Twitter Bot Detector

Social Sensing and Cyber Physical Systems Final Report

Brynna Conway, Erin Flynn, Leah Plofchan
University of Notre Dame
Notre Dame, Indiana
Bconway3@nd.edu
Eflynn3@nd.edu
lplofcha@nd.edu

***Abstract*— Twitter has become a very popular social network, even in the automated sphere. Twitter bots are computer programs that produce automated posts and can automatically follow Twitter users. While some bots are well-intentioned, many are malicious. This paper outlines our finished project and solution for a Twitter bot detector for Social Sensing and Cyber Physical Systems. We created a program that accurately detects bots based on a score given to Twitter accounts based off various characteristics. The solution included a preliminary check for malicious bots as well. Some of these characteristics have been identified by researchers and assessed as valid criterion for bots.**

Keywords—APIs, bot detection, malicious, social sensing, Twitter, Tweepy

I. Introduction

Over the past decade, the social networking and online news service, Twitter, has grown rapidly and become widely used worldwide. As of June 30th, 2016, Twitter reported 313 million active monthly users. The opinions, conversations, and stories posted on Twitter have a huge effect on the day-to-day lives of people all over the world. These effects spill into all spheres of society: politics, popular culture, current events, personal relationships, etc. The list goes on and on. For example, on the day of the 2016 United States presidential election, Twitter proved to be the largest source of breaking news, with over 40 million tweets sent by 10:00 P.M. ET that day.[1]

With this huge amount of influence comes an even greater challenge for Twitter to achieve their mission of being an outlet "for all people to be able to create and share ideas and information instantly, without barriers".[2] Unfortunately, a growing trend has the power to threaten this mission: Twitter bots.

Twitter bots are computer programs that produce automated posts and can automatically follow Twitter users. These bots, which can interact with real human users, are created per specific algorithms that are designed for specific content. These bots are different from human users in that they directly access Twitter's mainline, via Twitter's widely used application programming interface (A.P.I.) by parsing information and posting at will. These bots can be created in any programming language and can live on huge cloud servers, adding up to create a prolific bot creation environment.

These bots come in all shapes and sizes. They can take many different forms and have vastly different end goals. Many bots are harmless and some are even helpful. They can be created to provide comedic relief, create political satire, or filter out news for specific users. In addition, some Twitter bots are used for strictly business purposes, designed to sell products and advertise brands.

However, it is no surprise that, over the past few years, people have begun to abuse the power of bots in a malicious and harmful way. Twitter bots can congest feeds with repetitive, automated posts. These posts can propagate rumors, spam, malware, political slander, etc. For example, these bots can spread false opinions in a quick manner to influence public opinion per a specific agenda. As these malicious bots have become more and

[1] Isaac, Mike, and Sydney Ember. "For Election Day Influence, Twitter Ruled Social Media." *The New York Times*. The New York Times, 08 Nov. 2016. Web. 23 Feb. 2017.

[2] "Company | About." *Twitter*. Twitter, n.d. Web. 23 Feb. 2017.

more prevalent on Twitter, the challenge to regulate them has become more and more pressing.

The first step in regulating these harmful bots on Twitter is detection; there are various ways to identify these bots. Researchers from the University College London have analyzed the networks of thousands of bots to narrow down some of the common elements that exist in these computer programs.[3] Some of the common characteristics that they found are that many bots have very young accounts (i.e. created recently), follow many accounts but have few followers, have obscure user names, have little personal content on their profile, and have many tweets posted in a short amount of time.

This project uses these characteristics to detect these bots so that they can be easily shut down and no longer inhibit the user experience on Twitter. To achieve this, the program looks for patterns that indicate automated text, unrealistic number of human tweets, spamming URL tweets and examine the follower/following ratio of suspected accounts, etc. This program gives a user account a score that is a sum of weighted values functions that conclude to indicate the likelihood of the bot. We came to a final bot detection threshold of greater than 70 after running the program multiple times on hundreds of users. Finally, we have an initial classification of malicious users; users that spam/clog a normal user's Twitter news feed. We hope that this project can continue to be updated with better features and more sophisticated technologies to provide the best possible experiences for Twitter users.

