Twitter Bot Detection using Machine Learning

Ruchitha M. Shanmugha Sundar(001838207), Chirayu Desai(001837499)
Northeastern University
{midigarahallishanm.r, desai.ch}@husky.neu.edu

1 Introduction

Twitter being one of the initial online social media platform to allow bots to autonomously perform operations like tweeting, re-tweeting, liking, following and much more. An estimate of two-third of the tweeted links in popular websites are posted by these bots. In order to ensure the information shared across the platform through conversations is credible, there is a need to focus on identification of spam bots. This will be an effort to prevent malicious automation from disrupting user's experience on twitter. Motivated by the need to identify and filter out the spam bots we have come up with an approach to distinguish between genuine accounts and spam twitter bots. This approach involves recursive feature elimination to select features with strong discriminating power and deriving dynamic and static features using two or more basic features. This process is repeated for both both user meta-data as well as for tweets data-set. In order to improve the predictive capability of our model we calculate weighted probability by considering the probability outcome of model which is trained on user meta-data and another model which is trained on tweet data. This results in a robust model which can detect bots by leveraging only a minimum number of features allowing to exploit the additional features that consists of user meta-data. The goal is to have solution which can suggest the chances of an account being a spam-bot from the very beginning of the creation of the account. The code for this work can be found at this url.

Contributions
The contributions that we are aiming to address are these challenges:
(1) Our work focuses on both, tweet-level classification and account-level classification. For Tweet-level classification we are making use of numerical features derived from the tweet, along with the textual aspects of the tweet by making use of TF-IDF and Bag-Of-words. You can observe that there is significant improvement in the performance of the model that we achieve by combining these two aspects. We are leveraging the power of Stochastic gradient descent along with TF-IDF to achieve near to perfect classification.

(2) Next, Our work introduces a model that can be used to predict whether or not a given account is handled by a bot or not. We make use of Supervised learning models like Random-Forest classifier or SVM as they result in a model with high accuracy. This model uses all data that are of high relevance that is obtained by making use of truncate-SVD and features selected using trial and error method. This proposed models help us to reach the state-of-art performance in bot detection.

2 Related work

A successful Machine Learning solution will help in removing such bots from twitter or flagging them as such to the users in real time.
Initial works by Yang et al., focused on detecting bots, which look to mimic human behavior and spread unwelcome advertising and malware [2]. They used features like URLs per tweet and ratio of friends to followers .
Cresci et al. worked on identifying fake followers and social bots[3][4].
Lee et al. came up with solutions that used features like the fact that bots are usually created at the same time, and many more. They implemented the Random Forest classifier with boosting and bagging[5].

The study made by Isa at [6] proposes a novel approach to detect and filter unwanted tweets. Their approach involves real-time spam bot detection. LSA is used to capture the information on semantics and relevance of the words in a tweet using which they further combine with n-grams to tag tweets as spam or not. Isa and team further conducted series of experiments on different classifying models such as Extremely Randomised Trees, Gradient Boosting, Support vector machines.
Identifying twitter bots can be quite challenging on a social media platform as there is no way to fully identify what a bot looks like. Unlike other social media platforms, twitter allows automation and semi-automation in their platform. This just makes identification of bots a gruel-some task. Twitter being a social media has varied number of data points which makes feature selection for model
All the above mentioned work focuses either on the tweet-level bot detection or the account-level bot detection. But our approach makes use of comprehensive model making use of both tweet-level or user-level data. We are taking into consideration the fact that tweet has a major impact in determining the bot pattern.

3 Models

3.1 Model/ Method

Machine learning provide a bag full of tools that can be utilized to make sense out of the given data. For Twitter Account level classification, Support vector machines was one of our first model choice.The reason being, SVM's ability to handle complex data with non-linear relationships as opposed to regressors which can handle only linear relationships. Data Points of the given data-set are represented in an n-dimensional space where n is the number of features and then SVM constructs an hyper-plane that can divide the given data points into two groups with minimum error. The only issue we have with this Model is that it's computationally expensive as it involves identifying the right hyper-plane with maximum margin to separate the data points. We observed that an SVM model built using user meta-data along with the derived tweet-level and user-level features showed promising results. We were able to achieve an accuracy of nearly 100 percent using this model.

