Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2022) 30: 1269 – 1283
© TÜBİTAK
doi:10.55730/1300-0632.3848

Evaluation of social bot detection models

M. Buğra TORUSDAĞ[1,*], **Mucahid KUTLU**[2], **Ali Aydın SELÇUK**[2]
[1]Cyber Security Research and Development Department, TURKCELL, Istanbul, Turkey
[2]Department of Computer Engineering, Faculty of Engineering, TOBB University of Economics and Technology, Ankara, Turkey

Received: 10.07.2021 • **Accepted/Published Online:** 03.02.2022 • **Final Version:** 31.05.2022

Abstract: Social bots are employed to automatically perform online social network activities; thereby, they can also be utilized in spreading misinformation and malware. Therefore, many researchers have focused on the automatic detection of social bots to reduce their negative impact on society. However, it is challenging to evaluate and compare existing studies due to difficulties and limitations in sharing datasets and models. In this study, we conduct a comparative study and evaluate four different bot detection systems in various settings using 20 different public datasets. We show that high-quality datasets covering various social bots are critical for a reliable evaluation of bot detection methods. In addition, our experiments suggest that Botometer is preferable to others in order to detect social bots.

Key words: Twitter, Social Bot Detection, Evaluation, Reproducibility

1. Introduction

Using online social networks (OSN) has become one of the main daily habits for many people. OSNs are incredibly convenient for communicating with people, freely expressing opinions on any topic, and following news about any location on Earth. However, OSNs can also be used to harm others in various forms, such as sharing misinformation [1], cyberbullying [2], spreading malware, and others.

While we have to deal with harmful activities of people on OSN platforms, there is another critical factor amplifying these problems: *social bots*, which are basically programs used to perform OSN activities (e.g., posting messages and following other accounts) automatically. Artificially increasing the amount of a message or any other OSN activity can be used for many different goals. For instance, we can make a particular topic listed in the "trending topic" of Twitter to spread our message globally. In addition, we can artificially increase the number of followers of a particular account (e.g., a political candidate or a celebrity) to give an impression that the corresponding person is potentially influential on many other people. Furthermore, sophisticated bots can interact with other people and might impact their stance on various issues (e.g., vaccines and political elections) by spreading false information. For instance, a recent study reports that half of the accounts tweeting about Covid-19 are likely to be bots[1].

Due to the potential harms of social bots in our daily life, OSN companies focus on removing social bots from their platforms. For instance, Twitter suspends thousands of accounts used for propaganda activities from

[*] Correspondence: bugratorusdag@gmail.com

[1] NPR (2020). Corona Virus Live Update [online]. Website https://www.npr.org/sections/coronavirus-live-updates/2020/05/20/859814085/researchers-nearly-half-of-accounts-tweeting-about-coronavirus-are-likely-bots [accessed 28/11/2021].

time to time[2]. In addition, social scientists working with data gathered from social media platforms usually have to remove posts of social bots to analyze the data accurately. However, manually detecting social bots is highly time-consuming and costly. Therefore, many researchers have focused on detecting social bots automatically [3–6]. Bot detection studies have one challenging goal: to conduct a Turing Test automatically. This exciting and challenging problem has certain restrictions in the context of OSN platforms. While the investigator in the original Turing Test is able to ask questions to programs (or humans) interactively, we do not have this ability in detecting social bots. Basically, these models use tweets (e.g., [3, 6]), metadata about profiles (e.g., [5]), and social network information (e.g., [4]) to identify social bots.

While automatically applying a Turing Test is a very challenging problem, evaluation of these bot detection models is also highly demanding due to difficulties in labeling accounts, limitations in re-distribution of data, and the necessity of covering a wide range of social bots for a reliable evaluation and comparison against other systems. Therefore, in this work, we conduct a comprehensive set of experiments to evaluate and compare four existing social bot detection models using 20 different public datasets in various setups. In particular, we conduct experiments 1) with datasets used in the original studies and 2) with datasets previously not used for them.

The main findings in our experiments are as follows: Firstly, models achieve high prediction accuracy on datasets used in the original studies. However, their performance dramatically decreases when we use different datasets for testing. This suggests that we need robust models and more labeled datasets covering various types of social bots. Secondly, some datasets do not provide the data required by Cresci et al. [3], reducing our ability to compare systems. Therefore, we believe that datasets for social bot detection should be standardized to enable effective comparison of models. Thirdly, the dynamic structure of social accounts and restrictions in sharing datasets are the main obstacles to comparing systems. Therefore, we need static and public datasets for a fair comparison of systems. Lastly, Botometer seems to be the best performing one among the tested models. We share our code to maintain the reproducibility of our results[3].

The rest of the paper is organized as follows: We provide the literature review in Section 2. Section 3 explains the selected models. Section 4 describes the datasets used in our evaluation. We present our experiments in Section 5. We discuss the limitations of our work and challenges in the evaluation of bot detection systems in Section 6 and conclude in Section 7.

2. Related work

A number of researchers focus on the social bot problem from different aspects [7], such as detecting automatically-generated posts [8] and predicting victims vulnerable to social bot attacks [9]. However, the majority of researchers working on this problem developed solutions to detect bots. In this section, we discuss studies to detect bot accounts (Section 2.1) and studies evaluating these detection methods (Section 2.2).

2.1. Bot detection systems

A number of researchers focused on how to detect social bots automatically. The majority of the studies propose a machine learning-based solutions using several features such as social graphs [10, 11], meta-data about users (e.g., the number of followers and account creation date) [4, 12], and content of social media posts [4, 12, 13]. Machine learning models they utilize include Random Forest [4, 14, 15], CNN-LSTM [12], BoostOR [16] and

[2] Washington Post (2018). Technology [online]. Website https://www.washingtonpost.com/technology/2018/07/06/twitter-is-sweeping-out-fake-accounts-like-never-before-putting-user-growth-risk/ [accessed 28/11/2021].

[3] https://github.com/bugratorusdag/Bot-Detection-Systems

others. In addition to supervised models, prior work also investigated unsupervised methods [17, 18] assuming that social bots are similar in terms of post content or behavior, but different from real users.

The machine learning models rely on the quality of datasets used to train them. As we also show in our work, their performance can change dramatically on bot types that do not exist in their train sets. Therefore, prior work also investigated crowdsourcing to detect social bots [18, 19]. Cresci et al. [18] show that crowdworkers are able to detect traditional social bots, but not social spambots. In our work, we do not develop a new social bot detection model. However, we investigate the robustness of social bot detection models using several datasets and compare them.

Regarding the type of social media platforms, most of the studies are conducted on Twitter [7]. However, there are also studies using data from other platforms such as Reddit [11], Sina Weibo [20], and Facebook [19]. In our work, we use tweet datasets because of their availability.

