Social Network Analysis and Mining (2024) 14:45
https://doi.org/10.1007/s13278-024-01211-2

ORIGINAL ARTICLE

Assembling a multi-platform ensemble social bot detector with applications to US 2020 elections

Lynnette Hui Xian Ng[1] · **Kathleen M. Carley**[1]

Received: 24 March 2023 / Revised: 15 September 2023 / Accepted: 24 January 2024

Abstract
Bots have been in the spotlight for many social media studies, for they have been observed to be participating in the manipulation of information and opinions on social media. These studies analyzed the activity and influence of bots in a variety of contexts: elections, protests, health communication and so forth. Prior to this analyzes is the identification of bot accounts to segregate the class of social media users. In this work, we propose an ensemble method for bot detection, designing a multi-platform bot detection architecture to handle several problems along the bot detection pipeline: incomplete data input, minimal feature engineering, optimized classifiers for each data field, and also eliminate the need for a threshold value for classification determination. With these design decisions, we generalize our bot detection framework across Twitter, Reddit and Instagram. We also perform feature importance analysis, observing that the entropy of names and number of interactions (retweets/shares) are important factors in bot determination. Finally, we apply our multi-platform bot detector to the US 2020 presidential elections to identify and analyze bot activity across multiple social media platforms, showcasing the difference in online discourse of bots from different platforms.

Keywords Bot detection · Twitter · Reddit · Instagram · Social media · Interpretability · Machine learning · US 2020 elections

1 Introduction

Social media bots, which are automated accounts, have been shown to participate in election interference (Ferrara et al. 2016), opinion manipulation in vaccination efforts (Ng and Carley 2022) and even extremism campaigns (Ferrara et al. 2016). The field of social cybersecurity is concerned with the problem of identifying these bot accounts because the bot campaigns can lead to negative offline impacts like protests.

A suite of bot detection models have been developed to characterize users on social media space as bot or humans. These bot detection models use techniques from feature-based detection to temporal detection to graph-based detection. However, the training and inference of these bot detection models often involves huge feature spaces, i.e., 1000+ extracted user features (Yang et al. 2019); or extensive data collection, i.e., temporal methods require longitudinal data and graph-based methods require network data. While the increase in feature space often results in improved performance (Yang et al. 2019), data collection becomes harder. With data collection requirements come the issue of incomplete data input: missing fields in input data due to data collection limitations, change in data formats, or unavailability of field. Unfortunately, the prevailing models typically rely on the completeness of account data to make a prediction. This is because models are typically tuned by the union of data features, and are thus unable to make a prediction with incomplete data.

After the data input is passed through bot detection algorithms, a bot probability score is typically returned. This score is between 0 and 1 and indicates the likelihood of an account being a bot. A threshold value is usually defined, where if the score is above the threshold, the account is deemed as a bot; and as a human otherwise. However, the threshold is usually arbitrarily determined,

✉ Lynnette Hui Xian Ng
lynnetteng@cmu.edu

Kathleen M. Carley
carley@andrew.cmu.edu

[1] Center for Computational Analysis of Social and Organizational Systems, Carnegie Mellon University, 4665 Forbes Avenue, Pittsburgh, PA 15213, USA

Published online: 24 February 2024

and values used ranged from 0.2 to 0.7, leading to a false positive problem (Rauchfleisch and Kaiser 2020; Ng et al. 2022; Yang et al. 2019). In fact, for Elon Musk's bot estimate in July 2022 during his Twitter acquisition negotiations, one key question was: what was the threshold value Musk used? (Clayton 2022) The choice of a threshold value can affect the determination of the proportion of bots, which will be different should different analysts choose different thresholds.

In this paper, we address the aforementioned problems, by designing a multi-platform ensemble architecture. Our architecture uses a small set of features for bot detection, separating the features into data chunks that represent user, user metadata and content features. Not only does this enable fine-tuning of separate classifiers for each data field, it also handles the problem of incomplete data where prediction can be made with the remaining classifiers. We aggregate bot/human probabilities before taking the larger value, eliminating the need to determine a threshold value.

As a result of these design decisions, we are able to generalize our bot detection framework across multiple platforms: Twitter, Reddit, and Instagram. Many of the bot detection models are currently constructed for the Twitter platform and there are few that analyze bot activities on other social media platforms, much less multiple platforms within a single bot detection architecture. We leveraged on training separate models for each data field in a parallel fashion before combining data across platforms. In this paper we also aim to improve the running capability of bot detection classifiers. Therefore, we leverage mostly on simpler tree-based classifiers instead of focusing on deep-learning based or graph-based classifiers, with the intent that our bot detection classifier can be run on a variety of machines, from low-powered to high-powered machines, thereby facilitating the analysis and research of bot detection. Our tree-based classification ensemble runs extremely quickly, completing 759 users in 3.9 min on an Intel Xeon-1250 CPU, which can facilitate large-scale bot detection. Across a series of 7 Twitter, 1 Reddit and 1 Instagram datasets, we show that our model outperforms baselines with an average accuracy of 75.47%.

The layout of this paper is as follows: in Sect. 2 we provide a brief literature review of bot detection models and bot detection on multiple social media platforms. Then, in Sect. 3, we describe the construction of the bot detection model. After building our bot detection model, we applied it to a slice of the online discourse on two social media platforms in Sect. 4, illustrating the use of our bot detection model on multiple platforms. Finally, we discuss the observations in our paper in Sect. 5 and provide concluding remarks in the final section.

2 Literature review

Social media bots, or fondly called "bots", refer to social media users that are software-controlled and can automatically perform a series of tasks. These types of user accounts are of keen interest to the social cybersecurity community because they have been observed to perform malicious activities online, which can affect the peace of society. They have observed to be used to infiltrate political discourse and spread misinformation. During the 2010 US midterm elections, social bots were already observed to have been flooding the social media space with their support for some candidates and smear their opponents by injecting thousands of tweets pointing to websites with fake news (Ferrara et al. 2016). Bots are also use by countries for digital diplomacy, to put forth a desired narrative facing the online public (Jacobs et al. 2023; Ng and Carley 2023). Bots working together in a coordinated fashion have also been known to apply social pressure on to other users, causing them to change their opinion toward key topics. This was observed in the case of the 2021 coronavirus vaccination debate, where users surrounded by coordinating active bots change their stance toward the vaccine, potentially resulting in an anti-vaccine stance and the real-world refusal of the vaccine (Ng and Carley 2022).

A suite of bot detection algorithms have been developed for the detection of automated social media bot accounts in key events such as elections and protests. A bot detection algorithm classification is a binary classification task: classifying whether a user is a bot or a human. For such a task, there are essentially two main approaches: a supervised learning approach where data labels are known and the model is trained on the segregation of data labels, and an unsupervised learning approach where the model discovers hidden patterns within the dataset.

Supervised learning models work based on identifying distinct set of features for each class in a dataset labeled bot and humans. These detection algorithms can be grouped into three types: feature-based, temporal-based and graph-based algorithms. All three types of algorithm features can be combined to be fed into a machine learning classifier, as in the case of T-Bot, which uses profile-based, user activity based and social network based features in its classification (Gera and Sinha 2022). Feature-based algorithms are algorithms that apply machine learning algorithms to features engineered from user and content information (Yang et al. 2019; Yang et al. 2020). Examples of such features are: average number of hashtags used per post, number of URLs used per post, average number of punctuations used per post, number of interactions per post (i.e., retweet, quote tweet, shares, likes), sentiment of the post and so forth. The machine learning models built on

features range from logistic regression classifiers (Heidari et al. 2021; Kantepe and Ganiz 2017), to support vector machines (Pratama and Rakhmawati 2019), to neural network-based classifiers (Kudugunta and Ferrara 2018).

Temporal-based models characterize accounts through time series pattern analysis and behavior activity occurrence (Cresci et al. 2018; Mazza et al. 2019). Another strategy is to make use of the patterns of inter-arrival times between posts and extract features to represent the circadian rhythm and cultural and environmental influences of a user for use in the classification model (Cai et al. 2017). The time interval between posts can also be processed to derive parameters that characterize the burst patterns or information entropy of posts and use them as classification features (Wu et al. 2021). These temporal features that are derived are eventually fed into a machine learning model which differentiates whether the user is a bot or human. These time-series methods, however, require a good length of post data across time of each account, which can be difficult to acquire given the volume of accounts and the platform's rate limits.