Equation for SVM classifier is as follows:

for soft margin case.

$$y_i(w_i.x_i - b) >= 1$$

for hard margin case following is considered

$$max(0, 1 - y_i(w_i.x_i - b))$$

Where,
y_i is the i^{th} target
$(w_i.x_i - b)$ is the current output

In SVM, we wish to minimize the following equation.

$$\left[\frac{1}{N}\sum_{i=1}^{n} max(0, 1 - y_i(w_i.x_i - b))\right] + \lambda||w||^2$$

where the parameter λ determines the tradeoff between increasing the margin-size and ensuring that the x_i lie on the correct side of the margin.

We propose using a stochastic gradient descent classifier for making tweet data based predictions. This model will be used with the concept of incremental learning.

3.2 Features

Feature selection was used to reduce the dataset to consider just the robust features. Basic user features and account level features such as user-name, screen-name, location and description, account-verification-flag are considered to identify the accounts which lack basic information. Pairwise engagement features such as status count, number of tweets, favourite count, followers count, etc are also considered to gain insight on the dynamism of textual features . In addition to this we are deriving novel features by combining two or more basic features available in dataset such as interestingness $(favouritesCount/followersCount)$, followership $(followersCount/friendsCount)$, friendship $(friendsCount/followersCount)$ and names ratio $(screenNameLength/userNameLength)$. Other features are average tweet length, account age, digits in tweet names etc. Specifically for tweets level we have also looked at features like bag of words, Tf-idf and even explored doc2vec. Selecting user-name and screen name features is based on the fact that legitimate users contains user-names which are not long and contain very few numbers and special characters. Where as spam accounts is a random mix of characters and numbers. Similarly we are taking into consideration usage pattern as legitimate user usage pattern in random when compared to spam bots which is more systematic. Further, statistical properties that are derived using the user data, account information have been effective in detecting spam bots. By this exploratory analysis we gained an understanding on the textual features, composition of data and dynamic features to come up the concrete set of features that best suits the models mentioned below.

3.3 Feature Engineering

Though there are several methods that allow us to work on large feature set, it is proven by several researches that we can obtain high performance by using minimal number of features. This choice of limiting the number of features is to accommodate the model efficiency and Interpret-ability. Reduced number of features yields a model that is less prone to over-fitting caused by the outlier presence in data-set. Also, having just the features which are easy to interpret, allows to comprehend the data transformations that happen when dealing with models like Deep Learning and SVM, which are popularly known as unfathomable models.

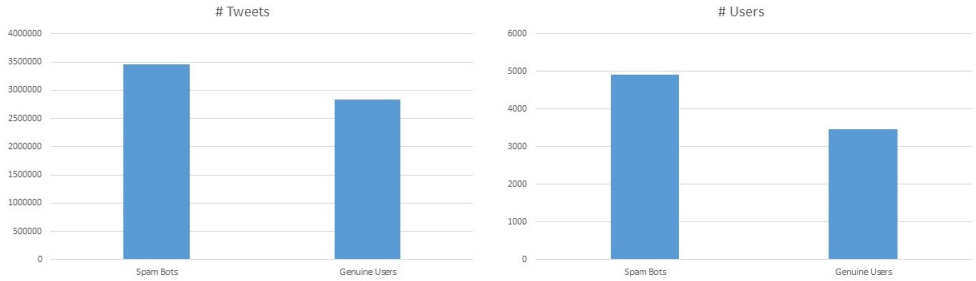
Table 1: Derived Features

Feature	Description
accountAge	total number of days from the start date of the account
digitsInScreenname	total number of digits in screen-name of the user.
digitCountName	total count of digits in user-name.
tweetsLength	length of the tweet test
userNameLen	length of the user name
screenNameLen	length of the screen name
nameSimilarity	similarity between the user name and screen name
friendship	ratio of total friends and follower friends
followship	ratio of total followers and friends
interestingness	ratio of favourite count and statuses count
activeness	ratio of status count and account age
nameRuser name lengthatio	ratio of screen name length and user name length

4 Experiments and Results

4.1 Datasets and Prepossessing

The datasets for our experiments have been curated from https://botometer.iuni.iu.edu/bot-repository/datasets.html. We have tried to curate datasets which have similar number of positive and negative samples, pretty much in line with actual twitter accounts. We have account information of 4912 spam-bots and 3474 genuine users. We also have 3457133 and 2839362 tweets collected from them respectively.
This size of the data is approximately 1.7 GB. In the initial data tweets and users were separated by classes like genuine users, spam bots, traditional bots etc. We created a unified dataset with users and labelled their accounts as bots or not and amalgamated the respective tweets by them in the same dataset.
At user level information like user name, handle, location, etc. are present. At tweet level we have focused on the content of the tweets.To clean the dataset we removed the columns with a few records, removed rows with dominant nan's.



