Network based Analysis of Reddit Troll Activity

Arka Sanka
University of Texas at Austin
arkasanka@utexas.edu

Sunny Sanyal
University of Texas at Austin
sanyal.sunny@utexas.edu

Abstract

In recent years, malicious political players have used trolls to disperse misinformation and fake news across all social media platforms, including Reddit. Based on the Reddit transparency report 2017, Russian trolls have been active during the 2016 US presidential elections. While trolls shape public opinion for various political discussions on Reddit, there is little understanding of how they operate and how their strategies evolve over time. This paper studies trolls' behavior across large political networks based on Reddit data. Our approach is two-fold; first, we build a random forest classifier that identifies active troll users in the r/politics subreddit. Second, we study the Russian trolls and the newly identified trolls for various network setups and perform both static and temporal analysis. Based on our experiments, we reveal some interesting characteristics of these trolls.

1. Introduction

Reddit is an online discussion platform that enables users to build communities to discuss various topics. Reddit relies on content propagation by users. Recently, there have been concerns about malicious entities' rising interference through paid disingenuous users (trolls) on Reddit's political discussions. These trolls peddle misinformation that corrupts honest political discourses on Reddit. For instance, Reddit has listed 944 suspicious Russian origin trolls who might have influenced the US elections in 2016. Recently trolls have also conducted a coordinated campaign to forward the Chinese Communist party's propaganda on Reddit[1]. Therefore it has become increasingly critical to study the dynamics of political trolls on various social media platforms. This paper studies troll behavior in various network settings for the 20 different subreddits, most affected by the Russian trolls. Particularly we seek answers to these questions; 1) how the presence of trolls impacts the overall network behavior? 2) How political troll operates? 3) how troll strategies evolve over time?

This is a challenging problem to solve as officially Reddit has only published a list of 944 Russian trolls that acts as ground truth in this paper. However, the distribution of these trolls across various subreddits in different periods (say years/months) is unknown. Moreover, the trolls are in a tiny minority in Reddit; hence the data is extremely sparse. Approaches used in Bot detection cannot to directly extended for detecting trolls as bots and trolls operate differently. Moreover, identifying bot is more straightforward as many bot users have "bot" in their names and display bot type behavior [1] more often than trolls. Trolls, on the other hand, are mostly sly human agents that conceal their identity by mimicking regular users. Hence it makes it altogether more interesting yet challenging to identify trolls.

To address this problem, we have developed a random forest classifier to detect trolls. We train our classifier on 2016 r/politics data and deploy our model on 2018 r/politics data. We have identified several political trolls in 2018 Reddit data, and we analyze their behavior to answer the research questions mentioned above posed in this paper. The main contributions of this work are a) the approach for troll detection based on random forest classifier. b) Analysis and visualization of the trolls in various network settings.

The article is organized in the following fashion. Section 2 discusses the background and prior work. Section 3 presents the details of the overall approach. In section 4, we discuss the implementation and the network analysis. Finally, in section 5, we conclude our paper by highlighting the significant findings and future work.

2. Prior Work

Several studies [3] have investigated bot and troll detection schemes for various social media platforms. However, the detection and analysis of trolls for the Reddit platform remained a much-understudied problem. This may be because social media platforms like Twitter and Facebook

[1] buzzfeed article

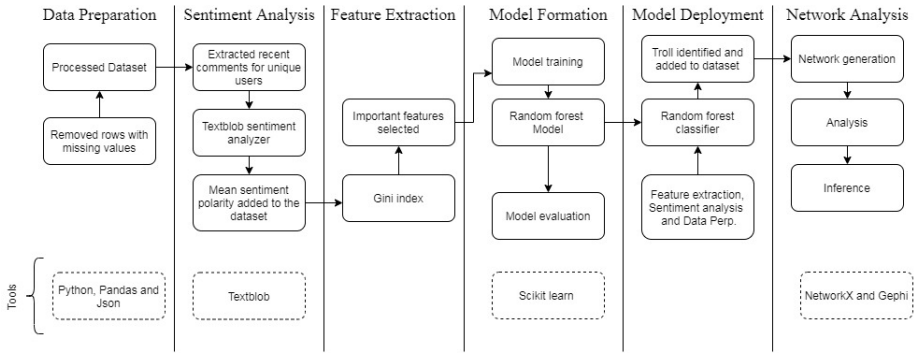


Figure 1. Overview of our approach

have more labeled troll data available to the public than Reddit. Reddit troll detection is performed by [4] [5]. Both these approaches have used individual comments to classify the users into troll and normal. [4] have used a random forest classifier to check individual comments. [5] on the other hand, uses a very powerful deep learning technique, BERT word embeddings, with RCNN and CNN to classify troll comments. Unlike these papers, our work classifies troll users based on their overall recent activities, where recent comments are just one feature among other several critical features. Moreover, we have achieved higher accuracy than [4].

