Identification of Twitter Bots Based on an Explainable Machine Learning Framework: The US 2020 Elections Case Study

Alexander Shevtsov,[1,3] **Christos Tzagkarakis,** [1] **Despoina Antonakaki,** [1] **Sotiris Ioannidis,** [2, 1]

[1] Institute of Computer Science, Foundation for Research and Technology - Hellas
[2] School of Electrical and Computer Engineering, Technical University of Crete
[3] Department of Computer Science, University of Crete
{shevtsov,tzagarak,despoina}@ics.forth.gr, sotiris@ece.tuc.gr

Abstract

Twitter is one of the most popular social networks attracting millions of users, while a considerable proportion of online discourse is captured. It provides a simple usage framework with short messages and an efficient application programming interface (API) enabling the research community to study and analyze several aspects of this social network. However, the Twitter usage simplicity can lead to malicious handling by various bots. The malicious handling phenomenon expands in online discourse, especially during the electoral periods, where except the legitimate bots used for dissemination and communication purposes, the goal is to manipulate the public opinion and the electorate towards a certain direction, specific ideology, or political party. This paper focuses on the design of a novel system for identifying Twitter bots based on labeled Twitter data. To this end, a supervised machine learning (ML) framework is adopted using an Extreme Gradient Boosting (XGBoost) algorithm, where the hyper-parameters are tuned via cross-validation. Our study also deploys Shapley Additive Explanations (SHAP) for explaining the ML model predictions by calculating feature importance, using the game theoretic-based Shapley values. Experimental evaluation on distinct Twitter datasets demonstrate the superiority of our approach, in terms of bot detection accuracy, when compared against a recent state-of-the-art Twitter bot detection method.

Introduction

Twitter is considered one of the most popular and widespread online social networks (OSNs) nowadays. It is used by millions of users and organizations to quickly share and discover information about a service, product, sports/social/political event etc. However, Twitter can be used as an intermediate system for malicious purposes, such as spreading fake news (Bovet and Makse 2019; Sharma et al. 2019) or manipulating public opinion (Badawy, Ferrara, and Lerman 2018).

Specifically, Twitter can be used to circulate propaganda (Neudert, Kollanyi, and Howard 2017; Jones 2019; Chatfield, Reddick, and Brajawidagda 2015), manipulate the public opinion (Bolsover and Howard 2019; Seo 2014) and influence the electorate towards a particular ideology or political party (Golovchenko et al. 2020; Howard, Kollanyi, and Woolley 2016). These tasks can be fully automated through a special organized group of agents, called *botnets*, which are groups of *sybil* accounts that collectively seek to influence ordinary users. In particular, a botnet is a group of bots, i.e., automated programs programmed to run certain tasks. A *sybil* account in OSNs is a fake identity, not necessarily representing a real person or created by the real person it represents (impersonation technique) (Alsaleh et al. 2014).

It has been observed that Twitter bots can also be exploited to spread fake news, rumors and hate speech (Founta et al. 2018; Fortuna and Nunes 2018; Burnap and Williams 2015) by instantly republishing low credibility Twitter content (Shao et al. 2018) via popular users and Twitter mentions (Stella, Ferrara, and De Domenico 2018).

In this work, we aim to build a machine learning (ML) framework over a large collected dataset, to detect bot Twitter accounts. We identify and analyze Twitter bots during the US 2020 Elections period. The current study provides answers to the following questions:

- Is it possible to implement and fine-tune a ML-based bot detection model to efficiently apply it to the US 2020 Elections dataset?
- Which types of features can be extracted from the Twitter application programming interface (API) to promote high performance?
- Is it possible to examine the ML model's generalization capability in terms of bot detection accuracy across several well-established datasets?
- Does the proposed ML model act as a black box or could the ML model's mechanism be "unlocked" in order to investigate how it yields its predictions?

Our analysis can help the research community to better understand the bot detection task and how it can be performed in different types of datasets, or within diverse domains. The presented methodology achieves a high bot detection accuracy on the US 2020 Elections dataset, while attaining increased generalization performance in terms of bot identification when applied on additional, well-established Twitter datasets. The ML model's outcome is also explained based on Shapley Additive Explanations (SHAP) method.

The rest of the paper is organized as follows: Section background explores related past works. A detailed description of the data collection process and the proposed methodology is given in Section methodology. Section experimental results evaluates the performance of our method, whilst in Section conclusions and future work summarizes the main outcomes and provides directions for future work.

Background

Twitter (social) bots can be used for malicious purposes spanning from junk news and fake news or rumor spreading (Sharma et al. 2019), to propaganda and astroturfing (Bovet and Makse 2019; Howard et al. 2017a; Neudert, Kollanyi, and Howard 2017; Howard et al. 2017b). Specifically, an application is developed in (Hui et al. 2020) to track information spreading on Twitter and tweets and accounts associated with suspicious campaigns.

Usually, a legitimate bots' usage is adopted, to perform automated communication or administration during the electoral periods (Howard, Woolley, and Calo 2018). However, Twitter bots have been extensively used for opinion hijacking during the Russian elections (Krebs 2011; Shane 2017; Lightfoot and Jacobs 2017; Illing 2018), the 2017 French presidential election (Ferrara 2017), the US elections (Howard, Woolley, and Calo 2018; Byrnes 2016; Rizoiu et al. 2018), the Catalan independence referendum (Stella, Ferrara, and De Domenico 2018), as well as in the Australian (Waugh et al. 2013), the Ukrainian (Hegelich and Janetzko 2016) and the Brazilian electoral processes (Arnaudo 2017). In (Luceri et al. 2019), the authors analyze 245,000 Twitter accounts during the US 2016 presidential election and 2018 midterm elections, and they detect approximately 31,000 bots. Forty-three million elections-based tweets related with the ongoing U.S. Congress investigation of Russian interference during the 2016 U.S. election campaigns are examined in (Badawy, Ferrara, and Lerman 2018), where it is estimated that 4.9% and 6.2% of liberal and conservative users, respectively, were bots, with reported precision and recall scores above 90%. In (Keller and Klinger 2019), the authors provide an analysis of the German parties' posts on Twitter from before and during the 2017 electoral period, revealing an increased amount of social bots (7.1% to 9.9%).

Additional Twitter bots analysis works examine a specific consequential period in Russian politics (February 2014 to December 2015) (Stukal et al. 2017) and apply sentiment analysis or attempt to predict the results of the elections (Ibrahim et al. 2015; Antonakaki et al. 2017).Other studies focusing on Twitter bots analysis include (Stukal et al. 2017) studying a specific consequential period in Russian politics (February 2014 to December 2015) and apply sentiment analysis or attempt to predict the results of the elections (Ibrahim et al. 2015; Antonakaki et al. 2017).

