How to Protect the Public Opinion Against New Types of Bots?

1st Jan Ludwig Reubold
KASTEL Security Research Labs
Karlsruhe Institute of Technology
Karlsruhe, Germany
jan.reubold@kit.edu

2nd Stephan Cornelius Escher
KASTEL Security Research Labs
Karlsruhe Institute of Technology
Karlsruhe, Germany
stephan.escher@kit.edu

3rd Christian Wressnegger
KASTEL Security Research Labs
Karlsruhe Institute of Technology
Karlsruhe, Germany
christian.wressnegger@kit.edu

4th Thorsten Strufe
KASTEL Security Research Labs
Karlsruhe Institute of Technology
Karlsruhe, Germany
thorsten.strufe@kit.edu

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020758

***Abstract*—Automated accounts affect political discourse in online social networks and therefore pose a threat to public opinion. They manipulate the regular flow of discussions by, for example, spreading false news, polluting content, or changing the popularity of users/content. Meanwhile, 56% of online social network users express concern about online false news. Therefore, reliable detection of automated accounts, among other things, is crucial for protecting political discourse in our society. However, the task is complex and challenging. Recent studies show that state-of-the-art bot detection algorithms have severe generalization problems.**

In this paper, we study features for reliable generalized identification of bot behavior on Twitter. Therefore, we propose an ensemble model that combines multiple neural network architectures. In a preprocessing step, we vectorize tweet texts using BERTweet. Exploiting behavioral features, the model then uncovers patterns in the metadata via a feed-forward and a convolutional component and incorporates the vectorized tweet texts via a recurrent unit. Extensive Leave-One-Botnet-Out (LOBO) evaluations on 20 real-world bot data sets show state-of-the-art performance in all experiments and outperform related approaches in average and peak performance.

***Index Terms*—Bot detection, Deep learning, Online social networks, Twitter**

I. Introduction

Democracy derives from the Greek words *dēmos* (people) and *kratos* (rule). In contrast to forms of government where a single person or a small group holds power, democracy means *rule of the people*. Its foundation, the *public opinion*, is its greatest strength and most vulnerable part at the same time.

Inevitably, people's opinions must encompass more than what they know and what they experience/observe. The opinions are thus based on imagination and what others report. Therefore, information from outside the horizon of our observations is essential, influencing our actions. However, it is often suppressed by censorship or source protection, or difficult to access due to physical and social barriers. Nonetheless, the information from beyond people's reach and how we assimilate it are an essential part of the foundation of democracy.

Information distribution saw three main developments, the inventions of newspapers, the Web, and online social networks. While newspapers opened a constant and reliable window to the outside world, the Web promised the *democratization of news* and online social networks (OSNs) brought us a step closer to it. Today, the costs of distributing or consuming information are close to negligible. Each evolution lowered the threshold of social and physical barriers and significantly increased news coverage and -perspectives. In consequence, it theoretically strengthened democracies.

Today, we live in a world of a sheer unlimited amount of information, perspectives, and opinions. In theory, we have technologically reached a near-perfect setting for democratic societies. Yet, while we have seen these massive developments in information distribution, we also observe increasingly complex difficulties in evaluating them. With each invention, new challenges concerning the *public opinion* arose.

A decade ago, the rise of Facebook marked the beginning of the era of the social web. Today, online social networks (OSNs), such as Facebook, Instagram, YouTube, and Twitter, are attracting enormous attention and have a nearly ubiquitous reach. Unfortunately, while we recognized the potential of online social platforms during times of political turmoil around the world, similar problems surfaced in everyday life as before.

After initial praise, research began to highlight the potential negative impact on democracies [1]. Some of the studies focused on automated accounts. According to these, bots are used to spread political propaganda, manipulate discussions, or

This work in part was supported by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy – EXC 2050/1 – Project ID 390696704 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden and by funding of the Helmholtz Association through the KASTEL Security Research Labs (HGF Topic 46.23).

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

influence the popularity of users/content, among other things [2], [3], [4]. In particular, the influence of malicious bots on political opinion-forming and political discussions poses a threat to democratic societies.

Today, on average, 56% of OSN users are concerned about online false news [5]. While false news is predominantly created by human authors [6], natural language processing (NLP) is catching up. GROVER, a language processing model based on the architecture of GPT-2[1] outputs automatically generated text that is more trustworthy than human-written false news [7]. These results suggest that the automatic generation of trustworthy propaganda on a large scale is within reach.

While both humans and bots are more likely to spread false news than factual news (cf. Vosoughi et al. [8]), bots significantly accelerate the spread of false news. Shao et al. [2] reported how bots target influential users via replies and mentions, reinforcing the early stage of extended spread.

Thus, in recent years, research and society have recognized that bots play a key role in the context of malicious behavior in OSNs. In the face of reported election meddling[2], reliable detection of automated accounts is an essential building block for healthy public opinion.

So, what is the state-of-the-art in social bot detection? In many analyses, scientists resort to heuristics. Often, suspended accounts are interpreted as bots. However, a recent study by Majo et al. [9] reports that less than 1% of the suspended accounts were suspected or potential bots. In line with other research, they found that suspended accounts pursued specific polarizing political agendas.

Taking a look at other fields that rely on reliable bot detection (e.g., social science), we see that the 'Botometer' is the go-to choice [10]. While often used by scientists, research shows how limited this approach is [10], [11]. Regarding Botometer, Twitter remarked that binary judgments have real potential to poison our public discourse.[3]

So, general bot detection seems to be an unsolved problem. But what is the issue? The detection of known bot types can be solved with a labeled data set and a state-of-the-art classification approach. However, while the authors of Botometer report about near-perfect detection performances [12], Echeverria et al. [11] argue that the established evaluation methods are rigged and, thus, reported performance results are misleading. When the goal is to distinguish between automated and manual accounts, the detection performance regarding known bot types may be interesting but is not a reliable statement of the detection rate of automated accounts in general. Additionally, this approach leads to an arms race between development and detection [13].

While Echeverria et al. [11] proposed a new evaluation scheme to measure the detection performance of unknown bot types, we lack an approach for reliable detection of these types of bots.