II. Related Work

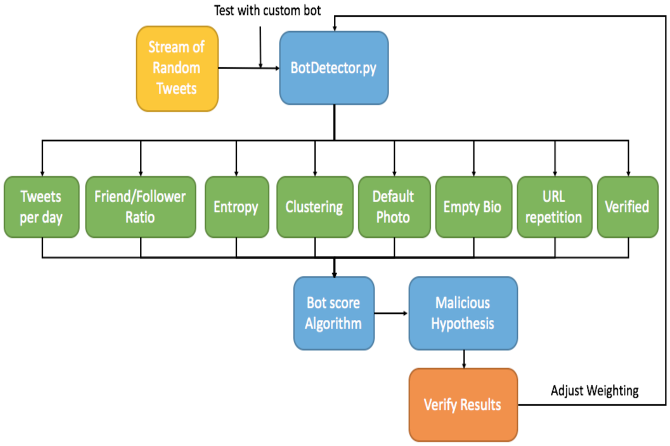
Since its release in May 2014, *BotOrNot* has been one of the primary programs for bot detection on twitter, made available to the public for private and commercial use. The founders of *BotOrNot,* C. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer [1] cite the dangers of bots on twitter. Funded by the U.S. Department of Defense, *BotOrNot*, uses over 1,000 predictive features to classify a user as a bot or a human. These six main features include: network, user, friends, temporal, content, sentiment. Network makes use of hashtags, mentions, and retweets to establish a user's network. The user information gives language, location, and account creation time. The friend information allows for mathematical analysis based on the number of followers, number of followees and number of posts. The temporal features look at timing patterns including tweet rate and inter-tweet time distribution. The content of the tweets provides linguistic cues using natural language processing to detect bot-like language. Lastly, sentiment gathers information on happiness level, emoticon scores and more. These features are used to apply a score to each account. The score will help determine if the user is automated or human, with 95% accuracy.

Z. Chu, S. Gianvecchio, and H. Wang [2] conducted research to determine and classify users on twitter. The work focuses on determining which users were bots, humans, or cyborgs (human-assisted bot, or bot-assisted human). Gathering data from over 500,000 users and 40 million tweets, Chu, Gianvecchio, and Wang used a four-part classification system. The first part uses an entropy component based on tweeting interval to measure behavior complexity. The second part, machine learning, makes use of tweet content to check if text patterns contain spam or not. Account properties act as the third part, using tweeting device and URL ration to detect abnormalities. Lastly, a decision useful account properties, such as tweeting device makeup and URL ration to detect abnormalities. Lastly, the decision maker uses Linear Discriminant Analysis on the data above to make the classification. The research concluded that overall the classification system was accurate in determining a bot versus a human. They found it was not as straight forward to identify a cyborg in the Twittersphere.

III. Problem Statement

Millions of the users tweeting daily are bots. You could be interacting with a bot and have no idea. While some of these users can provide comedic relief, or useful functionality like reporting the weather, others can be malicious. For this project, we have defined a malicious bot to be one that tweets the same content over and over and/or tweets excessively. This means that malicious bots are clogging other users' feeds daily, inhibiting the twitter experience.

IV. Solution



[3] "Massive networks of fake accounts found on Twitter." *BBC News*. BBC, 24 Jan. 2017. Web. 23 Feb. 2017.

Figure 1. Bot Detector program outline

A. Introduction

Our Twitter Bot Detector aims to solve this problem by identifying all bots on Twitter and then further classifying each bot as benign or malicious.
Figure 1 depicts the program flow of the bot detector, which was created in python using the Tweepy API to access twitter.

First, the program collects a large set of user IDs from a stream of tweets based on a filter. The user ID is then taken from the tweet to be passed into the program through eight classifying functions on those users, parsing through the users' data and tweets. These functions include verified, default photo, tweets per day, time of tweet entropy, friend/follower ratio, similarity clustering, empty bio, and URL repetition. Once passed through these functions, a score is calculated using the bot score algorithm. If classified as a bot, then a Boolean value of 0 or 1 is applied to each bot to signify if it was identified as malicious or benign. Lastly, these results are outputted to the screen.

With the outputted data, we manually checked if the users identified as bots appeared to be bots on twitter. During the initial testing stages, we used the results to reevaluate our bot score algorithm to better identify bots in the future. We then repeated the process. Initially, we used our own bot to test the functionality of the program and create the "perfect" bot.

B. Collecting User IDs

To get accurate results, the program needed a random sampling of user IDs from Twitter. This ensured lack of bias in our data collection. To make the sampling as random as possible, our program creates a Tweepy stream, collecting a specified number of recently posted tweets. The Tweepy API requires a filter to find specific tweets. Thus, the program filters for the words "the", "a", and "I", to spread a wide net for the types of tweets we want to collect.

