Zayed University

ZU Scholars

All Works

12-1-2021

Hybrid feature selection approach to identify optimal features of profile metadata to detect social bots in Twitter

Eiman Alothali
United Arab Emirates University

Kadhim Hayawi
Zayed University

Hany Alashwal
United Arab Emirates University

ORIGINAL ARTICLE

Hybrid feature selection approach to identify optimal features of profile metadata to detect social bots in Twitter

Eiman Alothali[1] · Kadhim Hayawi[2] · Hany Alashwal[1]

Abstract

The last few years have revealed that social bots in social networks have become more sophisticated in design as they adapt their features to avoid detection systems. The deceptive nature of bots to mimic human users is due to the advancement of artificial intelligence and chatbots, where these bots learn and adjust very quickly. Therefore, finding the optimal features needed to detect them is an area for further investigation. In this paper, we propose a hybrid feature selection (FS) method to evaluate profile metadata features to find these optimal features, which are evaluated using random forest, naïve Bayes, support vector machines, and neural networks. We found that the cross-validation attribute evaluation performance was the best when compared to other FS methods. Our results show that the random forest classifier with six optimal features achieved the best score of 94.3% for the area under the curve. The results maintained overall 89% accuracy, 83.8% precision, and 83.3% recall for the bot class. We found that using four features: *favorites_count*, *verified*, *statuses_count*, and *average_tweets_per_day,* achieves good performance metrics for bot detection (84.1% precision, 81.2% recall).

Keywords Bot detection · Feature selection · Supervised learning · Twitter

1 Introduction

The exponential growth of data from different sources has increased the number of features and attributes available for analytics. Consequently, selecting the appropriate features for the job is clearly necessary to reducing the dimensionality of data and improving the performance of machine-learning algorithms (Devi and Sabrigiriraj 2018; Shah and Patel 2016; Visalakshi and Radha 2014). The feature selection (FS) goal is to eliminate irrelevant and redundant data to improve prediction accuracy and reduce execution time (Visalakshi and Radha 2014). Additionally, FS can enhance the understanding of attributes and the interpretation of data.

There are four basic steps for selecting features, as addressed in Visalakshi and Radha (2014) (see Fig. 1). Step 1 refers to the generation process needed to create a subset of features starting from nulls. Step 2 includes the evaluation of the subset using filter and wrapper methods. In Step 3, the stop condition is checked to determine whether features are to be added or removed. Finally, in Step 4, validation procedures are applied to confirm whether selected features are valid. The objective of these steps is to obtain an optimal subset of predictive features to enhance the learning process for a given problem.
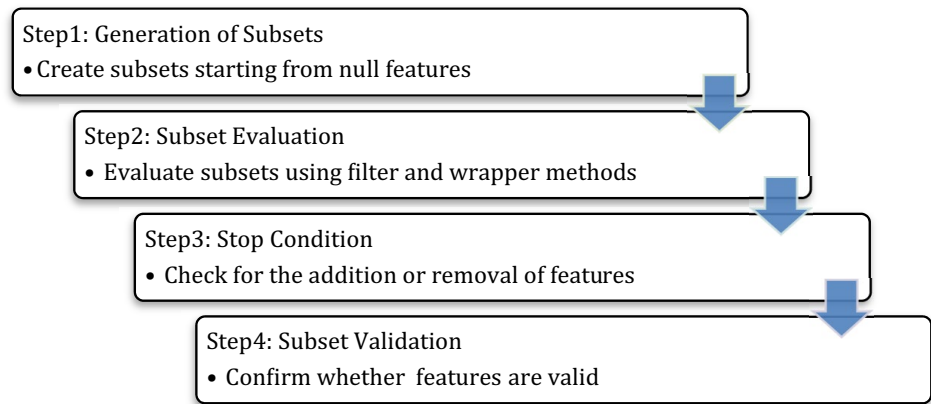
An existing problem in social networks is the identification of social bot accounts (Alothali et al. 2018). Social bots make online social networks vulnerable to adversaries. These programs automatically generate content, distribute it via a particular social network, and interact with users (Ferrara et al. 2016). Varol et al. (2017) found that between 9 and 15% of Twitter accounts were bots. Another study found that social bots were responsible for generating 35% of the content posted on Twitter (Abokhodair et al., 2015). Many studies have aimed to address the problems associated with the use of automated accounts on social networks (Subrahmanian et al. 2016; Grier et al. 2010; Stringhini et al. 2010; Wang 2010), which spread spam, worms, and phishing links, or they manipulate legitimate accounts by hijacking and deceiving users (Zhang et al. 2012; Rathore et al. 2017; Shafahi et al. 2016).

✉ Hany Alashwal
  halashwal@uaeu.ac.ae

1 College of Information Technology, United Arab Emirates University, AlAin, UAE

2 College of Technological Innovation, Zayed University, Abu Dhabi, UAE

Fig. 1 Basic steps to feature selection

Step1: Generation of Subsets
• Create subsets starting from null features

Step2: Subset Evaluation
• Evaluate subsets using filter and wrapper methods

Step3: Stop Condition
• Check for the addition or removal of features

Step4: Subset Validation
• Confirm whether features are valid

In this regard, finding the optimal features to be used to detect social bots on Twitter is an area rich for investigation, owing to their fast adaptability. For example, a past study applied deep learning for bot detection using pruned features, achieving almost 99% at the profile level and 96% at the tweet level (Kudugunta and Ferrara 2018). They used a public dataset that was collected in 2015 by (Cresci et al. 2017). Older generations of social bots had easily identifiable features, such as a randomized *screen_name* with no meaning (Beskow and Carley 2019). They also used default images known as "egg accounts," as this was the default initial image for any new account. However, more recently, these flaws have been resolved, and feature-detection accuracy has dropped. Thus, machine-learning models that detect bot features must be at least as adaptive as the bots (Cresci et al. 2019).

To address this issue, we investigated profile features for bots and human accounts on Twitter. We used available datasets that were discussed in Yang et al. (2020) and retrieved by Martin-Gutierrez et al. (2021) to examine these features. Furthermore, we used filter and wrapper methods to identify the best feature subsets and machine-learning methods (e.g., random forest (RF), naïve Bayes (NB), support vector machine (SVM), and neural network (NN)) for evaluation.

In this paper, we propose a hybrid technique to evaluate metadata features of profile accounts to enhance the recall scores of bot classes. We combine the strengths of the wrapper approach to minimize the number of features for the generated lists of filter-feature methods. Using wrapper approaches provides the best performing feature subset for a given learning algorithm but with high computational complexity, which increases directly with the number of features. Similarly, using filter methods separately to rank features independently without the engagement of any learning algorithm is insufficient for detecting active social bots. However, they can detect fake followers accounts because they remain inactive (Khalil et al. 2017). Therefore, the use of a hybrid approach helps reduce the space of features, and

it improves the time complexity by integrating both filter and wrapper techniques. Reducing the feature space is essential for real-time detection approaches for social bots to avoid the curse of dimensionality (Dadkhah et al. 2021; Ariyaluran et al. 2019).

Our work contributes to the literature in three ways:

- Profile features of bot detection are examined using filter and wrapper methods.
- A new algorithm is proposed to identify optimal features to detect social bots on Twitter.
- Feature subsets are evaluated using different machine-learning classifiers.

The rest of the paper is structured as follows: In Sect. 2, we present related works on FS with a subsection on bot detection features. In Sect. 3, we discuss our dataset and FS experiments. The results of the experiments are presented in Sect. 4. In Sect. 5, a discussion of the results is provided, and Sect. 6 concludes our work and presents future research directions.