In this paper, we contribute to the discussion on social bot detection with a novel approach for generic bot detection. Recent bot detection algorithms [12], [14] are optimized based on collected and labeled data sets of bot- and benign accounts. These models are thus trained and tested on the same pool of bots.

Echeverria et al. [11] emphasized that these approaches do not generalize. They overfit due to a feature selection process that focuses on the best combination of features for a given data set. As a result, they often include information that exploits artifacts in the data, which reduces generalization to other types of bots.

In this paper, we focus on the detection of unknown bot types. To detect unknown bots and break the arms race, we have to shift to general bot detection approaches. We pose the following question:

RQ: Is it possible to define/identify generic bot behavior that enables generalized bot detection on Twitter?

Therefore, we assume that bot behavior manifests itself in the form of patterns in aggregated activity data and consider only behavioral characteristics. We ignore information that only exploits artifacts of specific bot types in the data (e.g., username length or profile description), although this can improve performance in detecting specific bot families.

To achieve the best possible generalization, we use an ensemble of neural networks that filter different aspects of the available information. To measure and compare the performance and generalization capabilities of our approach, we use the evaluation strategy and data sets proposed and published in [11]. The results of extensive evaluations of Twitter data sets show that our model significantly outperforms the Botometer in terms of accuracy and stability, and generalizes therefore significantly more to new, previously unknown bot families.

II. Related Work

We start by looking at the current approaches to bot detection. Early approaches examined spam-related topics on the social web. Benevenuto et al. [15] collected a data set of Twitter usage. They manually labeled users as spammers or non-spammers and proposed an SVM classifier for detection.

To help human users understand who they are communicating with, Chu et al. [16] developed a model for identifying accounts as human, bot, or cyborg (i.e., bot-assisted human or human-assisted bot). Their approach consisted of a four-component model that combined entropy and machine learning-based information with account characteristics into a final decision-maker component.

To make social bot detectors available for the general public, Davis et al. [12] launched the Botometer (former BotOrNot) service in 2014. The free social bot assessment service uses more than 1000 features.

Then, in 2017, Cresci et al. [13] reported a new type of bot, called social bots. Empirical studies suggested that humans

[1] openai.com/blog/better-language-models/

[2] https://reut.rs/3AovisG

[3] https://blog.twitter.com/en_us/topics/company/2020/bot-or-not

and state-of-the-art detection approaches performed poorly in detecting these new bots because they closely mimicked benign user behavior. Research, therefore, examined the larger context and highlighted another promising approach to the task: collective behavior detection.

In-depth analyses of the cybercriminal ecosystem on social web platforms provided detailed information about the activities and scale of criminal accounts on Twitter and Facebook [17], [18], [19]. The researchers recognized that coordinated campaigns often operate through the same set of accounts. Therefore, in cooperation with Facebook [20], Renren [21], or YouTube [22], researchers proposed models that leverage detailed data on social network account activities. These models detect coordinated behavior patterns caused by malicious campaigning on the platforms.

For example, Chavoshi et al. [23] assumed that humans are not able to be highly synchronized over a long period. Therefore, they proposed an activity correlation model that does not require labeled data.

However, recent work has identified serious limitations of studies across the discipline [11], [24]. Echeverria et al. [11], for one, discussed the established evaluation scheme for bot detection approaches. They emphasized the lack of generalization when approaches are trained and tested on the same pool of bot data. Therefore, they proposed a Leave-One-Botnet-Out evaluation strategy (LOBO). Based on a collection of real-world data sets, the model measures the generalization capability of approaches by running tests on held-out bot types, i.e., bot types that were ignored during optimization. The results of the Botometer algorithm, e.g., suggest that modern approaches that use metadata do indeed fail in detecting new types of bots.

Vargas et al. [24] on the other, challenged the assumption that humans do not act in a highly synchronized manner. They showed that coordination is indeed not uncommon in Twitter communities. With a high detection rate for malicious coordination, 46% of legitimate coordinated activity was misclassified.

In this work, we, therefore, investigate whether generalized bot detection based on account activities rather than coordinated campaigns can achieve high detection rates in previously unknown bot families.

III. Designing a General Bot Detector

We can divide most bot detection models into two general groups. One uses the content and metadata of individual accounts on social networks [11], [12], [25]. The other uses coordinated activities in the network [20], [21], [22]. Recent studies have shown that the basic assumptions underlying coordinated behavior approaches may be flawed [24]. Therefore, we focus on the behavior of each account and ignore the coordinated behavior. However, Echeverria et al. [11] recently reported serious generalization problems with account-based metadata approaches. The introduction of variations in bot signatures – similar to encountering instances of new types – led to poor performance.

In our work, focused on unknown bots, we reconsider bot detection. We assume that the intentions of malicious activities leave detectable traces in the data. Therefore, we focus on metadata and ignore features that do not contain behavioral information. In particular, we ignore information that exploits artifacts of specific bot types. To measure the detection performance of unknown bots and potentially uncover generalization problems, we use the *Leave-One-Botnet-Out* (*LOBO*) evaluation strategy proposed by Echeverria et al. [11].

A. Behavioral Features

Our model distills the data to identify characteristic patterns of behavior. We rely on similar metadata to related approaches, but disregard features that do not contain information about the activity. Examples include account name length or profile descriptions.

We represent behavior by a set of 33 aggregated *user-*, *content-*, and *response-centric* features (see Table I). The *user-centric* data provide a general overview of an account, such as its overall lifetime, the number of Tweets published, or the number of friends and followers. It also contains information summarizing user activity by breaking down the total number of published Tweets into the number of Tweets, retweets, replies, and quotes. *Content-centric* information provides more details about an account's tweet activity. The data includes statistics about the content of Tweets, such as # of mentions shared, hashtags, or URLs. It also consists of a representation of an account's average Tweet (e.g., its Ø length or the Ø mentions, hashtags, and URLs). In this context, we define *domain diversity* as the number of unique hosts normalized to the total number of URLs. Finally, *response-centric* features contain information about the response to an account's activity. We measure response by the number of retweets, replies, and favorites an account or its average Tweet receives. In what follows, we refer to these features as metadata.

