Botocracy

A BERT-powered Text Model for Authoritarian Twitter Platform Manipulation

Hemanth Bharatha Chakravarthy **Em McGlone** [*]

hbharathachakravarthy@gmail.com mdmcglone@college.harvard.edu

April 18, 2022

Abstract

As politics moves onto Twitter, authoritarians must update their set of tools used to manipulate discourse and amplify sentiments. While much attention has been given to automated "robots" that violate platform rules and post junk content, inadequate attention is given to the paid armies of real users that strongmen leaders deploy to control trending charts and manipulate voter timelines. Our thesis is that the Indian paid armies of political Tweeters are exploited to add vitriol to the platform, drive nationalistic sentiments, and attack critics—that is, change the nature of the text corpus on Twitter. Thus, we are interested in predicting whether an account is a platform manipulator or not based on their Tweets' word embeddings and user metadata. An ideal dataset collected from scraping millions of Tweets during the peak of the 2019 Indian national election campaigns is used to subset-train the Google BERT base model and then test subsequent models built on the word embeddings. Through model tuning and evaluation, we arrive at a random forest natural language processing model for Twitter platform manipulation prediction. The model is trained on the most relevant principal components of Tweet word embeddings and metadata such as likes or days existed to predict a coordination indicator. The coordination indicator is predicted as true if the Tweet is associated with Tweets that were copy-pasted by multiple unique users. On the test set, the random forest model of choice has an accuracy of 74.07%, a sensitivity rate of 19.74%, and a specificity rate of 82.55%. This paper accompanies our NLP web application product, a Twitter "botocrat" detector, that is built upon this random forest model, and is available here.

[*] Final Project for Government 52: Models (Spring 2022) with Dr. Andrew Therriault at Harvard University.

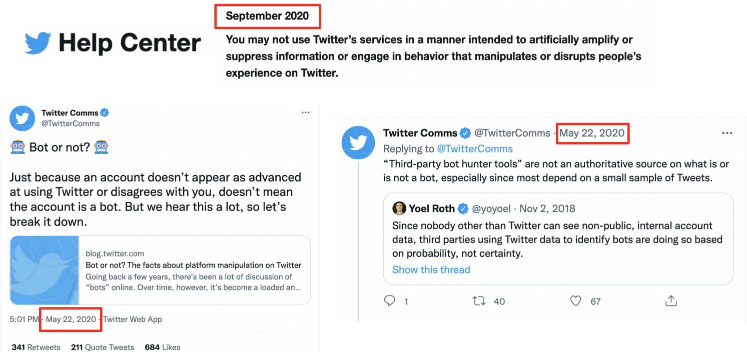
1 Overview

Control over the media has been a consistent ingredient of the autocratic modus operandi. However, as dissent and opposition move online on to social media, how do strongmen leaders still control and set narratives? Twitter is widely and increasingly becoming one of the primary forums to promote political campaigns and for public political discourse (Ausserhofer and Maireder 2013; Steffes and Burgee 2009; Chhabra 2020). There has been a significant academic study of the use of robots, especially for fake news production, on Twitter. In the Oxford Computational Propaganda Project, Howard and Kollanyi (2016) and Woolley and Howard (2018) find that the most engaging stories on social media during the UK-EU referendums and in the UK national elections were those produced by "junk news outlets" built upon robots Tweeting. Another project, Kollanyi, Howard, and Woolley (2016), finds evidence of political bots being used to change political views in the US before the 2016 election by posting misinformation and ad-hominem attacks.

Unfortunately, the study of Twitter thus far privileges easier to detect "robots," accounts run via automation without real unique users associated with them. In contrast, modern propaganda works manually on social media, with investigative journalism exposing the BJP IT Wing's online propaganda army, Tek Fog app and Google Sheets with Tweets meant to be copy-pasted by a vast base of real Twitter users, and paid Tweets (for some examples, see Sanghvi (2016), Devesh Kumar and Ayushman Kaul (2022), and Bose (2019)). Through paid armies of real, unique human Tweeters, the BJP is able to amplify their voice and manipulate political discourse on social media. Already, Twitter privileges discourse that is "simple, impulsive, and uncivil"—a phenomenon of vitriol observed even with accounts of leaders (Ott 2017). These forms of platform manipulation extend this vitriol, attacking dissenters and promoting visceral sentiments. They manipulate trends and coordinate to produce the daily timelines of everyday Indians.