2 Related work

We discuss related works under two broad categories. The first deals with FS methods. The second reviews different evaluations of the predictive features for bot detection.

2.1 FS methods

The large scale of massive data generated from different sources on the Internet presents challenges for data analysis and knowledge extraction. The challenge of FS is that the velocity of data generation is nonlinear, and most of the generated data are heterogeneous (Li and Liu 2017). As mentioned in Fig. 2, there are three general methods for FS: wrappers, filters, and embedded methods (Devi and

Fig. 2 Key feature selection methods
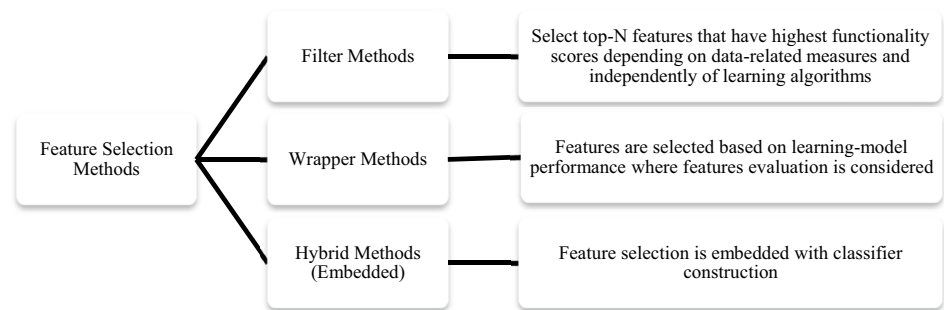
Sabrigiriraj 2018; Shah and Patel 2016). In the wrapper technique, features are selected based on learning-model performance, where the evaluation of features must be considered. In the study by Wald et al. (2013a), they used the wrapper technique to evaluate a classification model for Twitter data. They considered whether matching the learners inside and outside the wrapper was optimal, finding that the NB learner performed best and that the multi-layer perceptron could build consistent classification models for different choices of internal learners.

Using the filter technique, the goal is to find the top-N features having the highest functionality scores depending on data-related measures while being independent of learning algorithms. Using Twitter data, theoretical processes, and natural language-processing (NLP) methodologies, Ostrowski (2014) explored filtering techniques to improve trend detection and information extraction. The use of pointwise mutual information as feature technique with Bayesian classifier achieved improved results to support prediction models for unstructured data.

In embedded techniques, an optimal subset of features is constructed using classifier generation. These methods combine wrapper and filter approaches within the construction of the classifier. It has three types of embedded methods: pruning, built-in, and regularization (Tang et al. 2014). With pruning methods, the goal is to evaluate all features during training and eliminate some features while maintaining model performance. To elaborate, in this approach, the built-in mechanism for FS uses classifiers, such as a decision tree. Additionally, the regularization models aim to minimize fitting errors while eliminating coefficients close to zero. Therefore, each technique has certain limitations. For example, in the filter technique, the feature dependency is not considered, and in wrapper and embedded techniques, they are computationally slower than filters.

When FS is completed, feature extraction is performed (Shah and Patel 2016). In feature extraction, the goal is to extract a new set of features from the generated ones from the feature selection phase. Principal component analysis (PCA) is a popular method of feature extraction.

It is commonly used for the dimensional reduction of massive numbers of data. PCA is used to produce a lower-dimensional feature set from the original dataset using an orthogonal linear transformation (Shah and Patel 2016). A study by Morchid et al. (2014) analyzed tweet features using PCA to understand the behavior of massive retweets, focusing on nine features of user profile and tweet metadata to be transferred from the original space representation to a set of linearly uncorrelated variables (i.e., factors). Their SVM approach showed an 86.9% recall and a precision of 59.8%. Another study by Kondor et al. (2013) evaluated the feasibility of identifying the regional characteristics of language use using PCA on geo-tagged Twitter messages. They successfully separated low-rank and sparse data points and identified some main features in both.

2.2 Bot detection features in Twitter

As a social network, Twitter yields massive information about users, content, interaction, and networks. There are three approaches to detecting social bots: feature-based, graph-based, and crowdsourcing (Ferrara et al. 2016). In the feature-based approach, machine- and deep-learning algorithms are used to identify social bots based on account features, such as profile images and account age. It also evaluates account behavior by measuring the ratios of posting activities. In the graph-based detection approach, the network topology reveals that there is a fellowship relationship between accounts that can be addressed as network nodes. These connecting links provide information for bot detection (Minnich et al. 2017). In crowdsourcing, a human expert evaluates and manually labels an account as a bot or human (Gilani et al. 2016). This approach consumes time and human effort and is prone to human error.

Accordingly, FS is important for finding optimal features to distinguish humans from bot users in social

Fig. 3 Twitter features categories
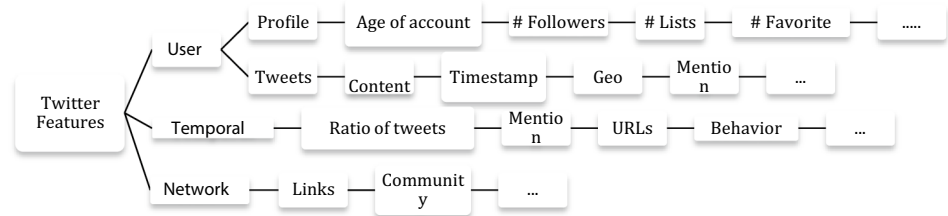
networks. The features that can be retrieved from a user profile include account and tweet features (Kudugunta and Ferrara 2018). Varol et al. (2017) identified more than 1000 features that can be extracted from Twitter. These features were used to build a botometer detection system that is available to the public.[1] A review study summarized 20 common features for bot detection that have been used to measure the likelihood of an account being a human or bot (Alothali et al. 2018). The extracted features from Twitter, as shown in Fig. 3, use network features to identify community features; user profile features are extracted from the metadata, such as profile images, screen names, and descriptions. Additionally, the temporal pattern features of an account, such as averages of tweeting and retweeting ratios, can reflect bot activity if it occurs with small inter-arrivals (Cai et al. 2017).

A recent study by Shukla et al. (2021) evaluated profile metadata features using an ensemble machine-learning algorithm. The proposed work used weight-of-evidence (WoE) encoding profile features using three ensemble learning algorithms: RF, AdaBoost, and artificial NNs. They used three FS methods: PCA, univariate FS, and the extra-tree classifier. The experimental results showed that a blending ensemble technique with extra-tree features achieved a score of 0.93 for the area under the receiver operating characteristic (ROC) curve (AUC). However, the recall score was 0.76, which indicates that the system was unable to identify many of the bots.

A study by (Martin–Gutierrez et al. 2021) proposed Bot-DenseNet for bot detection on Twitter using transfer-learning techniques to extract multilingual representations of text-based features for a user account. They used deep NNs to encode text-based features into vectors and concatenated them with the metadata features of the user account. The proposed work did well to overcome constraints of language as independently input text. Notably, many proposed works on bot detection that use NLP techniques focus on English-based accounts. The experimental results showed a 0.77 F1 score.

A recent study by Khalil et al. evaluated different features for bot detection using unsupervised learning (Khalil et al. 2020). They found that these features (i.e., follower-count, friends-count, favorite-count, listed-count, retweet-count, reply-count, hashtag-count, and mention-count) achieved 97.7% accuracy, 91% precision, 98% recall, and 94% f-measure using density-based spatial clustering of applications with noise. They used an available 2015 dataset collected by (Cresci et al. 2017). Dataset performance was good, but it lacked new examples of social bots and their new features and behaviors, which were a result of Twitter policy changes related to automation (Twitter.com 2020a). Another study by Wald et al. evaluated three forms of FS to predict user interaction with twitter bots via reply or mention (Wald et al. 2013b). These three FSs were filter-based feature ranking, filter-based subset evaluation, and wrapper-based subset selection. They found that feature ranking produced better models than either of the subset-based techniques, and the SVM classifier performed best with ranking compared with the five-nearest-neighbor and NB classifier.