In addition, we consider the published Tweets. Therefore, we convert the raw Tweets into numerical vectors through tokenization and a BERT model. BERT, Bidirectional Encoder Representations from Transformers [26], is a sequence transduction model that replaces the recurrent layers with multi-headed self-attention and represents the state-of-the-art for various NLP tasks. Transformers can be trained much faster than recurrent or convolutional neural networks. The variant we use for our experiments, BERTweet [27], is pre-trained based on English Tweets.

B. Model Architecture

In addition to feature selection, model architecture also plays a crucial role in abstraction. We chose a standard feed-forward neural network (FFNN) and a convolutional neural network (CNN) as candidates for processing the metadata. The latter is because it is capable of highlighting certain feature combinations. The candidates for text processing were

TABLE I: List of feature-sets used in our studies; *User centric*: a collection of statistics on the tweeting behavior of a user; *Content centric*: statistics on the content of a users' Tweets combined with a machine-readable summary of the content; *Response-centric*: statistics of how others reacted towards the content of an account.

Categories	Features
User-centric	# Tweets, Lifetime, Ø Duration (Tweets), # Statuses, # Friends, # Followers, # Favorites, # Listings
↪ Tweet-behavior	# Original Tweets, # Retweets, # Replies, # Quotes
	Ratio of Tweets, -Retweets, -Replies, -Quotes
Content-centric	# Mentions, # Hashtags, # URLs, # Domains
↪ Mean-Tweet	Ø Tweet-length, Ø Mentions, Ø Hashtags, Ø URLs, Domaindiversity, Ø Pictures, Ø Geo-locations
↪ Text	BERTweet (vectorized Tweets)
Response-centric	# retweeted Tweets, # received Replies, # favorited Tweets, Ø retweeted Tweets, Ø received Replies, Ø favorited Tweets

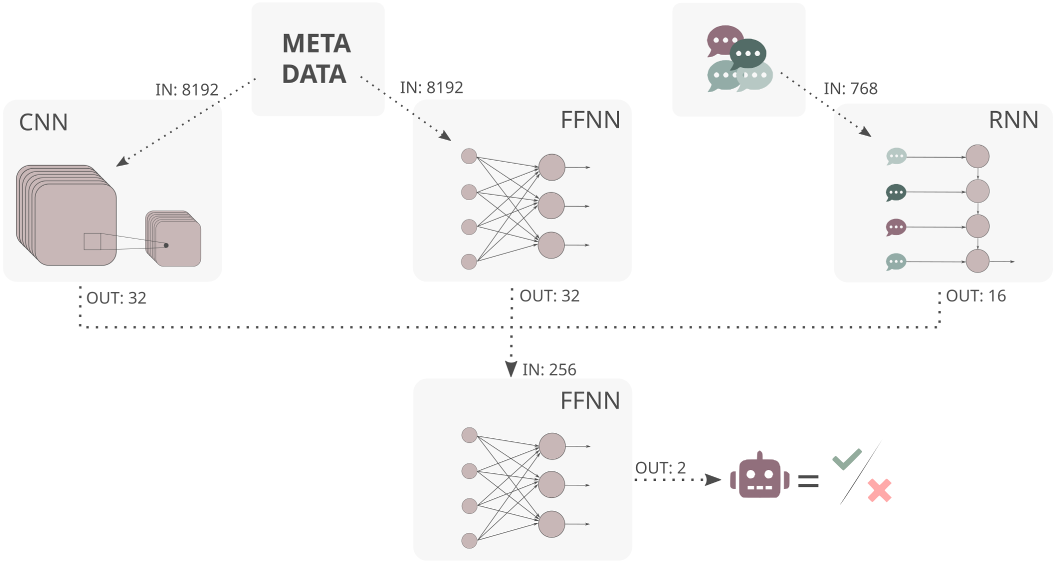


Fig. 1: Architecture of the proposed model; Meta-data is processed by a CNN and FFN separately; vectorized Tweets are run through an RNN; the outputs are combined by a final feed-forward unit.

a standard recurrent neural network (RNN) and a long short-term memory RNN (LSTM). We found that the metadata performance of different architectures varied depending on the bot types tested. Therefore, we assume that they provide different generalizations of the data. For texts, the simple RNN consistently performed better than the more sophisticated LSTM. Overall, we obtained the highest average and peak performance by combining the different metadata approaches with the RNN to form an ensemble model of neural networks. Figure 1 shows the final architecture.

The resulting model consists of (1) an FFNN, (2) a CNN, and (3) an RNN component combined by a final FFNN. Here, each of the 3 components processes the provided data and outputs a *summary*, i.e., a numerical vector. Depending on the architecture (FFNN vs. CNN), the model appears to take into account different aspects of the data.

Metadata is processed by an FFNN and a CNN in a normalized and standardized form. The FFNN component consists of 9 fully-connected layers arranged in a funnel shape. The 33-dimensional input vector (metadata) is connected with the first, 8192-dimensional (2^{13}) network layer. Accordingly, the layer sizes are $\{2^{13}, 2^{12}, \ldots, 2^{5}\}$, resulting in an 32-dimensional output vector.

In the CNN framework, the 33-dimensional input data is extended to 1024 dimensions using a linear layer. The convolutional network consists of a single 1D-convolutional layer with 30 output channels and a kernel size of 3. This is followed by a 1*D-MaxPool* layer, also with a size of 3. The output of the 30 channels is flattened and passed through a linear component-layer normalization combination. The length of the resulting output vector is 32.

Tweet texts are processed by an RNN. To obtain numerical vectors, Tweet texts are pre-processed with a transformer model. BERTweet transforms each Tweet into a 768-dimensional numerical vector. Then, a single RNN unit processes all transformed Tweets (768 × #Tweets) from a user and returns a 16-dimensional vector (final state of the RNN).

Combined are the three components by a final FFNN. Thus, we concatenate the outputs of the experts. The 80-dimensional (32 FFNN + 32 CNN + 16 RNN) input is fed into a 256-dimensional (2^8) network layer. We arranged the hidden layers in a funnel shape, with layer sizes $\{2^8, 2^7, \dots, 2^4\}$. A final linear layer (16 to 2) and a softmax unit give the final result.

