Spot the Bot: Developing a Bot Detection Algorithm for Twitter

Aishwarya Kura, Oravee Smithiphol, Matt Lampl, Mahnoor Ayub, and Sara Maillacheruvu

Problem Statement

Defined as automated accounts that can post, like, retweet, and follow other users, social bots have become a major concern for social media platforms and their users. As technology advances, it has become increasingly difficult for social media users to discern content generated by bots from content generated by humans.[1] In turn, this can make it more difficult for users to critically evaluate bot-generated content, increasing the potential for misinformation.

Moreover, bots can be deployed for a variety of purposes – many of which are malicious. The US Department of Homeland Security identified terrorism, hate speech, and harassment, as well as civic engagement, counter-terrorism, and commerce as use cases.[2] Other uses include political propaganda, spamming, and manipulation of public opinion. Thus, bots present huge threats not just to social media platforms, but to society in general. Because social media is inherently social – interwebbed in global social systems – bots have the potential to sow national and international discord, if not controlled. In fact, a team of researchers found that nearly one in five tweets (19%) posting US election-related content in 2016 were generated by bots.[3] They also found that 25% of accounts using QAnon hashtags belonged to bots, meaning that bots add significant noise to conspiracy theories.[4] Further, as generative AI platforms evolve and become more advanced, these developments are likely to make it easier for malicious actors to leverage bots to spew nefarious content. In a time when democracy around the world appears fragile, the need for dependable algorithms to identify bots – and mitigate their negative social impacts – is all the more crucial.

While we have outlined a tale of despair in this introduction, we are heartened by recent research, which inspires our own work. A 2021 study out of University of Pennsylvania and Stony Brook University discovered that while bots can produce human-like content, their networks display highly similar traits, as compared to human users (e.g., the distribution of age of "bot users" posting content would be narrowly distributed, compared to wider distributions for real users).[5] This is to say that bots may be able to generate "human" content, but their interwebbed nature belies the fact that they are non-human. This could be a crucial factor in managing and offlining bot networks, one which we leverage in our analysis.

[1] Lerner, Evan. "Social Media Bots May Appear Human, but Their Similar Personalities Give Them Away." 11 November 2021. Penn Today. Accessed April 30, 2023. https://penntoday.upenn.edu/news/social-media-bots-may-appear-human-their-similar-personalities-give-them-away.

[2] "Social Media Bots Overview." May 2018. Department of Homeland Security, National Protection and Programs Directorate, Office of Cyber and Infrastructure Analysis. https://niccs.cisa.gov/sites/default/files/documents/pdf/ncsam_socialmediabotsoverview_508.pdf?trackDocs=ncsam_socialmediabotsoverview_508.pdf.

[3] Guglielmi, Giorgia. 2020. "The Next-Generation Bots Interfering with the US Election." Nature 587 (7832): 21–21. https://doi.org/10.1038/d41586-020-03034-5.

[4] Ibid.

[5] Lerner, "Social Media Bots May Appear Human, but Their Similar Personalities Give Them Away."

Literature Review

Various methods have been employed to classify bots, such as machine learning, crowd-sourcing, community detection, and correlated accounts. The first known detection of automated accounts on Twitter was in 2010 when a three-class classification system was implemented to distinguish between human, bot, and cyborg using an ensemble model.[6] Early works primarily focused on studying automated accounts as spam or for spam prevention purposes. In 2011, a team from Texas A&M University used honey pots to label bots. These honey pots used bots to generate senseless content that was only meant to attract other bots.

In 2014, Indiana University and the University of Southern California created the Bot or Not online API service, now called Botometer, which used supervised machine learning to determine whether an account is a bot.[7] The Botometer model is now widely used in applied bot research. Earlier research had identified different semantic features of tweet sentiments and the various social contacts between human users and social bots. The Botometer program uses machine learning algorithms to extract over 1000 predictive features that identify suspicious behaviors, primarily by characterizing the account's profile, friends, social network, temporal activity patterns, language, and sentiments. The program then produces an ensemble classification score on a normalized scale that indicates the likelihood that a Twitter account is a bot. Scores closer to 1 indicate a higher probability of being a bot, while scores closer to 0 are more likely to belong to humans.[8]