3 Methods

In this section, we discuss the demonstration of our dataset and the approach that was followed to select predictive features to classify an account class as human or bot. We highlight the performance of these predictive features using four supervised machine-learning algorithms: RF, NB, NB, and NN.

3.1 Dataset

In our experiment, we used a public dataset available from Kaggle.com (Martín-Gutiérrez 2020), which was retrieved from previous studies that discussed bot detection in different events (Yang et al. 2020). It was available separately in the data repository link of Botometer (2020). The dataset contained 37,438 label instances that were classified as "bot" or "human." The total number of bots in this dataset was 12,425, and 25,013 were human user accounts. For our experiment, we used a stratified technique to obtain a 10% data size to reduce computational time and memory. The

---

[1] https://botometer.osome.iu.edu/

Table 1 Feature descriptions

#	Attributes (Features)	Description
1	Created_at	The date of creating an account
2	Default_profile	TRUE that the user has not altered the theme or background of their profile
3	Default_profile_image	TRUE that the user has not uploaded a profile image and the default image is used
4	Description	The user-defined description of their account
5	Favourites_count	The number of tweets a user has liked since the account creation date
6	Followers_count	The number of users (accounts) who are following this user account
7	Friends_count	Total number of accounts this user is following
8	Geo_enabled	TRUE if the current user attaches geographic data when tweeting or retweeting
9	Id	Unique identifier for the user account (64-bit)
10	Lang	The language that Twitter detects for a user account; if no language is detected (undefined)
11	Location	User-defined location in account profile
12	Profile_bg_image_url	The URL of the background image of an account
13	Screen_name	The name of the user as provided
14	Statuses_count	Total number of tweets and retweets issued by the user
15	Verified	Indication that the user has a verified account
16	Average_tweets_per_day	The average tweets and retweets that a user post per day (avg. = total posts/age of account in days)
17	Account_age_days	The total number of days since the account was created until the day of retrieving the data

Content adapted from Twitter.com (2020b)

total number of bots was 621, and that of humans was 1,250. We used the entire dataset for comparison with a benchmark study (Shukla et al. 2021) to validate our proposed technique.

The total number of features of the dataset included 18 attributes and the class output. These attributes represent some of the account profile retrieved metadata and their descriptions, as shown in Table 1. The descriptions are taken from a Twitter website data dictionary for developers (Twitter.com 2020b).

For the preprocessing task, we used normalization for a number of attribute data to save computation time and memory. We normalized the description attribute and *default_image_url* to be either zero or one to indicate it having a description or not being in the profile and having a default image address. We similarly normalized *default_profile*, *default_profile_image*, *geo_enable* and verified TRUE or FALSE to indicate the existence of such options in the profile.

3.2 Feature selection

In this section, we discuss the different feature selection methods applied to find the optimal feature subset. Our objective was to estimate the highest predictive score of bot accounts using a subset of features generated by different FS methods. After we ranked features based on weight and sorted them in descending order using filter FS methods, we defined the maximum number for each subset that can be investigated per method as in Algorithm 1.

We began by running the wrapper subset evaluation (Kohavi and John 1997) method to identify a ceiling number (k) from the 17 features, so each subset would have similar (k) number of features. The wrapper method used the best-first search approach to search the space of features and assess the prediction performance using a machine-learning algorithm. Because wrappers methods are a powerful approach for FSs, having a large number of features with exhaustive search capabilities is computationally expensive (Guyon and Elisseeff 2003).

After identifying (k), each ranked and sorted subset by filter FS methods is evaluated by the learning-model performance gradually using fewer features and by observing the performance. We stopped the iteration after recognizing a drop in performance when using a threshold of 0.05, which signifies that there was no improvement in performance for any of the evaluation metrics (i.e., precision (PTC), recall (RTC), and AUC).

We explored four different FS techniques for supervised learning. We used three filter methods and one wrapper method. These techniques included correlation attributes (CA), information gain (InfoGain), cross-validation attribute evaluation (CVAE), and wrapper subset evaluation (WSE). The correlation attribute method assesses the worthiness of an attribute by measuring Pearson's coefficient between it and the class. Using the information gain method, the worthiness of an attribute is evaluated by measuring the information gain with respect to the class. CVAE appraised the worthiness of an attribute by computing the value of the

Algorithm 1: *Finding the optimal predictive features*

Input:	Dataset D as a matrix of (N × M), where N is the number of features F=$\{f_1, f_2, ..., f_N\}$, and M is the number of Twitter accounts A= $\{a_1, a_2, ..., a_M\}$, *FSM* =*{m: m is a* filter feature selection method}, and CA = {c: c is a classifier algorithm}
Variables	*k:* to save total number of features, *b***:** to save k features, b_{Temp} : to save current k features **PTC**: to save Precision of Target Class, **RCT**: to save Recall of Target Class, **AUC**: to save Area Under the Curve. **PTC$_t$** : to save current Precision of Target Class, **RTC$_t$**: to save current Recall of Target Class, **AUC$_t$**: to save current Area Under the Curve
Output:	Finding the best features (b) for each m in FSM

1:   **Run** wrapper subset evaluation method
2:   Set $\boldsymbol{k}$ = total number of features in the subset generated by the wrapper method
3:   **for** m in FSM
4:      Run m on F to get all features ranked and sorted in descending order
5:      $\boldsymbol{b} = \{f_i : i = 1, ..., k\}$
6:      **for** c in CA
7:         **Run** c on $\boldsymbol{b}$ to get ***PTC, RTC, AUC***
8:         **do**
9:            $\boldsymbol{k = k - 1}$;   // **removing the least weight feature**
10:           $\boldsymbol{b_{Temp}} = \{f_i : i = 1, ..., k\}$
11:           **Run** c on $\boldsymbol{b_{Temp}}$ to get ***PTC$_t$, RTC$_t$, AUC$_t$***
12:         **If** $(PTC_t > PTC$ **and**
13:           $RTC_t > RTC$ **and**
14:            $AUC_t > AUC$)   **then**
15:             $PTC \leftarrow PTC_t$
16:             $RTC \leftarrow RTC_t$
17:             $AUC \leftarrow AUC_t$
18:             $\boldsymbol{b = b_{Temp}}$
19:         **else if** ($\frac{PTC - PTC_t}{PTC} < 5\%$ **and**
20:               $\frac{RTC - RTC_t}{RTC} < 5\%$ **and**
21:               $\frac{AUC - AUC_t}{AUC} < 5\%$) **then**
22:               **if** $AUC_t > AUC$ **then**
23:                  $PTC \leftarrow PTC_t$
24:                  $RTC \leftarrow RTC_t$
25:               $AUC \leftarrow AUC_t$
26:               $\boldsymbol{b = b_{Temp}}$
27:         **else** Break
28:        **while** ($k > 1$)
29:        **Output** b;
30:      **end for**   // classifiers
31:   **end for**   // features selection method

Table 2 Feature subsets list and rank for each technique

Technique		Search method	Features subset	
			Weight	Feature
Filter	Correlation attribute	Ranker	0.3555 0.2989 0.2492 0.2457 0.1798 0.1554 0.1093 0.0955	geo_enabled verified description default_profile account_age_days favorites_count id default_profile_image
	Information gain (entropy)		0.91686 0.91686 0.48263 0.23724 0.17872 0.15575 0.14144 0.10167	created_at screen_name location followers_count friends_count statuses_count favorites_count average_tweets_per_day
	CV attribute evaluation		7.936 7.5674 6.3993 6.3289 5.0376 4.4785 2.9434 2.9424	favorites_count verified statuses_count average_tweets_per_day lang id location geo_enabled
Wrapper	Wrapper subset evaluation	Random search	–	default_profile default_profile_image description followers_count location profile_bg_image_url screen_name verified

cross-validation value with respect to class. To evaluate the wrapper subset, the attribute was calculated using a learning scheme to estimate the accuracy of the learning model for a set of attributes.