Graph-based models which make use of an account's social network graph to enhance predictions with information inferred from the account's neighbors (Feng et al. 2021). This technique builds on the concept of homophily, that users tend to interact with other similar users. The technique thus makes use of a matrix that reveals the connections formed between users, assuming the connections are formed with confidence, i.e., the same type of users tend to form connections with each other. This matrix is then put through a machine learning model, for example a graph-based regression model, to differentiate the user classes. Since a graph-based approach constructs matrices based on connections between users, it can be extended for use across many social media platforms (Al-Qurishi et al. 2018). One drawback of graph-based models, however, is that while they can be fairly accurate in determining bot-likelihood from an account's friends (~85% accuracy (Feng et al. 2021)), collecting the other users that a user is following/ follows him can be time and resource intensive. Graph-based methods also works mostly on Twitter and Instagram data for those platforms do have a follow/following feature, but Reddit does not reveal the users that follow a user.

Unsupervised learning approaches use anomaly detection or time-series based methods to extract connectivity of suspicious accounts. DeBot (Chavoshi et al. 2016) is an unsupervised classification algorithm that makes use of temporal patterns to determine the presence of bot accounts, inferring the presence of bot accounts using time series spikes. BotWalk compares each new user to a seed of bot/human user using an ensemble anomaly detection method (Minnich et al. 2017). Time-series based methods include algorithms like MulBot and RTBust. MulBot infers bot accounts through multivariate time series statistics of the user posts as features (Mannocci et al. 2022). RTBust constructs a univariate time series based on the time difference between retweets, which then is fed into an LSTM autoencoder (Mazza et al. 2019).

Finally, ensemble-based classification models are approaches that combine multiple classification models together to increase the accuracy of differentiating a user. Sayyadiharikandeh et al. (2020) developed an Ensemble of Specialized Classifiers to detect different types of bots, like spam bots and fake follower bots. This ensemble is made up of multiple Random Forests classifiers and aggregated through a voting system. Similarly, Dimitriadis et al. (2021) trained Random Forest classifiers based on content, user, temporal and social network features to differentiate between different bot types (e.g.,, political bots, spam bots, social bots etc). The value of an ensemble-based approach is that it can work well to produce outputs for multiple binary classification tasks that have disparate outcomes and inputs, then aggregate them together for the final outcome.

Most of the bot detection algorithms are designed for the detection of bots on Twitter, a microblogging platform. One of the commonly used Twitter bot detection algorithm is Botometer, which uses over 1000 features extracted from social media profiles to perform its classification. It uses the Twitter API to query the platform live, in order to provide the latest update as to the bot likelihood of the account, but in doing so is unable to perform predictions for historical data Yang et al. (2020).

Reddit is a forum-like site which is organized by interests, termed subreddits. Temporal analysis methods have been used in Reddit bot detection to classify accounts in terms of their bot-likelihood based on their temporal bursts between comments and their network connectivity between subreddits (Hurtado et al. 2019). Another feature engineering method analyzes the presence of a user account making Reddit submissions with same titles and the comment activity to characterize bot activity (Saeed et al. 2022).

In terms of bots in the image-based social media platform Instagram, classification algorithms have been developed with logistic regression, naive bayes or support vector machines that take in profile features such as follower/following counts, number of digits in the account username and so forth in order to provide a bot/human differentiation (Akyon and Kalfaoglu 2019). Instagram bot accounts that impersonate politicians, news agencies and sports stars have also been differentiated through clustering algorithms that makes use of profile metrics like number of posts, comments, likes and so forth (Zarei et al. 2019).

Table 1 Statistics of datasets used. We use the aggregation of these datasets to construct a bot detection algorithm. Many of the datasets contain partial data due to the unavailability of data at collection time. Therefore, our bot detection algorithm needs to be able to handle incomplete data and provide its best-guess prediction

Dataset	Users (% Bots)	Data present				
		User name	Screen name	Description	Posts	User metadata
botometer-feedback-2019 Yang et al. (2019)	529 (27)	Y	Y	P	P	P
botwiki-2019 Yang et al. (2019)	704 (100)	Y	Y	P	P	P
cresci-rtbust-2019 Mazza et al. (2019)	759 (52)	Y	Y	P	P	P
cresci-stock-2018 Cresci et al. (2018)	25987 (71)	Y	Y	P	P	P
midterms-2018 Yang et al. (2020)	50538 (84)	Y	Y	P	P	P
political-bots-2019 Yang et al. (2019)	62 (100)	Y	Y	Y	N	Y
reddit-2022	667 (75)	Y	N	Y	Y	Y
instagram-2022	1862 (100)	Y	Y	P	N	Y

Y: Data field present for all users

N: Data field not present for all users

P: Data field present for partial subset of users

3 Methodology

This section specifies the building of the multi-platform social bot detector, beginning with the description of the training datasets, then the description of the machine learning algorithms used in the construction and evaluation of the bot detection model, and finally we perform an evaluation on an external dataset.

3.1 Data

In building our multi-platform social bot detector, we used datasets from Twitter, Reddit and Instagram. By using datasets across multiple social media platforms, and thereafter training the bot detection model on this aggregated dataset, we are able to build a bot detection model that is able to analyze multiple social media platforms.

We used the following datasets: (1) Seven Twitter datasets extracted from the OSOME bot repository.[1] These datasets contain only user information from Twitter profiles and we rehydrated them with the Twitter V2 API in June 2022. Some accounts have since been suspended prior to the rehydration and only partial user and content information are available. (2) Reddit dataset was self-curated through extracting the top 500 “bad bots" flagged by Reddit users on B0tRank.[2] The dataset is enhanced with the human users who reported the bots. (3) Instagram bot dataset was self-curated through a purchase of fake follower bots. These bots follow a public account to increase the number of followers, providing the illusion of account popularity, increasing the influence of the account. We then harmonized the naming conventions data fields of all three platforms. Twitter provides the most data fields, whereas Reddit and Instagram do not posses all the fields, hence we work with partial data.

For each of the social media accounts, we identified several fields that are important to be used in our bot detection model: user name, screen name, description, posts and user metadata (i.e.,, number of followers, number of following). However, not all datasets provided all the information. This is due to two reasons: profile suspension on Twitter or that the platform does not provide the information. In these cases, we identify these fields as partially available for the datasets. For it to be useful, our social bot detector needs to be able to handle datasets in which the fields are partially available, and make use of the available fields to make a best-guess decision on whether the account is a bot or not.

Table 1 summarizes the dataset bot/human composition and data field availability (present, not present, partially present). We use an aggregation of these datasets to construct a bot detection algorithm. Many of these datasets contain incomplete information about the users, due to the unavailability of data at collection time. Therefore, our bot detection algorithm needs to be able to provide its best-guess prediction under the circumstances of incomplete data. We also have datasets from three different social media platforms, and we aim to construct a bot detection algorithm that is generic enough to apply to all three platforms In our work, data was only collected from public accounts and no attempt was made to access or use information that was not publicly available from the social media sites.