Twitter Inc. itself only recently updated its privacy policy from one of responding to criticism with paltform manipulation with the idea that unique accounts coordinating is legitimate use to one of accepting something more subversive is happening here. Now, Twitter policy says, "You may not use Twitter's services in a manner intended to artificially amplify or suppress information or engage in behavior that manipulates or disrupts people's experience on Twitter" (March 2022).

If there are indeed unique users peculiarly Tweeting the same text and if these texts are qualitatively different (say, more vitriolic), we should be able to map text features onto coordination. We use word embeddings to predict the novelty of an user's Tweet: is it a Tweet that would likely be original and posted once, or does it resemble Tweets that manipulate the platform? These questions are investigated by extracting numeric vectors out of tweets and training them to predict a coordination indicator that is set at `TRUE` if the unique text was tweeted multiple times. The Google BERT based model is used to convert tweets into contextual word embeddings and principal component analysis is used to de-dimension the word embeddings (Devlin et al. 2018). Section 2 describes the data and sampling methods as well as the empirical strategy. Section 3 presents the model results and evaluates them. Section 4 compares model performance and discusses results of the top choice, a tuned random forest on BERT word embedding principal components. Section 5 concludes.



September 2020

You may not use Twitter's services in a manner intended to artificially amplify or suppress information or engage in behavior that manipulates or disrupts people's experience on Twitter.

Twitter Comms
@TwitterComms

🤖 Bot or not? 🤖

Just because an account doesn't appear as advanced at using Twitter or disagrees with you, doesn't mean the account is a bot. But we hear this a lot, so let's break it down.

blog.twitter.com
Bot or not? The facts about platform manipulation on Twitter
Going back a few years, there's been a lot of discussion of "bots" online. Over time, however, it's become a loaded an...

5:01 PM · May 22, 2020 · Twitter Web App

341 Retweets 211 Quote Tweets 684 Likes

Twitter Comms @TwitterComms · May 22, 2020
Replying to @TwitterComms
"Third-party bot hunter tools" are not an authoritative source on what is or is not a bot, especially since most depend on a small sample of Tweets.

Yoel Roth @yoyoel · Nov 2, 2018
Since nobody other than Twitter can see non-public, internal account data, third parties using Twitter data to identify bots are doing so based on probability, not certainty.
Show this thread

Figure 1: Twitter platform manipulation policy evolution

2 Empirical Strategy

2.1 Data description and feature engineering

The data is a proprietary dataset that Hemanth constructed in 2020 for a different paper. It consists of 981,154 tweets and 92 other profile attributes (e.g., location, follower count, name, description, etc.) of these 352,067 accounts across an arbitrarily chosen 11-day period from 2020-04-11 13:32:27 UTC to 2020-04-22 02:50:39 UTC. The data is constructed by scraping Twitter India trends from this time period and using the top phrases or hashtags as search queries for unverified tweets. The size of the data is used to sidestep selection bias issues.

The data is then wrangled by dropping all retweets and quote tweets, restricting the data to English tweets only. Then, the target vector `n_tweeted` is constructed by counting the number of unique Twitter users who Tweet the exact identical text. A `days_existed` feature for the metadata models is created by counting the number of days since the account was created. A `serial_dummy` feature is constructed as a boolean indicator representing if the account's screen name has the default 8-digit code generated by Twitter, showing low account maturity. The metadata features are engineered by taking the mean of the feature across those accounts who tweeted every unique body of text. The subsetted data at this stage has 90,985 rows of tweets.

Finally, the data is split into two mutually exclusive subsamples of 80 and 20 percent of the data size. From these subsamples I create a training and testing set respectively by randomly sampling from unique tweets by group of `n_tweeted`, taking the entire vector of the groups where the size of the population is smaller than the target group subample. In other words, to maintain a balance of platform manipulators and non-manipulators of varying degree in the subset training, the data is stratified by number of Tweets containing the same text and then Tweets are subsampled from here. This yields a training set of 3,556 and a testing set of 563.

