Bot Detection in Reddit

Aniket Ajay Sahoo
Department of ECE
Virginia Tech
Blacksburg, VA, USA
aniketsahoo@vt.edu

Sai Anish Chinthalapudi
Department of CSE
Virginia Tech
Blacksburg, VA, USA
saianish@vt.edu

Spoorthi Airody Suresh
Department of ECE
Virginia Tech
Blacksburg, VA, USA
spoorthi@vt.edu

***Abstract*—Online Social Networks (OSNs) have become an essential part of our everyday life. With billions of users signing up every day, these social media sites continue to develop and dominate the Internet. Bots are thought to account for 9% to 15% of these users. Bots are widely used and adaptable to a variety of social media sites, covering a wide range of topics. As social media bots become more popular and useful, harmful bot behavior is expected to rise as well. Thus it is important to identify bots, and to do this we propose a model that can separate bots and users by comparing parameters. Parameters are either readily available from Reddit like karma or upvotes, or need to be modified in order to be used, like comment similarity and sentiment. The latter are obtained using NLP libraries. The results are evaluated using a confusion matrix and are compared in order to recognize which feature most accurately identifies bots.**

***Index Terms*—Reddit, formatting, style, styling, insert**

I. Introduction

Online social networks can be defined as dedicated web services and applications that allow individuals to engage with other users and/or find individuals with similar interests to one's own [1]. People's daily lives have become increasingly reliant on such online social networks. There has been a significant increase in the number of people using social media as it allows users to connect to people, share opinions, reaching out to potential customers, enhance their careers. In 2021, there are about 4.2 billion active social media users and about 4.66 billion people accessing the internet every day [2].

Social Networks such as Facebook, Twitter, Reddit, and others have dominated our world. With such social media networks being free of cost, there has been a 60% increase in the number of users using social media from 2017 to 2021 [3], for instance, Reddit has 52 million daily active users compared to 36 million daily active users in 2020, which is a growth of 44% [4]. Even though there are significant benefits, many people are wary of social media because of privacy issues and/or misleading activities aimed at harming legitimate users. For example, due to the ease of setting up an account, the possibility that an individual might deceive one another increases. Social media provides new environments and technologies for potential deceivers. Online social deception attacks such as cyberbullying, identity theft have significantly increased in recent times. About 4.8 million identity theft and fraud reports were received by the Federal Trade Commission (FTC) in 2020 [5].

Online Social Networks are potential targets for exploitation by bots due of their popularity and comprehensive API [6]. A social bot can be defined as a computer algorithm that creates content and interacts with people on social media in an attempt to mimic and maybe change their behavior [7]. A bot can be used for beneficial purposes like collecting information from various websites like news, weather channels, blogs and posting them on social media. However, they can also be used for nefarious purposes like listening to human conversations, extracting the personal information of users. This category of bots are designed with the intend to harm. They can be used to spam, spread fake information, create fake identities or crowdturfing. For example, there have been speculations that bots may have inflated support for a political candidate during elections and slandering opponents by posting tweets pointing to fake news [8,9].There have been cases where due to the spreading of false news by the malicious bots, the benign bots ended up re-posting the news without verifying the facts or the credibility of the posts [10]. Bots have been used to infiltrate political debates, manipulate the financial market, steal personal information, and disseminate false information. The detection of social bots, be it benign or malicious is, therefore, an important research path.

II. Problem Statement

Bot detection has been established in popular social networking sites such as Facebook and Twitter. However, related work on Reddit is basically nonexistent. This is mainly because social networks like Twitter and Facebook are organized around users, that is, user-to-user subscription and hence network based bot detection approaches and features are extremely effective on these platforms. Reddit is organized around topics, that is, user-to-topic subscription and hence these methods and certain features cannot be applied in the same way.

