Preprocessing Framework for Twitter Bot Detection

Mücahit Kantepe, Murat Can Ganiz
Department of Computer Engineering
Marmara University
Istanbul, Turkey
mucahitkantepe@marun.edu.tr | murat.ganiz@marmara.edu.tr

Abstract— **One of the important problems in social media platforms like Twitter is the large number of social bots or sybil accounts which are controlled by automated agents, generally used for malicious activities. These include directing more visitors to certain websites which can be considered as spam, influence a community on a specific topic, spread misinformation, recruit people to illegal organizations, manipulating people for stock market actions, and blackmailing people to spread their private information by the power of these accounts. Consequently, social bot detection is of great importance to keep people safe from these harmful effects. In this study, we approach the social bot detection on Twitter as a supervised classification problem and use machine learning algorithms after extensive data preprocessing and feature extraction operations. Large number of features are extracted by analysis of Twitter user accounts for posted tweets, profile information and temporal behaviors. In order to obtain labeled data, we use accounts that are suspended by Twitter with the assumption that majority of these are social bot accounts. Our results demonstrate that our framework can distinguish between bot and normal accounts with reasonable accuracy.**

Keywords-component; sybil account; social bot; bot detection; feature engineering; model construction; machine learning.

I. INTRODUCTION

Today, social media platforms like Twitter have many accounts that are controlled by automated agents called bot or sybil accounts [1]. Mostly, people aim to have more visitors to their websites, influence community on a specific topic, recruit people to their organizations that might be an illegal organization, manipulating people for stock market actions, propagate some fake news and blackmailing people to spread their private information by the power of these accounts [2]. As a result, social bot detection framework becomes very crucial to keep people safe from sybil accounts [6] [7].

When these bot accounts are analyzed, it can be seen that there are various types; some of them very primitive and some of them are very complex that they are hard to diagnose even by humans. In order to avoid detection, they mimic human accounts, develop strategies to friend or follow human accounts and support each other as a large network to gather trust [3]. Additionally, a large group or network of these accounts can act collaboratively to change trending topics of Twitter for malicious purposes.

978-1-5386-0930-9/17/$31.00 ©

The sheer number of these bot accounts and their increasing complexity bestows a challenge for the manual detection of these accounts. The new user sign-up process seems and ideal place to detect and prevent these sybil and bot accounts. However, sign-up process is actually the Achilles heel of social networking sites in this context since they are under heavy business pressure to extend their user bases. Therefore, they can not employ complicated techniques to detect bots as they can discourage humans to sign-up. Furthermore, a user account registered by a human can later be used by a bot.

We approach the social bot detection on Twitter as a supervised classification problem and use machine learning algorithms after extensive data preprocessing and feature extraction operations. Large number of features are extracted by analyzing tweets of Twitter user accounts, profile information and temporal behaviors such as changes in profile and tweet frequencies. In order to obtain labelled data, we use accounts that are suspended by Twitter with the assumption that majority of these are spam or bot accounts[1].

In order to build a machine learning models and conduct experiments we collected data between February 2017 and June 2017 from Twitter using Twitter Streaming API. We focus on the users which tweets with trending topics of Turkey. Once the users who tweets with trending topics are identified, we fetch their tweets (up to 3200 tweets due to Twitter REST API limitation) which are written in Turkish and store them in a NoSQL database.

On this data, our preprocessing framework extracts 62 features for each user. These features are inspired from the publications resulted from DARPA's Twitter Bot Challenge [4]. These features are than fed into a machine learning classifier. Our results demonstrate that our framework can distinguish between bot and normal accounts with reasonable accuracy.

The rest of the paper is organized as follows. The background and related work are covered in Section II. We present our approach for detecting bot accounts in Section III. We follow this by summarizing our experimental results in Section IV. Section IV includes a conclusion and discussion of the future work.

[1] https://support.twitter.com/articles/15790

II. BACKGROUND AND RELATED WORK

There are various types of bot accounts exist in Twitter. Some of them are very primitive and some of them have very complex structures that mimic human behaviors. It becomes harder to discriminate synthetic behaviors from human ones. The task of detecting these bot accounts is a popular research topic especially 2016 US presidential campaign. One of the most interesting of these research studies was a contest named as "*The DARPA Twitter Bot Challenge"* [4]. Six teams are competed in this contest and a team called *Sentimetrix* got first place. There were six teams and all teams approached this task differently and employed very useful techniques. We take advantage of these different approaches [1] [4].