2.2 Summary statistics

Table 1: Training data description (N = 3,556)

Statistic	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
n_tweeted	8.786	15.417	1	1	6	73
serial_dummy	0.146	0.277	0	0	0.2	1
favorite_count	6.620	41.309	0	0	2	846
retweet_count	2.832	15.985	0	0	1	346
days_existed	2,090.689	1,073.608	725	1,230.5	2,684.7	5,274
followers_count	1,732.106	10,530.360	0	43.3	602.5	322,884
friends_count	856.938	4,089.422	0	117.6	698.5	177,897
statuses_count	20,675.750	494,887.300	1	708.6	7,952	29,436,682
coordination	0.719	0.450	0	0	1	1

Table 2: Testing data description (N = 563)

Statistic	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
n_tweeted	10.963	16.668	1	2	8	73
serial_dummy	0.151	0.249	0	0	0.2	1
favorite_count	10.751	58.603	0	0	2	599
retweet_count	4.329	25.010	0	0	1.3	325
days_existed	1,984.870	966.336	733	1,293.9	2,379	4,996
followers_count	1,509.110	6,783.217	0	55.1	565.4	92,149
friends_count	679.645	1,526.361	0	123.8	612.8	16,435
statuses_count	8,435.914	19,347.690	1	732	7,096.6	224,932
coordination	0.822	0.383	0	1	1	1

2.3 Strategy

Four models are tuned and tested as explained in Figure 2. These are a logistic regression, a ridge regression, a SVM model, and a random forest model.

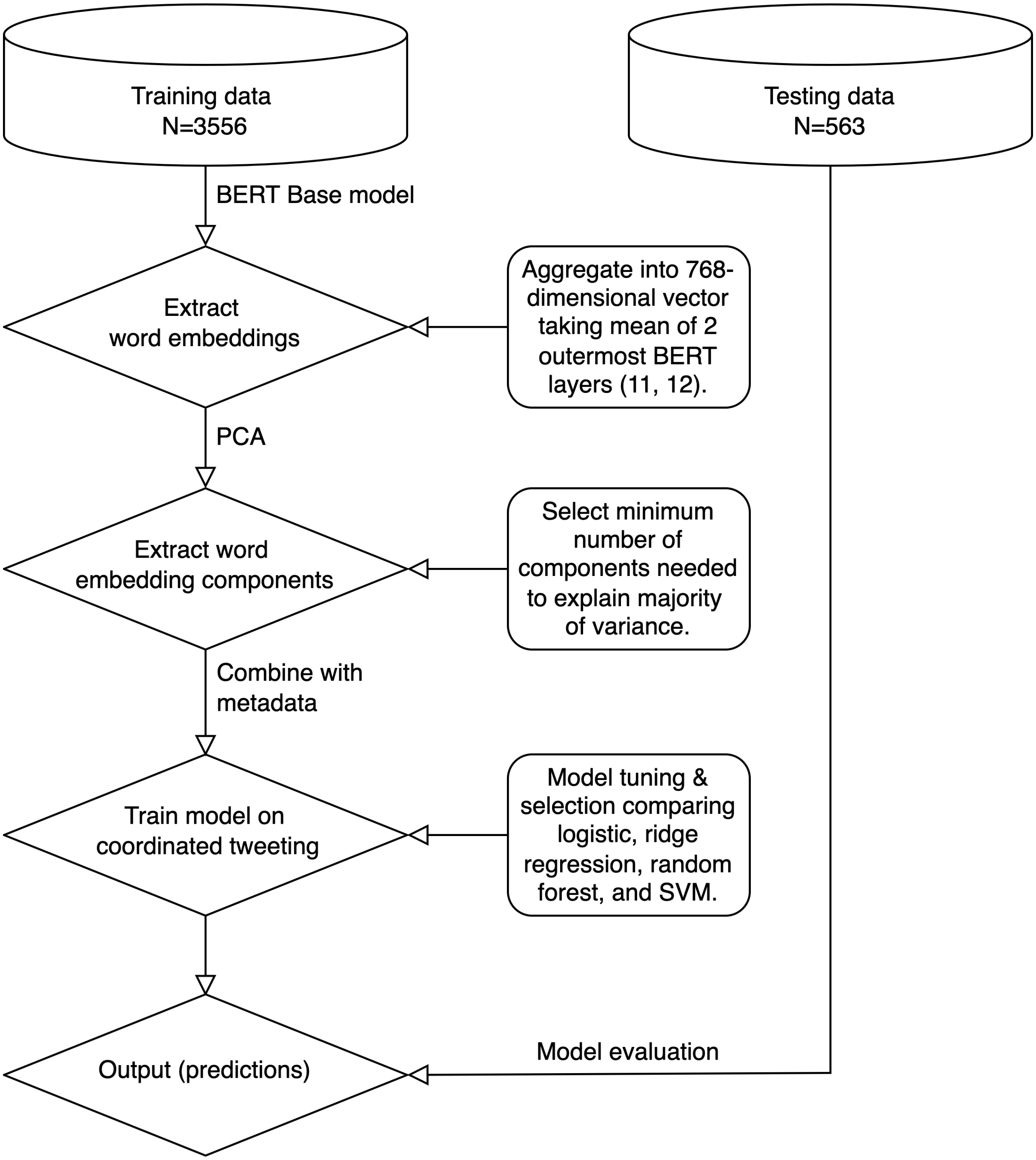
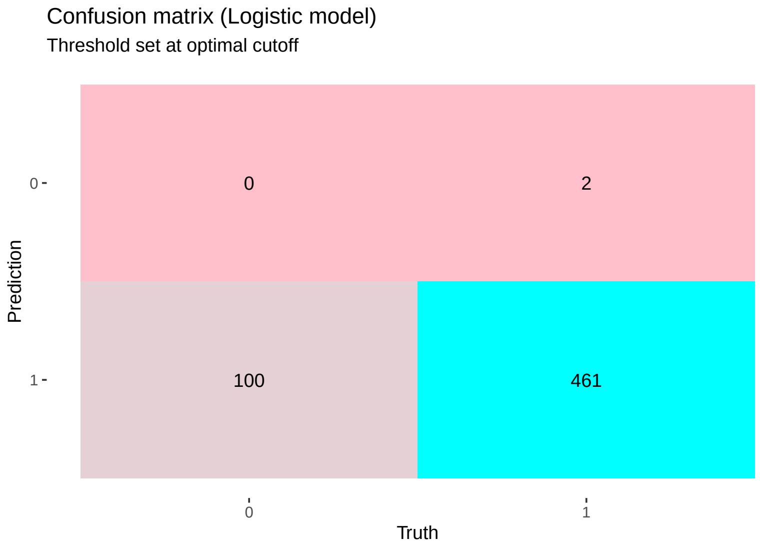