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

[2] https://botrank.pastimes.eu

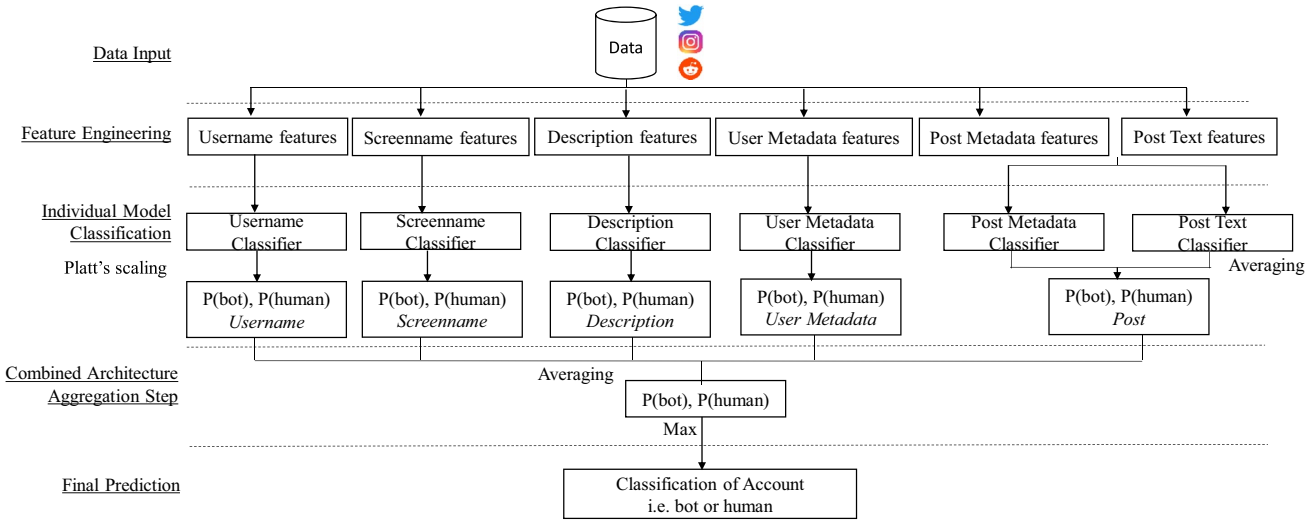


Fig. 1 Diagram of multi-platform bot detection ensemble. The ensemble is made up of six classifiers which extract and train/test on specialized features, providing a probability of bot/human. The probabilities are then aggregated together before the account's classification is determined by the higher of the two bot/human values

3.2 BotBuster for everyone: ensemble bot detection

We propose an ensemble method for multi-platform bot detection. This method is illustrated in Fig. 1. The bot detection pipeline in our proposed BotBuster For Everyone contains of five steps: data input, feature engineering, individual model classification, combined aggregation and final prediction. Each of the steps are described in further detail in the following subsections.

Our pipeline first begins with data input and feature engineering, to format the data from social media accounts and extract the salient features. Model training and testing is implemented using the scikit-learn Python package. Our ensemble method involves a two-step training/testing strategy: the individual model construction and the combined architecture construction. We use the accuracy score as an evaluation metric, in order to focus on correctly classified observations of both bot and human classes.

3.3 Data input

The data input step reads and processes user data, conforming the field names from each social media platforms to a common field mapping, thus dealing with multi-platform bot detection through commonality of data fields (i.e.,, a user on every platform has a userid and a username). It also provides an identifier at the initialization step to indicate which type of platform the data is being drawn from, so that the rest of the bot detection procedure can make use of the corresponding classifiers.

3.4 Feature engineering

The feature engineering step extracts a set of attribute from each data field for subsequent input into the field-specific classifiers. By tuning a classifier specific to each field, we are able to use a small set of features per field, keeping feature extraction and prediction time short. Table 2 includes a summary of the features extracted in this feature engineering step to be used by the individual models.

The features that are extracted from the social media accounts for use in our bot detection model are:

1. *Username:* A username is a singular unique word that identifies an account. It has been successfully used by its own to classify bots (Beskow and Carley 2019). We distill the username into the number of uppercase and lowercase letters and the number of digits and punctuations, and the measure of string entropy. We will elaborate on the calculation of string entropy for usernames later.
2. *Screenname:* Screennames are a longer name identifier for a user, and can contain multiple words and emojis. We use the same features as the Username field, but include the number of emojis, hashtags and words as additional features. Similar to username, we distill the screenname into the number of uppercase and lowercase letters and the number of digits and punctuations, and the measure of string entropy. We will elaborate on the calculation of string entropy for screennames later.
3. *Description:* The description is a short excerpt the user writes of himself. This field is broken down into words and the Term Frequency-Inverse Document Frequency

Table 2 Accuracy metrics of individual models

Data	Features used	Decision tree	Random forest	Gradient boosting	Ada boost
Username	string entropy, #uppercase letters, #lowercase letters, #digits, #punctuations, #emojis, #hashtags	**75.81**^{R,IG}	75.94^{IG}	72.72^{IG}	72.14^{R,IG}
Screenname	string entropy, #uppercase letters, #lowercase letters, #digits, #punctuations, #emojis, #hashtags, #words	**75.69**^{R,IG}	79.54^{IG}	72.08^{R,IG}	71.57^{IG}
Description*	TF-IDF	**70.48**^{IG}	69.84	81.59	79.26
User Metadata	#followers, #following, #listed, #posts, #likes, protected, verified	100^{IG}	74.80^{IG}	**100**^{R,IG}	100^{R,IG}
Posts**	TF-IDF, #likes, #retweets, #replies, #quotes	56.37	**81.02**^{R}	79.97	79.02^{R}

The final ensemble combination selected are highlighted in bold.

*No description data available for Reddit.

**No posts data available for Instagram.

R: Model gives non-zero accuracy for Reddit dataset IG: Model gives non-zero accuracy for Instagram dataset

(TFIDF) statistic of each word across the corpus of user descriptions are used as features.

4. *User Metadata:* User-based features have been successfully used in bot classification (Ferrara et al. 2016). We use the followers count, following count, total post count, total likes count and indicators of verified and protected accounts, if available from the data.
5. *Posts:* Data from each post is typically in the form of continuous text. This text is split into the main text and the corresponding post metadata. The main text is processed using the TFIDF statistic, while the metadata (number of likes, retweets, replies, quotes) is captured as a series of integers.

To calculate the entropy of usernames/screennames, we first collected 3.8million names from users who posted in the last year. Using this corpus of names, we constructed a frequency of characters used in the usernames, before using the dictionary to calculate username entropy. For a username X, its entropy $H(X)$ is: $H(X) = -\sum_{i=1}^{n} P(x_i) log_2 P(x_i)$, where $P(x_i)$ is the probability of the ith of the n characters appearing in the username. We curated our own list of names because the probability distributions of characters in social media usernames can differ from that of the English dictionary. Screenname entropy is calculated similarly.

3.5 Individual model classification

In our first step of training/testing, we constructed individual classification models for each data field. To do so, we first combine all the datasets to form a meta-training dataset in the following fashion. For each dataset, we partition it with a 80-20 train-test split with stratification by bot/human class. This ensures that there is the same proportion of bots/humans in each training/testing set. We chose to use a supervised classification approach because our datasets have already been collected and annotated by different groups of experts, and the use of supervised classification model means that the expected output is known beforehand.

All the training splits from each dataset are then combined into a meta-training dataset used for training the individual models, and the testing splits are combined into a meta-testing dataset for testing the individual models. Then, we performed experimentation across several tree-based classifiers: decision tree, random forest, gradient boosted trees and ada boosted trees. We selected these classifiers due to their speed, which will serve advantageous in bot/human classification of large-scale datasets when deployed in actual analysis studies. Past analysis studies that characterize bot activity have used datasets that are of sizes from 40,000 (Uyheng et al. 2021) to 240,000 (Luceri et al. 2019) to 2.7 million users (Ferrara 2017). With such dataset sizes, speed of classification is of concern when designing a bot detection model. Five-fold cross-validation is used in all our experiments and the average results are reported.

This step takes in features specific to each data field and run them through field-specific models. Each model returns a prediction of (bot, human) tuple, which contains two values, representing the probability of the user being a bot and a human respectively. Many bot detection classifiers make use of a threshold-based classification. If the resultant probability of a bot is above a certain threshold, the social media account is classified as a bot; if the resultant probability is below the same threshold, the social media account is classified as a human. Since the choice of the threshold value can affect the percentage of bots identified, we determined the bot/human class of the user through the higher of two values that represent the probability of a bot and a human.

Separating the data input for different classifiers enables the fine-tuning of classification models specific to the feature set of each data field. This structure also deals with the problem of incomplete data for a user. In the case of incomplete data, the pipeline performs classification using the rest of the individual classifiers for the data fields present.

The classifier for the missing field returns Null values. For example, if 3 data fields were present, the 3 corresponding classifiers will return a (bot, human) tuple, while 2 classifiers return null values.