In Reddit, users often use pseudonyms and can remain hidden without any public activity. Also, user page customization is quite limited. Most often you can find users using default profile pictures. This makes it even more difficult to distinguish bots from users. Datasets are easily available for platforms like

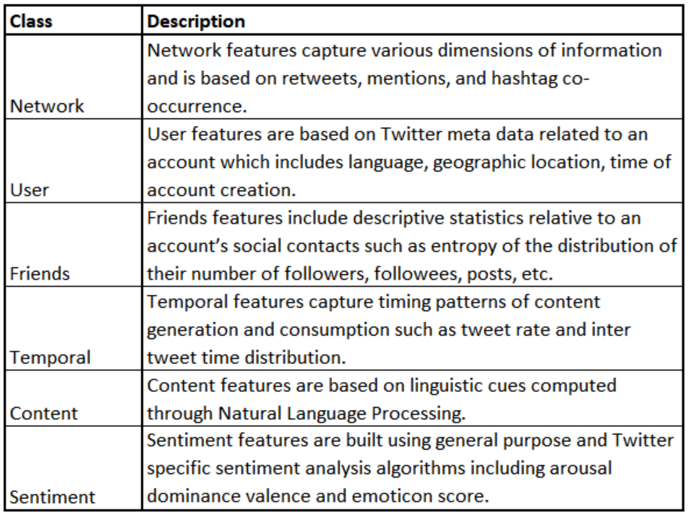
Twitter unlike Reddit. Hence limiting research with respect to bot detection on Reddit.

III. Related Work

In this section, we discuss an overview of the existing approaches of bot detection in Online Social Networks. The bot detection techniques can be broadly classified into three categories as follows:

- Structure-Based Bot Detection: bot detection systems based on social network topology information.
- Crowd Sourcing-Based Bot Detection: bot detection systems based on crowd-sourcing on user posts and profile analysis.
- Machine Learning-Based Bot Detection: bot detection systems based on feature-based machine learning methods.

BotOrNot is a publicly available Twitter Bot Detector that was made available in May 2016 and has been updated ever since. BotOrNot determines if a Twitter account is managed by a human or a computer. The BotOrNOt classifier uses Random Forest an ensemble supervised learning method. The performance was observed to be 0.95 Area under ROC curve. It takes a Twitter username and obtains the account's recent actions before calculating and returning a bot probability score. The bot probability score is calculated using a classification algorithm that takes into account a variety of factors as shown in Table 1.



Class	Description
Network	Network features capture various dimensions of information and is based on retweets, mentions, and hashtag co-occurrence.
User	User features are based on Twitter meta data related to an account which includes language, geographic location, time of account creation.
Friends	Friends features include descriptive statistics relative to an account's social contacts such as entropy of the distribution of their number of followers, followees, posts, etc.
Temporal	Temporal features capture timing patterns of content generation and consumption such as tweet rate and inter tweet time distribution.
Content	Content features are based on linguistic cues computed through Natural Language Processing.
Sentiment	Sentiment features are built using general purpose and Twitter specific sentiment analysis algorithms including arousal dominance valence and emoticon score.

Fig. 1. Example of a figure caption.

The approach used in BotOrNot does not apply to Reddit. This is because platforms like Twitter promote user-to-user subscription whereas Reddit is a user-to-topic subscription. The approach makes use of only one metric, that is, Area Under Curve (AUC) which is likely an overestimate given the age of the training data.

Another Twitter bot detection approach uses supervised machine learning techniques and compares its results with a custom classifier used to detect Twitter bots from a given training dataset. The dataset used for training is a set of known bots that discussed information regarding COVID-19. These Twitter bots tried to reach people by spreading fake news and malicious URLs regularly. Different algorithms such as Decision Tree, Logistic regression, K nearest neighbors and Naïve Bayes are implemented on the features extracted from the dataset and the algorithm with the highest accuracy is compared against a custom classifier. This approach considered a bag of bots words, that is, words that are commonly used by bots and the classifier based on this bag of words has been proposed.

The Reddit Pattern Detector focuses on detecting propaganda/political sentiment spreading accounts on Reddit. It uses a network-based approach to identify chains of such accounts and is used to detect both users and bots. The dataset used is based on all users, posts, comments made in a month on the subreddit ”The Donald”. Their hypothesis was that because of the Russian influence campaign, Trump's subreddit would be a good training ground. It used edge weight and downtime between comments as well as post-to-post and user-to-user networks to determine said patterns. The detector checks to see if “bot” is in the username to try and classify bots. The approach used for Reddit Pattern Detector is far too simplistic as having ”bot” in the username does not guarantee that the user is a bot. This approach is specific and can be applied to subreddits susceptible to propaganda.