Therefore, we used the size of the subset features generated by the wrapper subset evaluation to determine the maximum size of the subset for all filter FSs used (for our case, $k = 8$). We chose this size, because the wrapper method only reports the best subset performance, whereas other methods report all features in ranking order. Hence, we took the eight features that ranked first in correlation attributes, information gain, and cross-validation evaluation as shown in Table 2. These filter methods were chosen to measure correlations between the feature and class, the information weight of each feature, and the worthiness of a feature with regard to class attribute. The selected feature subset for each FS technique is presented in Table 2, where the feature method and the weight of the selected features are listed. The first eight ranked features of the three filter methods out of the 17 features were selected.

Different machine-learning classifiers were then used to evaluate the performance of these eight features. We used RF, NB, SVM, and NN and executed the learning model for a number of times ($k > 1$) to observe the changes in performance. Therefore, in each round we decreased the number of features by removing the least weighted feature k times, as shown in Table 3.

It is noteworthy that the total selected features from all four methods were 17. From the eight ranked features for each method, at least three were selected using the three FS methods. These features included *Favorites_count*, *location*, and *verification*. Similarly, nine features were selected using at least two methods, as shown in Table 4. Out of the 17 features, five were selected using only one FS method. These features were *account_age_days*, *created_at*, *friend_count*, *lang*, and *profile_bg_image_url*.

Table 3 Feature subsets performance for bot class

Feature selection method	Classifier	# Of Features	Bot		AUC	Bot PRC
			P	R		
Correlation attributes	RF	8	0.801	0.77	0.898	0.822
		7	0.8	0.768	0.9	0.822
		6	**0.807**	**0.768**	**0.903**	**0.822**
		5	0.682	0.694	0.824	0.691
	NB	8	0.591	0.636	0.769	0.572
		7	0.59	0.636	0.769	0.569
		6	**0.606**	**0.652**	**0.787**	**0.633**
		5	0.595	0.633	0.776	0.621
		4	0.589	0.604	0.78	0.628
		3	0.757	0.266	0.766	0.589
	SVM	8	1	0.375	0.688	0.583
		7	1	0.375	0.688	0.583
		6	**1**	**0.39**	**0.695**	**0.592**
		5	0.662	0.378	0.641	0.457
	NN	8	**0.658**	0.581	0.826	0.679
		7	**0.658**	0.557	**0.827**	**0.688**
		6	0.631	**0.652**	0.825	0.672
		5	0.65	0.514	0.792	0.64
Information gain	RF	8	1	0.37	0.908	0.854
		7	**1**	**0.37**	**0.913**	**0.862**
		6	1	0.37	0.902	0.846
		5	1	0.37	0.866	0.791
		4	1	0.37	0.812	0.724
	NB	8	**0.688**	0.277	0.798	0.586
		7	0.65	**0.551**	**0.812**	**0.637**
	SVM	8	**0.872**	**0.646**	**0.799**	**0.68**
		7	0.872	0.646	0.799	0.68
		6	0.77	0.638	0.772	0.612
	NN	8	0.952	0.383	0.861	0.791
		7	0.949	**0.388**	**0.865**	**0.799**
		6	0.941	0.383	0.858	0.789
		5	**0.956**	0.383	0.842	0.774
		4	0.956	0.385	0.845	0.776
		3	0.948	0.385	0.837	0.771
		2	1	0.37	0.805	0.729
CV attribute evaluation	RF	8	0.873	0.644	0.923	0.876
		7	**0.882**	0.649	0.93	0.885
		6	0.838	**0.833**	**0.943**	**0.91**
		5	0.825	0.812	0.94	0.899
		4	0.841	0.812	0.938	0.9
		3	0.844	0.802	0.932	0.892
		2	0.674	0.676	0.822	0.68
	NB	8	**0.577**	**0.127**	**0.794**	**0.564**
		7	0.504	0.093	0.753	0.517
	SVM	8	**1**	**0.395**	**0.697**	**0.595**
		7	1	0.395	0.697	0.595
		6	1	0.367	0.684	0.577
	NN	8	**0.708**	0.733	0.852	0.726
		7	0.652	**0.741**	**0.854**	**0.717**
Wrapper subset evaluation	RF	8	**0.992**	0.377	**0.914**	**0.865**
	NB		0.684	0.536	0.829	0.485
	SVM		0.648	**0.676**	0.747	0.546
	NN		0.879	0.514	0.89	0.831

Bold values represent best performance in the category

NB naïve Bayes, *NN* neural network, *RF* random forest, *SVM* support vector machine

Table 4 Features selected by all FS methods

	Favorites_count	location	verified	Default_profile	Geo_enabled	descrip-tion	Follow-ers_count	Screen_name	average_tweets_per_day	Default_profile_image	statuses_count	id	account_age_days	cre-ated_at	friends_count	lang	pro-file_bg_image_url
Correlation attribute	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓				
Information Gain	✓	✓	✓				✓	✓	✓		✓			✓	✓		
CV Attribute Evaluation	✓	✓	✓		✓				✓		✓	✓				✓	
Wrapper Subset Evaluation		✓	✓	✓		✓	✓	✓		✓							✓

4 Results

As mentioned, our goal was to be able to identify more bot accounts using a subset of features to improve the prediction rate and other evaluation measures, such as recall and precision. To evaluate the performance, we used RF and other classifiers, because they performed well in previous studies (Alothali et al. 2018). We trained our data with all features using a cross-validation approach. The results showed that the learning model using RF achieved an accuracy score of 79.2%. However, the model was unable to identify more than half of the bots' classes with a recall score of 37%, compared with identifying nearly all human accounts with a recall of 99.8% and average AUC score of 93.3%.

The learning model using NNs was similarly unable to identify bot classes. The result reported ROC area curve of 80–90% over all studied subsets of features with a recall score for bot class between 38.8 and 73.3% and a precision score of 70.8–95.6%.

Consequently, we experimented with the performance of the feature subsets mentioned in Sect. 3.2 separately to observe the learning-model performance using machine-learning algorithms, as shown in Table 3. It is noteworthy that the purpose was to find the optimal features that could help our learning model perform well to identify bot classes without affecting human classes. In particular, we dealt with imbalanced data having a ratio of bots to human of 33:67.

In this section, a report on the experimental results is presented. We address the performance of FS methods using RF, NB, NB, and NN algorithms.

4.1 Correlation attribute

The point of using the correlation attribute was to measure the relationship between a given feature and its class (Hall 2000). In this method, the best performance of different subsets for all chosen machine-learning algorithms showed that RF could maintain good scores for both recall and precision for both classes, bot and human. It achieved an ROC area score between 82.4 and 90.3%, as shown in Table 3. The results illustrate that the learning-model performance for RF using the subset with the first six ranked features achieved the best performance compared with other subsets having an average recall score of 86.2 and precision score of 86.1%. It is noteworthy that recall for bots using eight features slightly improved using RF with a score of 77% compared with 76.8% with six features. In terms of precision for bot class, the model performed better with six features, scoring 80.7%.

