.98	91.31	74.53
Precision	94.89	98.79	93.36
Recall	52.57	83.55	52.57
F1 Score	67.66	90.50	67.26

Table 4.1: Comparisons of results for three classifiers.

Dataset 2: Mixed set of 50% genuine accounts + 50% social spambots3.

	SVM Algorithm	**Random Forest**	**Logistic Regression**
Accuracy	56.07	69.28	68.34
Precision	81.69	96.72	91.75
Recall	12.50	38.14	38.36
F1 Score	21.68	54.71	54.10

Table 4.2: Comparisons of results for three classifiers.

Results of dataset 1 with existing as well as proposed behaviors features with string.

Algorithm/ Parameters	**SVM Algorithm**	**Random Forest**	**Logistic Regression**
Accuracy	92.90	99.04	97.70
Precision	100.00	84.65	98.20
Recall	85.80	78.75	97.70
F1 Score	92.30	80.04	97.70

Table 4.3: Comparisons of results for three classifiers.

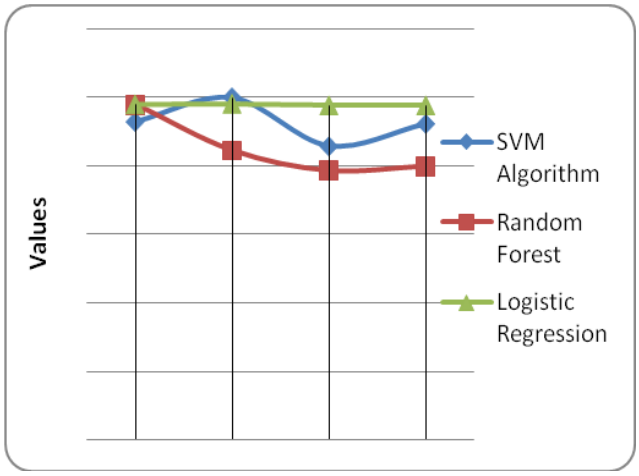

Figure 4.1: Comparison Chart for all three classifiers.

V. CONCLUSION

With the fast-growing popularity of online social networks (OSN), maintaining the security of OSN ecosystems becomes essential for the public. Among all the security threats facing OSN, malicious social bots have become the most common and detrimental. These bot programs are often employed to violate users' privacy, distribute spam, and disturb the financial market, posing a compelling need for effective social bot detection solutions.

Social media bots can change society's perspective in different aspects of life. This thesis analyses sentiment features and their effect on the accuracy of machine learning models for social media bot detection. Social bots can use tweet sentiment to create a backfire effect and confirmation bias to create a fake trend or change public opinion. This work is based on Tweets behaviour and social effects inherent in tweets' behaviour based on the string patterns. The new set of sentiment features are extracted from a tweet's posting patterns and used to train bot detection models.

REFERENCES

[1] M. Workman, "An empirical study of social media exchanges about a controversial topic: Confirmation bias and participant characteristics," social media in Society, pp. 381–400, 2018.

[2] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," SIGKDD Explorations, vol. 19, no. 1, pp. 22–36, 2017.

[3] Z. Rajabi, A. Shehu, and O. Uzuner,"A multi-channel bilstm-cnn model for multilabel emotion classification of informal text," in 2020 IEEE 14th International Conference on Semantic Computing (ICSC), pp. 303–306, 2020.

[4] Z. Rajabi, A. Shehu, and H. Purohit, "User behavior modelling for fake information mitigation on social web," in Social, Cultural, and Behavioral Modeling (R. Thomson, H. Bisgin, C. Dancy, and A. Hyder, eds.), (Cham), pp. 234–244, Springer International Publishing, 2019.

[5] H. Karbasian, H. Purohit, R. Handa, A. Malik, and A. Johri, "Real-time inference of user types to assist with more inclusive and diverse social media activism campaigns," in Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, pp. 171–177, 2018.

[6] A. Rajabi, C. Gunaratne, A. V. Mantzaris, and I. Garibay, "Modeling disinformation and the effort to counter it: A cautionary tale of when the treatment can be worse than the disease," in Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, pp. 1975–1977, 2020.

[7] L. Madahali and M. Hall, "Application of the Benford's law to social bots and information operations activities," in 2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), pp. 1–8, CyberSA, 2020.

[8] M. Heidari, J. H. J. Jones, and O. Uzuner, "Deep contextualized word embedding for text-based online user profiling to detect social bots on twitter," in IEEE 2020 International Conference on Data Mining Workshops (ICDMW), ICDMW 2020, 2020.

[9] F. Husain, J. Lee, S. Henry, and O. Uzuner, "Salamnet at semeval-2020 task12: Deep learning approach for Arabic offensive language detection," in International Workshop on Semantic Evaluation (SemEval) 2020, 2020.

[10] F. Husain and O. Uzuner, "Transfer learning approach for arabic offensive language detection system – Bert-based model," in 2021 4th International Conference on Computer Applications Information Security (ICCAIS) - Contemporary Computer Technologies and Applications, 2020.

[11] A. Sabzehzar, G. Burtch, Y. Hong, and T. Raghu, "The role of religion in online pro-social lending," in 40th International Conference on Information Systems, ICIS 2019, Association for Information Systems, 2019.