The authors in (Garimella and Weber 2017) investigate the political polarization on Twitter between 2009 and 2016, with an increased polarization of 10% and 20% being reported. The impact of Twitter bots during the first U.S. presidential debate of 2016 is studied in (Rizoiu et al. 2018), where a novel algorithm for estimating user influence from retweet cascades is introduced towards analyzing the role and user influence of bots versus humans. Moreover, a Twitter data analysis has been conducted in (Fraisier et al. 2018) related to the 2017 French presidential campaign. The authors built a large and complex dataset of 22,853 active Twitter profiles, during the campaign from November 2016 to May 2017. The analysis of political discourse on Twitter in elections dataset has been noted during the US 2016 presidential elections (Yaqub et al. 2017) as well. Opinion hijacking has been observed not only in politics, but also in anti-vaccination promotion movements (Broniatowski et al. 2018). Thus, it is important to quantify the spread of fake news on Twitter (Waugh et al. 2013) and the inherent variability (Vosoughi, Roy, and Aral 2018), in order to distinguish bots from human agents and legitimate users (Edwards et al. 2014).

It is evident that Twitter bot detection is a complex task, often requiring rigorous and solid treatment. Several ML-based solutions have been proposed such as the BotOrNot real-time detection system (Davis et al. 2016) using a total amount of 1200 different features in combination with a Random Forest classifier. An updated version of this system is described in (Yang et al. 2019) named as Botometer, which requires Twitter API keys to collect user information during the real-time computations, thus it is not efficient to use real-time labeling tools in the case of big datasets. BotSentinel (Bouzy 2021) on the other hand, is a non-real-time labeling tool, capable of processing large amounts of user accounts and storing the results in a database. BotSentinel's offline labeling methodology is adopted whenever a user account is being suspended or removed, whereas real-time labeling does not provide any suspended account information. Moreover, the offline implementation allows an increase in query rate limits since it does not involve any labeling computational costs.

A fundamental part of the bot detection pipeline corresponds to the computation of features based on Twitter data, and thus a plethora of different types of features have been proposed. Various features are based on content (Ahmed and Abulaish 2013; Gilani, Kochmar, and Crowcroft 2017; Lee, Caverlee, and Webb 2010; Davis et al. 2016; Varol et al. 2017), sentiment (Loyola-González et al. 2019; Dickerson, Kagan, and Subrahmanian 2014; Ferrara et al. 2016; Loyola-González et al. 2019), account information (Wald et al. 2013; Chu et al. 2012; Davis et al. 2016; Lee, Caverlee, and Webb 2010; Loyola-González et al. 2019), usage (Chu et al. 2012) and network characteristics (Feng et al. 2020; Keller et al. 2017; Cresci et al. 2017).

There is a growing number of ML and data (statistical) analysis-based Twitter bot identification tools. The most popular can be considered the Stweeler tool (Gilani et al. 2016), the Debot system (Chavoshi, Hamooni, and Mueen 2016)which takes into account synchronous bots spreading content, the TSD Sybil Detector (Alsaleh et al. 2014) that adopts a ML approach using 17 Twitter data-based features, and the Retweet-Buster (RTbust) (Mazza et al. 2019) which is an unsupervised learning tool combining feature extraction and clustering techniques.

Sentiment analysis has also been incorporated into the bot detection pipeline (Dickerson, Kagan, and Subrahmanian 2014; Loyola-González et al. 2019). A set of sentiment features is exploited by the BotOrNot tool in (Varol et al. 2017).

Hashtag	Tweet Counts
#VOTE	3,064,099
#Trump202	2,403,586
#Vote	2,200,954
#Election2020	1,906,959
#vote	1,838,645
#Biden	1,063,265
#Debate2020	839,717
#BidenHarris2020	781,697
#VoteBlueToSaveAmerica	746,896
#Trump	601,516

Table 1: Most popular HTs in our dataset. Tweets may contain multiple HTs so that the sum of tweets is not equal to the number of tweets in our collection.

The promising direction of ML-based Twitter bot detection can be reflected in DARPA competition, where six different research groups competed in performing bot identification, using anti-vaccination campaigns Twitter data (Subrahmanian et al. 2016).

Methodology

Dataset

In order to capture the US 2020 elections' Twitter dynamics shortly before the elections day (November 3rd 2020), we build a dataset where the most popular hashtags (HTs) related to the US 2020 elections are initially obtained. Twitter API is used to retrieve all the tweets containing these HTs, spanning from September 1st, 2020 to November 3rd, 2020, resulting in a dataset of 15.6 million tweets and 3.2 million users. The ten most popular HTs are shown in Table 1.

Twitter Users Labeling

The acquired dataset does not contain explicit knowledge whether a user is a bot or not. Since the goal of the current study is to provide a supervised ML-based solution for Twitter bot detection, it is crucial to obtain a bot vs. normal users labeled dataset. Unfortunately, it is not possible to collect accurate ground truth labels without using third-party bot labeling tools. The typical solution of ground truth generation corresponds to a manual/crowd-sourcing analysis, which requires a thorough inspection of Twitter accounts by human experts to identify the label of each account (via a majority voting rule). The manual labeling process is cumbersome due to the dataset size, potentially containing millions of users (in our case the dataset contains 3.2 million users), and the sophistication level of bot accounts which has risen during the last years.

As a means of overcoming the inherent restrictions of manual labeling, we utilize off-the-shelf ML-based techniques allowing us to scale up the labeling procedure. ML methods achieve higher accuracy in terms of ground truth labeling as compared with the manual/crowd-sourcing analysis, since they exploit Twitter data feature representations not evident to human experts. Here, we use the Botometer (Osome 2020; Varol et al. 2017) and BotSentinel (Bouzy 2021) online tools to obtain the user labeling information. To achieve highly confident results, we combine the set of labels provided as output by the Botometer and the BotSentinel tool, respectively. In particular, we compute the intersection of the two label sets. The intersection contains the labels that are equal in both label sets. The users identified as bots unanimously by both tools are labeled in our dataset as bots. The users identified as bots by only one of the two tools are marked as unlabeled.

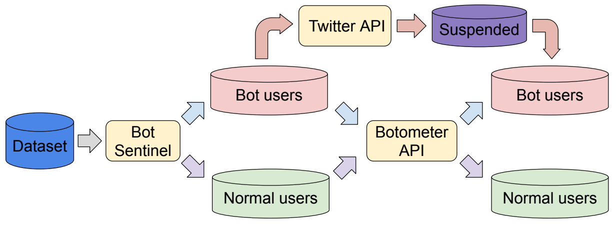

Figure 1: The bot vs. normal users labeling pipeline.

Both of the bot detection labeling tools yield an output score for each requested Twitter account. The Botometer score lies in the interval $[0, 5]$, while the BotSentinel score takes integer values in $\{0, \ldots, 100\}$. The higher the output score is, the higher the probability the requested account is a bot. A Twitter user is labeled as bot when the Botomoter and BotSentinel's output score is greater than 4.0 and 75, respectively. When the Botomoter and BotSentinel's output score is less than 1.0 and 25, respectively, the Twitter user is labeled as normal.

As mentioned above, since none of the two tools guarantees 100% bot identification accuracy, we aim at combining the scores from both tools and take into consideration the labels that are (mutually) equal. When an account is already suspended, Botometer cannot query Twitter API, and thus we perform a Twitter API check to identify whether an account is suspended or not.

We separate our dataset into two parts to minimize the time complexity of the labeling process. The first part contains data extracted during September 2020 and is utilized for user labeling, ML model fine-tuning, training, validation and testing purposes. The second part incorporates data from October 1st, 2020 until November 3rd, 2020 and is used to evaluate the generalization capability of the proposed ML-based Twitter bot identification system on unseen data.