Analysis of bot accounts show many behaviors that are different from human behaviors. These behaviors can be categorized as tweet syntax, tweet semantics, temporal behavior, user profile information and user network [5]. Tweet syntax includes information about contents of the tweets like hashtags, mentions, URL's, special characters, as well as statistics such as number of retweets and location information. Basically, these features are to detect if tweet content is generated by an automated agent.

Tweet semantics includes sentiment analysis of user's tweets, average of positive sentiment strength, and average of negative sentiment strength. The rationale behind this is the observation that bot accounts post strong positive comments about their promoted opinion or products and strong negative comments about their opposite opinions or products. Number of different languages is another feature that is considered because these accounts are automated, they can easily tweet in many different languages by reading these tweets or messages from a database.

Temporal behaviors include sentiment entropy and sentiment variance due to indication of inconsistency. The bot accounts might post flip-flopped tweets consecutively. On the other hand humans do not change their mind very frequently like this. Duration of longest session without sleep break (4 hours) indicates that the account owner did not slept for long time, this increases the probability of the account to be bot.

User profile features include information about user's profile image, description text, number of tweets, number of retweets, number of listed tweets, number of favorites, number of replies, number of mentions, location information, number of follower and following, ratio between follower and following, number of different sources used like mobile or desktop clients. Tweeting entropy is also important due to bot accounts tweet with closer amount of time between tweets, similarity of tweets gives information about tweets of bot accounts created by a generator. Lastly, number of similar tweets feature [5].

Periodic features indicate the difference of user profile settings over time. The change in the number of tweets, number of followers, number of following, number of favorites, location information, profile image, screen name, description and profile background settings are used as periodic features.

Most of these features are used in The DARPA Twitter Bot Challenge contest [4]. They are engineered to distinguish or separate bot accounts from the humans [1].

III. APPROACH

A. Data Collection

Our first task was collecting enough and relevant data for our analysis. We collected Twitter data for four months. In order to collect as much bot account as possible, we focus on the Twitter users who tweets with the trending topics. The basic assumption in here is that bot accounts prefer trending topics to provide more visibility to their tweets. We speculate that visibility is important for bot accounts because main characteristics of bot accounts are to reach as many users as possible. Trending topics offers this opportunity since it is read by many users. So the tweets about trending topics are our main source of data. However, we are not using only streamed tweets from Twitter trending topics but we also attempt to fetch the timeline (all tweets) of users. Certain limitations of Twitter public API prevents us to get all of them. As the first limitation, Twitter only provides last 3200 tweets of a user via its API. Second limitation is about rate limit, we had limited amount of API request per hour, and these were two main limitations that slows data collection process. We also record trending topics for each 10 minute windows and which trending topics keyword a user fetched for. Our aim is to analyze bot activity for trending topics.

In order to calculate periodic (temporal) features, we periodically get user profile information again by using Twitter Streaming API. Periodic control time was not deterministic due to our dependency of Twitter Streaming API.

We use Apache Spark's Streaming library to have more tolerance for errors in a data collection scenario that spans for months. Apache Spark [2] handles network exceptions by back-pressure propagation.

B. Feature Engineering

On this data, our preprocessing framework extracts 62 features for each user. These features are inspired from the publications resulted from DARPA's Twitter Bot Challenge [4]. These features are than fed into a machine learning classifier.

We can simply divide these features into three main categories as follows: user profile based features, tweet based features and periodic features. There were 14 user profile based features, 36 tweet based features and 12 periodic features. These features are listed in Table I.

[2] https://spark.apache.org/

TABLE I. LIST OF FEATURES

User Features	Is Verified Is Protected GPS Enabled Has Default Profile Has Default Profile Image	Has Description Description Has URL Extended Profile Number of Listed Number of Friends	Number of Followers Number of Tweets Number of Withheld Countries Days Passed from Last Tweet
Tweet Features	Average Sentiment Max. Same URL Max. Same URL Domain Max. Same Hashtag Longest Session Duration Tweeting Entropy Mentions Number of Similar Tweets Avg. Similarity Between Tweets Follower/Following Rate of Favorited Rate of Mobile Most Active Day	Rate of Media Rate of Hashtag Rate of Mention Average Retweeted Rate of Turkish Most Active Hour Average Replies Average Special Characters Average Short URLs Sentiment Variance Average Tweet Rate of GPS Rate of Reply Rate of URL	Average Number of Terms Average Number of Chars Rate of Truncated Rate of Quote Average Positive Sentiment Strength Average Negative Sentiment Strength Rate of End with Special Character Number of Different Languages Sentiment Entropy
Periodic Features	Screen Name Location Is Protected Profile Image	Background Color Friends Number Favorites Number UTC Offset	Language Number of Tweets GPS Enabled Follower Count

In order to calculate sentiment related features we need to implement a Sentiment Classifier. Basically, a sentiment classifier classifies a text, usually a comment, as positive or negative with a core or probability. We found a dataset that includes reviews and sentiment scores. We tokenized these sentences and trained with several different machine learning algorithms and got best score from Logistic Regression algorithm as 81% accuracy. This sentiment classifier model is used to calculate sentiment related features.