IV. Experiments

In this section, we report evaluation results focusing on whether it is possible to define/identify generic bot behavior to enable generalized bot detection? To this end, we compare our model to state-of-the-art algorithms to evaluate its generalization capabilities. We use the model proposed by Echeverria et al. [11] and the popular Botometer as baselines. The evaluations are performed according to the LOBO scheme with balanced data sets. In addition to comparing with state-of-the-art approaches, we investigate the performance of our model in more detail, i.e., considering additional metrics and using performance progression results.

A. Evaluation Methodology: Leave-One-Botnet-Out

We are interested in measuring the detection performance in the context of variations in the signatures of different bot types. These variations can lead to poor performance when encountering instances of new types of bots. By using a *metadata set* consisting of different real bots, such a scenario can be simulated.

1) Methodology: While previous approaches used the same data basis for optimization and evaluation, *Leave-One-Botnet-Out* (*LOBO*) [11] relies on a collection of different real-world bot data sets. Bot types range from traditional- to social spam bots, to honeypot bots, to bots that attack individuals. The evaluation process, which is similar to cross-validation in its approach, then proceeds as follows: We optimize a model based on a training set with data samples from all but one bot type, augmented by an equal number of samples of benign users. Performance is measured on a data set consisting of samples of the withheld bot type balanced with benign user samples. We repeat the process for all bot types.

In addition, Echeverria et al. [11] proposed a sub-sampling of bot data to ensure that each known bot type is represented with the same number of data samples during training. Realizing that one does not always have the advantage of a large bot data corpus, they set the sample size to 500 ($C500$) and excluded all bot data with less than 500 samples from the evaluations.

We use their evaluation method, with minor adjustments: While we follow their strategy of excluding data of bot types with less than 500 samples for training, we still include them in the measurement of detection performance.

2) Data: The subsequent evaluations use data from 20 real-world data sets. The different data sets contain bot types, ranging from political bots, phishing bots, content polluters, or fake followers to silent accounts. An overview can be found in Table II.

The *metadata set* was published by Echeverria et al. [11] and contains content from various bot data sets. Some of these are from research [13], [28], [29], while others were reported by journalists who fell victim to a botnet attack. The data is supplemented by an equal number of benign user samples. Each sample includes information on the user profile and published Tweets.

Spambots range from the simplest bot type (TSB) to sophisticated social spambots (SSB) that mimic real user behavior. The TSB data sets consist mainly of bots used for traditional spam campaigns (TSB1, TSB2), with two of them (TSB3, TSB4) specifically spreading job offers.

SSB records contain accounts that mimic the behavior of real users, which makes bots more difficult to detect. Here, SSB1 consists of spammers of paid apps for mobile devices, while others (SSB2) retweet content from an Italian politician and (SSB3) promote products on Amazon.

All data sets (TSB, SSB) were previously used by Cresci et al. [13] and Echeverria et al. [11].

Fake-follower bot types consist of accounts that can be purchased by customers to follow their accounts to push them in visibility. The corresponding data sets contain fake followers from different services (fastfollowerz (FSF), intertwitter (INT), and twittertechnology (TWT). These types of accounts can be identified by synchronized behavior, but are very difficult to detect by behavioral analysis. For more information see Cresci et al. [30].

Attack-bots are Twitter accounts that participated in an attack on two journalists, Brian Krebs and Ben Nimmo (Krebs, Nimmo), in 2017. The journalists logged and published a list of the Twitter accounts involved[4].

Campaign-bots are bots detected by a bot detection service (DeBot) [23], [31]. The service provides daily reports on bot activity, focusing on warped correlation in Tweet timings of different accounts. Echeverria et al. [11] used the API to query over 700 000 accounts that were identified as bots. Therefore, the data set represents a potentially noisy sample as it is based on real detection results.

Mixed-bots contain data sets that were labeled by humans or were captured by honeypots (Darpa [4]). Thus, they may contain different types of bots. The different manually labeled bot accounts are grouped by the size of their followings:

B1k → follower counts between 900 and 1 100.
B100k → follower counts between 90 000 and 110 000.
B1M → follower counts between 900 000 and 1 000 000.
B10M → follower counts over 9 000 000.

[4] https://krebsonsecurity.com/tag/twitter-bots/

TABLE II: Overview of the data sets, information on their size, whether they were used for the Botometer optimization, and how they are used in our experiments (Train/Test).

Name	Size	Boto	Tr	Te
Social Spambots 1 (SSB1)	551	✓	✓	✓
Social Spambots 2 (SSB2)	3 320	✓	✓	✓
Social Spambots 3 (SSB3)	458	✓	×	✓
Traditional Spambots 1 (TSB1)	872	✓	✓	✓
Traditional Spambots 2 (TSB2)	1	✓	×	✓
Traditional Spambots 3 (TSB3)	283	✓	×	✓
Traditional Spambots 4 (TSB4)	977	✓	✓	✓
Fake-followers FSF	33	✓	×	✓
Fake-followers INT	64	✓	×	✓
Fake-followers TWT	624	✓	✓	✓
Human Annotated 1k (B1k)	387	✓	×	✓
Human Annotated 100k (B100k)	534	✓	✓	✓
Human Annotated 1M (B1M)	229	✓	×	✓
Human Annotated 10M (B10M)	26	✓	×	✓
Darpa	2 521	×	✓	✓
Attack on Brian Krebs (Krebs)	728	×	✓	✓
Attack on Ben Nimmo (Nimmo)	1 558	×	✓	✓
StarWars Bots	357 000	×	✓	✓
Bursty Bots	500 000	×	✓	✓
DeBot	700 000	×	✓	✓

Other bots finally belong to none of the above categories. These two data sets (StarWars, Bursty) contain samples of discovered botnets, one quoting from Star Wars novels and the other luring users to dubious websites through mentions. The StarWars bots were all created during a small window of time and have only a small number of friends and followers. The Bursty bots, on the other hand, all have similar characteristics in terms of a lifetime (only a few Tweets shortly after account creation), with no friends or followers. Both were reported by Echeverria *et al.* in [28], [32].