2.2. Evaluation of bot detection models

Similar to our work, several studies focus on evaluating existing social bot detection models to detect their vulnerabilities. Boshmaf et al. [21] created social bots and used them to connect to users. In their large-scale infiltration study, they report that Facebook Immune System [22] is evaded with a success rate of up to 80%. Similarly, Elyashar et al. [23] target Facebook users of specific organizations and show that they can infiltrate targeted users with a 50-70% success rate. Freitas et al. [24] created 120 different social bots running on Twitter with different profiles and creating their own tweets. They report that Twitter has detected only 31% of these social bots.

In addition to evaluating the security mechanisms of social media platforms, prior work also explored the effectiveness of bot detection methods. Grimme et al. [25] created social bots to evaluate the performance of Botometer. In particular, they used three different social bots: 1) only retweeting bots, 2) bots that are partially controlled by humans, and 3) bots retweeting particular tweets and posting tweets from a pool of manually created ones.

Cresci et al. [26] also developed sophisticated social bots which can evade Cresci et al. [3]'s method. Sayyadiharikandeh et al. [27] evaluated the performance of Botometer using varying datasets and showed that its performance decrease when train and test sets are from different domains. In our work, we investigate the impact of train data and compare it with other detection methods.

Many social scientists also started to use social bot detection methods, especially Botometer due to its handy API, to conduct research based on data collected from social media platforms (e.g., [28, 29]). However, this massive use of bot detection methods in other disciplines also raised the questioning validity of findings due to the imperfect classification of these models. Martini et al. [30] investigate three bot detection methods commonly used by social scientists. They report that the accounts identified as bots by each method vary a lot, and recommend paying special attention to selecting bot detection methods. Gallwitz and Kreil [31] discuss the theoretical flaws of social bot detection methods and manually investigate how reliable classifications of these methods are in a real-world setting. They report that Botometer has a significant false-positive problem such that the majority of the accounts detected as "bot" by Botometer are actually operated by humans. Rauchfleisch and Kaiser [32] focus on analyzing Botometer's performance due to its popularity and report that its performance might vary across languages and over time. Due to the high false negative and false positive results of Botometer, they recommend manual assessment of its results when used in social science research. In our work, we compare Botometer against other methods and show that Botometer outperforms others, suggesting that Botometer is still the best choice among other detection systems. However, due to its questionable performance reported by prior work, its results should be taken with a grain of salt and analyzed carefully.

3. Evaluated bot detection systems

We use four different social bot detection systems in our study, covering unsupervised and supervised approaches with various sets of features and machine learning algorithms. Selected systems drew the attention of many researchers, yielding a high number of citations. In addition, their authors share codes or API for these studies, enabling us to use them in our experiments easily. Now, we explain these systems in detail.

3.1. Bot detection system of Cresci et al.

Instead of using user metadata and tweets separately, Cresci et al. [18] propose an unsupervised solution in a holistic manner. In particular, each user's timeline is transformed into sequences similar to DNA chains to represent the activity of users. They use two different strategies for this transformation. In the first strategy, each tweet is transferred to one of the nucleobases: Simple tweets → A, retweets → C, and reply tweets → T. The other strategy focuses on tweet content, and there exist six different types of content for this strategy: tweets with URL → A, tweets with hashtag → T, tweets with mention → C, tweets with media → G, tweets with more than one label → X, simple tweets → N. In order to classify accounts, Cresci et al. calculate similarities between DNA chains (i.e., timelines of users) using longest common subsequence (LCS) similarity. Subsequently, users are clustered based on their similarity scores. The decision line is set to where LCS between accounts starts decreasing dramatically. The accounts with high similarity are considered bots, assuming that bots will have similar activity patterns. As Cresci et al. show that the former strategy outperforms the latter, we report results only for the first strategy. We will refer to the bot detection system of Cresci et al. as $\mathbf{BDS_{Cresci}}$ in the rest of the paper.

3.2. Bot detection system of Efthimion et al.

Efthimion et al. [33] investigate the impact of several features, including message sharing frequency, the similarity of message contents based on Levenshtein distance, the number of followers, total number of posts, the ratio of followers to friends, absence/presence of user name, description, and profile image, whether the description contains a link, and others. As their primary model, they use a scaled version of Support Vector machines (SVM) with a set of profile-based features. The features they use are different variations of tweet counts, follower counts, and user bio information. We will refer to the bot detection system of Efthimion et al. as $\mathbf{BDS_{Efthimion}}$ in the rest of the paper.

3.3. Bot detection system of Kudugunta and Ferrara

Kudugunta and Ferrara [5] work on two classification tasks: 1) user-level classification to determine whether a bot controls an account, and 2) tweet-level classification to determine whether a tweet's author is a bot or not. They show that even a single tweet can be helpful to detect whether an account is a bot or not. User-level classification uses only ten different features, including the verified and protected status of accounts, the number of statuses, followers, friends, favorites, and listed counts, and whether the profile is in the default setting, geo-enabled, and using a background image. They also apply synthetic minority oversampling and generate additional labeled examples. They investigate different learning algorithms and show that Adaboost yields the highest performance for user-level classification in their experiments. We will refer to the bot detection system of Kudugunta and Ferrara as $\mathbf{BDS_{Kudugunta}}$ in the rest of the paper.

3.4. Botometer

Botometer [6] is considered one of the state-of-the-art bot detection models. Botometer has been evolving since its first version was published in 2016 [4], formerly known as BotorNot. It is now a commercial product with an API that provides the probability of being a bot for a given Twitter account. Botometer uses Random Forest classifier with more than 1000 features which can be grouped into six different groups: user, friend, content, sentiment, network, and timing of sharing tweets. Thus, Botometer needs both the user and tweet data of accounts to predict their likelihood of being a bot.

4. Datasets and social bots used for evaluation

Datasets used in the evaluation of bot detection models play a crucial role in assessing the performance of models and comparing them. Therefore, it is essential to have labeled datasets that accurately represent the real-world environment. Accordingly, the size of the dataset, label distribution, and types of social bots are all important factors impacting evaluation reliability. However, it is challenging to build a high-quality labeled dataset for bot detection models because there can be many different types of bots used in real life, and new types of social bots can always be developed. In addition, the labeling process has its own challenges. For example, manually judging whether a Twitter account is a social bot by just examining their profile is a challenging process. Therefore, we mostly rely on the best guess of assessors. Nevertheless, in order to achieve a reliable evaluation, we collected 20 different public datasets that cover different types of social bots. Now we explain the datasets used in this work.