Each individual model is then evaluated based on their overall accuracy and their ability to generalize across social media platforms. We selected the best model for each data field as the one with the highest accuracy and does not give a 0% accuracy score for the Reddit and Instagram datasets. After measuring overall accuracy on the meta-testing dataset, we partitioned out the non-Twitter users and perform an evaluation on them to quantify the models' accuracy on these datasets. This method sometimes sacrifices a little overall accuracy but ensures multi-platform generalizability. For example, for classification using the Description data field, the Gradient Boosting classifier performs the best at 81.59% accuracy, but it gives a 0% accuracy for the Instagram dataset. We then selected the Decision Tree classifier which performed at 70.48% overall accuracy but gave a non-zero accuracy for the Instagram dataset, and so will be able to evaluate user accounts originating from the Instagram platform.

The final set of chosen classifiers are: decision tree for username, screenname and description, gradient boosting classifier for user metadata and random forest for posts. After choosing these best classifiers, we retrained the individual models and their outputs are calibrated using Platt's scaling, adapting the idea from past work on specialized ensembles for different types of bots (Sayyadiharikandeh et al. 2020). This scaling is implemented using the Calibrated Classifier function.[3] Platt's scaling calibrates the outputs of each classifier into a probability distribution using logistic regression. This therefore makes the probability returned in the (bot, human) tuple in each of the six classifiers comparable.

3.6 Combined aggregation

The combined aggregation step aggregates the non-null (bot, human) probability scores for the individual classifiers. The final bot classification is determined by the larger of the values in the final (bot, human) tuple. In this fashion, the need for determining a suitable threshold to classify whether an account is a bot or a human is eliminated, reducing ambiguity of the classification. This step creates the ensemble model, combining the different individual models together, and can therefore better generalize data features as a whole (Sayyadiharikandeh et al. 2020).

No further model training is required to combine the individual models. The bot and human probabilities generated by the individual models are averaged out to produce a final bot/human probability. Testing occurs dataset by dataset, where all accounts of each dataset are evaluated for its bot probability, and the final accuracy is reported. We perform two evaluation metrics: the first, we evaluate model accuracy on the data points that the model can process, ignoring unprocessed data points. The second, we set the prediction of users that cannot be processed as the "human" class before making an overall accuracy comparison. This mimics the use of bot detection algorithms in analysis: any user not marked as a "bot" is typically not considered when analyzing bot behavior, and thus treated as "humans".

Although this method of evaluation means that some data points were previously seen by the algorithms, this preserves evaluation consistency as the baseline algorithms are also trained on the same datasets and we are unable to perform separate testing on them.

3.7 Model evaluation

3.7.1 Baseline algorithms

We compare our bot detection algorithm implementation against two commonly employed algorithms: BotHunter (Beskow and Carley 2018) and Botometer (Yang et al. 2019). Both algorithms are constructed using random forests for Twitter data. BotHunter uses a tiered approach that includes more user and content features as the tiers progress. Botometer is an ensemble method that relies on the real-time query of the profile. Botometer was also used by Elon Musk when estimating the proportion of Twitter bots (Clayton 2022).

3.7.2 Combined aggregation evaluation

On average, our combined framework performed with an overall accuracy of 75.47% across the nine datasets. A summary of our results are presented in Table 3, which presents the overall accuracy and the percentage of the dataset each algorithm processed. The overall accuracy is calculated by assuming that the unprocessed data are humans. This assumption mimics how one typically uses bot detection in an analysis: zooming in on the positively identified bot users and analyzing the rest of the users as a human or non-bot class separately. A more detailed summary of the results are presented in Table 6, together with other accuracy metrics that account for proportion of bot/humans in the training and testing datasets.

Our bot detection framework outperforms the overall accuracy of both baselines, which fares at 35.92% accuracy for the BotHunter baseline and 31.46% accuracy for Botometer baselines. The better performance of our ensemble framework can be attributed to the fact that it can process

[3] https://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html

Table 3 Summary of results for bot detection algorithms. BotHunter and botometer are unable to process all the data, while BotBuster for everyone method is able to. The overall accuracy is calculated by assuming the unprocessed data are humans

Dataset	BotHunter Overall accuracy (% processed)	Botometer Overall accuracy (% processed)	BotBuster for everyone Overall accuracy (% processed)
Botometer-feedback-2019	57.60 (61.44)	59.05 (71.07)	83.08 (100)
Botwiki-2019	53.12 (90.34)	48.12 (92.90)	91.60 (100)
Cresci-rtbust-2019	61.89 (74.97)	69.43 (78.78)	71.65 (100)
Cresci-stock-2018	37.20 (40.57)	39.25 (47.03)	74.61 (100)
Midterms-2018	13.20 (11.26)	14.15 (1.31)	85.23 (100)
Political-bots-2019	0 (0)	17.33(20.60)	74.54 (100)
Verified-2019	88.60 (100)	35.50 (98.15)	99.57 (100)
Reddit-2022	0.30 (0)	0 (0)	35.68 (100)
Instagram-2022	0 (0)	0 (0)	60.26 (100)
Average	34.62 (42.06)	31.42(45.52)	75.14 (100)

Table 4 Twibot-20 evaluation

Algorithm	% Processed	Overall sccuracy	MicroF1 score	MacroF1 score
BotHunter	99.15	45.98	41.48	49.02
Botometer	91.38	33.02	35.98	30.86
BotBuster For Everyone	100	57.32	44.62	50.69

partial data and data on non-Twitter platforms. While selecting individual models, there were some cases we sacrificed accuracy for non-zero accuracy on Reddit/Instagram dataset. The accuracy of each type of individual model is reported in Table 2. This shows that not all model architectures are equally adept at differentiating bot/human features across platforms, and can be overwhelmed by the larger volume of Twitter data. However, the accuracy scores of Reddit and Instagram datasets are lower, indicating that bot features may be slightly different on the three social media platforms.

Both baseline algorithms are unable to process partial data: BotHunter relies on the complete field set while Botometer relies on the survival of the account. However, our method breaks up the data into chunks for processing, allowing evaluation based on the available data. This is useful in the case of incomplete data collection or unavailable user during collection. Separating the classifiers into individual classifiers for each data field enables optimization of models for each data field, and allows us to piece together an ensemble of different types of classifiers.

3.7.3 External evaluation

To ensure the robustness of our bot detector, we perform an evaluation on an external dataset, a dataset that has not been used in the model training before. This measures how well the model does on a dataset which it has not seen the features before, adding value to its ability to perform on out-of-domain datasets.

We use Twibot-20 dataset (Feng et al. 2021) for this external evaluation. the Twibot-20 was collected in 2020 via a snowball sampling method from seed users across politics, business, entertainment and sports.

BotBuster For Everyone evaluated all the data points in the Twibot dataset and performed at 57.32% accuracy. Our architecture outperforms the baseline BotHunter and Botometer algorithms in terms of accuracy. In terms of the number of data points processed, BotBuster For Everyone is able to process all data points unlike BotHunter and Botometer, which are not able to evaluate all the data points due to missing data fields. Table 4 presents the statistics of the evaluation ran on the Twibot-20 dataset.