The Reddit Troll Detector aims to detect disruptive and non-disruptive users in Reddit. It uses predefined features for classifier along with statistical features such as comment history - all the comments made by the user, karma - ratio of number of up-votes to down-votes, comment rate or count – the number of comments by the user, mean comments per week, account age - age of the account since creation date. It uses reply networks to see how disruptive users are different from normal users. The Reddit Troll Detector aims to detect disruptive users and is more useful for detecting trolls rather than detecting bots. The detector uses a limited database.

Another bot detection method on Twitter uses machine learning to detect fake identities. It uses friends and follower networks to identify fake identities and psychology statistics. The dataset consists of fake identities out of which many were manually created and added to it. The bot detector uses features such as account age - age of the account; friends to follower ratio - number of friends to the number of followers ratio; profile photo, name, and description -whether the profile has a name, photo, URL, etc.; length of the profile username. The effectiveness is measured using the accuracy and precision scores. This approach mainly focuses on fake accounts by humans on Twitter and does not apply to Reddit due to the difference in the structure of social media. This approach cannot be used on social media platforms that do not use features like followers, tweets, likes, followers to friends ratio.

Criteria	Davis et al. (2016)[8]	Ramalingaiah et al. (2021)[10]	Hurtado et al. (2019)[11]	Ashford et al. (2020)[12]	Van Der Walt et al. (2018)[13]	Our Approach
Classifies Users	Yes	Yes	Yes	Yes	Yes	Yes
Detects Bots	Yes	Yes	Yes	No	Yes	Yes
Individual Bots	Yes	No	No	No	Yes	Yes
Applicable to Reddit	No	No	Yes	Yes	No	Yes
Distinguishes based on intent	No	No	Yes	Yes	No	No

TABLE I
COMPARISON OF EXISTING APPROACHES WITH OUR APPROACH

IV. COMMON LIMITATIONS OF EXISTING WORK

- Almost none of these approaches can be applied to bot detection on Reddit
- This is because social networks like Twitter/Facebook are organized around users (user to user subscription) while Reddit is organized around topics (user to topic subscription)
- Reddit based detection approaches are not specialized
- Their main goal is usually to identify accounts spreading misinformation/being disruptive without classifying them as bots or users
- None of them can actually identify individual bots

V. PROPOSED SOLUTION

- We propose a model that can separate bots and users by comparing parameters.
- Parameters are either readily available from Reddit like karma or upvotes, or need to be modified in order to be used, like comment similarity and sentiment. The latter are obtained using NLP libraries.
- The results are evaluated using a confusion matrix and are compared in order to recognize which feature most accurately identifies bots.

VI. NOVEL IDEAS

- While bot detectors do exist, specialized bot detectors using Reddit parameters are new
- Our approach builds on this by using modified parameters that we think will be more effective to distinguish bots from users that are not readily available from Reddit
- We also compare different popular classification methods to figure out which works best for this purpose and use different NLP methods to extract our modified parameters to compare their effectiveness as well

VII. METHODOLOGIES

A. Overview

In this section we discussed the dataset, features and ML algorithms used in the experiment

B. Dataset

The dataset used in the experiment is a labelled dataset consisting of more than 100000 instances which contains the posts made by both bots and normal users. Features are the columns in the dataset which can be used to train the models.