Table 1 presents details about datasets used in this study[4]. Datasets shown in rows 1-10 in Table 1 are used for training and testing Botometer [6]. **Botwiki** dataset consists of active bot accounts from botwiki.org. **Celebrity** dataset contains metadata for celebrities collected by CNetS team [34]. **Cresci Stock** has been constructed to analyze the impact of bots on stock markets by tracking specific cashtags [35, 36]. Accounts collected have been manually labeled. **Cresci Rtbust** [37] targets Italian retweeter bot accounts, which are used to increase the visibility and popularity of tweets and accounts artificially. Genuine and bot accounts have been manually labeled. **Pronbots** dataset consists of bots sharing scam tweets[5]. No detail about how data is collected and judged is provided, but later it has been used in other academic studies (e.g., [34]). **Botometer Feedback** consists of accounts selected based on the feedback from users of Botometer and then labeled manually by one of the authors of [34]. **Gilani** consists of manually annotated accounts grouped by their number of followers [38]. In particular, accounts have been grouped into four groups: celebrity status (i.e., more than 9M followers), very popular (i.e., between 900K and 1.1M followers), mid-level recognition (i.e., between 90K and 110K followers), and lower popularity (i.e., between 0.9K and 1.1K followers). Subsequently, four annotators manually judged each account and discussed each other to reach a final label. **Vendor-Purchased** have been constructed by purchasing fake followers from several companies [34]. **Verified** dataset contains accounts verified by Twitter. This dataset is mainly used to balance the unbalanced datasets such as Botwiki and Vendor-Purchased. **Political Bots** has been shared by a Twitter user (@josh_emerson) and includes bots sharing political content. **Fake Followers** consist of bot accounts used for artificially increasing the followers. The bots have been purchased by Yang et al. [34].

Datasets in rows 11-15, 17-19 of Table 1 have been created by Cresci et al. [18]. **Genuine Accounts** dataset contains only accounts controlled by humans. In the construction of the dataset, Cresci et al. [18]

[4] The datasets can be accessed at https://botometer.osome.iu.edu/bot-repository/datasets.html

[5] Github (2018). pronbot2 [online]. Website https://github.com/r0zetta/pronbot2 [accessed 28/11/2021].

randomly selected accounts and asked a question in natural language to understand whether they are operated by humans or not. Social Spambots (rows 13-15) consist of sophisticated bots with complex tweet structures and filled profile pages. **Social Spambots 1** consists of bots used in one of the Italian elections to propagate a candidate's campaign. Bots in **Social Spambots 2** constantly share a hashtag to promote a mobile phone application. Bots in **Social Spambots 3** advertise Amazon products by spamming URLs of products on sale. Traditional spambots (rows 16-19) are easier to detect than social spambots because of their simple profile structure. **Traditional Spambots 1** is a training set developed by Yang et al. [39]. It contains bots posting URLs linking to malicious content. Bots in **Traditional Spambots 2** post tweets by adding a few usernames to the tweets and URLs that offer money to deceive people. Bots in **Traditional Spambots 3 and 4** also have a simple profile structure because they repeatedly publish only job offers. **Cresci-17** is actually a combination of Fake Followers, Genuine Accounts, Social Spambots 1-2-3, and Traditional Spambots 1-2-3-4 datasets. Lastly, the **NBC Russian Trolls** dataset has been published by NBC News[6] and contains Russia-linked trolls which performed malicious activities during the 2016 U.S. Presidential Election to manipulate the opinion of voters. It is noteworthy that the accounts in this dataset are not bots. We use this dataset in our experiments because Efthimion et al. [33] also utilize it in their experiments, ignoring differences between trolls and bots. Nevertheless, evaluations on this dataset should be taken with a grain of salt.

Table 1. Datasets used in our study.

Row	Dataset Name	# of Bots	# of Humans	Provided Data Type	Used in Studies
1	Botwiki	698	0	Metadata	[6]
2	Celebrity	0	19997	metadata	[6]
3	Cresci Stock	7102	6174	metadata	[6]
4	Cresci Rtbust	353	340	metadata	[6]
5	Pronbots	17882	0	metadata	[6]
6	Botometer Feedback	139	380	metadata	[6]
7	Gilani-17	1090	1413	metadata	[6]
8	Vendor Purchased	1087	0	metadata	[6]
9	Verified	0	1987	metadata	[6]
10	Political Bots	62	0	metadata	[6]
11	Fake Followers	3351	0	metadata & tweets	[33], [6]
12	Genuine Accounts	0	1083[7]	Metadata & tweets	[3], [33], [5], [6]
13	Social Spambots 1	991	0	metadata & tweets	[3], [33], [5], [6]
14	Social Spambots 2	3457	0	metadata & tweets	[33], [5], [6]
15	Social Spambots 3	464	0	metadata & tweets	[3], [33], [5], [6]
16	Traditional Spambots 1	1000	0	metadata & tweets	[33], [6]
17	Traditional Spambots 2	100	0	metadata	[33], [6]
18	Traditional Spambots 3	403	0	metadata	[33], [6]
19	Traditional Spambots 4	1128	0	metadata	[6]
20	Cresci-17	10894	1083	metadata & tweets	[6]
21	NBC Russian Trolls	453	0	metadata & tweets	[33]

While we could compile a long list of datasets, the data type provided in each dataset varies significantly. For instance, datasets in rows 1-9 and 17-19 contain only metadata about users, not actual tweets of users. This

[6] NBC News (2018). Tech & Media [online]. Website https://www.nbcnews.com/tech/social-media/now-available-more-200-000-deleted-russian-troll-tweets-n844731 [accessed 28/11/2021].

[7] While there are 3474 human accounts in the original dataset, tweet data of some accounts are out-of-order. Therefore, we use the dataset in [34] which has tweets for 1083 human accounts.

limits our capability to use them in our evaluations because the information used by each model also varies significantly.

5. Experiments

In this section, we explain the implementation details of bot detection models (Section 5.1), describe the experimental setup (Section 5.2), and present results (Section 5.3).

5.1. Implementation details of bot detection models

Being a commercial product, Botometer is a handy tool to detect social bots. The Botometer API can easily analyze any existing Twitter account. However, this also makes conducting experiments extremely challenging. Firstly, its code is not available due to its commercial status. Secondly, it is not easy to implement it from scratch following the explanations in the respective papers because it utilizes more than one thousand features, and the model keeps changing with its new versions. Therefore, it is hard to confirm whether the same features explained in the paper are used in the actual API. Thus, in our experiments with Botometer, we use its API. Upon our request, Cresci et al. [18] kindly shared their code to calculate LCS between accounts. In this model, we also need to set the discrimination point between bots and humans (i.e., the threshold value to consider accounts as bots). We pick the threshold value when the similarity scores start to decrease dramatically, as done by the authors. However, it is still challenging to pick a single value by manually examining LCS figures. Therefore, we tried different values within the range where similarity scores decreased dramatically and picked the one yielding the highest scores. For the other models, we use the code shared by the authors of each model. We keep all hyper-parameters as they are in the original codes.