Step	Bot Users	Normal users	Total
Before labeling	-	-	$1.3M$
BotSentinel	10,324	25,546	35,870
Botometer	2,180	7,267	9,447
Suspended	2,389	0	2,389
Final	4,569	7,267	11,836

Table 2: The number of users during each phase of the labeling of our dataset (the symbol M corresponds to million).

Feature	Type	Feature	Type	Calculation
statuses_count	count	*screen_name_len*	count	
entities_count	count	*description_len*	count	
followers_count	count	*screen_name_likelihood*	real-valued	likelihood of screen_name
friends_count	count	**name_screen_sim**	real-valued	name and screen_name similarity
favourites_count	count	**tweet_retweet_ratio**	real-valued	statuses_count / retweet_count score
listed_count	count	*name_digits*	real-valued	number of digits in user name
name_len	count	*screen_name_digits*	real-valued	number of digits in user screen_name
geolocation	boolean	*tweets_by_age*	real-valued	statuses_count / user age
protected	boolean	*followers_by_age*	real-valued	followers_count / user age
location	boolean	*friends_by_age*	real-valued	friends_count / user age
background_img	boolean	*favourites_by_age*	real-valued	favourites_count / user age
default_profile	boolean	*listed_by_age*	real-valued	listed_count / user age
verified	boolean	*followers_friends*	real-valued	followers_count / friends_count

Table 3: Profile features extracted from Twitter user objects.

The dataset separation allows us to reduce the labeling process (computational) time without significant information loss, since the accounts remain active throughout the whole period of September and October. The first part has 1.3 million users and more than 5 million tweets and retweets, while the second part consists of 2.6 million users and 10.6 million tweets and retweets. A subset of users remain active during both periods, therefore it is obvious to notice the overlap between the two parts.

Figure 1 shows the labeling pipeline. Our dataset has 1.3 million users during the BotSentinel labeling step. Then, the Botometer tool receives as input 35,870 users, i.e., 10,324 bot users and 25,546 normal users (see Table 2), and outputs 9,447 users (2,180 Twitter accounts labeled as bots and 7,267 Twitter accounts marked as normal accounts). As a parallel step, we query the Twitter API and the response provides a set of 2,389 users labeled as suspended. Therefore, the final labeled set has 4,569 bot users and 7,267 normal users. Note that the overall labeling procedure is initialized with BotSentinel, since it does not impose any daily query limitations, in contrast with the Botometer. In the case of a Botomoter-based initialization step, the labeling outcome of the pipeline depicted in Figure 1 will be the same, but the processing time will grow dramatically and will require 650 days to terminate, due to the Botometer request limitations. In contrast, the computational time of the labeling process is 18 days if BotSentinel is used during the first labeling step.

We already mentioned that none of the existing labeling tools provide 100% accurate ground truth labels. To quantify the accuracy of our proposed labeling pipeline, we compare the labeling results of our pipeline against the Twitter bot detection algorithm after a period of six months. According to Twitter, the number of bot accounts is 4,569 (with 51% and 34% of the bot accounts being suspended and removed, respectively), while 7,267 Twitter accounts are identified as normal (with only 1.9% and 6.3% of the normal accounts being suspended and removed, respectively). Twitter's labeling mechanism incorporates a lag time, and thus it cannot be efficiently used to compute our ground truth labels. Specifically, we manually confirm that the lag time corresponds to approximately two months in the case of the US 2020 Elections.

Feature Extraction

Twitter API allows the collection of tweets including information such as tweet text, tweet post time, as well as metadata such as HTs, URLs, and mentions. In this paper, we also include the user profile information by retrieving user objects, where all different types of the retrieved Twitter content are utilized, leading to a total amount of 335 computed features. The features can be divided into four categories, namely, user profile, user context, user time, and user interaction.

Profile Features Twitter API retrieves user objects containing critical information to achieve accurate bot identification performance. The importance of user profile features is analyzed in various works (Chu et al. 2012; Wald et al. 2013; Gilani, Kochmar, and Crowcroft 2017; Yang et al. 2020). Typically, a user profile object includes user description, username, profile picture, and profile statistics (e.g., number of followers, friends, favourites, and listed). In this paper, bot vs. normal users distinction is promoted by enriching the features set through the extraction of profile features. For this, the user profile description and the user/screen name digits are taken into consideration. The computed user object-based features correspond to unedited parameters such as the number of followers, friends, favourites, lists, and description length. Flag type elements like location usage, account description, protected flag, geolocation usage, and background image usage are also estimated. Additional parameters such as the Jaccard similarity of the user and the account screen name are precomputed and included in the overall features set. Table 3 shows the list of the extracted feature set, where the feature names written in italics correspond to the statistical features described in (Yang et al. 2020) and the feature names written in bold correspond to our proposed features leading to a 26-dimensional profile feature space.

Feature	Description
N_tweet_mentioned_tfidf	TF-IDF score of the 3 most popular user mentions contained in tweets
N_tweet_mentioned_word	The 3 most popular mentions in user tweets as word features
N_tweet_hashtags_tfidf	TF-IDF score of the 3 most popular user HTs contained in tweets
N_tweet_hashtags_word	The 3 most popular HTs in user tweets as word features
N_retweet_mentioned_tfidf	TF-IDF score of the 3 most popular user mentions contained in RTs
N_retweet_mentioned_word	The 3 most popular mentions in user RTs as word features
N_retweet_hashtags_tfidf	TF-IDF score of the 3 most popular user HTs contained in RTs
N_retweet_hashtags_word	The 3 most popular HTs in user RTs as word features
N_tweet_word	The 3 most popular words used by user in tweets as word features
N_retweet_word	The 3 most popular words used by user in RTs as word feature
tweet_number_of_urls	Number of URLs in tweets, computed as average and standard deviation
retweet_number_of_urls	Number of URLs in RTs, computed as average and standard deviation
tweet_number_of_hashtags	Number of HTs in tweets, computed as average and standard deviation
retweet_number_of_hashtags	Number of HTs in RTs, computed as average and standard deviation
tweet_number_of_mentions	Number of mentions in tweets, computed as average and standard deviation
retweet_number_of_mentions	Number of mentions in RTs, computed as average and standard deviation

Table 4: Context features based on user tweets and RTs crawled by Twitter API.

Context Features The user profile feature set described in the previous Section reflects the statistics of the user's Twitter account from the first day of the subscription. However, the profile features lack semantic information regarding the actual content sent by the user. Thus, it is essential to incorporate contextual information such as user posts' content, most important user tweeted/re-tweeted topics, popular user HTs, and number of URLs usage per tweet. Table 4 summarizes the list of the estimated context features.

Tweet's context characteristics provide a diverse range of uniqueness because each user operates in a different form of expression. To estimate the most frequent words and entities, we compute a subset of the context features such as the three most popular words, mentions, and HTs per user (punctuation marks and stop words are removed since they do not provide important information). For each user, we discover the most frequent sentences (user mentions, hashtags, upper, and lower words). User mentions and hashtags may provide unique information that highlights the characteristics of a particular user, thus we compute the term frequency-inverse document frequency (TF-IDF) (Rajaraman and Ullman 2011) on the collected dataset. This allows us to identify the importance level of the user's hashtags and mentions. In particular, we compute the TF-IDF of the overall user mentions and hashtags, and we identify the three most frequent mentions/hashtags for each specific user. The final step is to compute the TF-IDF based on the overall frequency.