Training data
N=3556
Testing data
N=563
BERT Base model
Extract
word embeddings
Aggregate into 768-dimensional vector taking mean of 2 outermost BERT layers (11, 12).
PCA
Extract word
embedding components
Select minimum number of components needed to explain majority of variance.
Combine with metadata
Train model on
coordinated tweeting
Model tuning & selection comparing logistic, ridge regression, random forest, and SVM.
Model evaluation
Output (predictions)

Figure 2: Empirical Strategy

3 Results

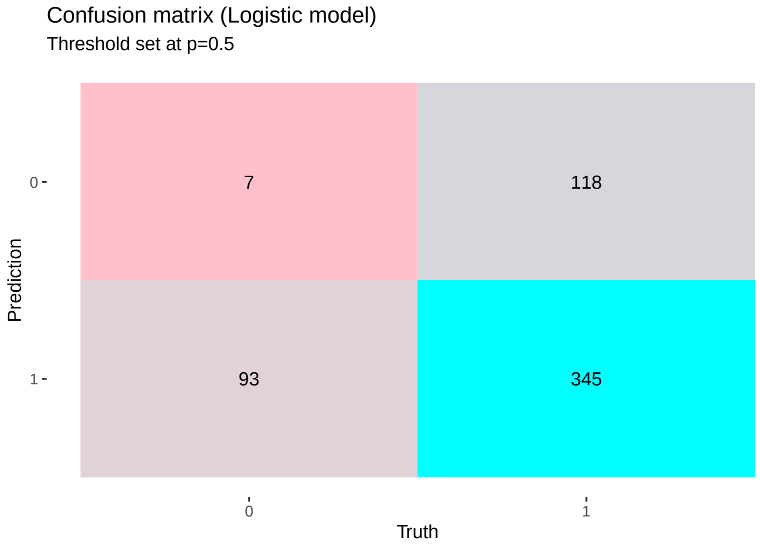
3.1 Logistic regression

The first logistic regression is trained on a target vector that is set as 1 if the `n_tweeted` > 1 and 0 otherwise. This model and all subsequent models include the metadata features alongside 38 components of the BERT word-embedding layers 11 and 12. The metadata features include the by Tweet text means of a screen name serial number dummy, favorites count, Retweet count, days existed, friends count, followers count, and statuses count. The logistic regression model performs reasonably well and has a sensitivity rate of 74.51%, a specificity rate of 7%, and a total misclassification error rate of 37.4% but if we set the optimal probability threshold, this falls to 18.12%. However, this "improvement" stems from the model almost always predicting 1. The ROC curve studies this further. The mean squared error is 0.375 on predicting the 0 or 1 dummy. Keeping in mind the right skew of the underlying distribution we work with, this is a good but rudimentary place to start. There is no need to drop down to a categorical target vector and lose richer information stored in `n_tweeted`.


Confusion matrix (Logistic model)

Threshold set at optimal cutoff

	0	1
0	0	2
1	100	461

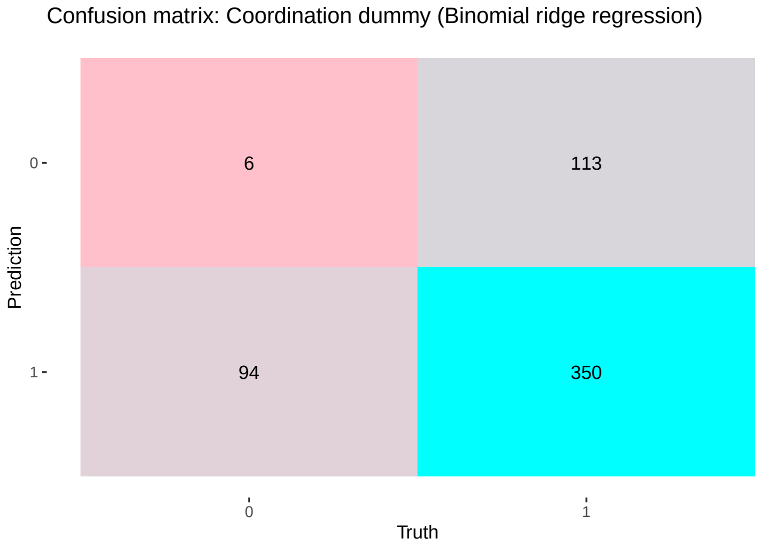

Confusion matrix (Logistic model)