We also analyzed more recent studies that contextualize the issues of bots in the face of the pandemic. The article "Social Bots' Sentiment Engagement in Health Emergencies: A Topic-Based Analysis of the COVID-19 Pandemic Discussions on Twitter" analyzes the use of social bots in shaping sentiment and engagement around COVID-19 discussions on Twitter. In this study, the threshold for Botometer was set to 0.5 to distinguish humans from bots, consistent with the sensitivity settings of earlier studies.[9] The study finds that social bots were used to spread both positive and negative sentiments around various COVID-19 related topics. For example, some social bots were programmed to promote positive attitudes toward vaccines, while others were used to spread negative sentiments about government policies. The authors suggest that the use of social bots during health emergencies can have significant implications, as it can influence public opinion, affect people's behavior, and even impact policy decisions.[10] The study also highlights the importance of topic-based analysis in studying the use of social bots. By analyzing social media discussions around specific topics, researchers can gain a

[6] Beskow, David, and Kathleen Carley. n.d. "Bot-Hunter: A Tiered Approach to Detecting & Characterizing Automated Activity on Twitter." Accessed June 8, 2022. http://www.casos.cs.cmu.edu/publications/papers/LB_5.pdf.

[7] Ibid.

[8] Ibid.

[9] Shi, Wen, Diyi Liu, Jing Yang, Jing Zhang, Sanmei Wen, and Jing Su. 2020. "Social Bots' Sentiment Engagement in Health Emergencies: A Topic-Based Analysis of the COVID-19 Pandemic Discussions on Twitter." International Journal of Environmental Research and Public Health 17 (22): 8701. https://doi.org/10.3390/ijerph17228701.

[10] Ibid.

better understanding of the role of social bots in shaping public opinion and influencing decision-making.

For our literature review we also studied the article "Bot-hunter: A Tiered Approach to Detecting & Characterizing Automated Activity on Twitter," which introduces a method to identify and analyze automated activity on Twitter. The authors propose a tiered approach that combines various techniques, including machine learning and manual verification, to distinguish between bots and human users. They also provide a framework to categorize different types of bots based on their behavior and purpose. Some key findings from their research include that a number of accounts used randomly generated alphanumeric 15-character strings for the screen name such as Wy3wU4HegLlvHgC (not a real account).[11] The team used this phenomenon to annotate a large bot training set. Additionally, they found that 60% of bot accounts had a profile image. Conducting a reverse Google lookup of the image, they found that many Twitter bot accounts also used the same profile image. This seems to provide evidence that these accounts are mass produced with the same stock photo or hijacked profile image.[12] The researchers evaluated several traditional supervised machine learning models on the feature space described above. They chose to evaluate Na¨ıve Bayes, Logistic Regression, Support Vector Machine (SVM), Decision Trees, and Random Forest models. All of these models have been used in previous bot research attempts. Given the baseline model performance, Random Forest model performed best and achieved AUC = 0.994 with tuning.[13]

Data Summary and EDA

Data Summary

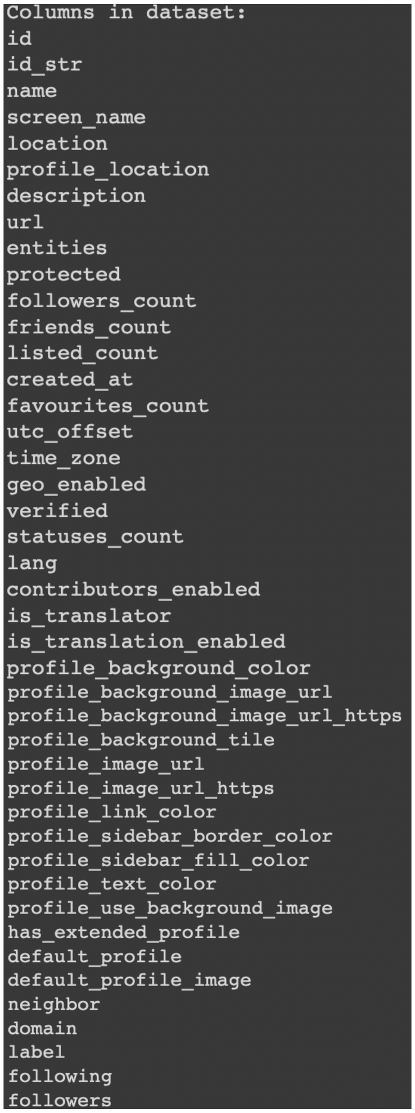
We focused our efforts on developing a bot detection algorithm for Twitter. Leveraging data from University of Indiana's well-known "Botometer" project, we requested and received access to **2020** Twitter data. The dataset included tweet text data as well as user and network metadata, which proved to be crucial in building our algorithm. There were 1,250,484 rows in the training dataset and 178,224 in the testing data set. In total, we analyzed about **1.42 million rows of tweets** in the overall dataset.