The NN classifier performed well with correlation attribute subsets after RF compared with NB or SVM classifiers. It achieved an ROC area scores between 79.2 and 82.7%

compared with NB and SVM. For NB, the AUC scores were between 76.6 and 78.7%, and 60.6% precision and 65.2% recall scores were attained for the six features. The performance of SVM was poor with ROC area scores between 64.1 and 69.5% and a recall score for bot class of 39% compared with 100% for the human class for a subset of six features.

The performance measure scores for correlation attribute subsets show that models generally worked well with subsets having six features for RF, NB, and SVM compared with other subsets, as shown in Table 3. For the NN, the overall good scores had a subset of seven features. However, the scores demonstrated that the learning models did not detect bot class.

4.2 Information gain

Information gain measures each feature's weight (Karegowda et al., 2010). Using this method to obtain predictive features showed that the learning model performed well when using the first seven ranked features compared to the correlation attribute method. The RF model displayed 91.3% ROC with an average recall score of 79.1% and precision of 84.1%. For NB, the best scores were with seven feature subsets with an ROC area of 81.2%, average recall of 75.3%, and average precision of 74.5%. The performance of SVM was the same for both subsets of eight and seven features with an ROC area of 79.9% and an average score of recall of 85.1% and precision of 85.3%. For the NN, the subset with seven features reported an ROC score of 86.5%. However, the recall of the bot class was approximately 38.3% for all subsets.

The RF model demonstrated good ROC, as shown in Fig. 4, for the first seven ranked feature subsets with 91.3% compared to other classifiers, but with an average F1 score of 75.7%. As the data were imbalanced, the F1 score for bot was 54.1%, which implied that the model was unable to detect around half of the bots class compared to the human class with an F1 score of 86.5%.

4.3 CV Attribute evaluation

This FS method performed best compared with the other FS methods using an RF classifier. The average accuracy was 86.9% for the RF model for different subsets with ROC area scores between 92.3 and 94.3% (Fig. 5). The best performance was achieved with a subset of the first six features ranked using an RF that showed precision scores of 83.8 and 83.3% recall for the bot class. The RF model result also maintained good performance for the human class (Fig. 6). The precision score was 91.7% with a recall score of 92%. Therefore, the overall best learning model using RF was the one that used a subset of six features, because it specified

the best recall score for the bot class compared with other feature subsets.

On the other hand, the performance for NB showed an ROC area score of 79.4% for a subset with eight features. For the SVM classifier, the highest performance was for the subset with seven and eight features with an ROC area of 69.7%. The NN classifier achieved the second highest score of recall for bot class with 74.1% with a subset of seven features and a score of ROC area of 85.4%. The precision for the bot class was 62.3%. However, the overall performance of the NN that maintains good precision and recall for the bot class was with a subset of eight features, as shown in Table 3.

4.4 Wrapper subset evaluation

The wrapper subset evaluation method was used because it is based on the evaluation of performance of different feature subsets (Maldonado and Weber 2009). The best subset performance was reported after the performance was evaluated. In our case, the reported features were for the subset that used eight features. The performance for RF showed a score of 37% recall and 99.2% for precision for the bot class with an AUC score of 91.4%. For NB, the highest recall score for bot class was 53.6% for eight features, with an AUC of 82.9%. For SVM, the results reported that the performance for a subset of eight features had a 74.7% AUC score, as shown in Table 3. The performance of the NN with eight features demonstrated the highest AUC score of 89.0% for NN performance during the entire experiment. However, the recall score was approximately 51.4% with a precision of 87.9 for the bot class.

5 Discussion

The experiments presented in the previous sections examined features of Twitter accounts to enhance the detection of social bots' accounts. We evaluated different subsets of features generated by different FS methods as shown in Table 3. Based on the findings, most of the classifiers were able to identify human accounts but not bot accounts. This was because of the misclassification of the bot class. However, we did enhance the learning model using six predictive features and the CV attribute evaluation method with a RF classifier as shown in Fig. 7. These features were *favorites _count*, *verified*, *statuses_count*, *average_tweets_per_day*, *lang*, and *id* (Table 2).

These predictive features improved the sensitivity score of the bot class from 64.9 to 83.3% while maintaining a 92% recall for the human class, as shown in Table 3. These good performance metrics indicate that these features can increase bot detection without impacting the human class, despite the bot class being the minority. Moreover, these

Fig. 4 Performance of area under the receiver operating characteristic curve (AUC) for information gain subsets with random forest (RF) classifier

ROC Curve - Information Gain with RF

AUC = 0.908 (8 Features)
AUC = 0.913 (7 Features)
AUC = 0.902 (6 Features)
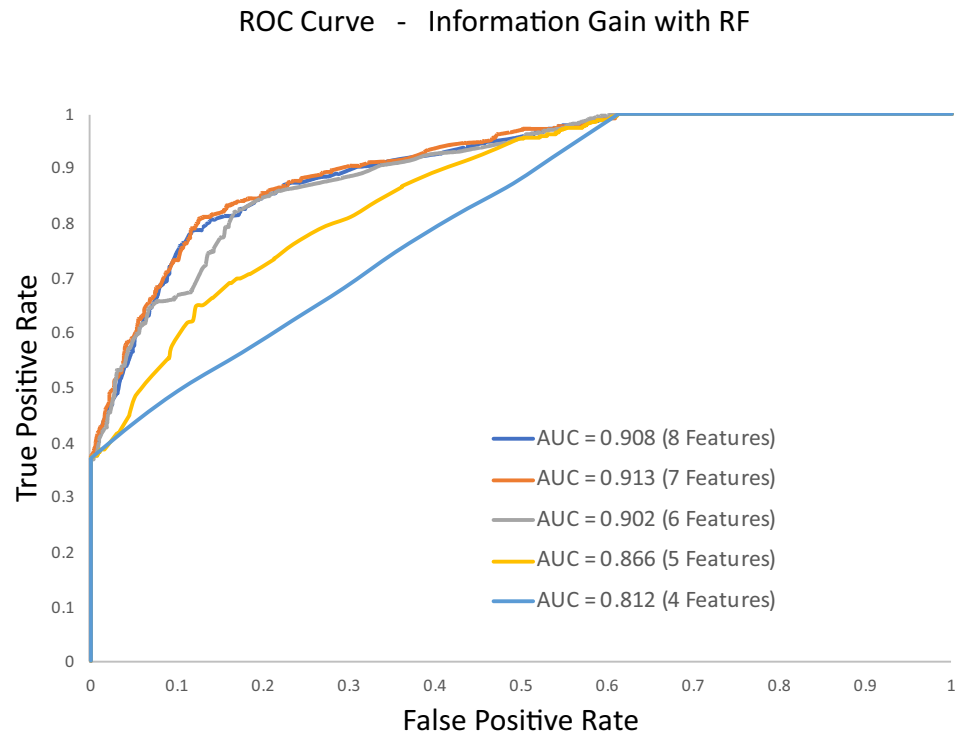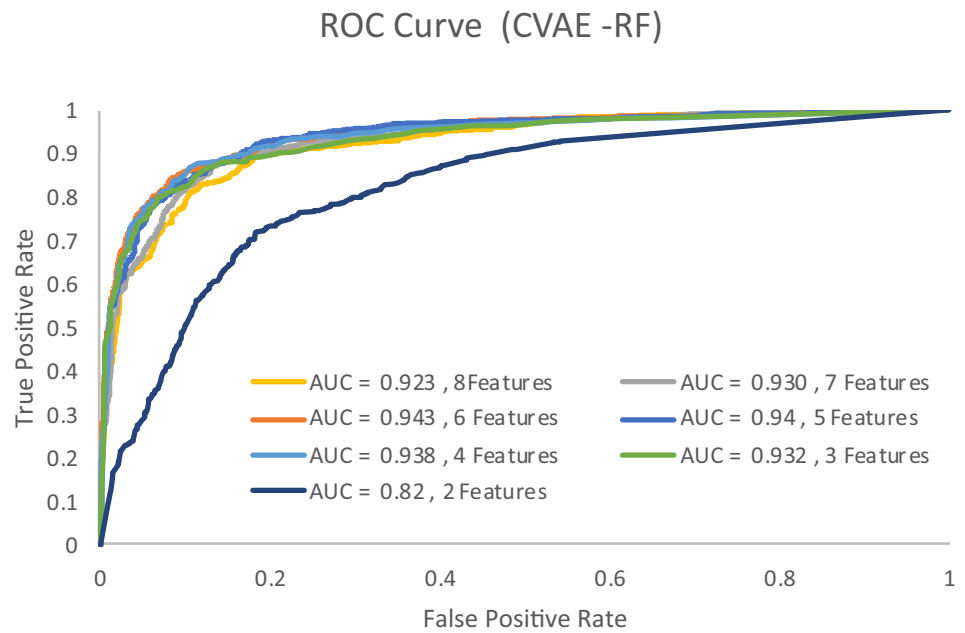AUC = 0.866 (5 Features)
AUC = 0.812 (4 Features)