Once the stream closes, the tweets are processed and the user ID of the tweet is collected and stored into an output file.

C. Verified

The first function checks if the user is verified or not. This function is run as the user IDs are being read in from the text file. If the user is verified, then their user ID will not be passed into any of the other functions. This is because there is a verification process that must be approved for a user to be verified; thus, if a user is verified it is not a bot as it has been vetted by twitter already.

D. Default Photo

The next function checks if the user has a photo or is using the default twitter photo. If the user has a default photo, they are more likely to be a bot than user's who do not have a photo due to the need to remain anonymous.

E. Tweets per day

The tweets per day function counts the number of tweets the user has posted in the last day. Depending on the number of tweets posted in that day, the user receives a bot score. As the number of tweets in a day gets higher, the weighting in the bot score gets higher. This is because bots often tweet many more times a day than humans, which also helps to identify a malicious bot who is clogging the twitter feed.

F. Friend/Follower Ratio

The friend/follower ratio function calculates the ratio between the number of friends a user has (the number of people that user follows) and the number of followers the user has. It also calculates the opposite ratio, followers to friends. Research shows that bots often have ratios very close to 1. This is because many bots will follow many people, and if they do not receive a "follow back" in a certain amount of time, then they will unfollow that user.

On the other hand, the ratio can be a very low number, close to 0. This is because there are many bots, like bots that update users on the weather, that have many followers because it is a benign and helpful bot. Other well-known comedic bots have many followers and follow very few users. Thus, a small ratio can identify bots. To avoid classifying famous people who also have many more followers than friends, we use the verified function as most celebrities are verified on twitter.

When a bot does not employ an algorithm to keep the ratio close to 1 they follow large amounts of users, but have very few friends. Thus, if the followers to friends is very low, then this user is also likely to be a bot.

The Friend/Follower ratio function takes all these different options into consideration and assigns a score to the user which is later evaluated in the bot score algorithm.

G. Time of Tweet Entropy

The next function checks the entropy of the times a user is tweeting. Often a bot has scheduled tweets, meaning they will tweet at a specified time every day. Thus, this function checks if a user has a high entropy or low entropy for the time of all the user's tweets processed. This is done using the NLTK library for language processing in python. Based on the resulting entropy, the bot score changes per the algorithm.

H. K-Means Clustering

To determine the level of variety of a user's tweet content, the bot detector includes a k-means clustering function. This function randomly selects tweets to serve as the initial centroids, and then loops through the user's tweets to assign each tweet to a centroid based on Jaccard distance, forming different clusters. By the k-means algorithm, this process will continue until the clusters converge, meaning that the centroids cease to change after each recalculation of clusters. However, to decrease the running time of the program, the re-clustering will stop after a maximum of fifteen reassignments. After the final clusters are determined, the function checks the size of each cluster. If a large outlier is found in the set of sizes, this indicates that a user may be posting many very similar tweets. To further ensure that the large cluster size indicates mass identical tweeting, the average Jaccard distance between the centroid of the outlier cluster and each tweet in the cluster is checked. If the average Jaccard distance is less than 0.2, then this is strong evidence that the user is a bot. The result of the k-means function is Boolean: the user either exhibits this characteristic or does not. Having this characteristic is very strong evidence that a user is a bot, although many bots do not mass tweet identical content.

I. Empty Bio

The empty bio function checks if the user has a bio or not on their profile by returning a Boolean value. Some bots do have bios, but a lot of bots will not put a bio on their profile. Thus, we can determine a higher bot score if the user does not have a bio.

J. URL Repetition

A twitter user who tweets or retweets the same URL or multiple URLs is likely to be a bot. Even more likely is that that bot is a malicious bot. Thus, this function checks the list of URLs that a user has tweeted and finds the frequency of each URL. Then depending on the number of unique URLs over the total number of URLs, the bot score algorithm assigns a value to the score.

K. Bot Score Algorithm

To classify the user as a bot or human, we used a bot score. Each function described above contributed to the bot score in some way. All functions except the k-means clustering, entropy, and URL functions were given a specific weighting. The ratio score given was worth 40% of the score, as was the tweets per day function. The empty bio and default photo functions were worth 10% of the score respectively. Each function yielded a score depending on the results of that function which was then multiplied by its weight and added to the total bot score.