3) Optimization: Our model consists of one CNN, one RNN, and two FFNN units.The hyper-parameters given are the result of an exhaustive model-selection process. All layers have a dropout ratio of 0.3 and use a *LeakyReLU* activation function, with only the recurrent layer using a *ReLU* function. The learning rate is fixed at 10^{-5}. Our models are trained to convergence, with a simulated annealing strategy to adjust the learning rate during training.

B. Performance Comparison

In this section, we report on the performance of the algorithms. Our goal is to measure their abstraction capabilities, i.e., their detection performances on new, previously unknown bot types. Unlike related work, the feature selection of our algorithm is limited to behavioral information to obtain more abstract representations. In addition to the baseline comparisons, we are also interested in the impact of the different information sources. Therefore, we report the performance of an FFNN model (referred to as *META*) fed only with metadata information. In initial experiments, we have already excluded the Tweet-text-only approaches as they showed worse performance.

TABLE III: General Performance: Results of evaluations of Botometer (Bmeter), Echeverrias model, our META model (only using meta information), and our ensemble of experts; evaluations are split into 6 groups of bot sets; average accuracy and standard deviation of the approaches are at the bottom.

Data Set	Botometer	Eche	**META**	**Expert**
SSB1	0.924	0.492	0.763	**0.949**
SSB2	**0.994**	0.007	0.871	0.938
SSB3	**0.941**	–	0.745	0.919
TSB1	**0.983**	0.022	0.750	0.846
TSB2	**1.0**	–	0.893	**1.0**
TSB3	0.661	–	0.566	**0.827**
TSB4	**0.978**	0.020	0.789	0.948
FSF	**1.0**	–	0.474	0.909
INT	**1.0**	–	0.563	0.891
TWT	**0.953**	0.888	0.698	0.757
B1k	0.209	–	0.821	**0.875**
B100k	0.109	0.660	0.717	**0.798**
B1M	0.013	–	0.688	**0.883**
B10M	0.000	–	0.476	**0.980**
Darpa	0.277	0.779	0.680	**0.835**
Krebs	0.831	–	**0.861**	0.817
Nimmo	0.591	**0.898**	0.807	0.754
StarWars	–	0.620	0.601	**0.949**
Bursty	0.028	**0.981**	0.898	0.975
DeBot	0.077	0.848	0.720	**0.862**
Avg. Acc.	0.609	0.565	0.719	**0.886**
$\hookrightarrow$ std	0.406	0.361	0.126	**0.071**

Besides the META model, all other algorithms are provided with the same information sources. Note, however, that Echeverria's model uses information extracted from Tweet texts but does not use NLP approaches for text understanding. The test data is only used for the final evaluation, not for model selection.

In our comparison, we include Echeverria's approach [11] as a representative of the algorithms using all features with current classification approaches for detecting unknown bots. We also compare against the Botometer to highlight the shortcomings of the current go-to approach. Note however that the approach can only be evaluated through the provided API. Thus, training data cannot be controlled and indeed violates the LOBO strategy. Nevertheless, due to its popularity as a bot detection service, we include the Botometer as a baseline. We report on Botometer results published by Echeverria et al. [11] using the model accessible through the public API. Note that experiments with bot types included in the training set can be interpreted as loose upper bounds. For an overview of the data used to optimize the Botometer model[5] see Tab. II.

[5] https://botometer.osome.iu.edu/bot-repository/datasets.html

TABLE IV: Average detection accuracy w.r.t. the bot categories.

Categories	Bmeter	Eche	**META**	**Expert**
Spam	**0.926**	0.135	0.768	0.918
Fake	**0.984**	0.888	0.578	0.852
Attackers	0.711	**0.898**	0.834	0.786
Campaigns	0.077	0.848	0.720	**0.862**
Mixed	0.122	0.720	0.676	**0.874**
Other	0.028	0.801	0.750	**0.962**
Avg. acc.	0.475	0.715	0.721	**0.876**
$\hookrightarrow$ std	0.408	0.266	0.080	**0.055**

Echeverria: Tab. III reports the evaluation performances. Overall, Echeverrias' model shows severe performance issues when detecting spambots (SSB, TSB). While the average accuracy on other bot types is 0.811, the performance on spambots is 0.135, only.

Botometer: The Botometer model performs best with an average accuracy of 0.697 on bot types known from training (remember: in violation with the LOBO evaluation strategy) (including acc. of SSB, TSB, FSF, INT, TWT, B1k, B100k, B1M, B10M). It performs poorly on human-annotated bots (B1k, B100k, B1M, B10M) with an average accuracy of 0.083, while showing peak performance on the other known bot types (Ø 0.943). When detecting unknown bot types, the model accuracy drops drastically to 0.361.

Interestingly, according to these results, the Botometer actually outperforms the model proposed by Echeverria et al. [11] in terms of average accuracy (0.609 to 0.565). We also note that the performance of both approaches varies significantly depending on the bot type (standard deviation (std): 0.406 and 0.361).

META model: Our metadata-based META model achieved the best detection rates compared to the other sub-models (CNN + metadata, RNN + text), with the Tweet texts model performing the worst. With the feed-forward component, we achieve accuracy between 0.474 and 0.898. While peak performance is significantly below baselines ($\leq$ 0.526) in some cases, the model is more stable and outperforms baselines in terms of average performance (across all bot types), i.e., 0.719 compared to 0.609 and 0.565. Moreover, a standard deviation of 0.126 indicates a better generalization than baselines. This is especially true for fake and spambots.

Ensemble of Experts: Our ensemble model additionally incorporates information from Tweet texts extracted using BERTweet. The model achieves an average accuracy of 0.886 with a standard deviation of 0.071. Compared to the Botometer model (cf Fig. 2) or Echeverria's model (cf Fig. 3), it increases the overall performance by 45.48% and 56.81%, respectively. In 11 of 20 experiments, it outperforms the baselines and never shows serious performance degradation. As for the peak performance, the worst performance degradation is 0.196 on TWT (compared to Botometer), while the largest gain is 0.980 on B10M.

TABLE V: Further performance measures: F_1 score, recall, and precision of the ensemble of experts; evaluations are split into 6 groups of bot sets.