The data was very large in size, nested in a zipfile, and in a JSON file format. Every datapoint in the JSON had 4 levels of information. Each level was in the the form of a nested dictionary. We have summarized the levels and data contained in each field below:-

Level 1

Contains four keys: "ID," "profile," "tweet," and "domain."
Each key has a corresponding value that provides more information about the user and their activity on Twitter.

[11] Beskow and Carley, "Bot-Hunter: A Tiered Approach to Detecting & Characterizing Automated Activity on Twitter."
[12] Ibid.
[13] Ibid.



```
Columns in dataset:
id
id_str
name
screen_name
location
profile_location
description
url
entities
protected
followers_count
friends_count
listed_count
created_at
favourites_count
utc_offset
time_zone
geo_enabled
verified
statuses_count
lang
contributors_enabled
is_translator
is_translation_enabled
profile_background_color
profile_background_image_url
profile_background_image_url_https
profile_background_tile
profile_image_url
profile_image_url_https
profile_link_color
profile_sidebar_border_color
profile_sidebar_fill_color
profile_text_color
profile_use_background_image
has_extended_profile
default_profile
default_profile_image
neighbor
domain
label
following
followers
```

<u>Level 2</u>

The "ID" key contains a string value that represents the user's unique identifier on Twitter.

The "profile" key contains a nested dictionary that includes information about the user's profile, such as their name, screen name, location, description, URL, and more.

The "tweet" key contains a string value that represents a specific tweet that the user retweeted.

The "domain" key contains a list of strings that represent different categories or domains that the tweet is related to.

<u>Level 3</u>

The "profile" nested dictionary contains multiple key-value pairs that represent different attributes of the user's profile. Some of these attributes include:

- "id": a string value that represents the user's unique identifier on Twitter.
- "name": a string value that represents the user's name.
- "screen_name": a string value that represents the user's screen name or handle.
- "location": a string value that represents the user's location.
- "description": a string value that represents the user's profile description.
- "url": a string value that represents the user's website URL.
- "followers_count": an integer value that represents the number of followers the user has.
- "friends_count": an integer value that represents the number of friends the user has.
- Each of these key-value pairs provides more information about the user and their profile on Twitter.

<u>Level 4</u>

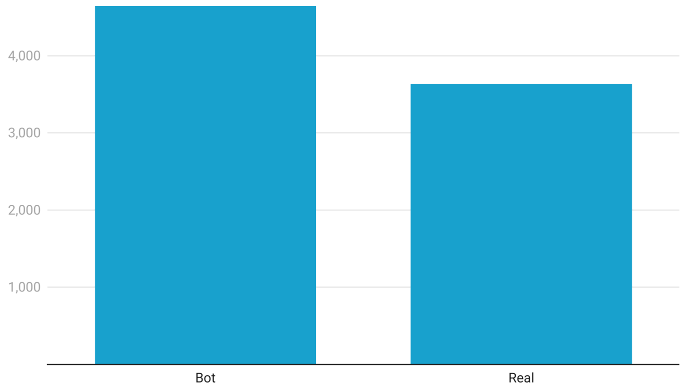
The "domain" list contains multiple string values that represent different categories or domains that the tweet is related to. Some of these values include:

- Politics, Business, Entertainment

Each of these string values provides more information about the content of the tweet and the topics that it relates to.