3.7.4 Full data fields evaluation

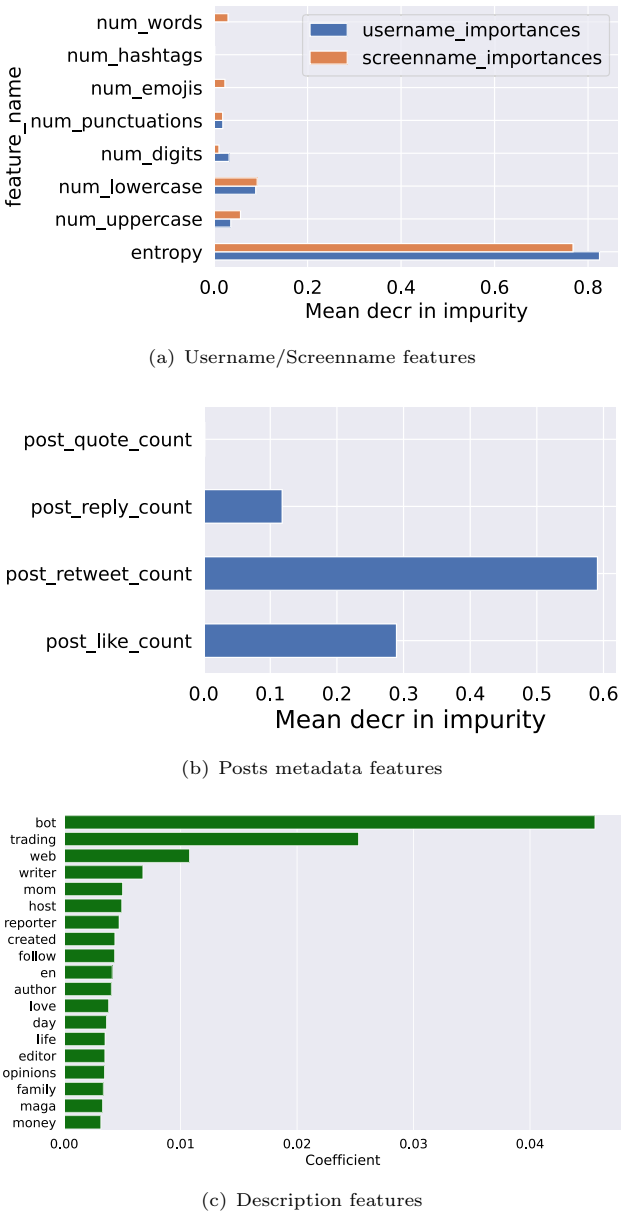
We also ask the question of how much does performance decrease in the ensemble classifier with all features in the input. For each dataset, we extracted out data points that have all the data items and ran the ensemble algorithm on those items. We report the accuracy of the ensemble classifier based on the proportion of correctly classified users out of the number of users that we are able to obtain the complete data set. We note that BotHunter and Botometer

Table 5 Summary of results for BotBuster for everyone for processing data points with full data fields. The bot detection setup does not lose a large amount of accuracy for handling incomplete data, yet it does improve the range of data that can be analyzed by the bot detector

Dataset	Full data fields (% Processed)	All data points (% processed)
botometer-feedback-2019	86.55 (54.30)	83.08 (100)
botwiki-2019	95.20 (84.94)	91.60 (100)
cresci-rtbust-2019	73.67 (41.90)	71.65 (100)
cresci-stock-2018	78.54 (26.40)	74.61 (100)
midterms-2018	87.93 (14.55)	85.23 (100)
political-bots-2019	NA (0)	74.54 (100)
verified-2019	99.67 (88.45)	99.57 (100)
reddit-2022	NA (0)	35.68 (100)
instagram-2022	NA (0)	60.26 (100)
Average	86.93 (51.76)	75.14 (100)

only processes data points with full data, while our method can process incomplete data.

Table 5 tabulates the accuracy of the ensemble algorithm where the full data can be evaluated. Although the accuracy of the classifier with full data is higher, but because of the proportion of users with incomplete data, we think that it is worth sacrificing greater accuracy to exploit incomplete data. Even so, the individual models are trained on data points where the data for that model is available, hence it is akin to building a classifier for the full data. In addition, if we were to require all data fields to be present before performing a classification, some datasets will not be analyzed. For example, Reddit does not have a "screen_name", which is required for most bot detection classifiers. Therefore, this breaks the ability of our bot detection classifier to handle multiple platforms. This analysis lends weight to the architecture of our bot detector for being constructed to handle incomplete data fields: it does not lose a large amount of accuracy for handling incomplete data, yet it does improve the range of data that can be analyzed by the bot detector.



(a) Username/Screenname features

(b) Posts metadata features

(c) Description features

Fig. 2 Feature importances. The most indicative feature of bot classification is the number of retweets/shares a post receives, followed by the number of likes and the number of replies

3.8 Feature importance analysis

In our feature extraction implementation, we kept the feature spaces small. Despite these, we are still able to achieve decent algorithm accuracy, showcasing that bot detection need only rely on a few key features for a decent accuracy. This provides directions for further bot account analysis: characterizing the defining features of bot accounts in contrast to human accounts.

We make some observations to our feature space in the username, screenname, post metadata and description classifiers. We extracted the feature importances of each of the estimators stored in Python's sklearn classifiers of the best performing classifiers for each data class. We graphed the results in Fig. 2.

For the username/screenname and post metadata features, they are numeric features, and hence the tree-based classifiers separate them through the decrease in impurity. The mean decrease in impurity calculates feature importances as the sum over the number of splits across the tree-based

classifier. The higher mean decrease in impurity, the more important the feature is in differentiating the final bot/human class. For username/screenname, the entropy of the name string plays a large factor in the determination of bot classification. This is consistent with previous studies that characterized the randomness of profile names as an indicator of automation (Beskow and Carley 2019). The R package Tweetbotornot primarily evaluates the bot likeliness of a user based on its username (Kearney 2018). It is also observed that the presence of digits in usernames and emojis in screennames are indications of bot/human classes of the account.

In terms of post metadata (Fig. 2), we observe that the most indicative feature of a bot classification is the number of retweets/shares a post receives, followed by the number of likes and the number of replies. This is consistent with the feature analysis of bot detection algorithm MulBot where retweets and replies are the more important features (Mannocci et al. 2022). This means that posts by bot accounts have a lot more shares than human accounts, possibly pointing to their ability to construct more viral posts or indications of bot networks working together to increase influences of posts of other bots within the network.

The description of an author is a string of words, and hence is treated differently by the Decision Tree classifier. In constructing the classifier, the description string is broken down into a bag of words, therefore the feature importances of the words are represented by coefficients, where the coefficient scores how important the word is within a description string. The first word is “bot", suggesting the incorporation of a heuristic to identify key signals of bot accounts such as words present in the description or account name (Livingstone 2022). Words representing a person’s identity (i.e., writer, mom, host, author, reporter, editor etc.) are extremely indicative words, suggesting connections between the expression of identities and bot likeliness of an account. This opens avenues for further investigation on the correlation on identity expression and automation.

4 Application of social bot detector

With the construction of the ensemble bot detection algorithm which we named BotBuster For Everyone, we applied it to a slice of the online discourse extracted from the US 2020 Elections from Twitter and Reddit. The 2020 United States presidential elections was held on 3 November 2020. In this election, Democratic president Joe Biden defeated incumbent Republican president Donald Trump. After the win by Biden, Trump and his supporters did not concede, and claimed voter and election fraud. Past research have been done to analyze users that have similarities across both platforms surround this incident of protest against voter fraud, and having bot detection capabilities can enhance these analysis by providing the perspective of the degree of automation of these users (Murdock et al. 2023).

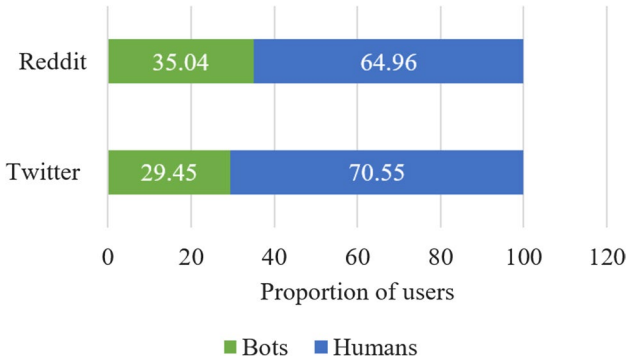


Fig. 3 Proportion of user types present in the US 2020 presidential elections. There is a higher proportion of bot users in Reddit than in Twitter

We perform a short study on the discourse of the protest of election voter fraud. We collected social media conversations for a week after the elections, from 3 November to 9 November 2020, analyzing the bot activity within this timeframe.

The Twitter data were collected with the Twitter V1 API and the Reddit data using the Pushshift API (Baumgartner et al. 2020). For Twitter data, we have 4,351,111 unique posts and 1,183,313 unique users. The discourse was not as active on Reddit, and we retrieved a smaller amount of data from Reddit, collecting a total of 4403 unique posts and 2449 unique users.