In the case of the k-means clustering, entropy, and URL functions, the score was incremented without any weighting. After research and experience with bots after initial trials, we decided to treat these scores as a type of "extra credit." This is because if a user had tweeted at the same time every day, the same tweets frequently, or the same URL frequently, it was very likely to be a bot. Yet if they didn't, that did not mean they were less likely to be a bot. Thus, they received bonus points instead of getting the score weighted. The maximum value a user could receive was 150 points.

Throughout testing, the threshold value for the bot score changed. That is, we adjusted the score a user needed to receive to be classified as a bot or a human user. In the final results, we decided on a threshold of 70 to indicate a bot.

L. Malicious Hypothesis

Not all twitter bots are malicious; some exist for helpful purposes such as entertainment or business. Malicious bots are the type that are most important to detect, because they are more intrusive and hinder a Twitter user's experience. To differentiate between malicious and harmful bots, the bot detector includes a Boolean malicious score for each detected bot. This score is assigned based on the result of two functions: the k-means clustering and the URL checking. If a user receives the extra credit for at least one of these functions, it is marked as malicious. This method of malicious classification was based on the understanding that a malicious bot is one that clogs a user's feed with excessive redundant content.

M. Custom Bot for Testing

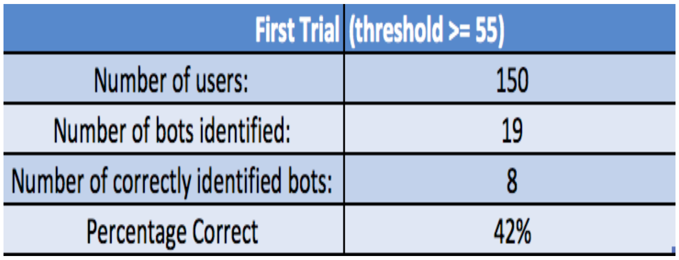
To create the perfect bot for initial testing purposes, we used Tweepy to simulate a bot on twitter. The bot, called JJ, followed large amounts of people to represent the follower/friend discrepancy. JJ also tweeted many times a day, and often almost the same tweet. The bot also did not have a profile photo or a bio. Thus, for testing purposes, JJ received a very high bot score. We could change JJ's behavior as needed to test bot conditions for our algorithm.

N. Verifying Results

To verify the results of the program, we performed manual checks of each twitter ID that resulted in a bot score greater than the threshold value. Manual checking is a difficult task for large amounts of users. Thus, we were only able to check the bots and verify if that user was a bot or a human. Due to the difficulty of manually checking all twitter users, the results described in the following sections focus on those users who yielded a bot score only.

V. EVALUATION

For the first trial of the program, we ran it on 150 randomly generated users that we streamed from Tweepy. For this trial, we hypothesized that the threshold for the bot detection would be around 50 points. After analyzing all the data from the first trial, we found that the threshold would be set at greater than or equal to 55 points. Any user ID that scores greater than or equal to 55 would be identified as a bot per our detection program. As seen in Figure 3 below, we put all the scores of the users that were identified as bots on a graph against an equal number of users that were identified as humans. By looking at this line graph, we see a clear gap between the human and bot line. This gap occurs at the 55-point line, thus verifying our threshold choice. With this threshold set, the bot detector program found 19 out of 150 (approximately 12.6%) users to be bots. This percentage aligns with the well-known statistic that 9-13% of users on Twitter are bots.[3] Of these 19 identified bots, after manually checking each user profile, we found that eight of them were actually bots. The table in Figure 2 shows that this gives a percentage rate of 42% for the first trial.



First Trial	(threshold >= 55)
Number of users:	150
Number of bots identified:	19
Number of correctly identified bots:	8
Percentage Correct	42%