The next step includes the use of the word2vec algorithm (Church 2017) to learn the word embeddings from the obtained Twitter dataset, allowing us to transform text-based features into a 10-dimensional space. The most frequent words, mentions, and HTs are transformed with the trained word2vec model. Note that the text-based features might differ between the user's original tweets and RTs, since they are usually written by a different user. Thus, text-based features are computed separately for each user's tweets and RTs.

Feature	Description
daily_rt	RTs % each week day
daily_tw	tweets % each week day
daily_rt_tw	tweets/RTs % each week day
daily_retweet_avg	average daily number of RTs
daily_tweet_avg	average daily number of tweets
hourly_rt	RTs % of daily hours
hourly_tw	tweets % of daily hours
hourly_rt_tw	tweets/RTs % of daily hours
retweet_time	time difference between original tweet and user RT, computed as min/max/avg/std

Table 5: Time-based features computed on user's tweets/RT objects. Min, max, avg, and std correspond to minimum, maximum, average, and standard deviation, respectively.

Time-Based Features The automated bot accounts follow a non-uniform time distribution activity (Zhang and Paxson 2011) due to either Twitter API time constraints regarding tweet posts within short time intervals or as a result of the job schedulers that invoke tasks at specific time intervals. In addition, the automated bots follow a non-uniform activity pattern whenever scripts are scheduled to start or stop running at the same timestamps. Thus, the automated bots behaviour can be detected by recognizing extremely non-uniform or highly uniform time patterns of tweet posts. On the other hand, normal users' tweets typically follow a diurnal pattern, which can be predictable for specific users. As mentioned in (Chu et al. 2010), the human activity follows a special pattern on Twitter since humans perform tweet posts at specific daily time intervals, while the activity appears to be lower during the weekends. Nevertheless, bots' activity pattern is more unpredictable because it does not follow the same activity level per day. The automated behaviour of bot

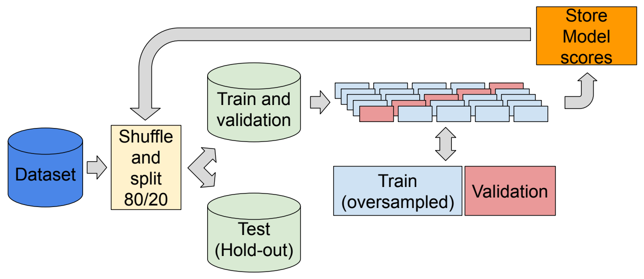

Figure 2: ML model selection pipeline.

accounts is constantly evolving, almost mimicking human users activity, making it a challenging task to detect bot accounts based solely on time-oriented features. However, due to the fact that not all automated behaviours are similar, we aim to compute and use time-based patterns as additional input to the proposed ML pipeline in order to enhance the bot vs. normal users detection accuracy.

In this paper, we extract multiple time-based features such as the RT time, as well as the hourly and daily activity. Regarding the RT times, we compute the difference between the original tweet and the RT time provided in the tweet object. We also measure the RT time distribution per user, where the minimum, maximum, average, and standard deviation values of the RT time are included in the feature set. As an account activity metric, the daily percentage of tweets and RTs is computed (i.e., we can identify during which days the users appear to be more active). Similar metrics are estimated during the active days and hours, and thus we can identify the exact hourly intervals of the day in which the user is vigorously posting tweets or RTs. Table 5 presents the set of time-based features.

Interaction Features The final set of extracted features is based on the RT network graph which models user interactions. The RT graph is estimated based on the collected dataset, with the nodes representing users and the directed edges defining a RT action from user i to user j. The edge weight indicates the number of RTs between the two users. The resulted graph represents the network of the RT connections in our dataset. Finally, we use Gephi (Bastian, Heymann, and Jacomy 2009) to compute the node statistics such as in-degree and out-degree of each node.

Experimental Results

In this Section, we examine the performance of our proposed system, with respect to the resulting bot vs. normal users detection accuracy.

ML Framework

As a main step towards building a robust and accurate ML-based bot identification system, we perform a model selection procedure by examining the bot vs. normal users classification accuracy of several state-of-the-art ML algorithms. In particular, we evaluate the performance of Random Forest (Breiman 2001), Support Vector Machine (SVM) (Cortes and Vapnik 1995) and Extreme Gradient Boosting (XGBoost) (Chen and Guestrin 2016) algorithm. Each ML method involves a different number of hyper-parameters, and thus it is of paramount importance to follow a hyper-parameter tuning procedure to identify the best (trained) version of each ML model and promote a fair models' comparison. Figure 2 illustrates the ML model selection pipeline based on a combination of $80/20$ train/test split (using random shuffle) and a 5-fold cross-validation scheme, i.e., the dataset is randomly shuffled, where 80% of the dataset is used for training/validation and the rest 20% (hold-out part) of the dataset is exploited for testing purposes. Each train/test split is performed in a stratified manner in order to have the same ratio of classes in both training and testing data. During the 5-fold cross-validation process, the synthetic minority oversampling technique (SMOTE) using Tomek links (Batista, Prati, and Monard 2004) is applied on the training folds to balance the two (bot vs. normal) distributions by oversampling the minority (bot) class distribution.

Model	F1	PR-AUC	ROC-AUC
XGBoost	0.919	0.967	0.979
Random Forest	0.908	0.955	0.973
SVM	0.889	0.941	0.964

Table 6: Testing accuracy during the model selection phase.

Additionally, we employ feature selection based on three different methods, namely, Lasso (Tibshirani 1994), Random Forest feature selection and model feature importance. Among the three feature selection methods, the model feature importance provided the features having the highest predictive accuracy (ScikitLearn 2022).

It is important to mention that some of the context features are vectors provided by the pre-trained word2vec model. The word2vec model provides a 10-dimensional space representation of the text, but only a few out of the ten dimensions are informative for the model. For this, we keep only the informative dimensions through the feature selection process. An example is presented in Figure 5, where the feature "*N1_retweet_hashtag_word_7*" represents the first most popular hashtag seen in the user retweets. According to the feature name, the seventh element of this particular word (embedding) vector corresponds to the most informative dimension.

Table 6 reports the F1 score and the two areas under the curve (AUC) scores, i.e., the precision-recall (PR) AUC value and the receiver operating characteristic (ROC) AUC value, averaged over ten repetitions. It can be seen that the XGBoost model achieves slightly better results on the testing data than SVM and Random Forest. Thus, we select XGBoost as the basic ML model applied in the next experimental evaluation phase.