In calculating text similarity feature, we randomly select 100 tweets for each user. In order to compare them we implemented a term frequency – inverse document frequency (TF-IDF) vectorizer. TF-IDF vectorizer takes documents and tokenizes them, calculate weights of term using the TF-IDF formula (Eq. 1) below, and then calculate the similarity between each sentence by using cosine similarity. In Eq. 1, $tf_{i,j}$ is the term frequency; number of occurences of term i in document j, df_i is the number of documents containing term i in the training set, and N is the total number of documents. Formula for cosine similarity is also given in Eq. 2.

$$\omega_{i,j} = tf_{i,j} \times log(N/df_i) \tag{1}$$

$$\cos(\theta) = \frac{A.B}{\| A \| . \| B \|} \quad \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2}$$

$$H = -\sum p(x) \log p(x) \tag{3}$$

C. Creating labelled training set

One of the main problems in applying supervised machine learning methods in the real-world problems is obtaining labeled data. Labeling data for creating a training set for supervised algorithms such as classification algorithms usually requires human effort and therefore it is a slow and expensive process. At first, we attempt to label Twitter users as bot or not by manual inspection but it turns out that it is a painstakingly slow process. Deciding if a user is a bot or not requires to read a large textual content and take into account of other non-textual properties of the user.

Based on this experience, we use unsupervised machine learning methods, namely clustering algorithms to reduce the effort spent in this manual process. By using K-Means clustering algorithm we divided users into different number of clusters starting from two. By manual investigations we noticed that bot like users are highly grouped in certain clusters. Although this approach shows promise, we observed that there are still many normal users in these bot clusters, which creates considerable noise.

As third and final method for creating a labeled training set, we decided to benefit from Twitter's own bot detection algorithm. Based on the assumption that most of the user account that are suspended are actually bot accounts, we label suspended accounts as bot, and others as normal. We do this by checking the status of previously collected user accounts. Our window of data collection

spans through 4 months. If a user which is collected in the first 3 months is suspended in the last month, it is labeled as bot. This method is the fastest method for us to label accounts. One drawback of this approach is the future status of the users that are labeled as normal. That is these users aren't suspended in the time frame of our data collection but it is possible that they can be suspended in the future, e.g. they may be sleeping bot accounts, held inactive by their masters for future use. In any case. In any case, our aim is to classify user accounts mostly based on their activity so we claim that these accounts can be considered as normal in the time frame of the experiments. As a result, we label 620 suspended users as bot and 2400 users as normal.

D. Feature Selection

We use 62 features that are inspired from the previous work. Increasing dimensionality in data science makes it harder to analyze data and construct good model. This is called curse of dimensionality. To avoid curse of dimensionality, we use different feature selection techniques. These techniques indicate the relevance between the feature and target class values. We use Information Gain (IG), Mutual Information (MI) and Chi-Square for this process.

TABLE II. FEATURE SELECTION RESULTS

Features	IG Rank	MI Rank	IG Score	MI Score
Max Same Hashtag	1	1	0,38	0,29
Longest Session	2	3	0,35	0,27
Tweeting Entropy	3	4	0,35	0,25
Most Active Day	4	2	0,31	0,29
Follower/Following	5	5	0,30	0,25
Number of Tweets	6	8	0,28	0,20
Most Active Hour	7	7	0,28	0,20
Average Similarity Between Tweets	8	6	0,28	0,21
Rate of Reply	9	9	0,26	0,19
Periodic Protected	10	11	0,24	0,13
Default Profile Image	11	10	0,24	0,15

Information Gain score for any feature is proportional to correlation of target class labels. If it closes to 1, that means the feature is linearly correlated with class attribute. Also Mutual Information score for any feature is proportional to correlation of target class labels. Our experiments with Information Gain and Mutual Information showed us the most important features for classification which are described below:

Maximum Same Hashtag: Amount of maximum hashtag occurrence in all tweets of a user.