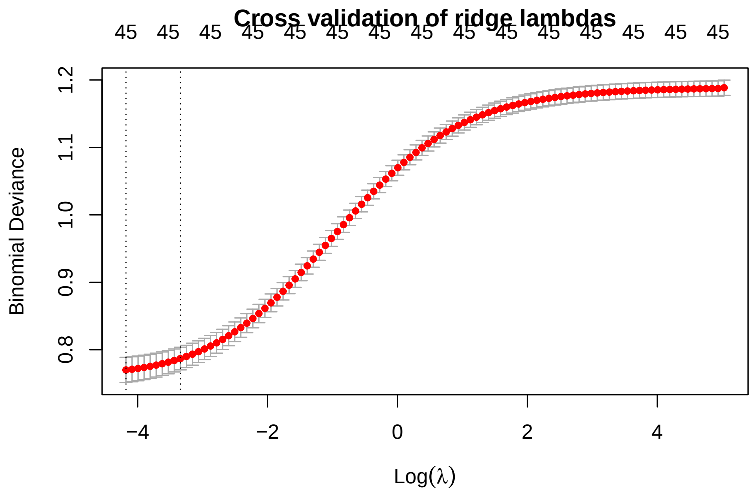
Threshold set at p=0.5

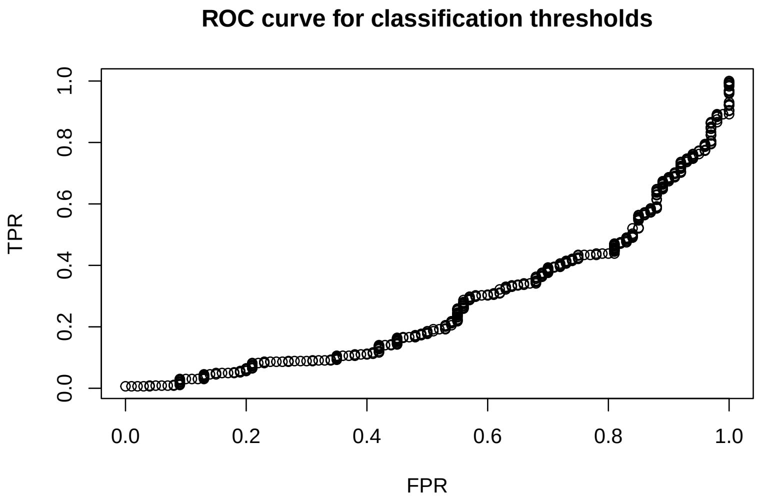
	0	1
0	7	118
1	93	345

3.2 Binomial ridge regression

We use the `GLMNET` package in R, built at Stanford, which provides extremely efficient methods for performing lasso and elastic-net regularized general linear models. We deploy this and perform a binomial ridge regression against the coordination dummy and test it. The model has a binomial deviance of 1.79, a total misclassification error rate of 36.77% (yielding an accuracy rate of 63.23%), sensitivity rate of 5.04%, and a specificity rate of 7.89%, and thus does worse than the simple logistic model when tested on a classification task. On the test data, the model has a R-squared of 0.3152 and the area under the ROC curve (AUC) is 0.271. The optimal lambda curve and the ROC curve for different classification thresholds are in the figures below. The ROC curve does worse than the 45 degree line for most low probabilities but convexly rises in the end.