5.2. Experimental setup

In our experiments, our main goal is to evaluate bot detection models at different setups and investigate their robustness. We consider two different setups: 1) experiments with datasets used in the papers of the selected models, and 2) experiments with other datasets. In all experiments, we compare all available models for the respective dataset. $\text{BDS}_{\text{Cresci}}$ requires actual tweets to transform user timelines to DNA chains. Therefore, we are not able to use it with datasets shown in rows 1-10, 17-19 of Table 1. $\text{BDS}_{\text{Efthimion}}$ and $\text{BDS}_{\text{Kudugunta}}$ have been used with all datasets. Botometer API works on only existing Twitter accounts. Unfortunately, many accounts in our datasets are suspended or deleted. This situation is very likely for bot detection datasets because Twitter periodically suspends bot accounts. In addition, the existing accounts will most likely have different profiles than the ones in the crawled datasets due to the profile changes over time. Alternatively, we could open a Twitter account and fill the profile and timeline as in accounts in our datasets. However, Botometer model requires too many pieces of information about accounts that are not available in our public datasets, such as the profile link color and information about a retweeted tweet's owner. Therefore, unfortunately, we could not conduct our own experiments with Botometer on the datasets we have. To mitigate this problem, we compare the results of Botometer reported in [27] and [6] with $\text{BDS}_{\text{Efthimion}}$ and $\text{BDS}_{\text{Kudugunta}}$ using the same experimental setup (see Section 5.3.3).

5.3. Experimental results

5.3.1. Evaluating with datasets used in the original papers

In this experiment, we use the datasets utilized in the evaluation of $\text{BDS}_{\text{Cresci}}$, $\text{BDS}_{\text{Efthimion}}$, and $\text{BDS}_{\text{Kudugunta}}$. This allows us to investigate the reproducibility of their reported results and compare them on different setups.

Furthermore, this enables us to investigate the impact of small changes on training and test datasets on the performance of the models because similar sets of datasets have been used in the evaluation of these three models. In particular, we construct four different datasets using the ones presented in Table 1: 1) combination of all datasets used in [3] (i.e., rows 12, 13, and 15 of Table 1), 2) combination of datasets that contain tweets and used in [33] (i.e., rows 11-16 and 20 of Table 1), 3) combination of all datasets used in [5] (i.e., rows 12-15 of Table 1), and 4) combination of all datasets used in [3, 5, 33] (i.e., rows 11-18 and 20 of Table 1). In all our experiments, datasets are split as %80 for training and %20 for testing.

$\text{BDS}_{\text{Cresci}}$ is based on clustering of accounts. Therefore, we use all accounts for clustering and then calculate the classification accuracy on the test data. We are not able to provide results for $\text{BDS}_{\text{Cresci}}$ on the fourth dataset because Traditional Spambots 2 and 3 datasets (used in forming the fourth dataset) do not have tweets required by $\text{BDS}_{\text{Cresci}}$. The results are shown in **Table 2**.

Table 2. Comparison of $\text{BDS}_{\text{Cresci}}$, $\text{BDS}_{\text{Efthimion}}$ and $\text{BDS}_{\text{Kudugunta}}$ with datasets used in the original studies.

	$\text{BDS}_{\text{Cresci}}$ [3]			**$\text{BDS}_{\text{Efthimion}}$** [33]			**$\text{BDS}_{\text{Kudugunta}}$** [5]		
Dataset	**Acc**	**F1**	**AUC**	**Acc**	**F1**	**AUC**	**Acc**	**F1**	**AUC**
Datasets in [3]	0.44	0.18	0.21	0.92	0.85	0.91	0.98	0.99	0.98
Datasets w/ tweets in [33]	0.38	0.41	0.27	0.94	0.95	0.94	0.99	0.99	0.99
Datasets in [5]	0.31	0.007	0.01	0.96	0.97	0.94	0.98	0.97	0.98
Datasets in [3, 5, 33]	–	–	–	0.94	0.96	0.95	0.98	0.96	0.98

Our observations are as follows. $\text{BDS}_{\text{Efthimion}}$ and $\text{BDS}_{\text{Kudugunta}}$ achieve very high classification performance. Remarkably, $\text{BDS}_{\text{Kudugunta}}$ outperforms the others and achieves almost perfect classification. Therefore, we can say that the results for $\text{BDS}_{\text{Efthimion}}$ and $\text{BDS}_{\text{Kudugunta}}$ are in line with their reported performance scores. On the other hand, the performance of $\text{BDS}_{\text{Cresci}}$ is dramatically lower than the others. Cresci et al. [3] report 0.95 F1 score in their experiments for $\text{BDS}_{\text{Cresci}}$. However, they use the whole dataset in their experiments while we calculate accuracy based on only 20% of the dataset to compare it with the others. We have also calculated its F1 score based on all accounts in the datasets to investigate this issue further. In this way, $\text{BDS}_{\text{Cresci}}$ also achieves 0.95 F1 score in the first dataset, confirming their findings. However, its F1 score for the second and third datasets are 0.45 and 0.31, respectively, when we use all data. Therefore, our experiments show that $\text{BDS}_{\text{Cresci}}$'s performance varies dramatically even when we use similar sets of datasets. This suggests that their assumption on the similar online activity of social bots does not hold.

5.3.2. Experiments with different datasets

As seen in Table 1, models we selected except Botometer use different combinations of datasets in rows 11-20 in their reported experiments, but do not use datasets in rows 1-10. Therefore, in this experiment, we use all datasets in rows 11-20 for training and the others for testing to see their performance on never-seen datasets. However, in this experiment, we cannot use $\text{BDS}_{\text{Cresci}}$ because datasets in rows 1-10 do not contain tweets of accounts. Thus, we report results only for $\text{BDS}_{\text{Efthimion}}$ and $\text{BDS}_{\text{Kudugunta}}$.

Some of the available datasets contain only bots (e.g., Botwiki) or only human accounts (e.g., celebrities). In order to reduce this extreme bias in label distribution, we conduct our experiments on different combinations of datasets in rows 1-10. **Table 3** presents the results.