Longest Session Duration: Longest tweeting duration without 4 hours break. Calculated by subtracting time information of consecutive tweets.

Tweeting Entropy: Entropy value of tweet times.

Most Active Day: The day which a user mostly tweeted.

Follower/Following: Ratio of follower number over following number.

Number of Tweets: Number of tweets belong to user.

Most Active Hour: The hour which a user mostly tweeted.

Average Similarity of Tweets: Average similarity value between last 100 tweets of user.

Reply per Tweet: Number of replies for each tweet over number of tweets.

Periodic Protected: Number of privacy settings difference.

Default Profile Image: User uses the default profile image.

E. Experiment Setup

We split our data into several different size training set and test set pairs using following percentages 60% - 40%, 70% - 30%, and 90% - 10%.

We use several supervised machine learning algorithms to construct to classification models that best distinguish bot accounts. These algorithms include Logistic Regression (LR), Multinomial Naïve Bayes (MNB), Support Vector Machines (SVM) and an ensemble learning method: Gradient Boosted Trees (GBT).

F. Experiment Results

After we collect a total of 1800 Twitter users and their tweets, we extract 62 features for each of them, and label them as suspended and non-suspended accounts. We got 600 suspended accounts and 1200 non-suspended accounts. We split our data as 70% as training and 30% test. As a result, Logistic Regression algorithm gave us 75% accuracy and 72% F1 score, Multinomial Naïve Bayes Algorithm gave us 78% accuracy and 77% F1 score, SVM gave us 82% accuracy and 75% F1 score and Gradient Boosted Tress gave us the best result as 86% accuracy and 83% F1 score.

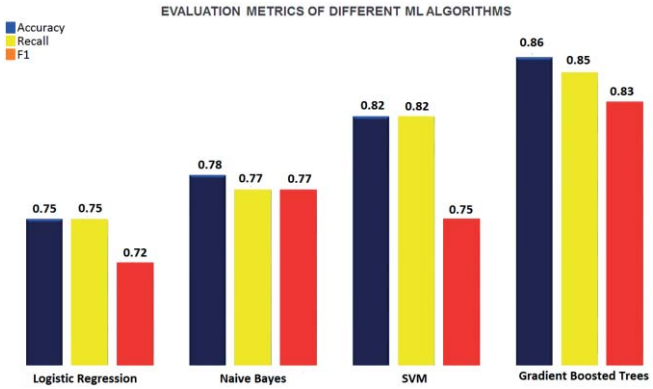

Figure 1. Evaluation metrics of different machine learning algorithms.

IV. CONCLUSION & FUTURE WORK

One of the important problems in social media platforms like Twitter is the large number of social bots or sybil accounts which are controlled by automated agents, generally used for malicious activities. In this study, we approach the social bot detection on Twitter as a supervised classification problem and use machine learning algorithms after extensive data preprocessing and feature extraction operations. Large number of features are extracted by analysis of Twitter user accounts for posted tweets, profile information and temporal behaviors.

Our results demonstrate that our framework can distinguish between bot and normal accounts with reasonable performance. We experiment with different machine learning algorithms with different sized train and test data and measure performance of the classification models using evaluation metrics such as accuracy, precision, recall and F1 score. We achieve 82% F1-score and %86 accuracy using Gradient Boosted Trees ensemble learning algorithm.

In the future, our classifier models can be improved by adding social network analysis features. Additionally, we plan to work on detecting which bot accounts are controlled with same software agents.

REFERENCES

[1] Ferrara, Emilio, et al. "The rise of social bots." *Communications of the ACM* 59.7 (2016): 96-104.

[2] Danezis, George, and Prateek Mittal. "SybilInfer: Detecting Sybil Nodes using Social Networks." *NDSS*. 2009.

[3] Davis, Clayton Allen, et al. "Botornot: A system to evaluate social bots." *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016.

[4] Subrahmanian, V. S., et al. "The DARPA Twitter bot challenge." *Computer* 49.6 (2016): 38-46.

[5] Chu, Zi, et al. "Detecting automation of twitter accounts: Are you a human, bot, or cyborg?." *IEEE Transactions on Dependable and Secure Computing* 9.6 (2012): 811-824.

[6] Lee, Kyumin, Brian David Eoff, and James Caverlee. "Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter." *ICWSM*. 2011.

[7] Dickerson, John P., Vadim Kagan, and V. S. Subrahmanian. "Using sentiment to detect bots on twitter: Are humans more opinionated than bots?." *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*. IEEE, 2014.