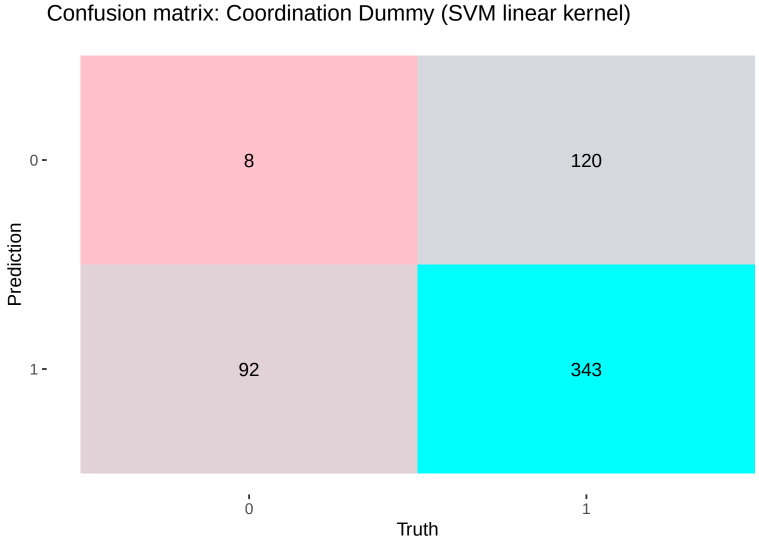


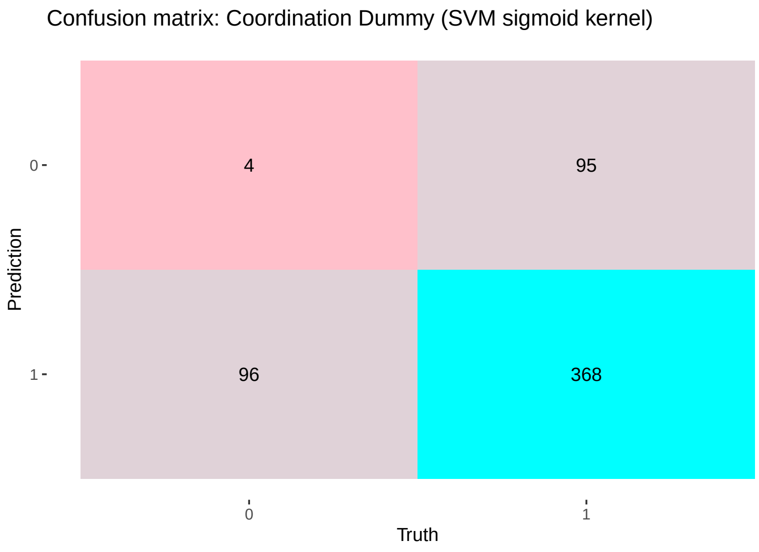


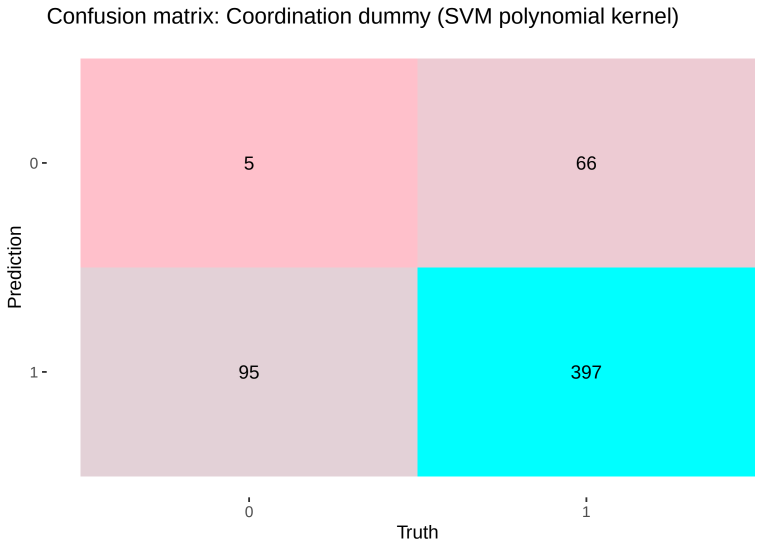


3.3 Support vector machine (SVM)

A C-classification SVM model is deployed on the same formula, and the top three kernel choices are shown as confusion matrices. The best performer is a polynomial kernel, which has an overall accuracy rate of 71.4%—a total misclassification error rate of 28.6%, a sensitivity rate of 7.04%, and a specificity rate of 8.07%.




Confusion matrix: Coordination Dummy (SVM sigmoid kernel)


Confusion matrix: Coordination dummy (SVM polynomial kernel)

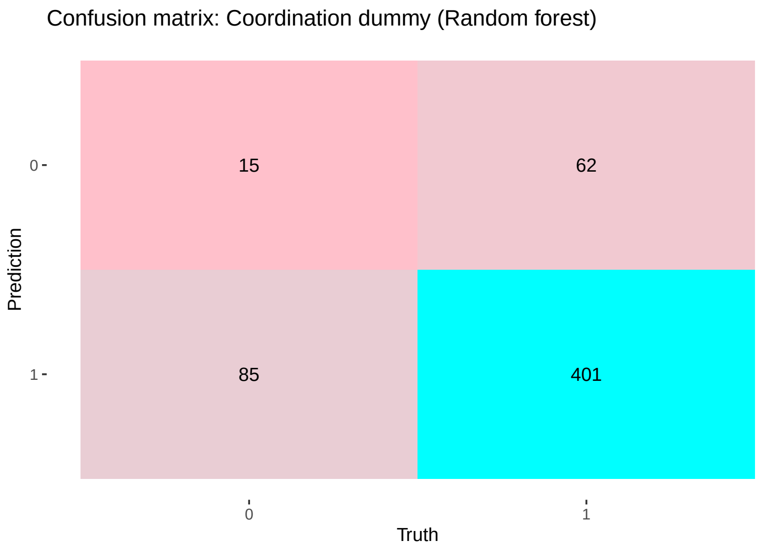
3.4 Random forest

3.4.1 Model tuning

The default random forest model sets optimal number of features to try per iteration at 23 with 10-fold cross-validation for an accuracy rate of 86.7% and a Kappa of 0.661 (indicating substantial agreement). By grid searching across features per iteration values from `1:100`, we arrive at the best optimal `mtry` of 24. Similarly, the optimal maximum number of nodes is set at 28 after grid searching 10:30. The accuracy of 800, 1000, and 2000 trees is similar and 1000 trees is chosen based on Kappa difference.