Figure 2: First Trial Results

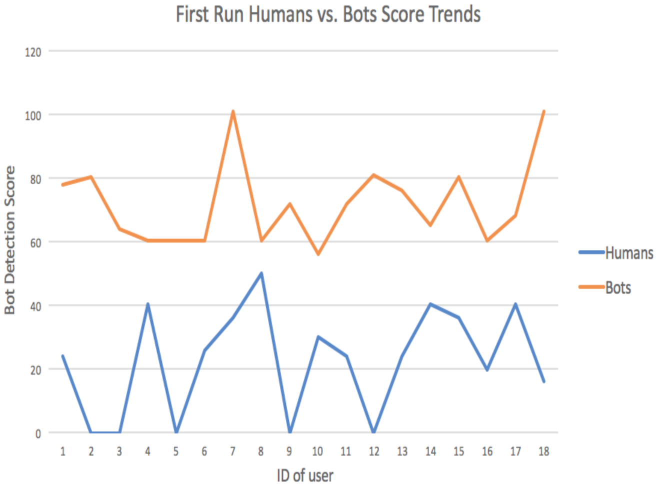
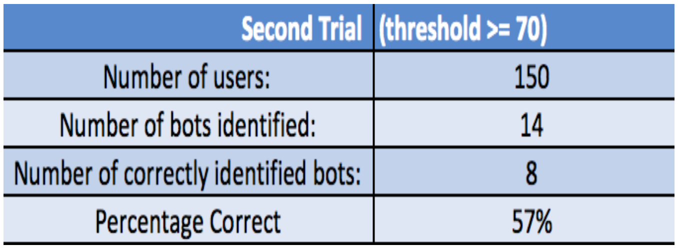


Figure 3: First Run Humans vs. Bots Score Trends

On the second trial, we again ran the program on the same previous 150 users. However, we did this after adding improvements to the bot score algorithm and creating new functions (i.e. time entropy function, URL function). With these new functions, we hoped to increase the accuracy rate of our program by finding more characteristics that could be attributed to a bot. By adding these functions, we needed to, thus, adjust the weightings for all the functions in relation to the final score. All these improvements can be seen in the results displayed in the following two figures.

In Figure 5 below, we again displayed all the scores of the 14 users identified as bots by our program against an equal number of users identified as humans. By looking at this graph, we could see a clear gap between line trends, allowing us to determine a bot detection threshold of greater than or equal to 70. With this threshold set, our bot detection program identified 14 users as bots. After our manual checks, we determined that only 8 of them were correctly identified. This gives us a 57% accuracy rate. As compared to our first trial, the accuracy rate increased 15%. This was a direct result of the significant improvements we made to the program after the first trial.



Second Trial	(threshold >= 70)
Number of users:	150
Number of bots identified:	14
Number of correctly identified bots:	8
Percentage Correct	57%

Figure 4: Second Trial Results

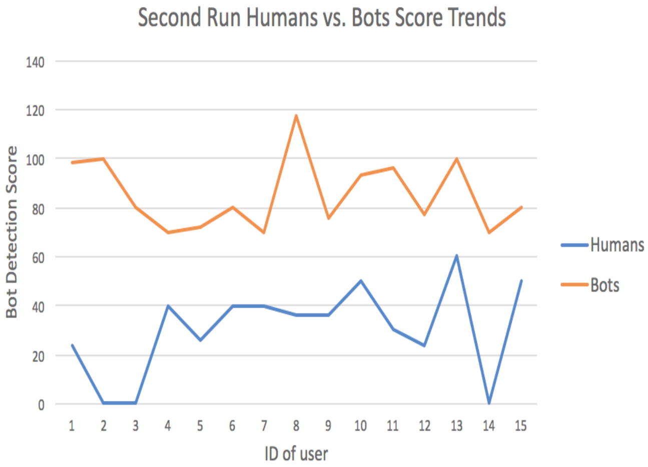
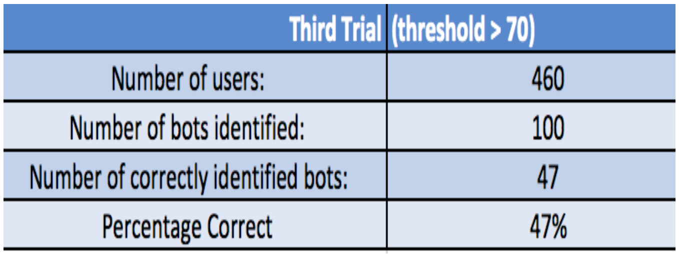


Figure 5: Second Run Humans vs. Bot Score Trends

After the second trial, we made a few more improvements/adjustments. After examining the results function by function, we determined that our number of tweets per day function could be improved. After researching more about the average number of tweets that users send out each day, we found that it is moderately suspicious when a tweets more than 70 times per day, and highly suspicious when a user tweets more than 140 times a day. Thus, we changed our function so that it would only add points to the final score if a user tweeted a minimum of more than 60 times per day. We also decided to run our program on many more users for this trial to see our program results on a greater scale. For this trial, we ran it on 460 users. In Figure 7, we can again see a clear gap between the humans and bots scores. From this graph, we decided to set the bot detection threshold at greater than 70 points. With this threshold, 100 users were identified as bots by our program. After our manual checks, 47 of those were accurate, giving us a 47% success rate. While this is a lower percentage than the second trial, it is important to note that we ran the program on 300 more users. Since our percentage of accuracy stayed relatively the same, we think this was a very successful run.