There are some features which are given in the dataset. The features included :

- *banned by:* It states whether the author is banned to post or comment or not. In the dataset used in the experiment we have all the values as false since not everyone will be able to see the banned users but only the admin/moderator of the subreddit can be able to view the banned users.
- *no follow:* It states whether the author has any followers or not
- *link id:* It is the id internal to reddit application to distinguish the posts or comments
- *gilded:* Reddit allows users to send other users awards purchased via Reddit coins, the more expensive ones give them access to 'Reddit Gold', which is premium Reddit with access to restricted subreddits. This feature shows whether the post has any awards or not.
- *author:* This feature has the name of the author
- *author verified:* This column states whether the author is verified or not
  Reddit **Karma** is a score you get when you post and comment on Reddit. Your total Karma is shown on your profile, and when someone clicks on your username, they'll see a breakdown of Post Karma and Comment Karma. Some people are so good at Reddit that their Karma runs into the millions.
- *author comment karma:* Karma score obtained from posting comments on posts
- *author link karma:* Karma score obtained from making posts
- *num comments:* The total comments made by the author
- *created utc:* This shows the time stamp at which user has posted.
- *ups:* It states the total number of upvotes which is equivalent to likes in other social media networks
- *downs:* It states the total number of downvotes recevied.

Some other features which reddit gives includes controversiality, quarantine, over 18. These features are used internally in the reddit application.

Apart from the features included in the dataset, we have created some new features which we think might be useful to detect the bots. These new features are created using python and NLP libraries like sequence matcher, text blob and word2vec.

The features created by us are :

- *comment difference:* It is the similarity score of between the text which has been posted by the author.
- *sentiment polarity:* Sentiment polarity score of the text which is posted by the author

- *comment score:* Number of votes received on the posted comment
- *recent num comments:* Number of comments made in the last 30 days

As per the literature review section in the paper But there are some popular features which have been used by other bot detection techniques on other social media sites which wont work on reddit. Those include:

- *default profile:* whether the user has altered their profile or not. It's not uncommon to have unchanged reddit profiles, most users on Reddit have never changed their profile picture
- *Friends count:* how many friends the user has It is a very important feature on other social networks but Reddit doesn't have an add friends feature.
- *Followers count:* how many followers the user has Reddit does allow users to follow other users, however we found that this didn't have a particularly big impact. This is likely because several users who don't make posts whereas simply comment on them and some lurkers just don't garner a following at all, but the most important thing is unlike Facebook or Twitter most users use Reddit under a pseudonym, so their friends are unlikely to recognise/follow them.

C. Pre-processing the dataset

The dataset which was used in the experiment has to be pre-processed before it can be used to train/test the model. Pre-processing steps includes:

- *Drop Columns:* Inclusion of columns which wont be any use will only hurt the performance of the model. So we have dropped some unnecessary columns which are internal to reddit application like banned by, quarantine, link id, num reports.
- *Null/Missing Values:* Some of the values in the dataset maybe missing or has NaN(NotANumber) values. These values cannot be processed by the machine learning algorithms due to which we have to deal with them. In the experiment we have replaced these values with 0 or an empty string based on the datatype of the column.
- *Duplicate values:* Duplicate values in a dataset might give us incorrect results/evaluations because if a duplicate value is both tested and trained then its not a valid machine learning model. So we have removed all the duplicates in the dataset.

D. ML Algorithms

After pre-processing the dataset, it must be learned using ML algorithms. Since the goal is to classify the users as bots or not it is a classification problem. So we have chosen supervised machine learning based classifiers. For this experiment We want to see how different models can be trained on the dataset so we have chosen 4 different classifiers.

1) **Decision Tree Classifier:** Decision tree builds classification models in the form of tree structures. It breaks down dataset into smaller and smaller subsets while associated tree is incrementally developed, final result being tree with decision nodes and leaf nodes. Each decision(non-leaf) node of the tree contains a split point that is a test on one or more attributes and determines how the data is partitioned.
2) **Random Forests Classifier:** Random forest contains many number of decision trees on random subsets of given dataset, it takes the average to improve the predictive accuracy of the dataset.
3) **Naive Bayes Classifier:** Naive Bayes predicts the membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class.
4) **Logistic Regression Classifier:** Logistic regression classifier predicts a dependent data variable by analyzing the relationship between one or more existing independent variable.

Regarding the reason on why we have chosen these algorithms is because we want to compare what type of ML algorithms works best i.e tree based vs probability based ML algorithms so we have picked the popularly used trees based classifiers like decision trees and random forests versus probability based classifiers like naive bayes and logistic regression.