[12] A. Sabzehzar, Y. Hong, and T. Raghu, "People don't change, their priorities do: Evidence of value homophily for disaster relief," in 41st International Conference on Information Systems, ICIS 2020, Association for Information Systems, 2020.

[13] S. Cresci, R. D. Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race," in Proceedings of the 26th International Conference on

World Wide Web Companion, Perth, Australia, April 3-7, 2017, pp. 963– 972, 2017.

[14] Z. Chen and D. Subramanian, "An unsupervised approach to detect spam campaigns that use botnets on twitter," CoRR, vol. abs/1804.05232, 2018.

[15] P. N. Howard, S. Woolley, and R. Calo, "Algorithms, bots, and political communication in US 2016 election: The challenge of automated political communication for election law and administration," Journal of Information Technology & Politics, 2018.

[16] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, "Botornot: A system to evaluate social bots," in Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume, pp. 273–274, 2016.

[17] W. M. Campbell, C. K. Dagli, and C. J. Weinstein, "Social network analysis with content and graphs," Lincoln Laboratory Journal, 2013.

[18] N. Chavoshi, H. Hamooni, A. Mueen, Temporal patterns in bot activities, in: Proceedings of the 26th International Conference on World Wide Web Companion, 2017.

[19] F. Amato, A. Castiglione, A. De Santo, V. Moscato, A. Picariello, F. Persia, G. Sperlí, Recognizing human behavior's in online social networks, Comput. Secur. 74 (2018) 355–370.

[20] C. CAI, L. Li, D. Zengi, Behavior enhanced deep bot detection in social media, in: IEEE International Conference on Intelligence and Security Informatics (ISI), 2017.

[21] S. Kudugunta, E. Ferrara, Deep neural networks for bot detection, Inform. Sci. 467 (2018) 312–322.

[22] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: Large-scale information network embedding, in: Proceedings of the 24th International Conference on World Wide Web, 2015.

[23] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[24] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi, Fame for sale: Efficient detection of fake Twitter followers, Decis. Support Syst. 80 (2015) 56–71.

[25] Sneha Kudugunta, Emilio Ferrara, Deep neural networks for bot detection, Information Sciences, Volume 467, 2018, Pages 312-322, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2018.08.019.

[26] Hoang, X., Nguyen, Q., 2018. Botnet detection based on machine learning techniques using DNS query data. Future Internet 10 (5), 43

[27] Strayer, W.T., Walsh, R., Livadas, C., Lapsley, D., 2006. Detecting botnets with tight command C and control. Source, 195–202.

[28] Choi, H., Lee, H., Lee, H., Kim, H., 2007. Botnet detection by monitoring group activities in DNS traffic. In: CIT 2007 7th IEEE Int. Conf. Comput. Inf. Technol., pp. 715–720.

[29] Chowdhury, S., Khanzadeh, M., Akula, R., Zhang, Zhang, S., Medal, H., Marufuzzaman, M., Bian, L., 2017. Botnet detection using graph-based feature clustering. J. Big Data 4 (1).

[30] Hostiadi, D.P., Ahmad, T., Wibisono, W., 2020. A new approach of botnet activity detection model based on time periodic analysis. In: 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), pp. 315–320.

[31] Hostiadi, D.P., Ahmad, T., 2021. Dataset for botnet group activity with adaptive generator. Data Br. 38, 107334.

[32] Dandy Pramana Hostiadi Tohari Ahmad," Hybrid model for bot group activity detection using similarity and correlation approaches based on network traffic flows analysis", Elsevier-2022.

[33] M. Golzadeh, A. Decan and N. Chidambaram, "On the Accuracy of Bot Detection Techniques," 2022 IEEE/ACM 4th International Workshop on Bots in Software Engineering (BotSE), 2022, pp. 1-5, doi: 10.1145/3528228.3528406.

[34] N. Cassee, C. Kitsanelis, E. Constantinou and A. Serebrenik, "Human, bot or both? A study on the capabilities of classification models on mixed accounts," 2021 IEEE InternationalConference on Software Maintenance and Evolution (ICSME), 2021, pp. 654-658, doi: 10.1109/ICSME52107.2021.00075.

[35] Jorge Rodríguez-Ruiz, Javier Israel Mata-Sanchez, Raul Monroy, Octavio Loyola-González, Armando Lopez-Cuevas, A one-class classification approach for bot detection on Twitter, Computers & Security, Volume 91, 2020,101715,ISSN 0167-4048, https://doi.org/10.1016/j.cose.2020.101715.

[36] Y. Feng, J. Li, L. Jiao and X. Wu, "BotFlowMon: Learning-based, Content-Agnostic Identification of Social Bot Traffic Flows," 2019 IEEE Conference on Communications and Network Security (CNS), 2019, pp. 169-177, doi: 10.1109/CNS.2019.8802706.

[37] F. Wei and U. T. Nguyen, "Twitter Bot Detection Using Bidirectional Long Short-Term Memory Neural Networks and Word Embeddings," 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), 2019, pp. 101-109, doi: 10.1109/TPS-ISA48467.2019.00021.