Third Trial	(threshold > 70)
Number of users:	460
Number of bots identified:	100
Number of correctly identified bots:	47
Percentage Correct	47%

Figure 6: Third Trial Results

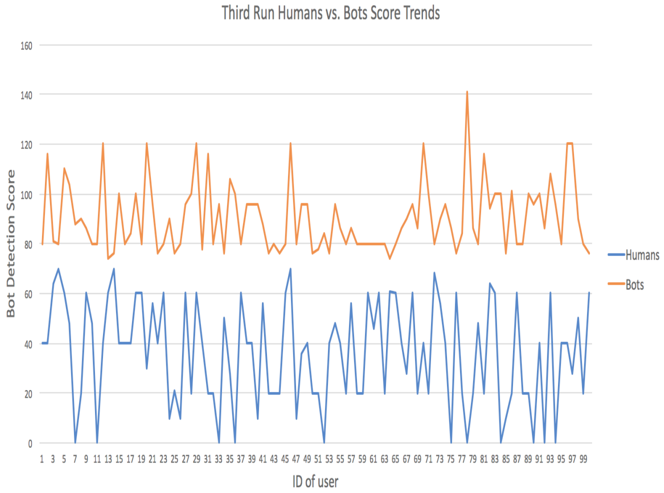


Figure 7: Third Run Humans vs. Bots Score Trends

Figure 8 below shows all three results of the three trials against each other. As explained before, while the second trial has the highest percentage, the third trial was run on many more users and thus is still successful.

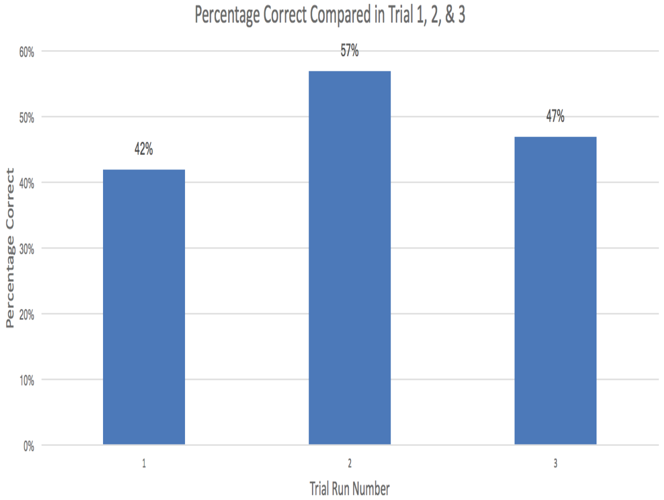
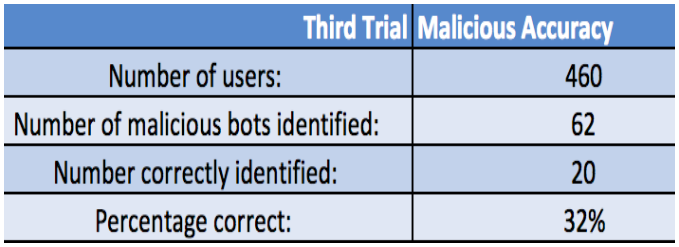


Figure 8: Percentage Correct

Finally, Figure 9 below shows the results of the third trial's testing of malicious bots. By adding a score to the k-means clustering and URL functions, we added a new classifying element to our program. It is important to note that we defined a malicious bot as a user that would clog another news feed with meaningless and unnecessary tweets that made little or no sense. Since we added it late on in the project, we did not hypothesize that it would be very successful. We were mainly adding this to begin the process of a much more complex process of classifying twitter users. We realize that much of this must do with analyzing the actual content of a tweet through processes like natural language processing. However, we still are relatively pleased with the initial accuracy of 32% correctly identified malicious bots.