General Model Comparison

We evaluate the generalization capability of the XGBoost model, which is already fine-tuned on the US 2020 Elections dataset (see in Subsection ML framework), against a general model (Yang et al. 2020) applied to detect bots on various

Dataset	# bots	# normal
cavarlee	15,483	14,833
varol-icwsm	733	1,495
cesci-17	7,049	2,764
pronbots	17,882	0
celebrity	0	5,918
vendor-purchased	1,087	0
botometer-feedback	139	380
political-bots	62	0
Gilani-17	1,090	1,413
Cresci-rtbust	353	340
cresci-stock	7,102	6,174
Midterm-18	42,446	8,092
Botwiki	698	0
verified	0	1,987
Total	94,124	43,396

Table 7: Publicly available labeled datasets used for bot detection performance evaluation in (Yang et al. 2020).

datasets. To perform this comparison, we collect the public datasets provided by (Yang et al. 2020) and we perform an experimental evaluation with similar specifications as described in ML framework Subsection. The authors in (Yang et al. 2020) utilize only the statistical features computed on the user objects without exploiting any further knowledge related with user interactions, RT times, or contextual information of user tweets. As a result, we extract and use the same feature set in both XGBoost and general model implementation to promote a fair comparison.

We follow the experimental strategy described in (Yang et al. 2020). Our proposed XGBoost model is trained over all possible combinations of publicly available Twitter datasets mentioned in Table 7. The dataset combinations that correspond to the best testing performance of each model are presented in Table 8. The first six rows of Table 8 indicate the different dataset combinations (check mark symbols) used as training data by the M196, M195, U1 and U2 models. The next four rows correspond to the identification accuracy of each model using as testing (unseen) data the Botwiki & verified, Midterm-18, Gilani-17 and Cresci-rtbust datasets, respectively. Note that the ROC-AUC scores over the four datasets correspond to the M196 and M195 dataset combinations used in (Yang et al. 2020) and to the U1, U2 dataset combinations trained by our model. The inherent information of the various combined datasets reflect the differences between bot vs. normal users. This information can be learned via the XGBoost model, achieving robust generalization capabilities.

The best ROC-AUC scores are achieved by our XGBoost model when using as training data the dataset combinations U1 (varol-icwsm, pronbots, botometer-feedback) and U2 (varol-icwsm, pronbots, botometer-feedback, political-bots). Table 8 also reports the best ROC-AUC scores achieved by the Random Forest model in (Yang et al. 2020) utilizing as training data the dataset combinations M196 (varol-icwsm, cesci-17, celebrity botometer-feedback, political-bots) and M195 (varol-icwsm, cesci-17, celebrity botometer-feedback).

Dataset	M196	M195	U1	U2
varol-icwsm	●	●	●	●
cesci-17	●	●		
pronbots			●	●
celebrity	●	●		
botometer-feedback	●	●	●	●
political-bots	●			●
Botwiki & verified	0.99	0.99	0.978	0.978
Midterm-18	0.99	0.99	0.954	0.951
Gilani-17	0.68	0.69	0.75	0.745
Cresci-rtbust	0.60	0.59	0.63	0.614
Average ROC-AUC	*0.815*	*0.815*	*0.828*	*0.822*

Table 8: Training dataset combinations and performance results between our XGBoost model and the method described in (Yang et al. 2020).

The results clearly indicate that our fine-tuned models (U1 and U2) achieve better results when compared with the general models (M195 and M196) in (Yang et al. 2020), as reflected on the average ROC-AUC scores. According to the best performance (U1 model) across the different datasets, the ROC-AUC values range from 0.97 to 0.63. In order to identify how the U1 and U2 trained models will perform on the US 2020 Elections dataset, we measure the ROC-AUC score for these two models. The models U1 and U2 achieved 0.609 and 0.618 ROC-AUC, respectively. For this purpose, we provide the performance of the model trained directly on the US 2020 Elections data.

Statistical vs. Context Features

We compare the proposed XGBoost model with the model introduced in (Yang et al. 2020) in light of the statistical features set. The authors in (Yang et al. 2020) exploit only the statistical features set. To promote a fair models' comparison we use the statistical features alone, including the number of followers, listed, favorites, and friends, as well as the com-

Feature set	Number of features
Statistical	6
Statistical General only	20
Context	204
Time	99
Graph	6
Total	335
Our model	228

Table 9: Number of features extracted by each feature category from the US 2020 Elections dataset. The statistical general set is a subcategory of the statistical features. For this reason, the statistical category above contains 6 unique and 20 features from the statistical general only feature category.

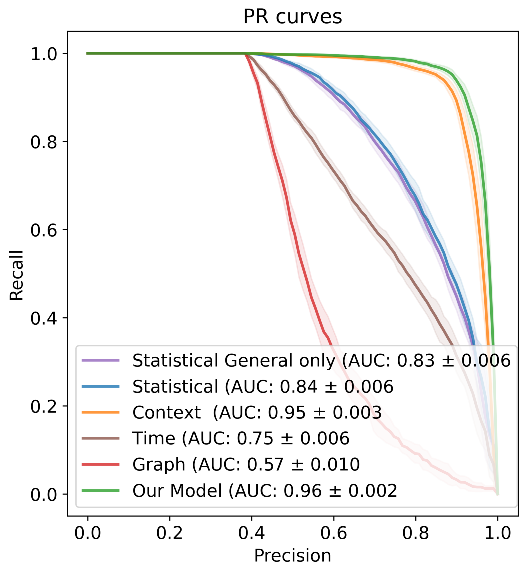

Figure 3: Mean PR curves: multiple types of features compared against our selected features.

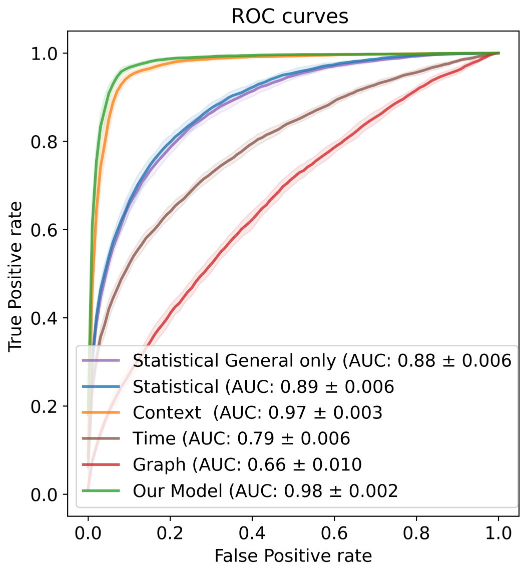

Figure 4: Mean ROC curves: multiple types of features compared against our selected features.

putation of the growth rate based on the user account age, the number of digits in the screen name, and the account screen name likelihood. The extracted set of features do not contain semantic information related with the posts content. The rest of the feature types described in Section feature extraction are utilized separately in our model during the training and validation steps. Each feature category presented in Table 9 is used separately and compared to each category's performance against PR and ROC curves with the features described in (Yang et al. 2020), as well as the best features that are selected by our model. Figure 3 presents the precision vs. recall performance of those features, with separate information of F1-score of the hold-out dataset portion. Figure 4 illustrates the ROC-AUC curve and the corresponding AUC score for each feature set. According to the ROC-AUC performance model, utilizing a mixture of multiple features with proper feature selection results in a better ROC-AUC performance model, since each feature set contains critical information for the model.