Data set	Accuracy	F_1	Precision	Recall	P/R ratio
SSB1	0.949	0.944	0.904	0.987	0.916
SSB2	0.938	0.932	0.878	0.994	0.883
SSB3	0.919	0.917	0.869	0.971	0.895
TSB1	0.846	0.848	0.850	0.846	1.005
TSB2	1.000	1.000	1.000	1.000	1.000
TSB3	0.827	0.828	0.831	0.825	1.007
TSB4	0.948	0.951	0.907	1.000	0.907
FSF	0.909	0.905	0.962	0.854	1.126
INT	0.891	0.859	0.951	0.784	1.213
TWT	0.757	0.724	0.873	0.619	1.410
B1k	0.875	0.867	0.934	0.809	1.155
B100k	0.798	0.780	0.895	0.691	1.295
B1M	0.883	0.872	0.947	0.807	1.173
B10M	0.980	0.964	0.990	0.941	1.052
Darpa	0.835	0.842	0.875	0.811	1.079
Krebs	0.817	0.783	0.893	0.698	1.279
Nimmo	0.754	0.714	0.856	0.612	1.399
StarWars	0.949	0.951	0.919	0.986	0.932
Bursty	0.975	0.944	0.905	0.986	0.918
DeBot	0.862	0.871	0.858	0.886	0.968

C. Bot Categories

Next, we report on the performance concerning the different bot categories (see Tab. IV). Our approach yields the most stable results with competitive performances in all categories. Interestingly, in contrast to the average performance results in the previous section, in this section, Echeverria's model significantly outperforms the Botometer and indeed achieves similar results to our approach, except in the spam category.

The results show the instability of the Botometer, which delivers top performance on spam and fake bots, but fails on campaign-, mixed-, and other bots. Our META model already delivers decent performance across the board ($0.578 - 0.834$), but fails to deliver consistent peak performance. The results of this model highlight the importance of feature selection and confirm our assumption that bots can be detected based on behavioral data.

Overall, the performance of our approach is significantly more stable than related work, suggesting better generalization capabilities.

D. Performance Details

In the following, we focus exclusively on the ensemble of experts model to obtain a differentiated understanding of its performance. Thus, more information on the bot detection results can be found in Tab. V.

While *accuracy* is a measure of the overall detection accuracy, we are interested in more detailed measurements. We consider the F_1 *score*, which provides information about the *precision* and *recall* of the model. Here, *recall* denotes the

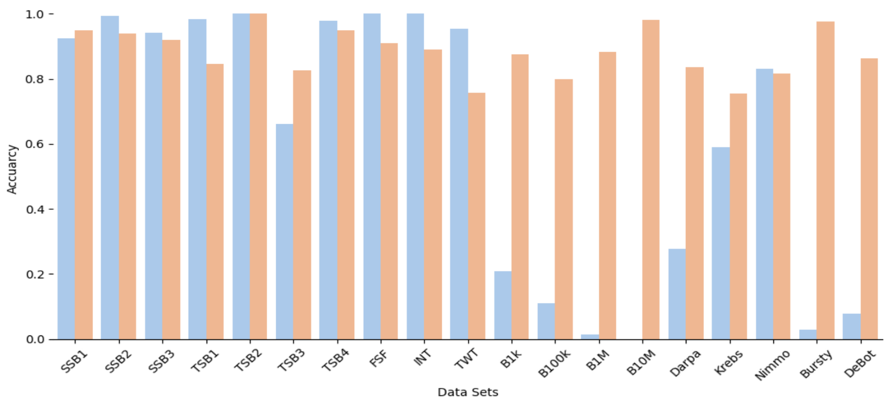


Fig. 2: Accuracy comparison between the ensemble of expert model (red) and the Botometer (blue).

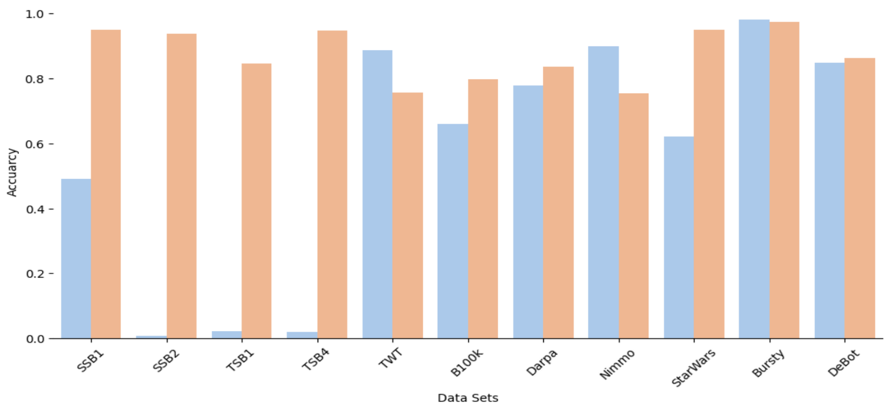


Fig. 3: Comparison between the ensemble of expert model (red) and Echeverria's approach (blue).

percentage of bots that were missed by the algorithm, while *precision* denotes the percentage of detected accounts that are actually bots. In general, misclassifying a user is a more serious error than overlooking a bot. Thus, *precision* is more crucial than *recall*.

We note that F_1 scores are similar to the accuracy measures. This is to be expected since we worked with balanced data sets. It confirms that the algorithm is generally balanced between detecting bots and detecting benign users. Nevertheless, we consider precision and recall separately.

Regarding this ratio (between precision and recall), we find that results vary slightly, with some showing similar results while others tend to have higher precision or recall. We note that the bots belonging to the same category show the same tendencies. In total, we achieve an average precision of 0.905 and an average recall of 0.856.