4.2 Baseline Models: Logistic Regression

We have used logistic regression to create two baseline models one at user account information level and the other at tweet level information level. This is basically the out of the box scikit learn model with default initializations.
All features directly available in the dataset including the content of the tweets and metadata related features. There features were converted into numeric representations suitable for the model to use.
The user account information level baseline used features like user id, name, screen name, statuses count, followers count, friends count, language, default profile, protected, verified, description, contributors enabled were used in this model. This initial implementation had an accuracy of 57.60
The tweet level baseline model focused on the content of the tweets by the users. The accuracy for tweet content based model: 96.13
The primary goal of these baseline experiments was to get an idea of the impact of the various features on the results and their influence in detecting spam-bots. We observed that the tweet content had a prominent impact in influencing classification results.

4.3 Models Considered:

Logistic-Regression Logistic Regression was our first choice of the model as bot detection involves binary data. But since the data-set consists of outliers. Logistic Regression works well for data-set with features which are independent as opposed to the feature set we have which has co-relevance. Also the non-linear relationships will not be covered by this model. These reasons resulted in a model which performs poorly.

Support-Vector-Machine Support-Vector-Machine is being used to handle non-linear relationships in the data. However this is computationally intensive which makes it not so popular model as it's training time is too long. In case of SVM, interpreting the data transformation along with the plane boundary interpretation is hard. Although this model has many cons, it still is proven to work well for lower gamma value. Accuracy achieved is near to 100 percent.

Multiple-Perceptron-Classifier Deep Learning model which involves feed-forward network.This is a Popular techniques that can derive patterns and trends from imprecise and complicated data.It is again used to learn non-linear relationships. From what we observed models performance differed for each data-set.

Random-Forest-Classifier Random forest classifier is an ensemble model which aggregates several decision tree to reduce the effect of noise. We Tuned hyper parameters for this model manually by building models for different number of estimators.

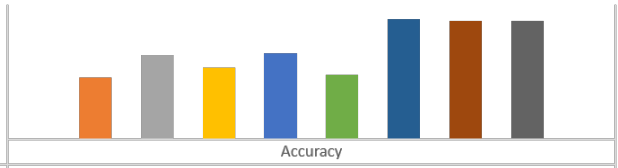
Stochastic-Gradient-Descent This is one of the Popular generalized algorithm which is used for training a wide range of models in machine learning. Primary reason for choosing this was that it is more flexible and more often than not converges faster for large data-sets. The one we have used also gives probabilistic class estimates.

Stochastic Gradient Descent with Incremental Knowledge Tweet data for each account was supplied in incremental snapshots for model training and prediction. This helps in identifying patterns at different temporal snapshots instances of data. This is done to measure the confidence of the system at different stages of the model.

4.4 Quantitative Results

Classification of Each Tweet

We started exploring different models to see their performance in classifying tweets themselves, the results are shown below:

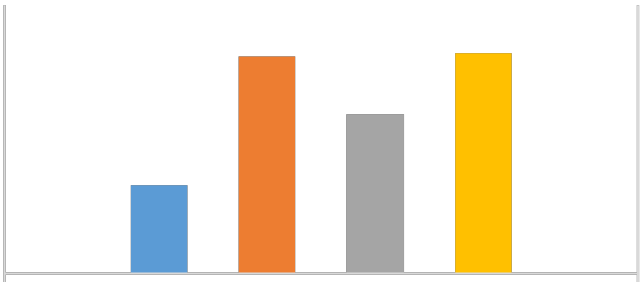


	Accuracy
MLPClassifier (hidden-layer - 5)	45.72%
MLPClassifier (hidden-layer - 20)	62.64%
MLPClassifier (hidden-layer - 50)	53.58%
MLPClassifier (hidden-layer - 100)	63.86%
MLPClassifier (hidden-layer - 200)	48.04%
RandomForestClassifier (n-estimators = 100)	89.25%
RandomForestClassifier (n-estimators = 200)	88.26%
RandomForestClassifier (n-estimators = 300)	88.06%

the primary observation was the fact that Random Forests are more stable for these tasks.