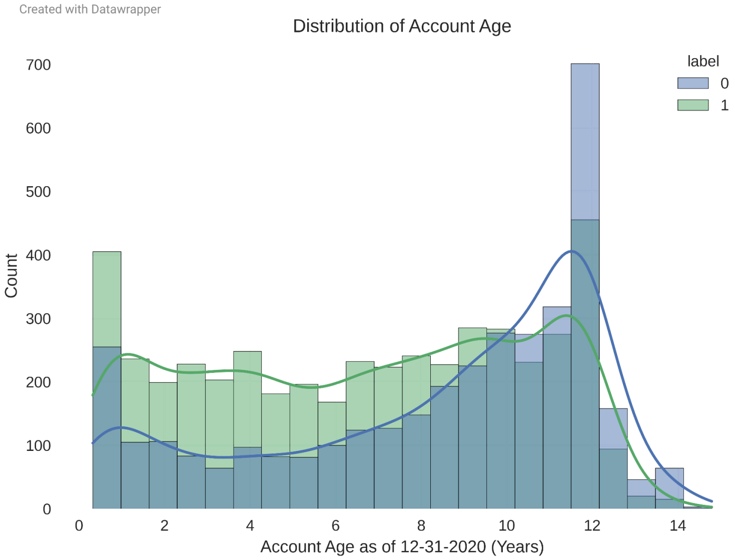
Exploratory Data Analysis

To start off the exploratory data analysis, the distribution between bot accounts and real accounts was visualized across the entire dataset. We found that the dataset was roughly 60% bots and 40% real accounts.

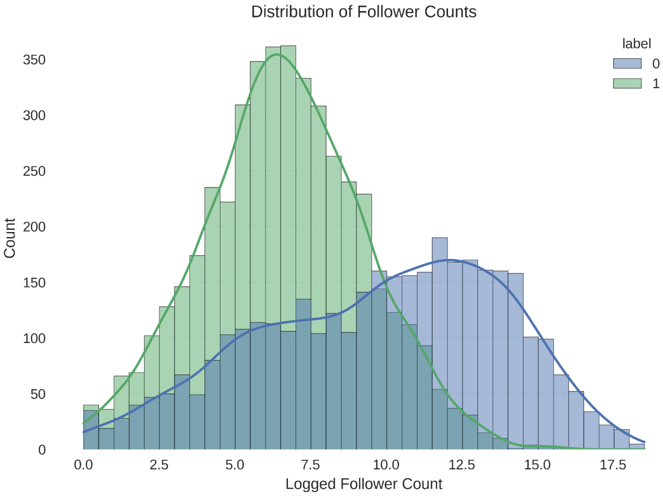


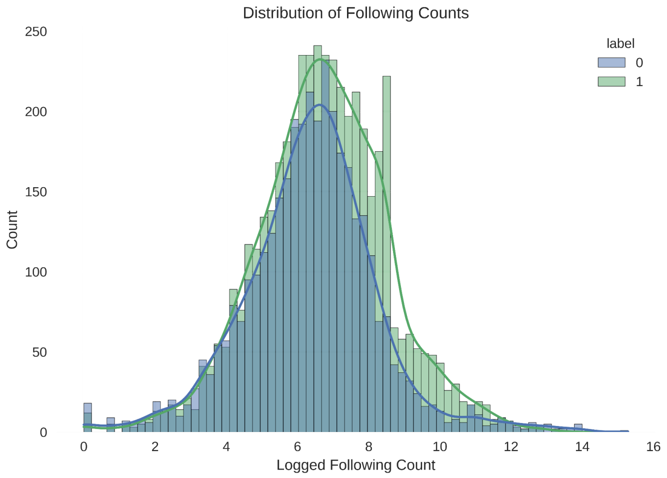
Created with Datawrapper

Next, we analyzed key variables in order to discern important variables for the modeling process later on. First, we ascertained account age by calculating the age of a given account on December 31st, 2020. We chose this date as our threshold because the dataset we used consisted solely of tweets from 2020. Bot accounts tend to skew just a bit younger than that of real accounts.