Fig. 5 Performance of area under the receiver operating characteristic curve (AUC) for cross-validation attribute evaluation (CVAE) subsets with random forest (RF) classifier

ROC Curve (CVAE -RF)

AUC = 0.923 , 8 Features   AUC = 0.930 , 7 Features
AUC = 0.943 , 6 Features   AUC = 0.94 , 5 Features
AUC = 0.938 , 4 Features   AUC = 0.932 , 3 Features
AUC = 0.82 , 2 Features

features represent some of the common profile features for detecting bot accounts in the literature (Alothali et al. 2018). For example, using only the first four of these features (i.e., *favorites_count*, *verified*, *statuses_count*, and *average_tweets_per_day*) achieved good performance metrics for bots (84.1% precision, 81.2% recall) and humans (90.8% precision, 92.4% recall) with an overall ROC score of 93.8%. However, all six features performed best overall.

These features indicate the number of characteristics used to identify social bot accounts. The *favorites_count* was once an indication for human accounts, as human users tend to receive more likes (Gilani et al. 2017). Therefore, the lower count of favorite or its absence might suggest a bot account. Bots typically generate similar content, and they are remotely guided via botnets. For example, the *verified* feature used to be an identification for a human or business
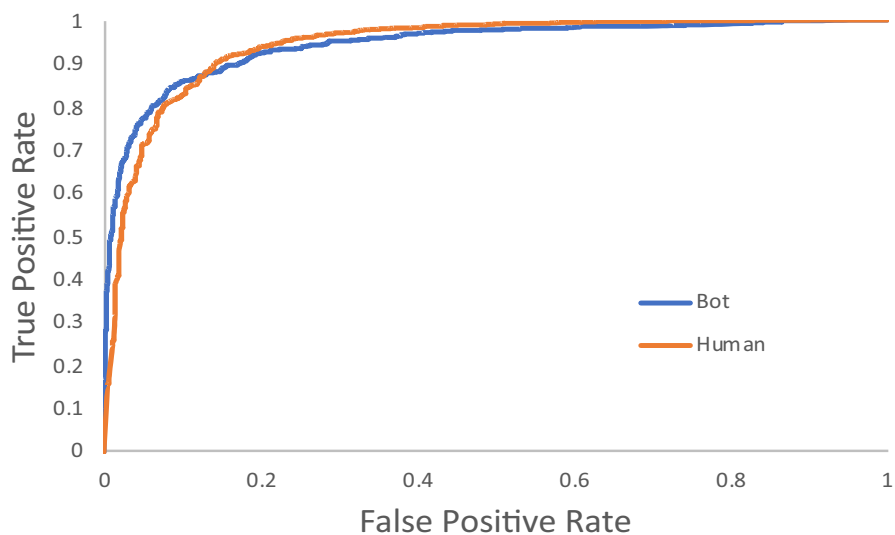
Fig. 6 Performance of cross-validation attribute evaluation (CVAE) subset with six features with random forest (RF) classifier for both human and bot class: **a** precision recall for bot and human; **b** receiver operating characteristic (ROC) curve for both human and bot for six features generated by CVAE and RF classifier

(A)
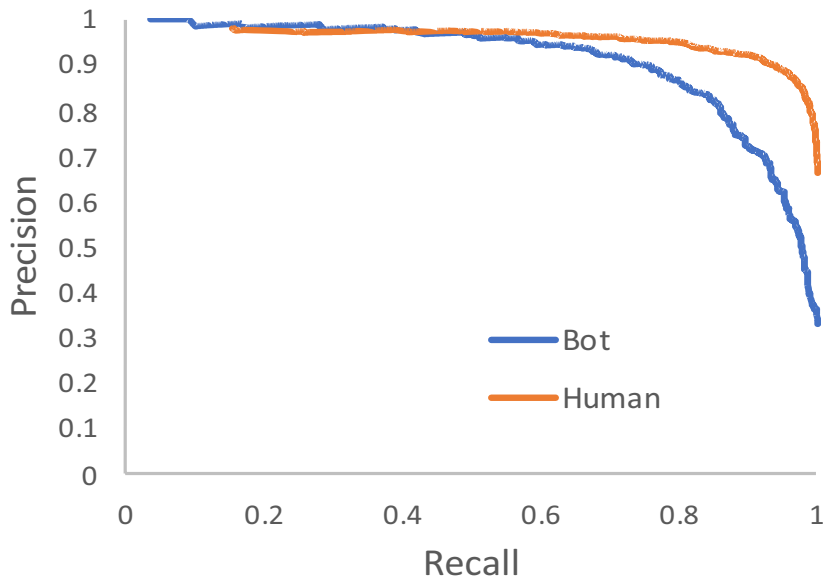
(B)