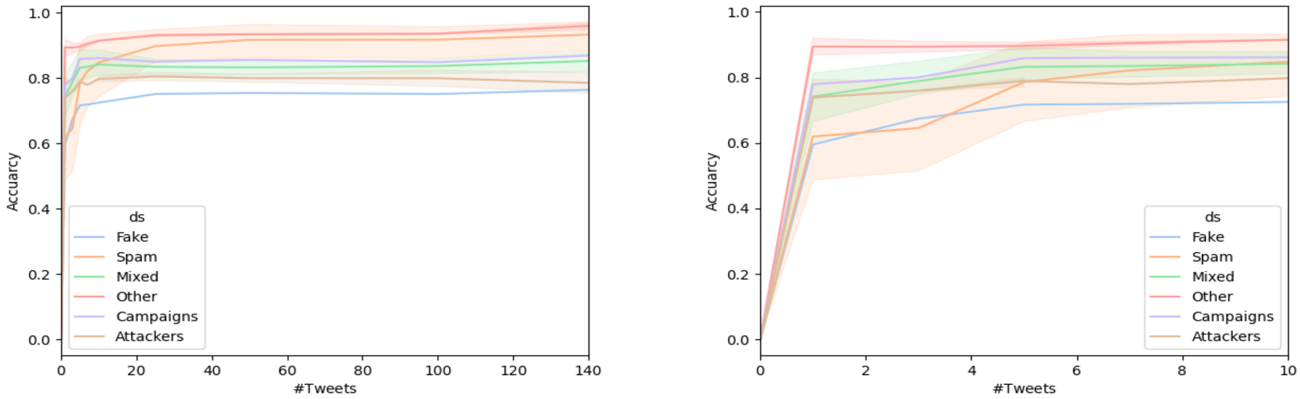
In general, *traditional spam bots* and *campaign bots* show balanced results. Experiments with social spam bots (SSB) and the *other bots* (StarWars, Bursty) show lower precision compared to recall. This is to be expected since social spam bots mimic human behavior and the other bots are non-commercial *concept bots* (StarWars) or are only active for a short period (Bursty). Finally, for the remaining bot types (fake followers, mixed bots), precision is the higher score. For human-labeled data sets, we have the most significant imbalance between precision (0.920) and recall (0.760).

E. Performance Progression

Last, we investigate the performance progression of the model as a function of the number of observed Tweets. To this end, we simulate data aggregation based on a theoretical number of observed Tweets. While we can adjust the number of Tweets, we need to interpolate the corresponding metadata. Note that by doing so, we introduce a bias towards better (averaged) metadata and avoid artifacts that might be caused by small amounts of data. We perform the evaluations on all experiments and limit the number of observed Tweets to $\{1, 3, 5, 7, 10, 25, 50, 100, 140\}$.

Fig. 5 shows the performance w.r.t. bot groups, while Fig. 4a shows the progression w.r.t. categories. In both cases, the experimental results indicate that the algorithm requires only a small number of Tweets to achieve a high level of accuracy. On average, this level is reached after observing 20 Tweets. Fake and social spam bots are an exception. Here, the algorithm takes longer to collect its 'sufficient statistics'.

If we take into account the performance for the first 10 observed Tweets (cf Fig. 4b), we see that the algorithm only needs a single Tweet from some bot types. Due to the



(a) Avg. accuracy progression per bot category.

(b) Accuracy progression of observing the first 10 Tweets.

Fig. 4: Performance progression with simulated data collection.

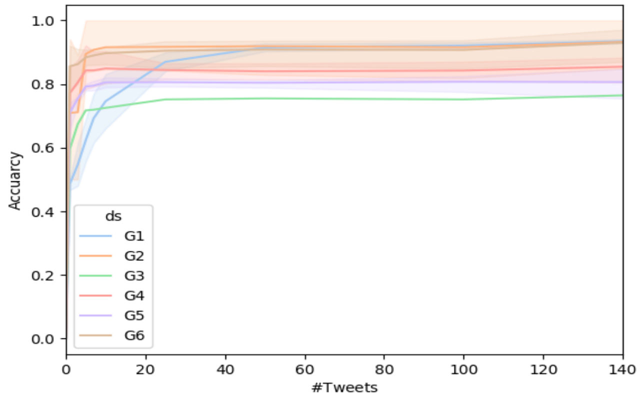


Fig. 5: Avg. accuracy per bot group; (1) SSB, (2), TSB (3), Fake (4), Mixed (5), Darpa + Attack, (6) Campaign + Other.

biased metadata, these results imply that for some bot types, statistics on behavior are sufficient for detection, while for others, Tweet texts provide valuable information. For example, the results suggest that the content of Tweets is particularly important for social spam bot detection, which explains the poor performance of Echeverria's model in this context. On the other hand, it seems that the detection of the StarWars- and Bursty bots do not benefit from Tweet content information.

V. Conclusion

The combination of the prevailing bot detection evaluations and performance-based feature selection has led to poor generalization performance in the past. While the Botometer achieves peak performance on known bots, the learned representation of bots seems too narrow to identify new bot types.

In this work, we investigated whether it is possible to identify generic bot behavior for generalized bot detection. We devised a model based on the assumption that bot behavior manifests as patterns in aggregated behavior in the form of statistics and the content of the Tweets. In particular, we ignored information that only exploited artifacts of specific bot types in the data. Experiments on a standard feed-forward model showed that selecting features that are limited to general behavior data increases the overall generalization performance of bot detection approaches. To achieve the best possible generalization, we developed an ensemble of neural networks to combine different aspects of the information. The results of our extensive experiments suggest that generic bot behavior can be extracted and used for reliable bot detection. Using more general features combined with a BERT model to incorporate textual information yields competitive performance with better consistency across bot types.

In general, it is complicated to classify the peak performances correctly, since most of the datasets are noisy. However, the difference regarding the generalizability of the learned bot representations is clear. The performance of our behavior-based approach significantly outperforms the others.

A look at the performance of the categories reveals the weakness of the currently preferred solution, the Botometer. With an average accuracy of 0.475 and a variance of 0.408, this method is unsuitable for detecting bots in general.

Echeverria's approach, on the other hand, shows much more consistent performance across categories. With 0.715 accuracy, it is much closer to the performance of our approach than the Botometer. However, the use of all available information seems to lead to a too narrow representation of general bot behavior. This is also indicated by a relatively high variance of 0.266.

Our approach shows very consistent performance across all bot types. As expected, it performed worst in detecting fake accounts, since the objective of fake accounts depends only very weakly on their behavior. Dedicated methods are preferable here.