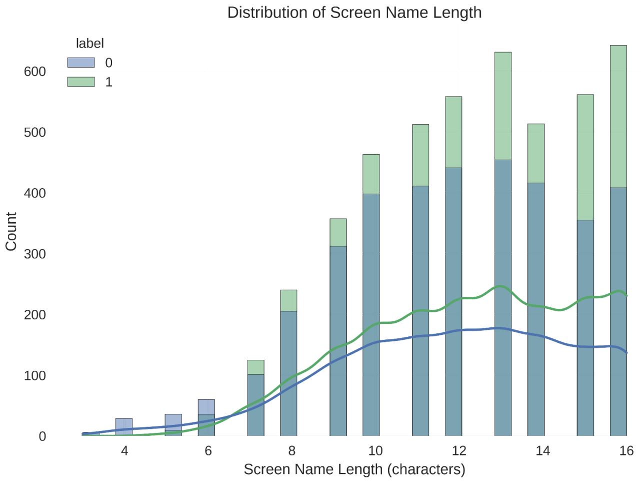


Second, in order to understand the network, we visualized the number of followers and the number of following or both bot and real accounts. As expected, bot accounts tended to have far fewer followers than real accounts. Conversely, both bot and real accounts had a similar number of accounts that they followed.

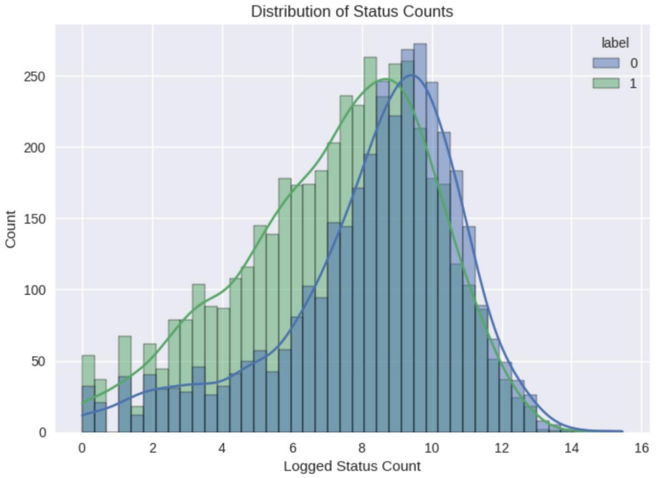


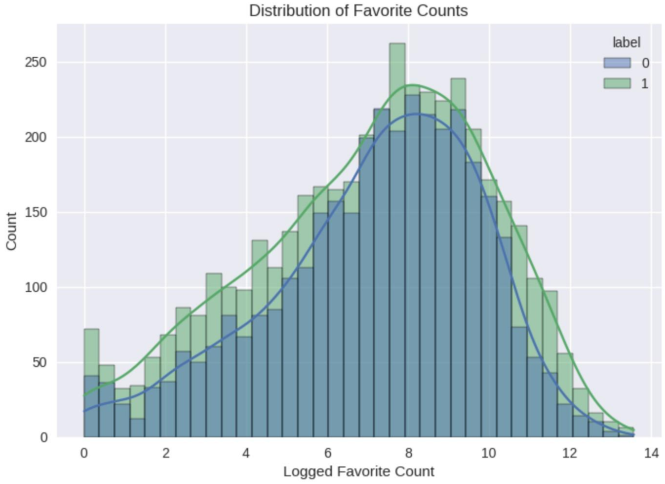


From our literature review, we expected that bot accounts might have longer account names than real accounts, as one study discovered that bots tend to have random alpha-numeric strings as their account name. Upon further investigation, however, bot and real accounts have roughly the same distribution in terms of screen name length.



Next, we examined the relationship between number of statuses (i.e., tweets). Bots tend to tweet fewer times than real accounts, though this could be due to the fact that bot accounts are also generally younger, and therefore, would have fewer tweets than older accounts. Additionally, the number of tweets favorited by bots and real accounts remains similar among the two groups.