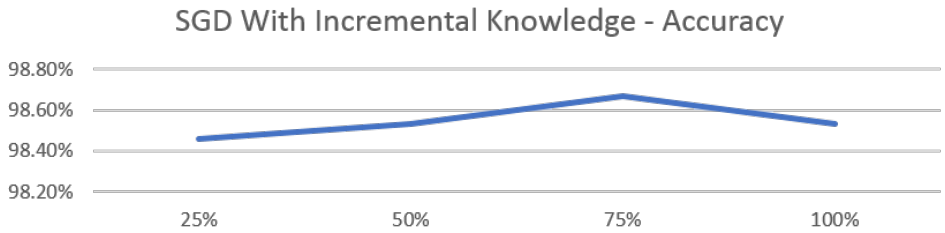
Aggregate Data Derived from tweets of each User Account

The next emphasis was on aggregate tweet data for each user account, the results are summarized below:



	Accuracy
Logistic Regression TF-IDF	97.86%
Logistic Regression Bag of Words	98.53%
SGD	98.23%
SGD Incremental Knowledge	98.55%

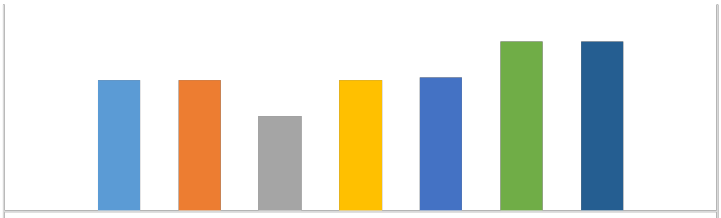
Firstly features like bag of words and TF-IDF were dominant influencers, boosting the model accuracy. We further worked on using Stochastic Gradient Descent with Incremental Knowledge to analyze the impact of providing the data to the model in incremental time snapshots. The model trained in this way had the best accuracy for prediction and confidence analysis based on tweet content, of course TF-IDF was the primary feature.



The above graph illustrates that when how the incremental model behaves when data is supplied in incremental snapshots of 25%. It is fairly stable ranging from 98.40 to 98.70%

Focus on user account metadata features

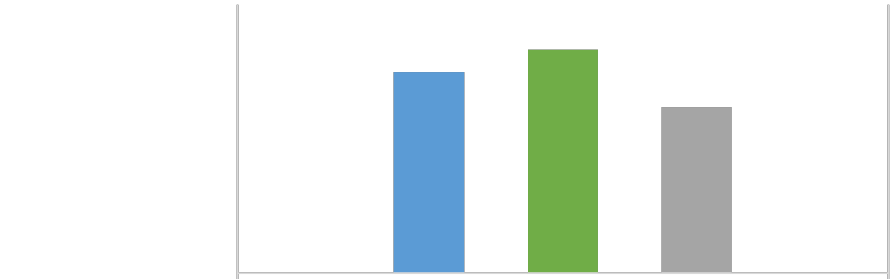
We created models that encompass the use of features available form user account metadata along with the content of tweets by users. TF-IDF was applied and the number of features for the model turned out to be too large. Consequently, we used Truncated SVD to do component analysis and pick 300 most relevant features. Again Random Forest was fairly good predictor for this feature. Detailed results are shown below:



	Accuracy
LogisticRegression	76.10%
Support vector machine (0.01)	76.10%
Support vector machine (0.1)	55.08%
MLPClassifier (hiddden-layers = 5)	76.10%
MLPClassifier (hiddden-layers = 50)	77.87%
RandomForestClassifier (n-estimators =100)	98.71%
RandomForestClassifier (n-estimators =200)	98.44%

Derived Features

We observed that the models were maxing out at 98.8 to 99.00 percent. Next we decided to derive features. The details are explained in the feature engineering section. When it all came together we were able to generate a model with 100% accuracy. The detailed results are in figure below:



	Accuracy
LogisticRegression	90.04%
Support vector machine (0.01)	100.00%
Support vector machine (0.1)	74.45%

4.5 Qualitative results

Calibration Plot

Calibration plot is used to measure the confidence on the prediction made by the model. By making use of this plot we can justify how well the predicted probabilities are calibrated and if a model is uncalibrated then how do we have them calibrated.