Our observations for the results in Table 3 are as follows. Firstly, the results for both systems are generally lower than the results in Table 2. For instance, in Table 2, $\text{BDS}_{\text{Efthimion}}$'s accuracy ranges between 0.92 and

Table 3. Accuracy (ACC), precision (PRE), recall (REC), F1, and AUC scores of BDS$_{\text{Efthimion}}$ and BDS$_{\text{Kudugunta}}$ when models are trained using datasets in rows 11-20 of Table 1 and tested on varying datasets.

	BDS$_{\text{Efthimion}}$					**BDS$_{\text{Kudugunta}}$**				
Test Dataset	**ACC**	**PRE**	**REC**	**F1**	**AUC**	**ACC**	**PRE**	**REC**	**F1**	**AUC**
Botwiki & Botometer Feedback (B+BF)	0.57	0.79	0.49	0.61	0.61	0.24	0.14	0.27	0.18	0.25
Celebrity & Pronbots (C+P)	0.83	0.89	0.72	0.79	0.82	0.88	0.96	0.8	0.87	0.88
Cresci Stock	0.65	0.67	0.68	0.68	0.65	0.59	0.79	0.16	0.27	0.56
Cresci Rtbust	0.56	0.66	0.29	0.41	0.57	0.46	0.34	0.12	0.18	0.45
Gilani	0.58	0.6	0.15	0.24	0.54	0.37	0.41	0.28	0.34	0.38
Vendor Purchased & Verified & Political Bots (VP+V+PB)	0.78	0.88	0.45	0.6	0.71	0.84	0.8	0.995	0.89	0.77

0.96, and BDS$_{\text{Kudugunta}}$ achieves almost perfect accuracy scores, ranging between 0.98 and 0.99. However, both models' accuracy scores never reach 0.9 in Table 3, and in some cases, their accuracy and F1 scores are very low. For instance, BDS$_{\text{Efthimion}}$ and BDS$_{\text{Kudugunta}}$ achieve 0.57 and 0.24 accuracy scores, respectively, on the B+BF dataset. This suggests that both models overfit in these training datasets and perform low on unseen data. Secondly, both models' performance on C+P and VP+V+PB datasets are noticeably higher than their performance on other datasets. This suggests that both models learn specific types of bots from the training data and perform well on particular datasets. Therefore, our experiments show that it is crucial to cover different bot types as much as possible during training. Lastly, we do not observe a clear winner in the comparison of both models. In terms of the F1 score, BDS$_{\text{Kudugunta}}$ outperforms BDS$_{\text{Efthimion}}$ in 3 datasets (VP+V+PB, Gilani, and C+P) while BDS$_{\text{Efthimion}}$ outperforms BDS$_{\text{Kudugunta}}$ in the other three datasets. However, considering the average performance on six datasets (ignoring differences in dataset sizes), BDS$_{\text{Efthimion}}$ outperforms BDS$_{\text{Kudugunta}}$.

5.3.3. Comparison with Botometer

As explained in Section 5.2, we are not able to run Botometer on our datasets. Therefore, in order to compare its reported performance with other models we selected, we assess the performance of BDS$_{\text{Efthimion}}$ and BDS$_{\text{Kudugunta}}$ using the same experimental setup in papers of Botometer. As there are several papers about Botometer and its versions, we first focus on the work of Sayyadiharikandeh et al. [27], in which the performance of Botometer's most recent version (i.e., v4) is analyzed. In particular, Sayyadiharikandeh et al. report precision and recall scores of Botometer.v4 in varying train and test sets (Figure 1 of [27]). We use the same experimental setup as theirs. In particular, we apply 5-fold cross-validation when the same dataset is used for training and testing. When different datasets are used for train and testing, the test data is divided into five-folds, and average performance is calculated for consistency with in-domain experiments, as Sayyadiharikandeh et al. do. In this experiment, they use four different datasets, yielding 16 different cases. However, we do not have access to two of these datasets[8]. Therefore, we could use only two datasets, namely Cresci-17[9] and Botometer Feedback, yielding four different train-test combinations. The results are shown in **Table 4**.

We observe that Botometer outperforms others in all cases in terms of precision. Regarding recall, Botometer outperforms others when Cresci-17 dataset is used for testing. However, it ranks second in both cases

[8] We tried to reach the authors for the datasets. However, unfortunately, we did not receive any response.

[9] We downloaded the dataset from the link authors share. However, the statistical numbers they share about this dataset do not match the data we downloaded. In particular, we have 10894 bots and 3474 human-controlled accounts; they report 7049 bots and 2764 humans (in Table 2 of [27]). Therefore, the results regarding Cresci-17 should be taken with a grain of salt.

Table 4. Comparison of Botometer.v4, BDS$_{\text{Efthimion}}$ and BDS$_{\text{Kudugunta}}$ using various train and test datasets. The results for Botometer is obtained from [27]. The best result for each case is highlighted.

	TEST SET					
	Datasets	**Models**	**Cresci-17**		**Botometer Feedback**	
			Precision	**Recall**	**Precision**	**Recall**
TRAIN SET	**Cresci-17**	BDS$_{\text{Efthimion}}$	0.95	0.83	0.42	0.36
		BDS$_{\text{Kudugunta}}$	0.92	0.81	0.39	0.15
		Botometer	0.98	0.98	0.62	0.34
	Botometer Feedback	BDS$_{\text{Efthimion}}$	0.81	0.32	0.65	0.42
		BDS$_{\text{Kudugunta}}$	0.73	0.59	0.75	0.81
		Botometer	0.97	0.79	0.84	0.70

when Botometer Feedback dataset is used for testing. As expected, the in-domain performance of Botometer and BDS$_{\text{Kudugunta}}$ is higher than their performance in cross-domain. Interestingly, when Botometer Feedback is used for training BDS$_{\text{Efthimion}}$, its in-domain performance is lower than its cross-domain performance.