Our Approach and Results

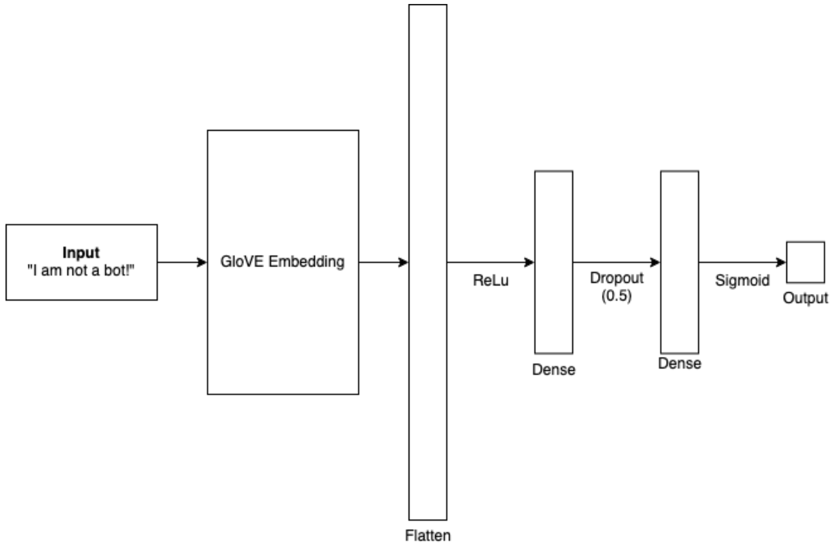
Described in detail below, we developed a four-tiered approach to identify bots in our dataset:

1) Tweet-based approach
2) Profile-based approach
3) Network-based approach
4) Combination of profile and network.

The input to our models included feature data extracted from the rows of Twitter data (account age, number of followers/following, etc.), as well as network features we engineered. The output of our models was a prediction of whether or not a given account was a bot. We take as our criteria of success the base rate of the dataset: if we were to predict every tweet were a bot, we would achieve a roughly 60% accuracy. As such, we aim to develop a model that exceeds the base rate, aiming for 70% accuracy.

Tweet-Based Approach

The first tier of our approach focused on using only the tweet data to create a classifier. We decided to use a neural network for this tier, as this is similar to approaches adopted by other researchers attempting to create bot detection algorithms. For this approach, each tweet was first encoded using a GLoVE embedding. The purpose of this is to encode each tweet as a numeric value, making it easier for the machine to understand the text. The reasoning behind using a GLoVE over, for example, Word2Vec was two-fold. First, the GLoVE matrix was trained using twitter data, making it suitable for our problem. Second, the GLoVE embedding we used was only 25 dimensions, making it much more feasible given our limited computing resources. For each tweet, the resulting embedding would be the mean value across those 25 dimensions for each word in the tweet. From there, the vector was run through a multi-layer perceptron neural network, with two dense layers, and with an output layer that was run through a sigmoid activation function. The resulting output was a number between 0 and 1 that effectively was the probability of a tweet coming from a bot account.



Results

Using a tweet-based approach did not yield results that were much higher than the base rate. The accuracy, precision, and recall of the model were .61, .60, and .88 respectively. This is not much higher than the base rate of simply classifying each tweet as a bot tweet. Therefore, we turned to different approaches to improve the model's accuracy.

Profile-Based Approach

Feature selection

The features that we are interested are as follow:

- **Account age:** Our dataset comes with 'created_at' which indicates the date of account creation. We use this information to calculate the age of an account up until 2020. In our dataset, we found that the average account age of bots tends to be lower than that of humans. We calculate the correlation between this number and the bot label outcome. We get -0.2 as a coefficient, which is significant
- **Number of statuses:** 'Statuses_count' indicates the number of tweets and retweets that an account has issued. We found that bot accounts appear to have this number higher than human accounts.
- **Number of favorites:** 'Favourites_count' indicates approximately how many times a Tweet has been liked by Twitter users. We found that bot accounts appear to receive slightly less favorite than user accounts.
- **Screen name length:** Our dataset comes with 'screen_name' variable which is a string of a display name of an account. We calculated the length of the screen name and used this as one of our features. According to the Bot-hunter paper, bot tends to have longer screen name length, although our EDA says otherwise: that bot and human accounts have similar screen name length.