The following figure-1 shows the calibration plot for SVM model which is trained for different gamma values. Higher gamma value will result in a model which was built by taking just the support vector into consideration, leaving aside the other data-points. This results in model under-fitting resulting in poor performance of the model. We can notice from the below graphs that the model with lower gamma value aligns with the diagonal, indicating stronger model confidence. As opposed to this with increasing gamma value we notice that the plot deviates itself from the diagonal, indicating model is under-confident.

The following figure-2 shows the calibration plot for Random Forest Classifier which is trained for different estimator counts. You can notice that with increase in estimator count the plot deviates from the diagonal, making it less confident. For n-estimator =10, the plot aligns itself with the diagonal, showing that model has strong confidence with lower Brier score.

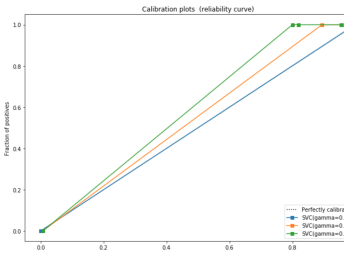


Figure 1:

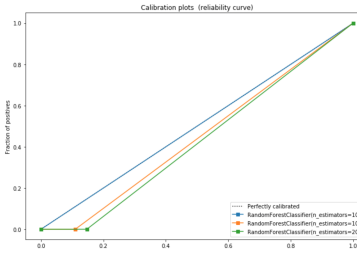


Figure 2:

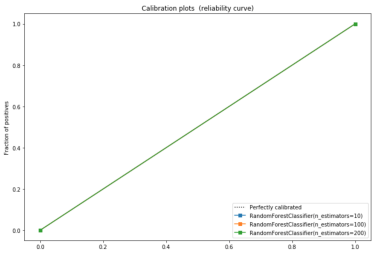


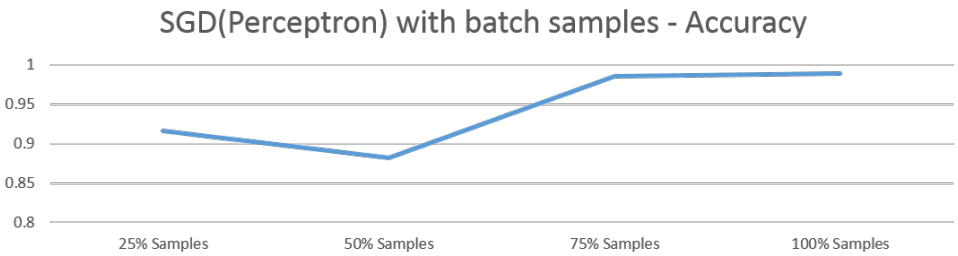
Figure 3:

An Interesting observation made was that for larger data samples considered, the confidence of the system remains same for all estimator counts. Larger or smaller value of n-estimators has no impact when data considered is large. This can be observed in Figure-3

The above method was employed to manually test the confidence of the system before having them built. This was done as part of parameter tuning, as searchCVGrid was taking too long to run for our models.

Model Learning

In order to observe the model learning for stochastic gradient descent we provided bathes of samples to train the model incrementally and checked the accuracy the figure below shows this:



The model showed a steady growth with stabilization after learning from 75% training samples, with an accuracy of 98.86%.

Model Prediction The eventual model for user account prediction has 100% accuracy, prediction snapshots are shown below:

ID	Name	Screen Name	Statuses	Followers	Friends	time zone	Classified as
191839658	pocahontas	farida	202968	2248	981	Greenland	Genuine User
465196345	Filippa Varelli	filippavarelli	120	0	0	Athens	Bot

Tweets	Classified as
This age/face recognition thing..no reason platforms can't have changing avatars of our actual faces to increase affect/better communication Only upside of the moment I can think of is that network news hasn't booked their #Baltimore panels with Bill Cosby.#fb	Genuine User
http://t.co/HyI5EQKz6Q Pink Floyd - Hey you [HQ] http://t.co/Pz648ODUKy Minestra di ceci castagne e baccala http://t.co/RbhGlv9DKM	Bot

5 Conclusion and Future work

In our analysis we found that the random forest classifier works consistently across all variants of the dataset. Making use of features derived using textual data, especially TF-IDF worked well for tweet level classification.