Generalization Performance: US 2020 Elections Dataset

The combination of multiple features provide the best bot identification accuracy as it is experimentally evaluated in Subsection statistical vs. context features, where the number of multiple combined features is 228. We use this set of features to investigate the generalization capability of our XGBoost model on the US 2020 Elections dataset. In particular, we divide the US 2020 Elections dataset into two parts as mentioned in Subsection twitter users labeling, i.e., the first part corresponds to the time interval between September 1st and September 30th, while the second part corresponds to the interval between October 1st and November 3rd. The experimental specification is the same as that adopted in Subsection ML framework. The only difference is that the train/test split now is $70/30$, where 70% of the September dataset is used for train/validation of the XGBoost model, while the rest 30% is used for testing with the F1 score equal to 0.916 and the ROC-AUC score is 0.98. The difference between these results and the ones depicted in Figure 3 is due to the random data shuffling. The second dataset (October 1st to November 3rd) is also used as testing data in order to evaluate the bot identification performance of the already trained (on the 70% data of September) XGBoost model on unseen data that correspond to an extended time horizon. The XGBoost model achieves an average of 0.896 F1 score and 0.977 ROC-AUC. The aforementioned results clearly indicate that our proposed ML model achieves impressive generalization capabilities by identifying bot accounts on future data based on past training samples.

Model Explainability

One of the ultimate goals of the current paper is to "unlock" the proposed ML model mechanism in order to better understand how the model yields its predictions. We use SHapley Additive exPlanations (SHAP) values proposed in (Lundberg and Lee 2017) since they present several advantageous characteristics. First and most importantly, SHAP values are model-agnostic, i.e., they are not bound to any particular type of ML model. Secondly, SHAP values present properties of local accuracy, consistency, and missingness, which are not found simultaneously in other methods. Lastly, SHAP implementation is actively supported by an open-source community[1], it is well documented and straightforward to use.

Before proceeding to the SHAP values explanation, let us first, provide a description of the concept of Shapley value.

[1]https://shap.readthedocs.io/

More specifically, Shapley introduced a game-theoretic approach for assigning fair payouts to players depending on their contribution to the total gain (Shapley 1953). Within a predictive modeling task, this translates to assigning an importance numerical value to features that depend on their contribution to a prediction. Thus, in the predictive ML context, a Shapley value can be defined as the average marginal contribution of a feature value across all possible feature coalitions. Based on this definition, a Shapley value for a given feature can be interpreted as the difference between the mean prediction for the whole dataset and the actual prediction.

The Shapley values are represented as a linear model of feature coalitions by the SHAP method (Lundberg and Lee 2017). SHAP values exploit the game theory's Shapley interaction index, which allows allocating payouts, i.e., importance, not just to individual players, i.e., features, but also among all pairs of them. As a result, SHAP values can explain the modeling of local interaction effects, and allow the possibility of providing new insights into the ML model's features.

Figure 5 shows the summary plot for SHAP values related with the features extracted from the US 2020 Elections dataset. The top twenty features with the highest impact at the XGBoost model's output are depicted. For each feature, one point corresponds to a single Twitter user. A point's position along the x-axis (i.e., the actual SHAP value) represents the impact that a feature had on the model's output for that specific Twitter user. Mathematically, this corresponds to the malicious behaviour risk relative across Twitter users (i.e., a Twitter user with a higher SHAP value has a higher risk being malicious relative to a Twitter user with a lower SHAP value). Features are arranged along the y-axis based on their importance, which is given by the mean of their absolute Shapley values. The higher the feature is positioned in the plot, the more important it is for the XGBoost model.

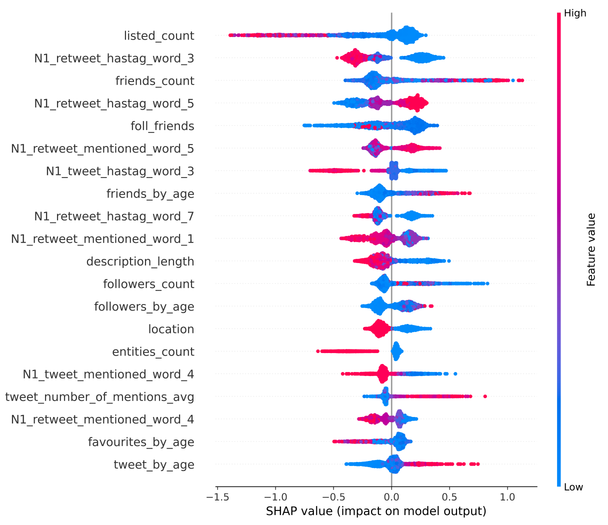

Figure 5: US 2020 Elections dataset: summary plot for SHAP values. The top twenty features with the highest impact at the XGBoost model's output are depicted.

A further analysis of the results in Figure 5 indicates that the top twenty features with the highest impact on the XGBoost model's output correspond to statistical, time, and graph-based features. In particular, features such as Twitter lists and average number of mentions in user tweets appear to have a high impact in XGBoost model's output. We expect that a combination of features with the highest output impact could provide the best possible bot identification performance. This statement can be confirmed by the results mentioned in Subsection generalization performance: US 2020 Elections dataset.

Based on the SHAP values summary plot depicted in Figure 5, it is obvious that "*listed_count*" corresponds to the feature with the highest impact at XGBoost model's bot vs. normal user detection. As shown in Figure 5 bot users tend to not belong to Twitter lists, whereas normal users could be members of more than one list. We can also deduce that bot users have lower values of "*favourites_by_age*" (also known as likes), which means that bot users tend to ignore the like button of other users' posts. This could be explained by the complexity of bot account implementation. Finally, we notice that bot users have high values of "*friends_by_age*" feature, which means that they tend to connect to more accounts within a short period of time. This activity is obvious since bot accounts try to gain high visibility and expand to larger parts of the Twitter network. Presented explanations confirm our initial intuitive explanations regarding the difference between normal and bot accounts activity.

Conclusions and Future Work

This paper introduces a novel methodology based on a supervised machine learning (ML) framework for identifying bot vs. normal Twitter users by extracting a wide range of features. Specifically, the proposed system incorporates the extraction and labeling of multiple features, with the ground truth labels estimated through the combination of two online bot detection tools' output. Following a thorough ML analysis involving train/validation/test split, feature selection, oversampling, and hyper-parameters tuning, we establish the Extreme Gradient Boosting (XGBoost) algorithm as the best ML model along with a specific set of features. The selected XGBoost model when trained on a wide range of combined features spanning from profile and context features to time-based and interaction features achieves the highest bot detection accuracy.

The generalization capability of the proposed ML system is extensively examined through an experimental evaluation process, and compared with a recently introduced general model (Yang et al. 2020). Finally, the obtained explanations revealed meaningful insights from a Twitter data analysis point of view about the reasoning process behind the XGBoost model's decisions. Future work concerns the extension of the proposed methodology by performing text analysis on the tweet corpus posted by the bot users in order to identify the shared type of content during the US 2020 Elections period.

The code and the employed datasets are available in a GitHub repository (USBotDetection 2022).

Acknowledgements

We would like to thank the reviewers for their valuable comments. This document is the result of the research projects CONCORDIA (grant number 830927), CyberSANE (grant number 833683) and PUZZLE (grant number 883540) co-funded by the European Commission, with (EUROPEAN COMMISSION Directorate-General Communications Networks, Content and Technology).