We apply our constructed BotBuster for Everyone to identify bot users in the datasets. The same bot detection model is applied to both Twitter and Reddit data, extracting and analyzing the bots present in both datasets. Figure 3 shows the proportion of user types extracted from these datasets. There is a higher proportion of bot users that are present in the Reddit conversation (35.04%) as compared to Twitter (29.45%). This observation shows that the election discussion by bots is focused on Reddit, in which the subreddit and reply structures do facilitate discussions, as compared to Twitter discussions.

We then separated the post texts written by each type of user in each of the platforms, and perform Latent Dirichlet Allocation (LDA) to identify the key narrative themes within the texts. The LDA algorithm returns a list of keywords relating to each theme, after which, the authors manually looked through and interpreted the themes, combining

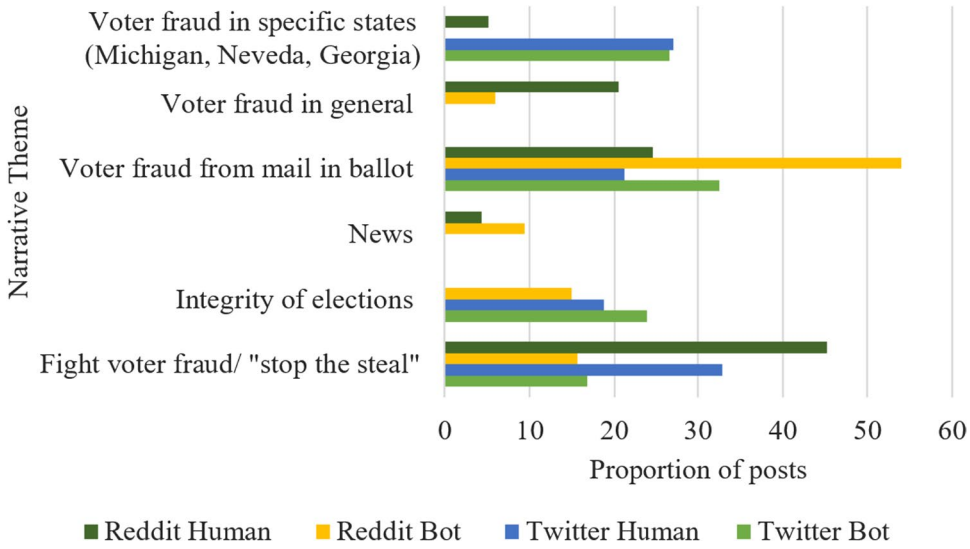


Fig. 4 Proportion of narrative themes present per user type in the US 2020 presidential elections. There are different focuses of each of the user class: bots disseminate information, while human users advocate for action

them where necessary. Six key themes emerged among the discourse: voter fraud in specific states, in particular the hotly contested states of Michigan, Neveda and Georgia; voter fraud in general; voter fraud from mail-in ballots; news regarding the elections and protest; the questioning of the integrity of the elections; and the call to fighting voter fraud, in particular using the catchphrase “stop the steal".

Figure 4 represents the proportion of posts by narrative themes. The narrative of voter fraud from mail in ballot and the call to fight voter fraud is present throughout both platforms, and echoed by both classes of users. The narrative of voter fraud by mail-in ballot is most echoed by Reddit bots followed by Twitter bots, while the calling out to fight voter fraud is most echoed by Reddit humans then Twitter humans. This shows the different focus of each of the user class: bots disseminate disinformation, i.e., insinuating that mailed-in ballots were rigged, and thus the elections were rigged; while human users advocate for action.

In this work, we analyzed only a small proportion of online discourse using our multi-platform bot detection model as an illustration that the model can be used to identify bots on multiple social media platforms. A subsequent step stemming from this identification analysis is to include a study into the social media accounts identified as bots and their differences with human accounts, however, that is outside the scope of this paper.

5 Discussion

In this work, we built BotBuster For Everyone, a multi-platform social media bot detector. This model identifies bot accounts from three main platforms: Twitter, Reddit and Instagram. The input format for these platforms that are currently built-in are: Twitter V1 API, Twitter V2 API, Reddit Pushshift API and Instagram data from CrowdTangle. There is also a “custom" format option, where users can edit a JSON file to specify the mapping between the field names of their data to the bot detector’s input fields.

5.1 Handling incomplete data fields

One highlight of our bot detection model is its ability to process data where not all fields are present. In our experiment of Full Data Fields Evaluation, we observe that while the bot detection algorithm performs better when processing data where the full data fields are present, it only does so slightly better. Therefore, we put forth that it is worth sacrificing a little accuracy for a wider use of the bot detection algorithm. Further, the ability to handle incomplete data fields lends the algorithm the ability to handle multiple platforms, for our experiments show that Twitter has the larger feature set that can be extracted from the platform, while other platforms have a smaller feature set. The ability to process datasets where the data features are missing for some data points or are not present as fields in the platform allows analysis of a lot more data points especially historically collected data, and reduces the collection burden.

5.2 Multi-platform generalizability

Our bot detection model can identify bots from multiple social media platforms, reducing the need to source and run multiple bot detection models for cross-platform studies, thus saving time and aggregating results. Leveraging on the fact that bots across different social media platforms have similar features, we are able to generalize our framework across three social media platforms. While there have been many bot detectors built for the Twitter platform, there are very few built for Reddit and Instagram. Our model contributes to the small set of bot detectors built for Reddit

and Instagram, while aggregating training data from the bot detection repositories for Twitter. The aggregation step combines heterogenous datasets as training data which teaches the models bot/human features at different time periods and behavioral patterns, making the model more generalizable to detect different types of bots (Hayawi et al. 2022). With the ability to analyze multiple platforms, our bot detection model thus provides the opportunity to perform cross-platform user and discourse analysis on social media.

Bot accounts are also observed to work together as sophisticated and coordinated bot networks, which have been observed in the 2021 French protests (Ng and Carley 2023), and during the 2014 Crimean water crisis (Khaund et al. 2021). Literature on identification of bot networks usually involve identifying bots singly before inferring their coordination with each other (Khaund et al. 2021; Pacheco et al. 2021). Because of the coordination features, where the group of bots are motivated by a single intent, they leave behind more automation than single bots, which allows detection through bot detection mechanisms (Cresci 2020). As such, our ensemble algorithm can aid in the identification of bot networks through the initial step of bot classification of single users.

5.3 Specialized fine-tuned classifiers

Our ensemble-based bot detection framework that fine-tunes specialized classifiers for each data class before aggregating the probabilities. This means that each classifier is specially designed to fit for the corresponding data class, making it more accurate for the data input from the data class.

The separate fine-tuning mechanism allows the overall bot detection architecture to handle cases of incomplete data. When there is incomplete data, that is, the data fields are not present in the input data, the corresponding specialized classifiers are unable to make a prediction. The rest of the specialized classifiers can still make a prediction on the prevailing data as they are trained separately and are analyzing different data fields, thus being unaffected by the lack of data from one field. This provides the ability of our bot detection model to provide predictions for as much users as possible, rather than only users with the complete suite of data fields.

Separating the classifiers also allows further interpretability of each classifier. Through analyzing each of the classifiers, we can highlight indicative features of bot accounts, such as randomness of usernames and the presence of identity terms within a user's description.

In addition, the use of tree-based algorithms in the construction of the ensembles illustrate the principle of Occam's razor: that the models can be simple enough to be valid. Our aggregation of tree-based classifiers perform almost as well as deep-learning based classifiers, e.g.,, an accuracy score of 71.65% for the cresci-rtbust-2019 dataset vs 72% by LSTM-based model DeeProBot (Hayawi et al. 2022); 83.08% for the botometer-feedback-2019 dataset vs 78.4% for a concatenation of multiple LSTM models (Arin and Kutlu 2023). With similar accuracy ranges, we infer that in terms of differentiating bot and human classes and features, simpler classifiers work equally as well as complex classifiers. In fact, the simplicity of tree-based classifiers means that such a bot detection algorithm can be easily run and do not require heavy GPU-processing that deep learning based ones do.