Zannettou et al. [6] have analyzed the influence of state-sponsored trolling and its impact on politics. This paper has analyzed trolls from Twitter, Reddit, and other platforms based on geolocation, traits, and language. Unlike [6], our approach is two-fold as we detect and analyze Reddit trolls.

3. Approach

This section discusses the classification algorithm used, including dataset, sentiment analysis and features extraction. The main approach is illustrated in Fig. 1.

3.1 Dataset

The 944 Russian trolls have made 21321 posts, and a large number of these posts and comments are made in 20 different subreddits [6] between 2015 and 2018. The classifier dataset[2] consists of 99494 user comments from the r/politics subreddit (year 2016) with 15028 bot comments, 6551 troll comments, and 77915 regular users' comments. The dataset is labeled with both troll and bot users, besides other attributes. The classifier data is divided into training and test datasets in a ratio of 70:30, respectively. We have used a second dataset (deployment dataset) that belongs to r/politics (September 2018) as a deployment dataset. This dataset has 1.9 million user comments; moreover, the users in this dataset are not labeled; here, we do not have the ground truth for trolls. We also used a third dataset (troll dataset) containing the comments and the posts made by the 944 Russian trolls between 2015 and 2018. We used this dataset for the analysis in section 4. The first two datasets are pre-processed to eliminate the automoderator bot, comments on comments, and rows with missing values. We have used classifier dataset for training and testing our classifier and deployment dataset for discovering more trolls. Finally we use the troll dataset and deplyment dataset for network analysis, temporal analysis and word frequency analysis.

[2] google drive

3.2 Sentiment Analysis

Sentiment analysis is performed using the Textblob library that returns the polarity of a sentence. Polarity lies between [-1,1], where -1 implies a negative sentiment and 1 implies a positive sentiment. Textblob uses a lexicon-based sentiment analysis approach where every word in a text message is assigned an individual score, and the sentiment polarity is the average of all the sentiments. Interestingly, TextBlob has semantic labels that help with fine-grained analysis. For instance, emoticons, exclamation marks, etc. For each unique user, we extracted 20 recent comments (reverse chronologically) and computed the average sentiment polarity. This average sentiment polarity is assigned to every user and is a critical feature for the classifier.

3.3 Random Forest Classifer

Random forest classifier, as the name suggests, consists of a large number of individual decision trees that operate as an ensemble. The decision trees map input features to output classes based on specific rules. The tree model is trained recursively by extracting the critical features that decrease the overall Gini impurity. Each decision tree returns a class; moreover, the random forest classifier returns the class (the

final classification output) that is the mode of the classes (in our case) or the mean prediction of the individual trees. Typically decision trees are prone to overfitting and sensitive to minor perturbations or changes in input. The Random forest approach combines several such trees in an ensemble; hence it effectively circumvents the demerits found in the individual decision trees and is highly robust. Refer to Figure 2.

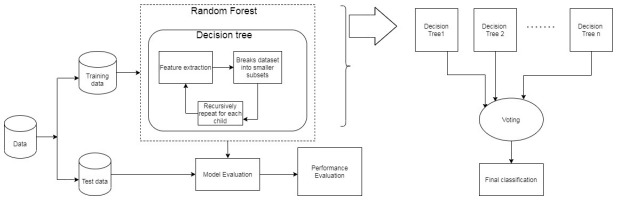


Figure 2. The machine learning pipeline for Random forest classifier

The number of decision trees, commonly known as estimators used in a random forest classifier ensemble, impacts the model performance. Although it may not overfit yet, a higher number of trees adversely affects the computation time without any significant improvement in the performance. We have chosen the number of estimators as 20 after experimenting with the trade-off between performance and computational run time. Our model is based on bootstrapping. We have employed the scikit learn random forest classifier to build the classifier.

3.4 Feature Extraction

We consider the Gini index or Gini impurity as a metric for feature selection. Gini index measures the degree of probability of a particular feature being wrongly classified when chosen randomly, as shown in eq. 1.

$$Gini = 1 - \sum_{i=1}^{n} p_i^2 \tag{1}$$

The Gini index takes values between 0 and 1; 0 implies a pure classification with all nodes belonging to a single class, and 1 implies that elements are randomly distributed among various classes.