Third Trial	Malicious Accuracy
Number of users:	460
Number of malicious bots identified:	62
Number correctly identified:	20
Percentage correct:	32%

Figure 9: Malicious Prediction Accuracy

VI. DISCUSSION/LIMITATIONS

A. Discussion on Results

Overall, we are pleased with the results of the Bot Detector. While we hope for higher success rates in the future, we believe that as a starting point, we could identify a significant number of bots.

It is important to note the variables that could have influenced the data collected. First, social media and humans are very unpredictable. There are many humans who use Twitter every day, often tweeting many times a day. Many users also use twitter to promote a brand, thus may retweet the same content daily. These are just a few examples of humans that may appear to be bots on Twitter. Thus, finding characteristics that identify a bot and not humans can be difficult. We employed the bot score weighting algorithm to try to minimize the confusion between bots and humans who act like bots, although this was not perfect.

It is also important to note that the last set of 460 tweets were collected during an awards show in Korea. It appears that to cast votes for the award show, users must tweet their vote. Thus, many users we found had just created their account and had tweeted the same tweet many times. So, while these users were not actually bots, they were identified as bots because of their bot-like behavior. This accounts for why the overall correct percentage is less than the percentage in the second round.

We believe these results are very promising and open many options for future improvements to the project.

B. Discussion of Future Work

This project has great potential for becoming a full functioning bot detector and helpful tool for many people. Thus, we have a few goals for the future of the project.

We hope to add many more functions to check characteristics of results. The more functions, the better overall classification of the bot. It will provide a more rounded score and hopefully eliminate the bot-like humans from the classification. Functions that could be added include one that follows tweeted URLs and determines the safety and validity of the links and one that uses natural language processing to determine if a grammatically correct tweet has real meaning or not. Continued research and interaction with bots on Twitter will yield more characteristics which the bot detector can consider.

To better identify the appropriate threshold value, it would be helpful to add a machine learning component. This would minimize human error in the results and allow the computer to continue to update with better thresholds to correctly identify more bots.

It would also be beneficial to create a more encompassing bot detector which could also classify cyborgs. Along with classifying cyborgs, it would be beneficial to do more work on malicious bots. Currently, we use a very basic definition of a malicious bot. Extending this definition and identifying more characteristics would lead to more accurate classification of malicious and benign bots.

Lastly, when the bot detector is ready to be deployed, we will create a web application for people to use the bot detector. A user would be able to input the twitter name and receive a likelihood of that user being a bot as the output.

VII. CONCLUSION

Bots are very prevalent on Twitter, which is unsurprising when considering the massive number of Twitter accounts in existence. Bots are automated accounts that produce content through computer programs, but can interact with other users in the same way as any human might. Though some Twitter bots exist for harmless purposes, such as weather updating or entertainment, some are malicious and clog human users' feeds with unhelpful, redundant content. Twitter aims to be a platform "for all people to be able to create and share ideas and information instantly, without barriers," though this goal can be restricted if humans are unknowingly interacting with inhuman accounts. To improve the user experience on Twitter and ensure that users are aware of with whom they are communicating, it is important to be able to detect bots. Though bots typically attempt to simulate human behavior, they often demonstrate several characteristics that help identify them as bots. These behaviors range from not having a user bio to tweeting identical content at the same time every day. Our bot detector solution harnesses these bot characteristics to identify and classify bots using a bot scoring algorithm. Because humans and bots can be similar and human social media behavior can be difficult to generalize, the current detector occasionally mistakes humans for bots. Through improvements such as a machine learning component and added functions, this bot detector has the potential to be an extremely useful tool for a wide range of people. Twitter is used across industries and by individuals around the world, so the bot detector could be used for a large variety of purposes, both businesses related and personal.

References

[1] Davis, C.A., Varol, O., Ferrara, E., Flammini, A. and Menczer, F. BotOrNot: A system to evaluate social bots. In *Proceedings of the 25th International World Wide Web Conference Companion* (2016); http://dx.doi.org/10.1145/2872518.2889302 Forthcoming. Preprint arXiv:1602.00975.

[2] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Detecting automation of Twitter accounts: Are you a human, bot, or cyborg?" IEEE Transactions on Dependable and Secure Computing, pp. 811-824, 2012.

[3] Nimmo, Ben. "Human, Bot or Cyborg?" *Medium*. Digital Forensic Research Lab , 23 Dec. 2016. Web. 07 May 2017.