Model selection

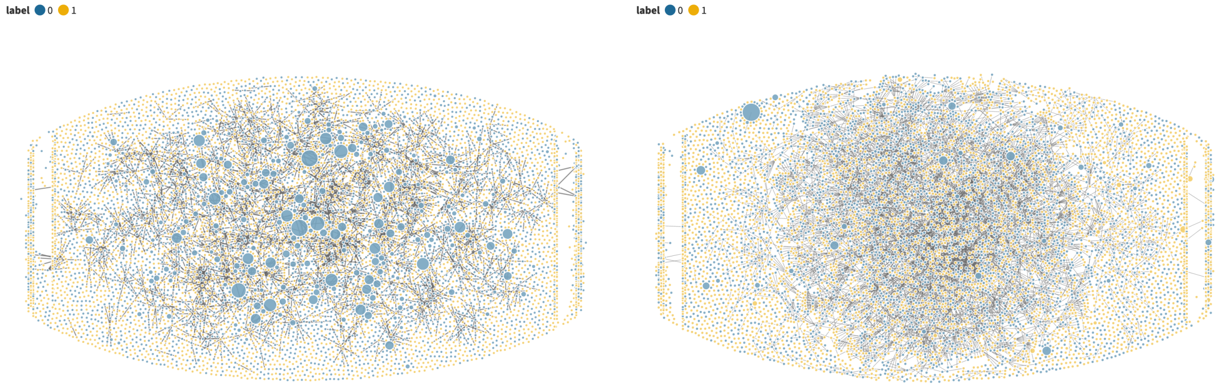
The Bot-hunter research paper stated that the two top performing models for bot classification are Random Forest and Logistic Regression. As such, we used these two models for this task. We saw that the Random Forest performs slightly better than the Logistic Regression and report those results here.

Results

The accuracy, precision, and recall of the model were 0.65, 0.69, and 0.73 respectively. Accuracy and precision are slightly higher than the tweet-based approach and is higher than the base rate. Nevertheless, we are interested in exploring more approaches to further finetune our model

Network-Based Approach

Next, we are interested in seeing how the way profiles connect with each other can be used to determine whether the account is a bot or not.



Network of followers (left) and Network of following (right)

Feature selection
The first two features we select are 1) number of people an account follows (friends_count) and 2) number of followers (followers_count). The latter is significant in bot detection, as we find that bots tend to have less followers; the correlation coefficient is -0.17

Next, we analyze the network of these accounts and how each of them impacts the network. Graph features are common centrality measures used in network analysis. They help to identify the importance of nodes in the graph based on different criteria. A profile account is called a node in a network. When an account follows another account then the node has a directed edge (or link) to another node.

Here is a brief explanation of some measures that can be used in network analysis:

1. **Degree centrality**: Degree centrality is a measure of the importance of a node in a network based on the number of edges it has to other nodes. If a node hasa high degree of centrality, it means that it connects with many other nodes. Degree centrality can be computed by dividing the number of links that a node has by the total number of possible links in the network.
2. **PageRank:** PageRank is a centrality measure originally developed by Google for ranking webpages. It measures the probability that a random walker (or surfer) starting at a given node will visit that node, after following a large number of links in the graph.
3. **Closeness centrality**: Closeness centrality measures how close a node is to all other nodes in the graph. The closeness centrality of a node is the inverse of the sum of the shortest path distances between that node and all other nodes in the network. Nodes with high closeness centrality are able to reach other nodes in the network quickly. In this context, it means that a node with high closeness centrality connects with diverse nodes that allow it to have a short path distance to all other nodes in the network. This means that an account follows diverse users in different communities in a network.

4. **Eigenvector centrality**: This is a measure of a node's importance based on the importance of its neighboring nodes. A node has high eigenvector centrality if it is connected to other nodes that are themselves important. Eigenvector centrality can be used to identify influential nodes in a network. Note that in this context, the importance here is determined by the number of links a node has which mean a number of accounts a node follows.
5. **Clustering coefficient**: This measures the degree to which nodes in a graph tend to cluster together.
6. **Betweenness centrality**: Betweenness centrality measures the number of times a node appears on the shortest path between any two other nodes in the graph. Nodes with high betweenness centrality are important for maintaining the structure of the network.
7. **Community detection**: This involves identifying groups of nodes in a graph that are highly interconnected with each other, but less connected to nodes outside the group. For example, we may assume that bots tend to be part of larger, more diverse communities, whereas real users tend to be part of smaller communities or clusters.

Nevertheless, as graph analysis is usually computational expensive, we needed to balance between the efficiency and the usefulness of the features. For example, betweenness centrality and community detection require high computational resources so we do not use them in our model. We identified the following features as meaningful as well as efficient features and used them in our model. They all have a positive correlation coefficient of +0.22 with a bot label.