References

Ahmed, F.; and Abulaish, M. 2013. A generic statistical approach for spam detection in online social networks. *Computer Communications*, 36(10-11): 1120–1129.

Alsaleh, M.; Alarifi, A.; Al-Salman, A. M.; Alfayez, M.; and Almuhaysin, A. 2014. Tsd: Detecting sybil accounts in twitter. In *2014 13th International Conference on Machine Learning and Applications*, 463–469. IEEE.

Antonakaki, D.; Spiliotopoulos, D.; V. Samaras, C.; Pratikakis, P.; Ioannidis, S.; and Fragopoulou, P. 2017. Social media analysis during political turbulence. *PloS one*, 12(10): e0186836.

Arnaudo, D. 2017. Computational propaganda in Brazil: Social bots during elections. *Computational propaganda research project - University of Oxford*.

Badawy, A.; Ferrara, E.; and Lerman, K. 2018. Analyzing the digital traces of political manipulation: The 2016 Russian interference Twitter campaign. In *2018 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)*, 258–265. IEEE.

Bastian, M.; Heymann, S.; and Jacomy, M. 2009. Gephi: An Open Source Software for Exploring and Manipulating Networks. *International AAAI Conference on Weblogs and Social Media*.

Batista, G. E. A. P. A.; Prati, R. C.; and Monard, M. C. 2004. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *SIGKDD Explor. Newsl.*, 6(1): 20–29.

Bolsover, G.; and Howard, P. 2019. Chinese computational propaganda: automation, algorithms and the manipulation of information about Chinese politics on Twitter and Weibo. *Information, communication & society*, 22(14): 2063–2080.

Bouzy, C. 2021. Bot Sentinel. https://rb.gy/wfymew. Accessed: 2021-04-19.

Bovet, A.; and Makse, H. A. 2019. Influence of fake news in Twitter during the 2016 US presidential election. *Nature Communications*, 10(1): 7.

Breiman, L. 2001. Random Forests. *Machine Learning*, 45(1): 5–32.

Broniatowski, D. A.; Jamison, A. M.; Qi, S.; AlKulaib, L.; Chen, T.; Benton, A.; Quinn, S. C.; and Dredze, M. 2018. Weaponized health communication: Twitter bots and Russian trolls amplify the vaccine debate. *American journal of public health*, 108(10): 1378–1384.

Burnap, P.; and Williams, M. L. 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet*, 7(2): 223–242.

Byrnes, N. 2016. How the bot-y politic influenced this election. *Technology Rev.*, 100(10).

Chatfield, A. T.; Reddick, C. G.; and Brajawidagda, U. 2015. Tweeting Propaganda, Radicalization and Recruitment: Islamic State Supporters Multi-Sided Twitter Networks. In *Proceedings of the 16th Annual International Conference on Digital Government Research*, dg.o '15, 239–249. New York, NY, USA: Association for Computing Machinery. ISBN 9781450336000.

Chavoshi, N.; Hamooni, H.; and Mueen, A. 2016. Identifying correlated bots in twitter. In *International conference on social informatics*, 14–21. Springer, Springer.

Chen, T.; and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 785–794. New York, NY, USA: Association for Computing Machinery. ISBN 9781450342322.

Chu, Z.; Gianvecchio, S.; Wang, H.; and Jajodia, S. 2010. Who is tweeting on Twitter: human, bot, or cyborg? In *Proceedings of the 26th annual computer security applications conference*, 21–30.

Chu, Z.; Gianvecchio, S.; Wang, H.; and Jajodia, S. 2012. Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Transactions on Dependable and Secure Computing*, 9(6): 811–824.

Church, K. W. 2017. Word2Vec. *Natural Language Engineering*, 23(1): 155–162.

Cortes, C.; and Vapnik, V. 1995. Support-vector networks. *Machine learning*, 20(3): 273–297.

Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; and Tesconi, M. 2017. 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*, 963–972. ACM.

Davis, C. A.; Varol, O.; Ferrara, E.; Flammini, A.; and Menczer, F. 2016. BotOrNot: A System to Evaluate Social Bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*, 273–274.

Dickerson, J. P.; Kagan, V.; and Subrahmanian, V. S. 2014. Using sentiment to detect bots on Twitter: Are humans more opinionated than bots? In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, 620–627. IEEE/ACM.

Edwards, C.; Edwards, A.; Spence, P. R.; and Shelton, A. K. 2014. Is that a bot running the social media feed? Testing the differences in perceptions of communication quality for a human agent and a bot agent on Twitter. *Computers in Human Behavior*, 33: 372–376.

Feng, Y.; Li, J.; Jiao, L.; and Wu, X. 2020. Towards Learning-Based, Content-Agnostic Detection of Social Bot Traffic. *IEEE Transactions on Dependable and Secure Computing*, 7(3): 3.

Ferrara, E. 2017. Disinformation and social bot operations in the run up to the 2017 French presidential election. *First Monday*.

Ferrara, E.; Varol, O.; Davis, C.; Menczer, F.; and Flammini, A. 2016. The rise of social bots. *Communications of the ACM*, 59(7): 96–104.

Fortuna, P.; and Nunes, S. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4): 1–30.

Founta, A.; Djouvas, C.; Chatzakou, D.; Leontiadis, I.; Blackburn, J.; Stringhini, G.; Vakali, A.; Sirivianos, M.; and Kourtellis, N. 2018. Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. *CoRR*, abs/1802.00393: 5.

Fraisier, O.; Cabanac, G.; Pitarch, Y.; Besançon, R.; and Boughanem, M. 2018. #Élysée2017fr: The 2017 French Presidential Campaign on Twitter. In *International AAAI Conference on Web and Social Media (ICWSM '18)*.

Garimella, K.; and Weber, I. 2017. A Long-Term Analysis of Polarization on Twitter. *icwsm*, 145–152.

Gilani, Z.; Kochmar, E.; and Crowcroft, J. 2017. Classification of twitter accounts into automated agents and human users. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, 489–496. IEEE/ACM.

Gilani, Z.; Wang, L.; Crowcroft, J.; Almeida, M.; and Farahbakhsh, R. 2016. Stweeler: A Framework for Twitter Bot Analysis. In *Proceedings of the 25th International Conference Companion on World Wide Web*, 37–38.

Golovchenko, Y.; Buntain, C.; Eady, G.; Brown, M. A.; and Tucker, J. A. 2020. Cross-Platform State Propaganda: Russian Trolls on Twitter and YouTube During the 2016 US Presidential Election. *The International Journal of Press/Politics*, 21(3): 1940161220912682.

Hegelich, S.; and Janetzko, D. 2016. Are social bots on Twitter political actors? Empirical evidence from a Ukrainian social botnet. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 10.

Howard, P. N.; Bolsover, G.; Kollanyi, B.; Bradshaw, S.; and Neudert, L.-M. 2017a. Junk news and bots during the US election: What were Michigan voters sharing over Twitter. *CompProp, OII, Data Memo*, 21(3): 8.