In terms of error type, we have an average precision of 0.905 and an average recall of 0.855. Thus, a bot is overlooked more often (Type 2 error) than a user is detected as a bot (Type 1 error). Only in the case of SSB, Debot, StarWars and Bursty does the more expensive type 1 error occur more frequently.

Here, another striking finding is the imbalance between precision and recall for the bot types labeled by humans. While we have an average precision of 0.920, the recall is significantly worse at 0.760.

The performance difference between our method and Echeverria's method concerning social spambots suggests the importance of tweet content in detecting these bot types. Here, semantic understanding of textual information appears to be critical for consistent competitive performance.

The results of this work suggest that using behavioral information can lead to a reliable and consistent detection procedure. Especially in networks like Facebook, which offer the user a larger action space, this seems to be possible with sufficient accuracy. Twitter presents a more difficult task because users here have only very limited options for action.

The approaches investigated here require high-quality labeled data. Obtaining this data is an expensive and lengthy process. However, as long as clustering approaches are far behind these methods in terms of performance, this is the only realistic way.

References

[1] B. D. Loader and D. Mercea, "Networking democracy? social media innovations and participatory politics," *Information, communication & society*, pp. 757–769, 2011.

[2] C. Shao, G. L. Ciampaglia, O. Varol, K.-C. Yang, A. Flammini, and F. Menczer, "The spread of low-credibility content by social bots," *Nature communications*, pp. 1–9, 2018.

[3] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Communications of the ACM*, pp. 96–104, 2016.

[4] V. S. Subrahmanian, A. Azaria, S. Durst, V. Kagan, A. Galstyan, K. Lerman, L. Zhu, E. Ferrara, A. Flammini, and F. Menczer, "The darpa twitter bot challenge," *Computer*, pp. 38–46, 2016.

[5] S. Hölig and U. Hasebrink, "Reuters institute digital news report," *Ergebnisse für Deutschland. Arbeitspapiere des Hans-Bredow-Instituts*, 2020.

[6] C. J. Vargo *et al.*, "The agenda-setting power of fake news: a big data analysis of the online media landscape from 2014 to 2016," *New media & society*, pp. 2028–2049, 2018.

[7] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, "Defending against neural fake news," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 9054–9065.

[8] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, pp. 1146–1151, 2018.

[9] S. Majó-Vázquez, M. Congosto, T. Nicholls, and R. K. Nielsen, "The role of suspended accounts in political discussion on social media: Analysis of the 2017 french, uk and german elections," *Social Media+ Society*, 2021.

[10] A. Rauchfleisch and J. Kaiser, "The false positive problem of automatic bot detection in social science research," *Berkman Klein Center Research Publication*, 2020.

[11] J. Echeverria, E. De Cristofaro, N. Kourtellis, I. Leontiadis, G. Stringhini, and S. Zhou, "LOBO: Evaluation of generalization deficiencies in twitter bot classifiers," in *Proceedings of the 34th Annual Computer Security Applications Conference*. ACM, 2018, pp. 137–146.

[12] 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 companion on world wide web*, 2016, pp. 273–274.

[13] S. Cresci, R. Di 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*, 2017, pp. 963–972.

[14] K.-C. Yang, O. Varol, P.-M. Hui, and F. Menczer, "Scalable and generalizable social bot detection through data selection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 1096–1103.

[15] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on twitter," in *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, 2010, p. 12.

[16] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Who is tweeting on twitter: Human, bot, or cyborg?" in *Proceedings of the 26th Annual Computer Security Applications Conference on - ACSAC '10*, 2010, p. 21.

[17] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu, "Analyzing spammers' social networks for fun and profit: A case study of cyber criminal ecosystem on twitter," in *Proceedings of the 21st international conference on World Wide Web*, 2012, pp. 71–80.

[18] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson, "Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse," in *22nd USENIX Security Symposium (USENIX Security 13)*, 2013, pp. 195–210.

[19] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Zhao, "Detecting and characterizing social spam campaigns," *ACM SIGCOMM conference on Internet measurement*, p. 13, 2010.

[20] Q. Cao, X. Yang, J. Yu, and C. Palow, "Uncovering large groups of active malicious accounts in online social networks," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*, 2014, pp. 477–488.

[21] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao, "You are how you click: Clickstream analysis for sybil detection," in *22nd USENIX Security Symposium (USENIX Security 13)*, 2013, pp. 241–256.

[22] Y. Li, O. Martinez, X. Chen, Y. Li, and J. E. Hopcroft, "In a world that counts: Clustering and detecting fake social engagement at scale," in *Proceedings of the 25th International Conference on World Wide Web*, 2016, pp. 111–120.

[23] N. Chavoshi, H. Hamooni, and A. Mueen, "Debot: Twitter bot detection via warped correlation," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE Computer Society, 2016, pp. 817–822.

[24] L. Vargas, P. Emami, and P. Traynor, "On the detection of disinformation campaign activity with network analysis," in *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, 2020, pp. 133–146.

[25] S. Kudugunta and E. Ferrara, "Deep neural networks for bot detection," *Information Sciences*, pp. 312–322, 2018.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019, pp. 4171–4186.

[27] D. Q. Nguyen, T. Vu, and A. T. Nguyen, "BERTweet: A pre-trained language model for English Tweets," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 9–14.

[28] J. Echeverria and S. Zhou, "Discovery, retrieval, and analysis of the 'star wars' botnet in twitter," in *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*, 2017, pp. 1–8.

[29] Z. Gilani, R. Farahbakhsh, G. Tyson, L. Wang, and J. Crowcroft, "Of bots and humans (on twitter)," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, 2017, pp. 349–354.

[30] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "Fame for sale: Efficient detection of fake twitter followers," *Decision Support Systems*, pp. 56–71, 2015.

[31] N. Chavoshi, H. Hamooni, and A. Mueen, "Identifying correlated bots in twitter," in *International Conference on Social Informatics*. Springer, 2016, pp. 14–21.

[32] J. Echeverria, C. Besel, and S. Zhou, "Discovery of the twitter bursty botnet," in *Data Science for Cyber-Security*. World Scientific, 2019, pp. 145–159.