E. Metrics

The following metrics are used to measure the **effectiveness** of the classifiers.

- *Confusion Matrix:* The confusion matrix is made of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). They are the basic components for other accuracy metrics, such as precision and recall.
- *Accuracy:* Accuracy measures correct detection for true positives and true negatives. However, when the datasets are not balanced such as too large true positives with too small true negatives or vice-versa, this metric may mislead. It is given by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- *Precision:* Precision estimates the true positives over positives detected including true positives and false positives. It quantifies the number of correct positive predictions made. Essentially the ability of the model to return only the bots This metric is estimated by:

$$Precision = \frac{TP}{TP + FP}$$

- *Recall:* Recall captures the true positives over the actual positives include true positives and false negatives. It quantifies the number of correct positive predictions made out of all positive predictions that could have been made. Since we are focusing on bot detection, it is important to know how many of the total bots were identified This metric is estimated by:

$$Recall = \frac{TP}{TP + FN}$$

- *Area Under the Curve (AUC) of ROC:* ROC is a probability curve and AUC represents the degree or measure of separability. AUC is calculated by the the area under the ROC curve. It measures the probability of a classifier to correctly identify a true-positive data. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1.
- *Feature Importance:* Feature importance is the score given to input features based on how useful they are at predicting the target variable. Feature importance's are calculated for the classifiers which have shown more accuracy and recall.

  In addition the following metrics are used to measure the **efficiency** of the classifiers.
- *Runtime:* In this paper we have taken two different types of runtime. One is the time taken to pre-process the data and add the extra features like comment difference and sentiment polarity. It will be same for all the classifiers as the pre-processed dataset is used to train/test the model. Another is the time taken by the different classifier algorithms to learn and test the approach.

F. Experimental Setup

To run the experiment we have used python programming language using python jupyter notebook IDE. To train and test the ML algorithms scikit learn framework is used.

G. Splitting the dataset

While splitting the dataset and evaluation of the dataset, we have used k-fold cross validation and splitting the

k-fold Cross-validation is a re-sampling method that uses different portions of the data to test and train a model on different iterations. It is mainly used in classification, and one wants to estimate how accurately a predictive model will perform in practice.

We have chosen k value between 3 and 10 as they are the mostly commonly used range for cross validation and increasing the value of k, increases the cost in running the algorithms.

We did not observe much changes with changing the values and so we have agreed to use k=5 also keeping in mind about the cost. So the results obtained shows the generalized values on the dataset.

On top of the cross validation we have split the whole dataset in the ratio of 60:40, 65:35, 70:30, 75:25 to see whether the model is generic or not.

VIII. Results

A. Confusion Matrices

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It also makes it easy to compare different classifier's true positives, true negatives, false positives and false negatives.

Decision Tree	Predicted		
True	Normal	Bot	All
Normal	29156	6	29162
Bot	420	3567	3987
All	29576	3573	33149

Random Forest	Predicted		
True	Normal	Bot	All
Normal	29161	1	29162
Bot	490	3497	3987
All	29651	3498	33149

Naive Bayesian	Predicted		
True	Normal	Bot	All
Normal	27659	1503	29162
Bot	860	3127	3987
All	28519	4630	33149

Logistic Regression	Predicted		
True	Normal	Bot	All
Normal	26230	2932	29162
Bot	1103	2884	3987
All	27333	5816	33149

TABLE II
Confusion Matrices

From the confusion matrices in Table I we can see that the Decision Tree Classifier identified the most number of bots correctly, with Random Forest not too far behind and with fewer false positives. While the probabilistic approaches do identify quite a few bots, they also identify thousands of false negatives, drastically lowering their Recall.