In order to further investigate the performance of models, we check the prediction of BDS$_{\text{Kudugunta}}$ and BDS$_{\text{Efthimion}}$ for each account in the test sets. We do not cover Botometer in this analysis because we do not have access to its output. Our main observations are as follows. Firstly, note that we have two different cases for each model and test set because of using different train datasets. In both cases of testing BDS$_{\text{Efthimion}}$ on Botometer Feedback dataset, 190 (out of 517) accounts have been classified correctly, while 48 accounts have been misclassified. Therefore, the prediction of BDS$_{\text{Efthimion}}$ changes for 289 accounts (i.e., 48% of Botometer Feedback) as the train data changes. We observe this significant change in predictions when we also test BDS$_{\text{Efthimion}}$ with Cresci-17 dataset, too. On the other hand, in both cases of testing BDS$_{\text{Kudugunta}}$ on Botometer Feedback dataset, 356 (out of 517) accounts have been classified correctly while 60 accounts have been misclassified, i.e., predictions for 101 accounts (i.e., 19.5% of the dataset) change as train data changes. Therefore, we can say that BDS$_{\text{Kudugunta}}$'s predictions are more stable than of BDS$_{\text{Efthimion}}$. Secondly, we noticed that features of BDS$_{\text{Efthimion}}$ are not able to distinguish many accounts, causing the same feature vector for different accounts. In particular, there are only 80 (out of 517) unique feature vectors in Botometer Feedback and 120 (out of 14368) in Cresci-17 datasets. This might also be the reason for its higher cross-domain performance than its in-domain performance. We do not observe this situation in BDS$_{\text{Kudugunta}}$ because it has many non-binary features. Regarding BDS$_{\text{Kudugunta}}$, we noticed that tweet, follower, and friend counts are noticeably higher for accounts that are misclassified in all cases than accurately classified ones. Therefore, it shows that BDS$_{\text{Kudugunta}}$ is more likely to make mistakes for accounts with a higher number of tweets, followers, and friends.

Overall, our results suggest that Botometer.v4 is preferable to other models. In addition, while BDS$_{\text{Efthimion}}$ outperforms BDS$_{\text{Kudugunta}}$ in many cases in terms of precision and recall, its results seem less reliable due to the limited representation of data and high variance in its predictions. Nevertheless, the comparison relies on only two datasets, and the statistical numbers about Cresci-17 reported in [27] do not match ours, as mentioned before. Therefore, we also conduct another set of experiments with more datasets and compare Botometer.v3 with BDS$_{\text{Efthimion}}$ and BDS$_{\text{Kudugunta}}$ based on its reported results in Figure 2 of [6]. In particular, seven different datasets are formed using 15 datasets in Table 1. Then, for each dataset, we train models using the other six datasets separately and calculate their AUC score. We use the same setup and evaluate BDS$_{\text{Efthimion}}$ and

BDS$_{\text{Kudugunta}}$. **Table** 5 shows the results. Botometer, BDS$_{\text{Efthimion}}$, and BDS$_{\text{Kudugunta}}$ achieve the highest AUC scores on 30, 8, and 4 cases (out of 42), respectively. This might be because Botometer employs many different features based on users' tweets and social network, while BDS$_{\text{Efthimion}}$ and BDS$_{\text{Kudugunta}}$ use only profile-based features. We note that in these experiments, we use the previous version of Botometer. However, it still outperforms the other two models. We also observe that the performances of models vary dramatically across cases. For instance, Botometer achieves 0.99 AUC score on Vendor & Verified dataset when trained with Bowtwiki and Verified datasets. However, its performance decreases to 0.45 when trained with Gilani-17 dataset and tested on Cresci Stock and Cresci Rtbust datasets. Therefore, our experiments point out the importance of 1) high-quality training datasets covering various social bots, and 2) using various datasets for an accurate evaluation of models. Overall, our experiments suggest that Botometer is preferable to other methods to detect social bots.

Table 5. Comparison of Botometer, BDS$_{\text{Efthimion}}$ and BDS$_{\text{Kudugunta}}$ using various train and test datasets. AUC results are shown. The results for Botometer is obtained from [6]. Each dataset is shown based on the row number in Table 1. The best result for each case is highlighted.

	TEST DATASETS								
	DATASETS	**Models**	**B-V**	**CS**	**CR**	**PF**	**G**	**VV**	**C**
TRAINING DATASETS	**Botwiki&Verified (B-V)**	BDS$_{\text{Efthimion}}$	-	0.67	0.5	0.45	0.54	0.68	0.88
		BDS$_{\text{Kudugunta}}$		0.54	0.58	0.53	0.55	0.89	0.77
		Botometer		0.64	0.71	0.5	0.61	0.99	0.87
	Cresci Stock (CS)	BDS$_{\text{Efthimion}}$	0.65	-	0.72	0.47	0.5	0.58	0.5
		BDS$_{\text{Kudugunta}}$	0.48		0.59	0.48	0.49	0.53	0.5
		Botometer	0.97		0.66	0.51	0.65	0.94	0.74
	Cresci Rtbust (CR)	BDS$_{\text{Efthimion}}$	0.67	0.68	-	0.49	0.49	0.72	0.5
		BDS$_{\text{Kudugunta}}$	0.88	0.64		0.54	0.46	0.79	0.73
		Botometer	0.97	0.7		0.6	0.57	0.83	0.68
	Political&Feedback (PF)	BDS$_{\text{Efthimion}}$	0.43	0.49	0.47	-	0.61	0.56	0.35
		BDS$_{\text{Kudugunta}}$	0.42	0.47	0.47		0.64	0.62	0.69
		Botometer	0.94	0.54	0.5		0.73	0.95	0.93
	Gilani-17 (G)	BDS$_{\text{Efthimion}}$	0.81	0.53	0.43	0.59	-	0.56	0.53
		BDS$_{\text{Kudugunta}}$	0.68	0.42	0.52	0.57		0.5	0.39
		Botometer	0.94	0.45	0.45	0.7		0.72	0.88
	Vendor&Verified (VV)	BDS$_{\text{Efthimion}}$	0.65	0.7	0.73	0.5	0.54	-	0.8
		BDS$_{\text{Kudugunta}}$	0.95	0.52	0.52	0.58	0.51		0.51
		Botometer	1.00	0.64	0.71	0.57	0.53		0.82
	Cresci-17 (C)	BDS$_{\text{Efthimion}}$	0.88	0.59	0.54	0.58	0.57	0.75	-
		BDS$_{\text{Kudugunta}}$	0.81	0.5	0.51	0.52	0.51	0.76	
		Botometer	0.97	0.58	0.57	0.72	0.67	0.95	

6. Discussion

Social bot detection is one of the most attractive research topics of recent years, and it is likely to continue due to two main factors.

Evolving nature of social bots. There is a rivalry between developers of social bots and bot detection systems. Therefore, social bot developers will always search for vulnerabilities of detection systems and modify their bots accordingly. This evolving nature of social bots also requires a continuous upgrade of detection systems, yielding a "never-ending clash", as Cresci [40] points out.

Advances in the field of natural language processing. Text generation models such as GPT-3 [41] can generate texts similar to human-authored ones. However, it still has serious flaws, providing hints about the actual "author" of texts such as maintaining consistency in genders and personality, lack of commonsense reasoning, and others [42]. Social bots using text generating models like GPT-3 will also require more advanced detection systems. We will need a more advanced analysis of texts to catch the hints of generated texts.

While we need more advanced detection systems in the future, reliable evaluation is crucial for an effective solution. However, there are several challenges regarding the evaluation of bot detection systems.