3.4.2 Evaluation

On the test set, the random forest model has a total prediction accuracy rate of 74.07% (at a total misclassification error rate 25.93%), making it the best performing model of those evaluated. It has a sensitivity rate of 19.74% and a specificity rate of 82.55%.



4 Model Evaluation

	Statistic	***Logistic***	***Ridge***	***SVM***	***Random forest***
		(P>0.5)	***(Optimal λ)***	***(P(x) kernel)***	***(Tuned)***
1	*Accuracy*	0.63	0.62	0.71	0.74
2	*McNemar p-value*	0.1	0.21	0.03	0.05
3	*Sensitivity*	0.06	0.05	0.07	0.19
4	*Specificity*	0.79	0.79	0.8	0.83
5	*Prevalence*	0.22	0.21	0.13	0.14
6	*Detection rate*	0.01	0.01	0.01	0.03
7	*Detection prevalence*	0.18	0.18	0.18	0.18
8	*Balanced accuracy*	0.42	0.42	0.44	0.51
	Model rank	**3**	**4**	**2**	**1**

Thus, we arrive at a random forest natural language processing model for Twitter coordination prediction. Thus, the final model has an accuracy rate of 74.07% and a Kappa of 0.017.

5 Conclusion

There are a few obvious limitations with the current modelling approach that can be further improved. While the corpora of text used to subset train BERT appears sufficient to us, a next version of this product would be capable of handling multilingual Tweets given that it is meant to be deployed in as multilingual a country as India and given BERT's powerful ability to handle multilingualism. Second, further investigation is needed into the loss of information from aggregating the 11th and 12th layers (which are actually the 10th and 11th layers—penultimate and one before) by taking a mean and into the loss of information from then de-dimensioning the word embeddings using PCA. It is also plausible that a Naive Bayes Classifier, a Gradient Boosted Tree, and a Neural Net might be plausible classification models to test in the future. In conclusion, we are satisfied with the novelty and relevance of our question—the ability to use Tweet text (alongside metadata) to predict platform manipulation rather than automated users—and with the accuracy of our best model at 74%.

References

Ausserhofer, Julian, and Axel Maireder. 2013. "National Politics on Twitter: Structures and Topics of a Networked Public Sphere." *Information, Communication & Society* 16 (3): 291–314.

Bose, Meghnad. 2019. "BJP Minister Criticises Modi Govt by Copy-Pasting from Edited Doc." *TheQuint*. https://www.thequint.com/elections/social-dangal/bjp-minister-criticises-modi-govt-copy-pastes-edited-doc.

Chhabra, Radhika. 2020. "Twitter Diplomacy: A Brief Analysis." *Observer Research Foundation Https://Www. Orfonline. Org/Research/Twitter-Diplomacy-a-Brief-Analysis-60462/# _Ftnref1*.

Devesh Kumar, and Ayushman Kaul. 2022. "Tek Fog: An App With BJP Footprints for Cyber Troops to Automate Hate, Manipulate Trends." *The Wire*. https://thewire.in/tekfog/en/1.html.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. "Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding." *arXiv Preprint arXiv:1810.04805*.

Howard, Philip N, and Bence Kollanyi. 2016. "Bots,# StrongerIn, and# Brexit: Computational Propaganda During the UK-EU Referendum." *Available at SSRN 2798311*.

Kollanyi, Bence, Philip N Howard, and Samuel C Woolley. 2016. "Bots and Automation over Twitter During the First US Presidential Debate." *Comprop Data Memo* 1: 1–4.

Ott, Brian L. 2017. "The Age of Twitter: Donald j. Trump and the Politics of Debasement." *Critical Studies in Media Communication* 34 (1): 59–68.

Sanghvi, Vir. 2016. "I Am a Troll: Inside the Secret World of BJP's Digital Army." *Business Standard India*, December. https://www.business-standard.com/article/beyond-business/i-am-a-troll-inside-the-secret-world-of-bjp-s-digital-army-116122801182_1.html.

Steffes, Erin M, and Lawrence E Burgee. 2009. "Social Ties and Online Word of Mouth." *Internet Research*.

Woolley, Samuel C, and Philip N Howard. 2018. *Computational Propaganda: Political Parties, Politicians, and Political Manipulation on Social Media*. Oxford University Press.