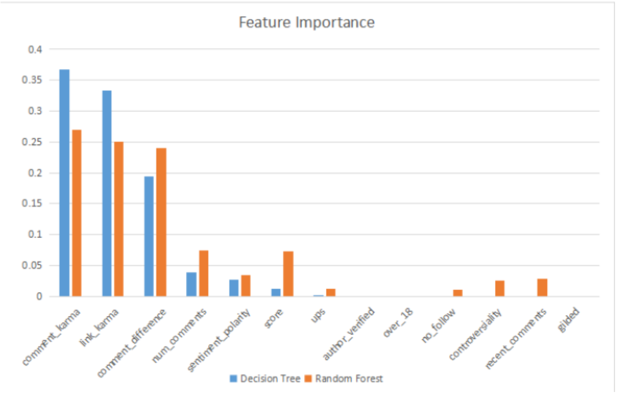
B. Important Features

The features deemed the most important by the tree classifiers essentially allow them to distinguish between bots and users better. The top three by some margin were comment_karma, link_karma and comment_difference.Both comment_karma and link_karma are Reddit exclusive features and the best way to earn Karma, and as such, their effectiveness is reasonable. comment_difference, however, is one of the features we extracted from users' recent comments with the assumption that bots tend to spam messages often. How high up on the graph this feature is verifies our sentiment.

Unlike the other Reddit exclusive features, we were surprised to see that gilded was absolutely of no importance. Our assumption was that since gilding another user requires people to use real money to buy them awards, this could be a strong feature to identify users. However, after some research, we think that people committed to passing their bots off as users are willing to buy them such awards so they appear more genuine. Posts or comments with multiple awards bought this way could seem far more legitimate, allowing them to make malicious links seem verified. Sentiment didn't have the impact we thought it would either - this will likely have more of an impact when it comes to distinguishing between good and bad bots.

Decision Tree		Random Forest	
Feature	Importance	Feature	Importance
comment_karma	0.367	comment_karma	0.271
link_karma	0.334	link_karma	0.255
comment_difference	0.195	comment_difference	0.248

TABLE III
IMPORTANT FEATURES



C. Performance Metrics and Dataset Splits

Table III compares the above metrics to identify which classifier is the most effective at distinguishing bots from users. It also investigates the effects of using different splits of the database as training and testing sets.

It is fairly straightforward to conclude that the tree classifiers are more effective all around, with the decision tree narrowly edging out the random forest. Due to the massive amount of false positives from Table I, the probabilistic classifiers have significantly lower accuracy. This is because probabilistic approaches are not as effective when one of the binary classes is rare, which relatively speaking, bots are. In such cases, they tend to become biased and give less than desirable results, as shown in Table III.

Also noticeable is that the more the data used for training, the better the results. However, while there isn't too much of a difference when the metrics are high, like for the tree classifiers, the drops are a lot more apparent for the probabilistic approaches, which are shown to drop to 54% and 41% accuracy at a 55/45 split for Naive Bayesian and Logistic Regression respectively.

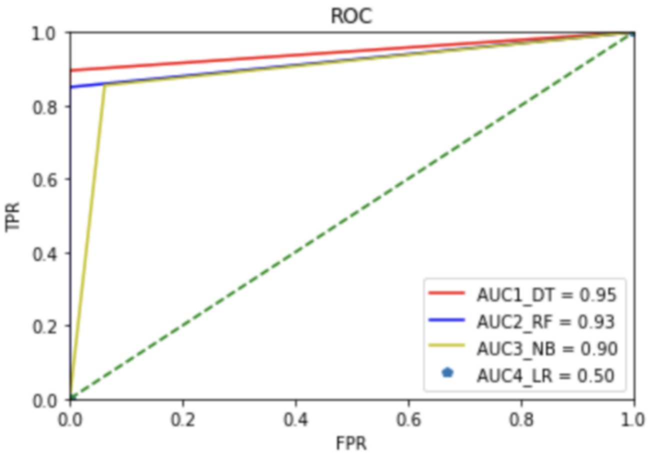
We also have the AUC of ROC, which determines how good our models are at separating bots from users. Decision Tree clocks in at 0.95, which is the same as BotOrNot - this is a very high number and shows extremely good separability. Random forest isn't too far behind at 0.93, and neither is Naive Bayesian. However Logistic Regression, which is most prone to bias of our classifiers is at 0.50, which makes it basically unsuitable for this task.