Annotation quality. Detecting whether an account is operated by a human or a bot is challenging, even for humans. This is because, unlike the traditional Turing test, annotators do not interact with the accounts. Therefore, annotators rely on the only data available in the profiles of accounts, possibly yielding incorrect annotations in some cases. In order to deal with this problem, prior work purchased real bots (e.g., Fake Followers and Vendor-Purchased datasets in our study) or user accounts that explicitly mention that they are controlled by bots (e.g., Botwiki dataset). However, most of the datasets rely on manual annotation based on inspection of user profiles. Although manual annotation might yield noisy labels, we still think that manual annotation is a reasonable choice for dataset creation to catch up with fast-evolving social bots. Furthermore, manually creating actual bots just for research studies might also be helpful to increase the quality of the datasets.

Limited coverage of social bots. Social bots have different automated activities depending on their target and the social media platform they are running. In addition, considering that bots are products of human imagination, social bot developers can quickly develop social bots with different behaviors than others. The diversity of social bots has a significant impact on the evaluation of bot detection systems. As we also observe in our experiments, models' performances vary across datasets. Therefore, we need datasets covering a wide range of social bots and continue developing datasets to cover bots of the future. While there are several datasets used by prior work, they are still not enough to cover all of them. For instance, the majority of the datasets focus on bots running on Twitter, not covering other social media platforms. Furthermore, many existing studies use a limited number of datasets to evaluate their systems, reducing the reliability of the reported performances. In our work, we evaluate systems on various datasets and observe that their performances vary dramatically on different datasets (See results in Table 2 and Table 3).

Inaccessible datasets. While we need datasets covering a wide range of social bots for a reliable evaluation, many datasets are not publicly available, mainly due to restrictions in sharing social media content. For instance, Twitter prohibits the re-distribution of its crawled content. Therefore, researchers can only share user or tweet IDs for their datasets, such that other researchers can re-download the data. However, tweets and user accounts can be deleted over time, preventing a fair comparison across studies due to having datasets with different sets of tweets and accounts. In our study, we also observed this problem such that we could not conduct experiments on some of the datasets. Therefore, it is vital to construct public and stable datasets enabling reliable evaluation and fair comparison of systems.

Limited share of models. Another challenge for comparison across models is that many studies do not share their code, and explanations in the papers are often not enough to implement them from scratch. In addition, many studies do not compare their work with others, making it challenging to identify the best-performing approaches. For instance, Kudugunta and Ferrara [5] do not compare their work with any other prior work.

The main goal of our work is to evaluate bot detection systems on a wide range of datasets and conduct a comparative study, reducing the research gap in this field. However, there are certain limitations of our work.

Generally, our work covers four studies for bot detection, and we could not cover all existing datasets due to limitations in data sharing. Therefore, a comprehensive study covering a higher number of bot detection systems and datasets is still needed. We leave this as future work.

7. Conclusion

In this work, we perform a comprehensive re-evaluation of four social bot detection systems, which are developed by Cresci et al. [3], Efthimion et al. [33], Kudugunta and Ferrara [5], and Yang et al. [6], using twenty different public datasets. Unfortunately, we are not able to use all models in all settings due to the lack of necessary data to run a model in some of the datasets. Therefore, we first compare the first three models with datasets used in experiments in the respective papers. Secondly, we evaluate models of Efthimion et al. [33] and Kudugunta and Ferrara [5] on datasets not used in these studies. Finally, we compare the last three models with several train-test dataset pairs.

Our comprehensive experiments confirm that studies achieve high prediction accuracy when they are evaluated with datasets used in those studies. However, their performance decreases dramatically when we use different datasets for testing. This points out the necessity of utilizing diverse datasets for reliable evaluation. In addition, our experiments suggest that Botometer is the best performing model among the models we investigate.

There are several research directions we would like to seek in the future. Firstly, we plan to evaluate the performance of bot detection methods for bots that utilize sophisticated text generation models such as GPT-3 [41]. Therefore, we will construct an annotated dataset covering bots sharing messages on various topics in different languages and performing miscellaneous social media activities. Furthermore, we plan to extend the studies we compare. In particular, we will also use bot detection systems based on crowdsourcing [18] and network topology [10]

Acknowledgment

This study was funded by the Scientific and Technological Research Council of Turkey (TUBITAK) ARDEB 3501 Grant No 120E514. The statements made herein are solely the responsibility of the authors.

References

[1] Vosoughi S, Roy D, Aral S. The spread of true and false news online. Science. 2018; 359 (6380):1146-51. doi: 10.1126/science.aap9559

[2] MacAvaney S, Yao HR, Yang E, Russell K, Goharian N et al. Hate speech detection: Challenges and solutions. PloS one. 2019; 14 (8):e0221152. doi: 10.1371/journal.pone.0221152

[3] Cresci S, Di Pietro R, Petrocchi M, Spognardi A, Tesconi M. Social fingerprinting: detection of spambot groups through DNA-inspired behavioral modeling. IEEE Transactions on Dependable and Secure Computing. 2017; 15(4):561–576. doi: 10.1109/TDSC.2017.2681672

[4] Davis CA, Varol O, Ferrara E, Flammini A, Menczer F. Botornot: A system to evaluate social bots. In: 25th International Conference Companion on World Wide Web; 2016. pp. 273–274.

[5] Kudugunta S, Ferrara E. Deep neural networks for bot detection. Information Sciences. 2018; 467:312–322. doi: 10.1016/j.ins.2018.08.019

[6] Yang KC, Varol O, Hui PM, Menczer F. Scalable and generalizable social bot detection through data selection. In: AAAI Conference on Artificial Intelligence. vol. 34; 2020. pp. 1096–1103.

[7] Orabi M, Mouheb D, Al Aghbari Z, Kamel I. Detection of bots in social media: a systematic review. Information Processing & Management. 2020; 57(4):102250. doi: 10.1016/j.ipm.2020.102250

[8] Almerekhi H, Elsayed T. Detecting automatically-generated arabic tweets. In: Asia Information Retrieval Symposium; 2015. pp. 123–134.

[9] Halawa H, Beznosov K, Boshmaf Y, Coskun B, Ripeanu M, Santos-Neto E. Harvesting the low-hanging fruits: defending against automated large-scale cyber-intrusions by focusing on the vulnerable population. In: 2016 New Security Paradigms Workshop; 2016. pp. 11–22.

[10] Cornelissen LA, Barnett RJ, Schoonwinkel P, Eichstadt BD, Magodla HB. A network topology approach to bot classification. In: Annual Conference of the South African Institute of Computer Scientists and Information Technologists; 2018. pp. 79–88.