5.4 Eliminating the need for threshold selection

Reflecting both the probability of bot and human as an output of the framework not only eliminates the need for selecting a threshold value for bot/human classification. Past work has shown that the proportion of bots can differ greatly between commonly used classification thresholds (50%, 75%, 80%), as much as a 15% difference. The classification threshold means that above which the user is deemed as a bot and below which it is deemed as a human (Adel Alipour et al. 2022). The elimination of threshold value thus removes the ambiguity of the selection of users as bots, and be helpful in increasing the consistency of bot detection. This also allows an analyst to objectively see the range of bot likeliness for the account as compared to its human likeliness. While typically we use the class that is indicated as the higher of the two values for the final classification, providing both likelihood values gives the analyst a chance to decide the bot/human difference is too small and thus use other features, including manual inspection, to determine the classification.

5.5 Providing multi-platform perspectives

We applied our bot detection model toward the US 2020 presidential elections and used it to understand the differences in discourse that happened on multiple social media platforms. This is especially useful because social media discussions are not isolated to a single platform, and to gain a full perspective of the online chatter, one must analyze multiple platforms. Our bot detector provides the capability to analyze multiple platforms at once, segregating the automated bot agents, allowing for subsequent analysis for better understanding the discussion on the event.

5.6 Limitations and future work

One key limitation of constructing bot algorithms is obtaining a representative set of annotated data. Several of the Twitter datasets are skewed toward financial and election bots. The classifiers trained on this data will be overfitted toward these topics, leaving bots that operate under other themes undetected. The Reddit dataset relies on crowdsourced ranking of bots via majority voting, which is subject

Table 6 Summary of final results for BotBuster for everyone algorithm. The overall accuracy is calculated by assuming the unprocessed data are humans

Dataset	BotBuster For everyone			
	% processed	Accuracy	Micro-F1 score	Macro-F1 score
botometer-feedback-2019	100	83.08	83.08	78.81
botwiki-2019	100	91.60	34.34	48.55
cresci-rtbust-2019	100	71.65	70.83	70.13
cresci-stock-2018	100	74.61	74.43	74.38
midterms-2018	100	85.23	89.73	49.46
political-bots-2019	100	74.54	78.64	68.45
verified-2019	100	99.57	99.42	50.29
reddit-2022	100	34.68	51.20	33.86
instagram-2022	100	60.26	69.67	37.61
Average	100	50.69	44.62	56.84

Table 7 Summary of final results for BotHunter algorithm. The overall accuracy is calculated by assuming the unprocessed data are humans

Dataset	BotHunter				
	% processed	Accuracy (processed)	Accuracy (overall)	Micro-F1 score (Overall)	Macro-F1 score (Overall)
botometer-feedback-2019	61.44	74.10	57.60	70.65	69.75
botwiki-2019	90.34	53.13	53.12	69.39	34.69
cresci-rtbust-2019	74.97	62.90	91.89	62.98	65.87
cresci-stock-2018	40.57	37.36	37.20	48.09	35.93
midterms-2018	100	11.26	15.30	13.20	9.02
political-bots-2019	0	0	0	0	0
verified-2019	88.60	100	100	100	100
reddit-2022	0	0	0	0	0
instagram-2022	0	0	0	0	0
Average	52.45	57.13	40.37	40.01	34.55

Table 8 Summary of final results for Botometer algorithm. The overall accuracy is calculated by assuming the unprocessed data are humans

Dataset	Botometer				
	% processed	Accuracy (processed)	Accuracy (overall)	Micro-F1 score (Overall)	Macro-F1 score (Overall)
botometer-feedback-2019	71.07	53.68	59.05	60.95	63.89
botwiki-2019	92.90	92.89	48.12	50.07	46.37
cresci-rtbust-2019	78.78	38.12	69.43	69.42	69.38
cresci-stock-2018	47.03	38.12	39.25	44.10	41.39
midterms-2018	47.03	11.90	14.15	12.40	7.22
political-bots-2019	1.31	20.60	17.33	21.48	20.96
verified-2019	20.60	30.20	35.50	25.71	27.53
reddit-2022	0	0	0	0	0
instagram-2022	0	0	0	0	0
Average	45.52	31.72	31.43	31.57	30.75

to social influence. Lastly, although the purchased Instagram dataset has no ambiguity of bot classification, it contains only users of the positive bot class and requires further curation for a balanced dataset. It also only contains one type of bot, the follower bot, and more work is required to characterize and consolidate a dataset spanning different bot types.

Future work involves sampling representative bot datasets both within and across platforms to improve generalizability of the classifiers across the scope of social media. Bot/human account features on social media are continually evolving. For example, past work has observed that across the years of bot detection data development and collection, the linguistic features of posts of bots/humans have been observed to evolve (Ng and Carley 2023). In addition, new technologies such as generative language technologies can change the behavior and the feature set of both bot and human users (Arin and Kutlu 2023). Therefore, future versions of this supervised bot detector involves continual updating and training of the model in order to keep up to date with the latest bot/human account similarities and differences. We acknowledge that the limitation to supervised learning bot detection models means that researchers need to continually observe the space, but it also opens avenues for research and observation of bot/human behavior changes.

6 Conclusion

In this work, we constructed a multi-platform social bot detector, Botbuster For Everyone, which works through assembling an ensemble of tree-based models. Each tree model is specific to a feature extracted from a social media profile, and is individually trained. The outputs of each model are aggregated together to return a probability of whether an account is a bot or a human. Breaking down the account evaluation by features provides BotBuster for Everyone the capability to deal with incomplete data where prediction can be made using available data and classifiers, and incorporate data from multiple platforms by using similarities in data field names. The use of tree-based classification models provides the ability to perform bot detection and classification quickly on lighter-weight computing hardware with CPUs, thus increasing the access of bot detection models.

We also applied BotBuster for Everyone on a dataset of US 2020 Elections discourse, analyzing the topic differences between bot and humans on Reddit and Twitter. We found that there is a higher proportion of bot users in the collected Reddit conversation than Twitter users, and that bot accounts are typically used for (dis)information dissemination and human users for advocating action.

Bot accounts can threaten the health of our social, and even economic, systems. As such, there is a need for continual efforts in bot detection research to detect these accounts at scale. As social media discourse diversify across platforms, there must be a bot detection model that is able to seamlessly identify automated accounts across multiple platforms quickly and at scale. We designed Botbuster For Everyone, a bot detection framework which generalizes across three social media platforms. We hope that our work provides inspiration for future research on bot detection and investigation.

Appendix A. Full accuracy metrics

Tables 6, 7 and 8 presents the full set of accuracy metrics for the aggregated bot detection classifier and the baseline classifiers.

Acknowledgements The research for this paper was supported by the following grants: Cognizant Center of Excellence Content Moderation Research Program, Office of Naval Research (Bothunter, N000141812108), Scalable Technologies for Social Cybersecurity/ARMY (W911NF20D0002), Air Force Research Laboratory/CyberFit (FA86502126244). The views and conclusions are those of the authors and should not be interpreted as representing the official policies, either expressed or implied.

Author contributions LHXN conceptualized and conducted the experiments, and wrote the manuscript. KMC reviewed the manuscript. All authors approved of the final manuscript.

Funding Open Access funding provided by Carnegie Mellon University.

Data availibility Data used for training the bot detection models are publicly available. Data collected for the US 2020 elections can be made available through the corresponding author, in accordance to Twitter's terms and conditions. Code to the bot detection model will be made publicly available upon acceptance of this paper (it is currently in this Github repository which is marked private https://github.com/quarbby/BotBuster-4-Everyone).

Declarations

Conflict of interest The authors declare that there are no conflicting interests.