Classifier	Accuracy 70/30	Accuracy 60/40	Accuracy 55/45
Decision Tree	0.9871	0.9854	0.9646
Random Forest	0.9851	0.9827	0.9612
Naive Bayesian	0.9287	0.9193	0.8831
Logistic Regression	0.8783	0.8623	0.8426

Classifier	Precision 70/30	Precision 60/40	Precision 55/45
Decision Tree	0.9983	0.9914	0.9894
Random Forest	0.99	0.9859	0.9802
Naive Bayesian	0.6754	0.6198	0.5437
Logistic Regression	0.4959	0.4644	0.4177

Classifier	Recall 70/30	Recall 60/40	Recall 55/45
Decision Tree	0.8946	0.8813	0.8692
Random Forest	0.8771	0.8647	0.8566
Naive Bayesian	0.7843	0.7389	0.682
Logistic Regression	0.7234	0.6692	0.6119

TABLE IV
ACCURACY, PRECISION, RECALL WITH DIFFERENT DATASET SPLITS



D. Efficiency Metric - Preprocessing and Runtime

Our model takes about 3 minutes per 10,000 rows to process. Given that our dataset is around a 100,000 rows, preprocessing takes about half an hour.

As far as runtime goes, Naive Bayesian is as expected the fastest of the classifiers and Random Forest is far slower than the rest - about fourteen times slower than decision trees. This essentially means while decision tree has a slight edge in the performance metrics, it is far better efficiency wise and is thus the better classifier.

Preprocessing Time/10000 items	**∼178s**
Algorithm	**Runtime(s)**
Decision Tree	0.2534
Random Forest	3.5336
Naive Bayesian	0.0718
Logistic Regression	0.7825

TABLE V
EFFICIENCY METRIC - RUNTIME

E. NLP Experiments

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of

	Prec	Rec	Acc	AUC
Our Approach	0.998	0.895	0.987	0.95
BotOrNot	-	-	-	0.95
Supervised Bot Detector	-	-	0.96	1.00
Reddit Pattern Detector	-	-	-	-
Reddit Troll Detector	0.88	0.95	0.96	-
Network Based	-	-	0.87	-

TABLE VI
COMPARISION WITH EXISTING WORKS

artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can. It can be used for many things, we have used it for sentiment analysis and comment difference.

We used TextBlob and word2Vec packages to do this - our results are from the TextBlob package as it's results made sentiment one of the better features for the random forest classifier. Comment difference on the other hand was consistently one of the top features regardless of which package was used. While the other extracted features had some effect on classification, they were far less impactful.

1) Comparison with Existing Works: Comparing our results with other SOTA research on similar subjects, we can see that we are either on par with or better in terms of performance metrics. Unfortunately none of them mentioned any efficiency metrics.

While the Supervised Twitter Bot Detector does show an incredible AUC of 1.0, their approach is laid out over a small dataset of 3000 entries, and have mentioned that their approach fails if the database is any larger.

IX. CONCLUSION

In this paper we showed that our bot detector can successfully distinguish between real users and bots with reasonable accuracy (90%). When comparing probabilistic and tree classifiers, our data tells us that tree classifiers are far better for binary classification when one of the events is relatively rarer. Among the tree classifiers, while performance-wise both the decision tree and random forest classifiers were close, the efficiency metric made it clear that decision tree was the superior classifier for this task as it is about seven times faster.

We also verified that network based features are not as important on Reddit while they are a primary feature when it comes to bot detection on social networks like Twitter/FB. Comment Difference, an extracted feature, was a key feature when it came to detecting bots - our assumption for it's effectiveness is the fact that bots often resort to spamming and thus have similar comments. Changing the NLP methods we used to extract this didn't show great variance in results, however.

Our method could be used to identify bots in propaganda prone subreddits like r/Politics.

X. LIMITATIONS AND FUTURE WORK

While we checked the sentiment of comments using NLP which gave some interesting results, a concrete way to distinguish between 'good' and 'bad' bots is something that needs to be added - this could be extended to classify users as 'good' and 'bad' as well. Using more advanced NLP packages like GloVe to do this could yield better results.