[11] Hurtado S, Ray P, Marculescu R. Bot detection in reddit political discussion. In: Fourth International Workshop on Social Sensing; 2019. pp. 30–35.

[12] Ping H, Qin S. A social bots detection model based on deep learning algorithm. In: 2018 IEEE 18th International Conference on Communication Technology; 2018. pp. 1435–1439.

[13] Wang Y, Wu C, Zheng K, Wang X. Social bot detection using tweets similarity. In: International Conference on Security and Privacy in Communication Systems; 2018. pp. 63–78.

[14] Jr SB, Campos GF, Tavares GM, Igawa RA, Jr MLP, et al. Detection of human, legitimate bot, and malicious bot in online social networks based on wavelets. ACM Transactions on Multimedia Computing, Communications, and Applications. 2018; 14(1s):1–17. doi: 10.1145/3183506

[15] Igawa RA, Barbon Jr S, Paulo KCS, Kido GS, Guido RC, et al. Account classification in online social networks with LBCA and wavelets. Information Sciences. 2016; 332:72–83. doi: 10.1016/j.ins.2015.10.039

[16] Morstatter F, Wu L, Nazer TH, Carley KM, Liu H. A new approach to bot detection: striking the balance between precision and recall. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining; 2016. pp. 533–540.

[17] Chavoshi N, Hamooni H, Mueen A. Identifying correlated bots in twitter. In: International conference on social informatics; 2016. pp. 14–21.

[18] Cresci S, Di Pietro R, Petrocchi M, Spognardi A, Tesconi M. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In: 26th international conference on world wide web companion; 2017. pp. 963–972.

[19] Wang G, Mohanlal M, Wilson C, Wang X, Metzger M, Zheng H et al. Social Turing Tests: Crowdsourcing Sybil Detection. In: 20th Annual Network and Distributed System Security Symposium; 2013.

[20] Pan J, Liu Y, Liu X, Hu H. Discriminating bot accounts based solely on temporal features of microblog behavior. Physica A: Statistical Mechanics and its Applications. 2016; 450:193–204. doi: 10.1016/j.physa.2015.12.148

[21] Boshmaf Y, Muslukhov I, Beznosov K, Ripeanu M. Design and analysis of a social botnet. Computer Networks. 2013; 57 (2):556–578. doi: 10.1016/j.comnet.2012.06.006

[22] Stein T, Chen E, Mangla K. Facebook immune system. In: 4th Workshop on Social Network Systems; 2011. pp. 1–8.

[23] Elyashar A, Fire M, Kagan D, Elovici Y. Homing socialbots: intrusion on a specific organization's employee using socialbots. In: 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining; 2013. pp. 1358–1365.

[24] Freitas C, Benevenuto F, Ghosh S, Veloso A. Reverse engineering socialbot infiltration strategies in twitter. In: 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining; 2015. pp. 25–32.

[25] Grimme C, Preuss M, Adam L, Trautmann H. Social bots: Human-like by means of human control? Big data. 2017; 5 (4):279–293. doi: 10.1089/big.2017.0044

[26] Cresci S, Petrocchi M, Spognardi A, Tognazzi S. On the capability of evolved spambots to evade detection via genetic engineering. Online Social Networks and Media. 2019; 9:1–16. doi: 10.1016/j.osnem.2018.10.005

[27] Sayyadiharikandeh M, Varol O, Yang KC, Flammini A, Menczer F. Detection of novel social bots by ensembles of specialized classifiers. In: 29th ACM International Conference on Information & Knowledge Management; 2020. pp. 2725–2732.

[28] Giglietto F, Righetti N, Rossi L, Marino G. It takes a village to manipulate the media: coordinated link sharing behavior during 2018 and 2019 Italian elections. Information, Communication & Society. 2020; 23 (6):867–891. doi: 10.1080/1369118X.2020.1739732

[29] Pasquetto IV, Swire-Thompson B, Amazeen MA, Benevenuto F, Brashier NM, et al. Tackling misinformation: What researchers could do with social media data. The Harvard Kennedy School Misinformation Review. 2020. doi: 10.37016/mr-2020-49

[30] Martini F, Samula P, Keller TR, Klinger U. Bot, or not? Comparing three methods for detecting social bots in five political discourses. Big Data & Society. 2021; 8 (2):20539517211033566. doi: 10.1177/20539517211033566

[31] Gallwitz F, Kreil M. The Rise and Fall of “Social Bot” Research. SSRN. 2021.

[32] Rauchfleisch A, Kaiser J. The False positive problem of automatic bot detection in social science research. PloS one. 2020; 15(10):e0241045. doi: 10.2139/ssrn.3565233

[33] Efthimion PG, Payne S, Proferes N. Supervised machine learning bot detection techniques to identify social twitter bots. SMU Data Science Review. 2018; 1 (2):5.

[34] Yang KC, Varol O, Davis CA, Ferrara E, Flammini A, Menczer F. Arming the public with artificial intelligence to counter social bots. Human Behavior and Emerging Technologies. 2019; 1 (1):48–61.

[35] Cresci S, Lillo F, Regoli D, Tardelli S, Tesconi M. $ FAKE: Evidence of spam and bot activity in stock microblogs on Twitter. In: 12th International AAAI Conference on Web and Social Media; 2018. pp. 580–583.

[36] Cresci S, Lillo F, Regoli D, Tardelli S, Tesconi M. Cashtag piggybacking: Uncovering spam and bot activity in stock microblogs on Twitter. ACM Transactions on the Web. 2019; 13 (2):1–27. doi: 10.1145/3313184

[37] Mazza M, Cresci S, Avvenuti M, Quattrociocchi W, Tesconi M. Rtbust: Exploiting temporal patterns for botnet detection on twitter. In: 10th ACM Conference on Web Science; 2019. pp. 183–192.

[38] Gilani Z, Farahbakhsh R, Tyson G, Wang L, Crowcroft J. Of bots and humans (on twitter). In: 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017; 2017. pp. 349–354.

[39] Yang C, Harkreader R, Gu G. Empirical evaluation and new design for fighting evolving twitter spammers. IEEE Transactions on Information Forensics and Security. 2013; 8 (8):1280–1293. doi: 10.1109/TIFS.2013.2267732

[40] Cresci S. Detecting malicious social bots: story of a never-ending clash. In: 1st Multidisciplinary International Symposium on Disinformation in Open Online Media; 2019. pp. 77–88

[41] Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J et al. Language models are few-shot learners. arXiv preprint arXiv:200514165. 2020.

[42] Elkins K, Chun J. Can GPT-3 Pass a Writer’s Turing Test? Journal of Cultural Analytics. 2020; 1(1):17212 doi: 10.22148/001c.17212