1. **Degree centrality**
2. **Eigenvector centrality**
3. **Closeness centrality**

Some features that we’ve experimented with but that do not provide better accuracy are *PageRank* and *Average shortest path length.* As such, we eliminated them from the model.

Results

The accuracy, precision, and recall of the model were 0.71, 0.73, and 0.80 respectively. All of them are improved from the previous approaches.

Combination of Profile-Based and Network-Based

Finally we train the model with all the features we use in profile-based approach and network-based approach.

Results

The accuracy, precision, and recall of the model were 0.75, 0.74, and 0.87 respectively. All of them are improved from all the previous approaches.

Conclusion

By testing different data inputs – i.e., only using tweet text data and including network data – we saw that the results generated by text-based methods alone paled in comparison to approaches that leveraged user metadata. Crucially, this reinforces the need for approaches that leverage user and network metadata; text-based approaches are not enough. Moreover, as generative AI becomes more advanced, it is highly likely that the content generated by bots will be less stereotypically “bot-like” – misspelled words, odd punctuation, etc. – but rather will be higher in grammatical, syntactical, and orthographical quality, better mimicking that of humans. These advancements further emphasize the need to leverage the particularities of bot networks via metadata and non-textual elements to identify – and control – bots.

As we wrap up this iteration of the project, we now turn to consider project next steps. First, we would like to gather more metadata and network data to further strengthen and enrich our analysis. This may enable our models to perform better across evaluative metrics of interest, such as accuracy and sensitivity. In addition, we would like to test additional AI models to see if they are better suited to the task at hand. While we selected these models based on the literature review and are pleased with their high performance, we are interested in achieving even better results. Moreover, we are interested in seeing how well our models generalize both to other platforms – such as Facebook – as well as other national contexts. We suspect that bot networks may differ in nature on other platforms – perhaps, the distribution of age varies – meaning that platform-specific models are required; this is likely not a one-model-fits-all problem. In the same vein, we are curious if the features that helped our models perform well on the dataset at hand vary across geographic contexts. We have the same suspicions as regards platforms. Nonetheless, understanding cross-country and cross-platforms differences in bot networks is a rich area of research that may help, in the long run, to develop more comprehensive solutions to tackle bots.

We also consider challenges, surprises, and alternative pathways we might have taken. Originally, we intended to develop a fake news detector. However, after initial conversations with Dr. Steier, we learned that this approach would likely not produce novel results or have much applicability in a rapidly-evolving online world. In fact, our results buttressed this: the text-based iterations of the models performed the worst. While we are glad to have pivoted our approach, this did set us back in terms of identifying a new dataset, cleaning said dataset, and more. Further, the new dataset that we identified contained a series of large files, which took a while for us to figure out how to open. This is all to say that data infrastructure issues posed an issue for us, but we were able to surmount them in the end. Lastly, if we were to develop this project again, we would 1) test our models on other, non-Twitter datasets; 2) allocate more time to resolving data infrastructure issues; and 3) start with bot detection, instead of fake news identification, first.

We would like to extend a warm note of thanks to Dr. David Steier for his assistance in office hours, as well as to researchers at Indiana University who shared access to the dataset with us.

Contributions

Below, we have outlined team members' primary contributions to the project. The workload was distributed equally among all team members.

	Primary Task 1	**Primary Task 2**	**Primary Task 3**
Oravee	Identified potential project topics and gathered project resources.	Project co-manager. Planned meetings, ensured team met deadlines, and scoped project.	Drafted report section: Our Approach and Results, Conclusion.
Aishwarya	Exploratory data analysis.	Project co-manager. Planned meetings, ensured team met deadlines, and scoped project.	Drafted report sections: Data Overview, Conclusion.
Matt	Exploratory data analysis. Created visualizations for final slide deck and report.	Scoped and identified data size issues. Proposed and established solutions to these issues.	Drafted report section: Data Overview and EDA, Our Approach and Results.
Mahnoor	Led literature review for fake news topic before we switched projects.	Led literature review for bot detection and fake news topic.	Drafted report: Literature Review, Conclusion.
Sara	Drafted project proposal.	Created final presentation, and edited and outlined report.	Drafted report: Problem Statement, Conclusion.