A user-friendly UI like Botornot's would help people with no knowledge of coding to use our model, which would greatly enhance accessibility. Speaking of which, our bot classifier is limited to Reddit, if the principles were to be combined with those of network based platforms for a multi-platform bot detector, something which does not currently exist, it would not only be accessible to people from every platform but it would also be incredibly useful.

XI. RESEARCH QUESTIONS

A. Why hasn't there been too much work specializing in Reddit bot detection when this has been established in other popular social networks?

- Social networks like Twitter/Facebook are organized around users (user to user subscription)
- This means network based bot detection approaches are the best ways to detect bots on such platforms
- Reddit is organized around topics (user to topic subscription) and as such these methods do not work on Reddit bots
- While readily available features can be used to identify bots on Reddit (like Karma), our experiment proves that it is more efficient to use extracted features like comment difference
- Datasets aren't as easily available unlike on platforms like Twitter
- While popular, compared to Twitter/ FB/ Instagram/ YouTube/ TikTok/ Snapchat, Reddit has only a fraction of users - work on other platforms may be more appealing

B. Which feature was best used to identify bots on Reddit?

- We find that comment_karma and link_karma (karma received from upvotes on a post) are the best features to identify bots on Reddit
- These are both inbuilt features and are exclusive to Reddit
- comment_difference, which is one of the modified features we added using NLP packages, is also not far behind and is thus a significant feature as well
- Unfortunately our other modified features aside from sentiment_polarity which still showed some significance, had low scores
- Most other features were essentially insignificant

REFERENCES

[1] Danah M. Boyd, Nicole B. Ellison, "Social Network Sites: Definition, History, and Scholarship", Journal of Computer-Mediated Communication, Volume 13, Issue 1, 1 October 2007, Pages 210–230.

[2] J. Johnson, "Global digital population as of January 2021" Internet: https://www.statista.com/statistics/617136/digital-population-worldwide/ [Dec. 12, 2021]

[3] Brian Dean,"Social Network Usage Growth Statistics: How Many People Use Social Media in 2021?" Internet: https://backlinko.com/social-media-users [Dec. 12, 2021]

[4] Brian Dean,"Social Network Usage Growth Statistics: How Many People Use Social Media in 2021?" Internet: https://backlinko.com/reddit-usersreddit-statistics [Dec. 12, 2021]

[5] Internet "https://www.iii.org/fact-statistic/facts-statistics-identity-theft-and-cybercrime"

[6] S. K. Dehade and A. M. Bagade, "A review on detecting automation on Twitter accounts," Eur. J. Adv. Eng. Technol, vol. 2, pp. 69-72, 2015.

[7] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, Alessandro Flammini, "Rise of Social Bots" Communications of the ACM, July 2016, Vol. 59 No. 7, Pages 96-104 10.1145/2818717

[8] Davis, Clayton Allen, et al. "Botornot: A system to evaluate social bots." Proceedings of the 25th international conference companion on world wide web. 2016.

[9] Rauchfleisch A, Kaiser J. The False positive problem of automatic bot detection in social science research. PLoS One. 2020;15(10):e0241045. Published 2020 Oct 22. doi:10.1371/journal.pone.0241045

[10] A Ramalingaiah et al 2021 J. Phys.: Conf. Ser. 1950 012006

[11] Sofia Hurtado, Poushali Ray, and Radu Marculescu. 2019. Bot Detection in Reddit Political Discussion. In Proceedings of the Fourth International Workshop on Social Sensing (SocialSense'19). Association for Computing Machinery, New York, NY, USA, 30–35. DOI:https://doi.org/10.1145/3313294.3313386

[12] J. R. Ashford, L. D. Turner, R. M. Whitaker, A. Preece and D. Felmlee, "Assessing temporal and spatial features in detecting disruptive users on Reddit," 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2020, pp. 892-896, doi:10.1109/ASONAM49781.2020.9381426.

[13] E. Van Der Walt and J. Eloff, "Using Machine Learning to Detect Fake Identities: Bots vs Humans," in IEEE Access, vol. 6, pp. 6540-6549, 2018, doi: 10.1109/ACCESS.2018.2796018.