Howard, P. N.; Bradshaw, S.; Kollanyi, B.; and Bolsolver, G. 2017b. Junk News and Bots during the French Presidential Election: What Are French Voters Sharing Over Twitter In Round Two? *ComProp data memo*, 21(3): 8.

Howard, P. N.; Kollanyi, B.; and Woolley, S. 2016. Bots and Automation over Twitter during the US Election. *Computational Propaganda Project: Working Paper Series*, 21: 8.

Howard, P. N.; Woolley, S.; and Calo, R. 2018. Algorithms, bots, and political communication in the US 2016 election: The challenge of automated political communication for election law and administration. *Journal of information technology & politics*, 15(2): 81–93.

Hui, P.-M.; Yang, K.-C.; Torres-Lugo, C.; and Menczer, F. 2020. BotSlayer: DIY Real-Time Influence Campaign Detection. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, 980–982.

Ibrahim, M.; Abdillah, O.; Wicaksono, A. F.; and Adriani, M. 2015. Buzzer Detection and Sentiment Analysis for Predicting Presidential Election Results in a Twitter Nation. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 1348–1353.

Illing, S. 2018. Cambridge Analytica, the shady data firm that might be a key Trump-Russia link, explained. https://rb.gy/zhcpm0. Accessed: 2021-04-19.

Jones, M. O. 2019. The gulf information war— propaganda, fake news, and fake trends: The weaponization of twitter bots in the gulf crisis. *International journal of communication*, 13: 27.

Keller, F. B.; Schoch, D.; Stier, S.; and Yang, J. 2017. How to manipulate social media: Analyzing political astroturfing using ground truth data from South Korea. In *Eleventh International AAAI Conference on Web and Social Media*, 811–824. AAAI.

Keller, T. R.; and Klinger, U. 2019. Social Bots in Election Campaigns: Theoretical, Empirical, and Methodological Implications. *Political Communication*, 36(1): 171–189.

Krebs, B. 2011. Twitter bots drown out anti-Kremlin tweets. https://rb.gy/xpaoze.

Lee, K.; Caverlee, J.; and Webb, S. 2010. Uncovering social spammers: social honeypots+ machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 435–442. ACM.

Lightfoot, S.; and Jacobs, S. 2017. Political propaganda spread through social bots. *Media, Culture, & Global Politics*, 8: 1–22.

Loyola-González, O.; Monroy, R.; Rodríguez, J.; López-Cuevas, A.; and Mata-Sánchez, J. I. 2019. Contrast pattern-based classification for bot detection on Twitter. *IEEE Access*, 7: 45800–45817.

Luceri, L.; Deb, A.; Giordano, S.; and Ferrara, E. 2019. Evolution of bot and human behavior during elections. *First Monday*.

Lundberg, S.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*.

Mazza, M.; Cresci, S.; Avvenuti, M.; Quattrociocchi, W.; and Tesconi, M. 2019. Rtbust: Exploiting temporal patterns for botnet detection on twitter. In *Proceedings of the 10th ACM Conference on Web Science*, 183–192. ACM.

Neudert, L.; Kollanyi, B.; and Howard, P. N. 2017. Junk news and bots during the german parliamentary election: What are german voters sharing over twitter? *Computational Propaganda - University of Oxford*, 21(3): 8.

Osome. 2020. Botometer (formerly BotOrNot) checks the activity of a Twitter account and gives it a score. Higher scores mean more bot-like activity. https://rb.gy/0hmj4x.

Rajaraman, A.; and Ullman, J. D. 2011. *Data Mining*, 1–17. Cambridge University Press.

Rizoiu, M.-A.; Graham, T.; Zhang, R.; Zhang, Y.; Ackland, R.; and Xie, L. 2018. #DebateNight: The Role and Influence of Socialbots on Twitter During the 1st 2016 U.S. Presidential Debate. In *International AAAI Conference on Web and Social Media (ICWSM '18)*.

ScikitLearn. 2022. Scikit-learn function: SelectFromModel. https://rb.gy/kipojk.

Seo, H. 2014. Visual propaganda in the age of social media: An empirical analysis of Twitter images during the 2012 Israeli–Hamas conflict. *Visual Communication Quarterly*, 21(3): 150–161.

Shane, S. 2017. The fake Americans Russia created to influence the election. *The New York Times*, 7(09).

Shao, C.; Ciampaglia, G. L.; Varol, O.; Yang, K.-C.; Flammini, A.; and Menczer, F. 2018. The spread of low-credibility content by social bots. *Nature communications*, 9(1): 1–9.

Shapley, L. S. 1953. A value for n-person games. *Contrib. Theory Games*, 2: 307–317.

Sharma, K.; Qian, F.; Jiang, H.; Ruchansky, N.; Zhang, M.; and Liu, Y. 2019. Combating fake news: A survey on identification and mitigation techniques. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(3): 1–42.

Stella, M.; Ferrara, E.; and De Domenico, M. 2018. Bots increase exposure to negative and inflammatory content in online social systems. *Proceedings of the National Academy of Sciences*, 115(49): 12435–12440.

Stukal, D.; Sanovich, S.; Bonneau, R.; and Tucker, J. A. 2017. Detecting bots on Russian political Twitter. *Big data*, 5(4): 310–324.

Subrahmanian, V.; Azaria, A.; Durst, S.; Kagan, V.; Galstyan, A.; Lerman, K.; Zhu, L.; Ferrara, E.; Flammini, A.; and Menczer, F. 2016. The DARPA Twitter Bot Challenge. *Computer*, 49(6): 38–46.

Tibshirani, R. 1994. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58: 267–288.

USBotDetection. 2022. GitHub Repository including code and dataset. https://github.com/alexdrk14/USBotDetection.

Varol, O.; Ferrara, E.; Davis, C. A.; Menczer, F.; and Flammini, A. 2017. Online human-bot interactions: Detection, estimation, and characterization. In *International AAAI Conference on Web and Social Media (ICWSM '17)*, 280–289.

Vosoughi, S.; Roy, D.; and Aral, S. 2018. The spread of true and false news online. *Science*, 359(6380): 1146–1151.

Wald, R.; Khoshgoftaar, T. M.; Napolitano, A.; and Sumner, C. 2013. Predicting susceptibility to social bots on twitter. In *2013 IEEE 14th International Conference on Information Reuse & Integration (IRI)*, 6–13. IEEE.

Waugh, B.; Abdipanah, M.; Hashemi, O.; Rahman, S. A.; and Cook, D. M. 2013. The Influence and Deception of Twitter: the authenticity of the narrative and slacktivism in the Australian electoral process. In *14th Australian Information Warfare Conference*, 28–38.

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

Yang, K.-C.; Varol, O.; Hui, P.-M.; and Menczer, F. 2020. Scalable and Generalizable Social Bot Detection through Data Selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1096–1103. AAAI.

Yaqub, U.; Chun, S. A.; Atluri, V.; and Vaidya, J. 2017. Analysis of political discourse on twitter in the context of the 2016 US presidential elections. *Government Information Quarterly*, 34(4): 613–626.

Zhang, C. M.; and Paxson, V. 2011. Detecting and analyzing automated activity on twitter. In *International Conference on Passive and Active Network Measurement*, 102–111. Springer.