Fig. 7 Comparison of receiver operating characteristic (ROC) curve between best performances of classifiers for different features subsets. Random forest (RF) achieved best with six features subset via cross-validation attribute evaluation (CVAE). Neural network (NN) achieved best with wrapper subset evaluation (WSE) with eight features. Naïve Bayes (NB) achieved best with WSE and eight features. Support vector machine (SVM) achieved best with information gain (InfoGain) with eight and seven features, respectively
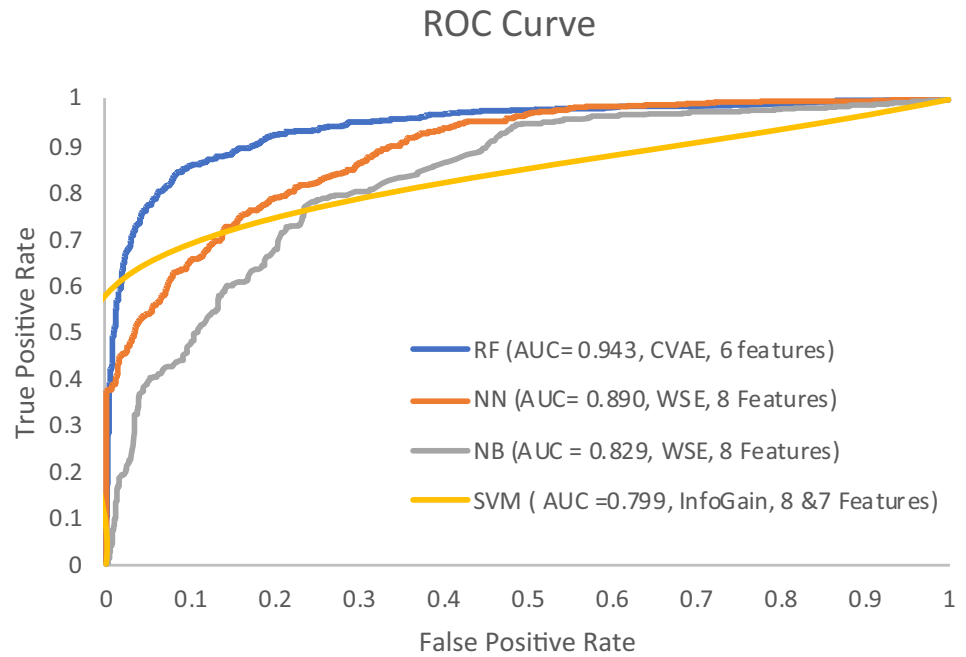
Table 5 Our approach compared to benchmark-related work

Ref#	Algorithm	Feature selection method	# of features	Accuracy	Precision	Recall	F1 score	AUC
Shukla et al. (2021)	Random forest	Univariate	10	0.84	0.82	0.67	0.74	0.9
		PCA	10	0.81	0.74	0.67	0.7	0.88
		Extra-tree	8	0.85	0.8	0.75	0.77	0.86
Our		CVAE	6	**0.86**	**0.85**	**0.84**	**0.84**	**0.91**

Bold values represent best performance in the category

AUC area under the receiver operating characteristic curve, *CVAE* cross-validation attribute evaluation, *PCA* principal component analysis

account. The *status_count* feature, which designates tweets, retweets, replies, and mentions, can be used to identify bot account, because humans tend to post less content (Gilani et al. 2019). The *average_tweets_per_day* feature, which includes tweets and retweets, is another, because bots are typically aggressive with retweeting content (Gilani et al. 2019).

Finally, the *lang* and *id* features should be further investigated, especially because bots can be targeted in any language. For *id*, it is interesting to note that accounts used to be sequential. However, Twitter changed them to be generated based on time (Twitter.com 2020c). This can help detect botnets, because they tend to be newer accounts.

To validate our approach, as shown in Table 5, we used the whole dataset and compared the performance measures' AUC, precision, recall, and F1 score for the CVAE features subset with six features using RF with a recent benchmark study that used the same dataset and classifier (Shukla et al., 2021). WoE encoding was used to recognize unique

values in nominal feature attributes. This framework has three stages, after which performance is evaluated. When the encoding feature is completed, FS is followed. Three different FS methods are used: PCA, univariate, and extra-tree models for a number of machine-learning algorithms, including RF. For this stage, the experimental results show that the extra-tree model using eight encoded features out of 12 performed better than the PCA and univariate FS methods. Extra-tree achieved a score of 0.91 AUC, but the recall score was 0.75, and the F1 score was 0.77. This highlights that the learning model is still unable to identify the social bot accounts compared with our results, as shown in Table 5. However, the framework using blending ensemble techniques with WoE encoding showed a better improvement in AUC score, achieving 0.93 and 0.81 for F1 score for the same features. However, the recall score improved slightly by 0.01%.

6 Conclusion

Detecting social bots on Twitter is a challenging task, especially with sophisticated design and behaviors that mimic human users. Therefore, identifying optimal features to enhance the detection of social bots is important. In this study, we explored four feature selection methods using a public dataset. Additionally, we investigated Twitter features of profile metadata to improve bot detection. Based on the results, the study presented a hybrid technique to identify the predictive features of profile metadata on Twitter. We used filter and wrapper methods to identify the optimal feature subset. We found that the CVAE method with a subset of six features performed best among the subsets. The RF classifier performed better, followed by the NN in general when identifying bot accounts. NB and SVM performed poorly in our experiments. Our results using RF showed 89% accuracy with a 94.3% ROC with a precision score of 83.8 and an 83.3% recall for bot class. We also achieved a good ROC score (93.8%) using four features from the CVAE method: *favorites _count*, *verified*, *statuses_count*, and *average_tweets_per_day*. Using the first three of these features can still achieve good performance metrics for bots (84.4% precision, 80.2% recall) and a ROC of 93.2%.

In future work, we will investigate the features of tweets to identify social bots. Those from both profiles and tweets will then be used to create an online detection system for social bots on Twitter. We intend to use deep-learning techniques to detect social bots in the future.

Declarations

References

Abokhodair, N, Daisy Y, McDonald DW (2015) Dissecting a social botnet. In: Proceedings of the 18th ACM conference on computer supported cooperative work and social computing, New York, NY, USA: ACM, 839–51. https://doi.org/10.1145/2675133.2675208

Alothali E, Nazar Z, Mohamed EA, Hany A (2018) Detecting social bots on Twitter: a literature review. In: 2018 international conference on innovations in information technology (IIT), IEEE, 175–80. https://doi.org/10.1109/INNOVATIONS.2018.8605995

H Ariyaluran A Riyaz N Fariza G Abdullah IAT Hashem A Ejaz I Muhammad 2019 Real-time big data processing for anomaly detection: a survey Int J Inf Manag 45 289 307 https://doi.org/10.1016/j.ijinfomgt.2018.08.006

DM Beskow KM Carley 2019 Its all in a name: detecting and labeling bots by their name Comput Math Organ Theory 25 1 24 35 https://doi.org/10.1007/s10588-018-09290-1

Botometer (2020) Datasets 2020. https://botometer.osome.iu.edu/bot-repository/datasets.html

Cai C, Linjing L, Daniel Z (2017) Behavior enhanced deep bot detection in social media. In: 2017 IEEE International conference on intelligence and security informatics (ISI), IEEE, 128–30. https://doi.org/10.1109/ISI.2017.8004887.

Cresci, S, Di Pietro R, Marinella P, Angelo S, Maurizio T (2017) The Paradigm-shift of social spambots. In: Proceedings of the 26th international conference on world wide web companion—WWW '17 companion, New York, New York, USA: ACM Press, 963–72. https://doi.org/10.1145/3041021.3055135

Cresci S, Marinella P, Angelo S, Stefano T (2019) Better safe than sorry. In: Proceedings of the 10th ACM conference on web science—WebSci. New York, New York, USA: ACM Press, 19:47–56. https://doi.org/10.1145/3292522.3326030

S Dadkhah S Farzaneh MM Yadollahi Z Xichen AG Ali 2021 A real-time hostile activities analyses and detection system Appl Soft Comput 104 107175 https://doi.org/10.1016/j.asoc.2021.107175

Devi SG, Sabrigiriraj M (2018) Feature selection, online feature selection techniques for big data classification: a review. In: 2018 international conference on current trends towards converging technologies (ICCTCT), IEEE, 1–9. https://doi.org/10.1109/ICCTCT.2018.8550928.