Only tweet content is not a sufficient identifier, we need to incorporate other features as well. Model built using user metadata, derived user features and tweet related data was able to correctly classify the user account with 100 % accuracy.

We intend to explore datasets of bots belonging to various other categories along with spam and traditional bots as part of future work.

Our eventual goal is to implement a plugin that can be used by the Twitter platform to classify the user account to bot or not a bot when you land on a Twitter user page. Also be able to classify the tweets when being pointed out.

References

[1] Gorwa, Robert. "Twitter has a bot problem and Wikipedia might be the solution". Quartz Media, 2017. https://qz.com/1108092/twitter-has-a-serious-bot-problem-and-wikipedia-mighthave-the-solution/

[2] C. Yang, R. C. Harkreader, and G. Gu, Die Free or Live Hard? Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 318–337.[Online]. Available: https://link.springer.com/content/pdf/10.1007

[3] S. Cresci, R. D. Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "Fame for sale: efficient detection of fake twitter followers," CoRR, vol. abs/1509.04098, 2015. [Online]. Available: http://arxiv.org/abs/1509.04098

[4] S. Cresci, R. D. Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "he paradigm-shift of social spambots: Evidence, theories, and tools for the arms race," CoRR, vol.abs/1701.03017, 2017. [Online]. Available: http://arxiv.org/abs/1701.03017

[5] K. Lee, B. D. Eoff, and J. Caverlee, "Seven months with the devils: a long-term study of content polluters on twitter," in In AAAI Intl Conference on Weblogs and Social Media (ICWSM), 2011.

[6] Isa Inuwa-Dutse, MarkLiptrott, Ioannis Korkontzelos, "Detection of spam posting accounts on Twitter". Available: https://reader.elsevier.com/reader/sd/pii/S0925231218308798

Appendix

Table below contains a brief summary of results:

Details	Model	Accuracy	Recall	Precision
TFIDF + Each Tweet Data	MLPClassifier (hidden-layer - 5)	0.4572	0.4572	0.4572
	MLPClassifier (hidden-layer - 20)	0.6264	0.6264	0.6264
	MLPClassifier (hidden-layer - 50)	0.5358	0.5358	0.5358
	MLPClassifier (hidden-layer - 100)	0.6386	0.6386	0.6386
	MLPClassifier (hidden-layer - 200)	0.4804	0.4804	0.4804
	RandomForestClassifier (n-estimators = 100)	0.8925	0.8925	0.8925
	RandomForestClassifier (n-estimators = 200)	0.8826	0.8826	0.8826
	RandomForestClassifier (n-estimators = 300)	0.8806	0.8806	0.8806
TFIDF + Each User Aggregated Tweet Data	Logistic Regression TF-IDF	0.9786	0.9800	0.9800
	Logistic Regression Bag of Words	0.9853	0.9800	0.9800
	SGD	0.9823	0.9800	0.9800
	SGD Incremental Knowledge	0.9855	0.9900	0.9900
	LogisticRegression	0.7610	0.7610	0.7610
	Support vector machine (0.01)	0.7610	0.7610	0.7610
	Support vector machine (0.1)	0.5508	0.5508	0.5508
User Metadata + Aggregated Tweets (SVD Truncated)	MLPClassifier (hidden-layers = 5)	0.7610	0.7610	0.7610
	MLPClassifier (hidden-layers = 50)	0.7787	0.7787	0.7787
	RandomForestClassifier (n-estimators =100)	0.9871	0.9871	0.9871
	RandomForestClassifier (n-estimators =200)	0.9844	0.9844	0.9844
	LogisticRegression	0.9004	0.9004	0.9004
User Metadata + User Derived Features + Tweet Related Data	Support vector machine (0.01)	1.0000	1.0000	1.0000
	Support vector machine (0.1)	0.7445	0.7445	0.7445