References

Adel Alipour S, Orji R, Zincir-Heywood N (2022) Security of social networks: lessons learned on twitter bot analysis in the literature. In: Proceedings of the 17th international conference on availability, reliability and security, pp 1–9

Akyon FC, Kalfaoglu ME (2019) Instagram fake and automated account detection. In: 2019 Innovations in Intelligent systems and applications conference (ASYU), pp 1–7. IEEE

Al-Qurishi M, Alrubaian M, Rahman SMM, Alamri A, Hassan MM (2018) A prediction system of sybil attack in social network using deep-regression model. Future Gener Comput Syst 87:743–753

Arin E, Kutlu M (2023) Deep learning based social bot detection on twitter. IEEE Transact Inf Forensics Sec 18:1763–1772

Baumgartner J, Zannettou S, Keegan B, Squire M, Blackburn J (2020) The pushshift reddit dataset. In: Proceedings of the international AAAI conference on web and social media, vol 14, pp 830–839

Beskow DM, Carley KM (2019) Its all in a name: detecting and labeling bots by their name. Comput Math Organ Theory 25(1):24–35

Beskow DM, Carley KM (2018) Bot-hunter: a tiered approach to detecting & characterizing automated activity on twitter. In: Conference paper. SBP-BRiMS: international conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation, vol 3, p 3

Cai C, Li L, Zengi D (2017) Behavior enhanced deep bot detection in social media. In: 2017 IEEE international conference on intelligence and security informatics (ISI), pp 128–130. IEEE

Chavoshi N, Hamooni H, Mueen A (2016) Debot: twitter bot detection via warped correlation. In: Icdm, vol 18, pp 28–65

Clayton J (2022) Doubts cast over Elon Musk's Twitter bot claims. BBC. https://www.bbc.com/news/technology-62571733

Cresci S (2020) A decade of social bot detection. Commun ACM 63(10):72–83

Cresci S, Lillo F, Regoli D, Tardelli S, Tesconi M (2018) Fake: Evidence of spam and bot activity in stock microblogs on twitter. In: Twelfth international AAAI conference on web and social media

Dimitriadis I, Georgiou K, Vakali A (2021) Social botomics: a systematic ensemble ml approach for explainable and multi-class bot detection. Appl Sci 11(21):9857

Feng S, Wan H, Wang N, Li J, Luo M (2021) Twibot-20: a comprehensive twitter bot detection benchmark. In: Proceedings of the 30th ACM international conference on information & knowledge management, pp 4485–4494

Ferrara E (2017) Disinformation and social bot operations in the run up to the 2017 French presidential election. First Monday. https://doi.org/10.5210/fm.v22i8.8005

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

Ferrara E, Wang W-Q, Varol O, Flammini A, Galstyan A (2016) Predicting online extremism, content adopters, and interaction reciprocity. In: International conference on social informatics, pp 22–39. Springer

Gera S, Sinha A (2022) T-bot: Ai-based social media bot detection model for trend-centric twitter network. Soc Netw Anal Min 12(1):76

Hayawi K, Mathew S, Venugopal N, Masud MM, Ho P-H (2022) Deeprobot: a hybrid deep neural network model for social bot detection based on user profile data. Soc Netw Anal Min 12(1):43

Heidari M, James Jr, H, Uzuner O (2021) An empirical study of machine learning algorithms for social media bot detection. In: 2021 IEEE international IOT, electronics and mechatronics conference (IEMTRONICS), pp 1–5. IEEE

Hurtado S, Ray P, Marculescu R (2019) Bot detection in reddit political discussion. In: Proceedings of the fourth international workshop on social sensing, pp 30–35

Kantepe M, Ganiz MC (2017) Preprocessing framework for twitter bot detection. In: 2017 International conference on computer science and engineering (ubmk), pp 630–634. IEEE

Kearney MW (2018) GitHub - mkearney/tweetbotornot: R package for detecting Twitter bots via machine learning — github.com. https://github.com/mkearney/Tweetbotornot. [Accessed 06-09-2023]

Khaund T, Kirdemir B, Agarwal N, Liu H, Morstatter F (2021) Social bots and their coordination during online campaigns: a survey. IEEE Transact Comput Soc Syst 9(2):530–545

Kudugunta S, Ferrara E (2018) Deep neural networks for bot detection. Inf Sci 467:312–322

Livingstone RM (2022) Trump bots and algorithmic experimentation on twitter. First Monday. https://doi.org/10.5210/fm.v27i11.12392

Luceri L, Deb A, Giordano S, Ferrara E (2019) Evolution of bot and human behavior during elections. First Monday. https://doi.org/10.5210/fm.v24i9.10213

Mannocci L, Cresci S, Monreale A, Vakali A, Tesconi M (2022) Mulbot: Unsupervised bot detection based on multivariate time series. In: 2022 IEEE international conference on big data (Big Data), pp 1485–1494. IEEE

Mazza M, Cresci S, Avvenuti M, Quattrociocchi W, Tesconi M (2019) Rtbust: exploiting temporal patterns for botnet detection on twitter. In: Proceedings of the 10th ACM conference on web science, pp 183–192

Minnich A, Chavoshi N, Koutra D, Mueen A (2017) Botwalk: efficient adaptive exploration of twitter bot networks. In: Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017, pp 467–474

Murdock I, Carley KM, Yağan O (2023) Identifying cross-platform user relationships in 2020 us election fraud and protest discussions. Online Soc Netw Med 33:100245

Ng LHX, Carley KM (2022) Pro or anti? a social influence model of online stance flipping. IEEE Transact Netw Sci Eng 10(1):3–19

Ng LHX, Carley KM (2023) Do you hear the people sing? comparison of synchronized URL and narrative themes in 2020 and 2023 French protests. Front Big Data. https://doi.org/10.3389/fdata.2023.1221744

Ng LHX, Robertson DC, Carley KM (2022) Stabilizing a supervised bot detection algorithm: How much data is needed for consistent predictions? Online Soc Netw Med 28:100198

Ng LHX, Carley KM (2023) Botbuster: Multi-platform bot detection using a mixture of experts. In: Proceedings of the international AAAI conference on web and social media, vol 17, pp 686–697

Pacheco D, Hui P-M, Torres-Lugo C, Truong BT, Flammini A, Menczer F (2021) Uncovering coordinated networks on social media: methods and case studies. In: Proceedings of the international AAAI conference on web and social media, vol 15, pp 455–466

Pratama PG, Rakhmawati NA (2019) Social bot detection on 2019 Indonesia president candidate's supporter's tweets. Procedia Comput Sci 161:813–820

Rauchfleisch A, Kaiser J (2020) The false positive problem of automatic bot detection in social science research. PloS one 15(10):0241045

Saeed MH, Ali S, Blackburn J, De Cristofaro E, Zannettou S, Stringhini G (2022) Trollmagnifier: detecting state-sponsored troll accounts on reddit. In: 2022 IEEE symposium on security and privacy (SP), pp 2161–2175. IEEE

Sayyadiharikandeh M, Varol O, Yang K-C, Flammini A, Menczer F (2020) Detection of novel social bots by ensembles of specialized classifiers. In: Proceedings of the 29th ACM international conference on information & knowledge management, pp 2725–2732

Uyheng J, Ng LHX, Carley KM (2021) Active, aggressive, but to little avail: characterizing bot activity during the 2020 Singaporean elections. Comput Math Organ Theory 27(3):324–342

Wu Y, Fang Y, Shang S, Jin J, Wei L, Wang H (2021) A novel framework for detecting social bots with deep neural networks and active learning. Knowl Based Syst 211:106525

Yang K-C, Varol O, Davis CA, Ferrara E, Flammini A, Menczer F (2019) Arming the public with artificial intelligence to counter social bots. Human Behav Emerg Technol 1(1):48–61

Yang K-C, Varol O, Hui P-M, Menczer F (2020) Scalable and generalizable social bot detection through data selection. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 1096–1103

Zarei K, Farahbakhsh R, Crespi N (2019) Typification of impersonated accounts on instagram. In: 2019 IEEE 38th international performance computing and communications conference (IPCCC), pp 1–6. IEEE

Charity S, Jacobs Lynnette Hui Xian, Ng Kathleen M, Carley Robert, Thomson Samer, Al-khateeb Annetta, Burger Patrick, Park Aryn, A. Pyke (2023) Social Cultural and Behavioral Modeling 16th International Conference SBP-BRiMS 2023 Pittsburgh PA USA September 20–22 2023 Proceedings Tracking China's Cross-Strait Bot Networks Against Taiwan Springer Nature Switzerland Cham 115-125

Lynnette Hui Xian, Ng Kathleen M, Carley (2023) Deflating the Chinese balloon: types of Twitter bots in US-China balloon incident Abstract EPJ Data Science 12(1) https://doi.org/10.1140/epjds/s13688-023-00440-3