E Ferrara O Varol C Davis F Menczer A Flammini 2016 The rise of social bots Commun ACM 59 7 96 104 https://doi.org/10.1145/2818717

Z Gilani R Farahbakhsh G Tyson J Crowcroft 2019 A large-scale behavioural analysis of bots and humans on Twitter ACM Trans Web 13 1 1 23 https://doi.org/10.1145/3298789

Gilani Z, Liang W, Jon C, Mario A, Reza F (2016) Stweeler: a framework for Twitter bot analysis. In: Proceedings of the 25th international conference companion on world wide web—WWW '16 companion, New York, New York, USA: ACM Press, 37–38. https://doi.org/10.1145/2872518.2889360

Gilani, Z, Reza F, Gareth T, Liang W, Jon C (2017) Of bots and humans (on Twitter). In: Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017, New York, NY, USA: ACM, 349–54. https://doi.org/10.1145/3110025.3110090

Grier C, Kurt T, Vern P, Michael Z (2010) @spam. In: Proceedings of the 17th ACM conference on computer and communications security—CCS '10, 27. New York, New York, USA: ACM Press. https://doi.org/10.1145/1866307.1866311

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(null):1157–1182

Hall, MA (2000) Correlation-based feature selection of discrete and numeric class machine learning

AG Karegowda AS Manjunath MA Jayaram 2010 Comparative study of attribute selection using gain ratio and correlation based feature selection Int J Inf Technol Knowl Manag 2 2 271 277

A Khalil H Hassan N Al-Qirim 2017 Detecting fake followers in Twitter a machine learning approach Int J Mach Learn Comput 7 6 198 202 https://doi.org/10.18178/ijmlc.2017.7.6.646

Khalil H, Muhammad USK, Mazhar A (2020) Feature selection for unsupervised bot detection. In: 2020 3rd international conference on computing, mathematics and engineering technologies (ICoMET), abs/1703.0:1–7. IEEE. https://doi.org/10.1109/iCoMET48670.2020.9074131.

Kohavi R, George HJ (1997) Wrappers for feature subset selection. Artif Intel 97(1):273–324. https://doi.org/10.1016/S0004-3702(97)00043-X

Kondor D, Istvan C, Laszlo D, Janos S, Norbert B, Tamas H, Tamas S, Zsofia K, Gabor V (2013) Using robust PCA to estimate regional characteristics of language use from geo-tagged Twitter messages. In: 2013 IEEE 4th international conference on cognitive info-communications (CogInfoCom), IEEE, 393–98. https://doi.org/10.1109/CogInfoCom.2013.6719277

S Kudugunta E Ferrara 2018 Deep neural networks for bot detection Inf Sci 467 312 322 https://doi.org/10.1016/j.ins.2018.08.019

J Li H Liu 2017 Challenges of feature selection for big data analytics IEEE Intell Syst 32 2 9 15 https://doi.org/10.1109/MIS.2017.38

S Maldonado R Weber 2009 A wrapper method for feature selection using support vector machines Inf Sci 179 13 2208 2217 https://doi.org/10.1016/j.ins.2009.02.014

Martín-Gutiérrez D (2020) Twitter bots accounts. Kaggle.Com. 2020. https://doi.org/10.34740/KAGGLE/DSV/1623389

D Martin-Gutierrez G Hernandez-Penaloza AB Hernandez A Lozano-Diez F Alvarez 2021 A deep learning approach for robust detection of bots in twitter using transformers IEEE Access 9 54591 54601 https://doi.org/10.1109/ACCESS.2021.3068659

Minnich A, Nikan C, Danai K, Abdullah M (2017) "BotWalk." In: Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017, New York, NY, USA: ACM, 467–74. https://doi.org/10.1145/3110025.3110163

Morchid M, Richard D, Pierre-Michel B, Georges L, Juan-Manuel T-M (2014) Feature selection using principal component analysis for massive retweet detection. Pattern Recogn Lett 49:33–39. https://doi.org/10.1016/j.patrec.2014.05.020

Ostrowski DA (2014) Feature selection for Twitter classification. In: 2014 IEEE international conference on semantic computing, IEEE, 267–72. https://doi.org/10.1109/ICSC.2014.50

S Rathore PK Sharma V Loia Y-S Jeong JH Park 2017 Social network security: issues, challenges, threats, and solutions Inf Sci 421 43 69 https://doi.org/10.1016/j.ins.2017.08.063

Shafahi M, Leon K, Hamideh A (2016) Phishing through social bots on Twitter. In: 2016 IEEE international conference on big data (big data), IEEE, 3703–12. https://doi.org/10.1109/BigData.2016.7841038

Shah FP, Vibha P (2016) A review on feature selection and feature extraction for text classification. In: 2016 international conference on wireless communications, signal processing and networking (WiSPNET), IEEE, 2264–2268. https://doi.org/10.1109/WiSPNET.2016.7566545

Shukla H, Nakshatra J, Balaji P (2021) Enhanced Twitter bot detection using ensemble machine learning. In: 2021 6th international conference on inventive computation technologies (ICICT), IEEE, 930–36. https://doi.org/10.1109/ICICT50816.2021.9358734

Stringhini G, Christopher K, Giovanni V (2010) Detecting spammers on social networks. In: Proceedings of the 26th annual computer security applications conference on—ACSAC '10, New York, New York, USA: ACM Press, 1. https://doi.org/10.1145/1920261.1920263

VS Subrahmanian A Azaria S Durst V Kagan A Galstyan K Lerman L Zhu E Ferrara A Flammini F Menczer 2016 The DARPA Twitter bot challenge Computer 49 6 38 46 https://doi.org/10.1109/MC.2016.183

Tang J, Salem A, Huan L (2014) Feature selection for classification: a review. In: Aggarwal CC (ed). Data classification: algorithms and applications, Chapman and Hall/CRC. https://doi.org/10.1201/b17320

Twitter.com (2020a) Automation rules 2020. https://help.twitter.com/en/rules-and-policies/twitter-automation

Twitter.com (2020b) Data dictionary. https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/overview/user-object

Twitter.com (2020c) Twitter IDs. Twitter Inc 2020. https://developer.twitter.com/en/docs/twitter-ids

Varol O, Emilio F, Davis CA, Filippo M, Alessandro F (2017) Human-bot interactions: detection, estimation, and characterization. CoRR abs/1703.0. http://arxiv.org/abs/1703.03107

Visalakshi S, Radha V (2014) A literature review of feature selection techniques and applications: review of feature selection in data mining. In: 2014 IEEE international conference on computational intelligence and computing research,. IEEE, 1–6. https://doi.org/10.1109/ICCIC.2014.7238499

Wald R, Khoshgoftaar TM, Napolitano A (2013a) Should the same learners be used both within wrapper feature selection and for building classification models? In: 2013 IEEE 25th international conference on tools with artificial intelligence, IEEE, 439–45. https://doi.org/10.1109/ICTAI.2013.72

Wald R, Taghi K, Amri N (2013b) Filter- and wrapper-based feature selection for predicting user interaction with Twitter Bots. In: 2013 IEEE 14th international conference on information reuse and integration (IRI), IEEE, 416–23. https://doi.org/10.1109/IRI.2013.6642501

Wang AH (2010) Detecting spam bots in online social networking sites: a machine learning approach. In: DBSec, Springer, 10:335–42. https://doi.org/10.1007/978-3-642-13739-6_25

K-C Yang O Varol P-M Hui F Menczer 2020 Scalable and generalizable social bot detection through data selection Proc AAAI Conf Artif Intell 34 01 1096 1103 https://doi.org/10.1609/aaai.v34i01.5460

Zhang, X, Shaoping Z, Wenxin L (2012) Detecting spam and promoting campaigns in the twitter social network. In: 2012 IEEE 12th international conference on data mining, IEEE, 1194–99. https://doi.org/10.1109/ICDM.2012.28