We use comment score(difference between up-votes and down-votes for a comment), comment sentiment polarity(as explained in section 3.1), up-votes, gilded to extract features. For each user, we compute the following features for no more than 20 recent comments or fewer,

- **Number of comments** is the number of recent comments made by a user.
- **Average responses** is the average of number of comments on target posts the user commented on.
- **Average sentiment polarity** is the average of comment polarities obtained using Textblob sentiment analyzer.
- **Minimum sentiment polarity** is the average of comment polarities obtained using Textblob sentiment analyzer.
- **Average score** is the average of comment scores obtained from Reddit data.
- **Minimum score** is the minimum of comment scores obtained from Reddit data.
- **Percentage of negative scores** is the fraction of negative comment scores obtained from Reddit data.
- **Average up-votes** is the average of up-votes from Reddit data.
- **Average gilded** is the average of number of times a user was gilded.

The implementation is highly scalable as we are using generator based python code for faster execution. An important limitation of this paper is that for deployment dataset there is no ground truth. Hence the trolls identified using our classifier may not be highly accurate. As trolling is a subjective term. The network analysis is discussed in the following section.

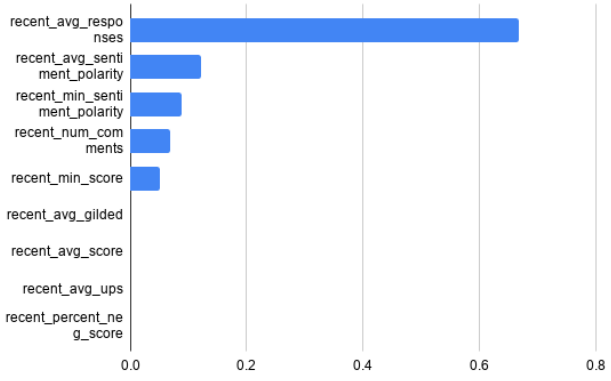


Figure 3. Feature importance visualization

4. Experimental Setup and Results

4.1 Performance Metrics for Random forest Classifier

There are three performance metrics commonly used to evaluate the performance of the model, namely, precision, recall, and F1 score. The precision is the number of true positives divided by the sum of the true positives and false positives; it is known to measure the accuracy of the classifier. The recall is defined as the of true positives divided by the sum of true positives and false negatives. Recall can be thought of as the classifier's ability to classify the full set of positives. The F1 score is the weighted average of recall and precision. The confusion matrix for our classifier is shown in Fig. 4.

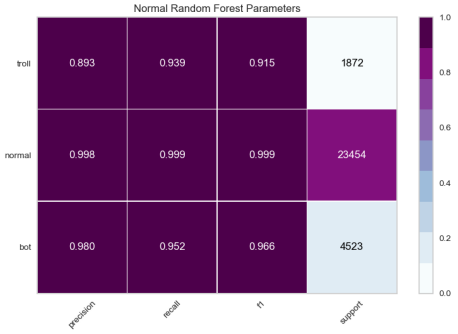
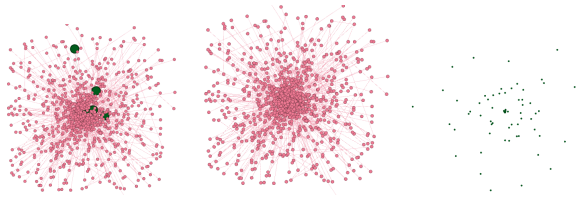


Figure 4. Class-wise precision recall and F1 score for random forest classifier

4.2 Author to Author network



(a) User and troll network, nodes=19320, edges=1715078, clustering coefficient= 1,Avg. path length=1, Avg. degree=177.5

(b) User only network, nodes=19243, edges=1707183, clustering coefficient= 1,Avg. path length=1, Avg. degree=177.4

(c) Troll only network, nodes=77, edges=9, clustering coefficient= 0,Avg. path length=1, Avg. degree=0.234

Figure 5. User to User network with trolls in green and users in pink.

After identifying trolls using Random forest classifier, we proceed to network analysis. Figure 5 represents three author networks where authors are connected if they comment on the same post. The weight of the edge is number of posts both the authors commented on. Figure 5(a) shows author to author network with both normal users and trolls included, figure 5(b) shows the author author network with normal users and figure 5(c) shows the network of trolls. Pink nodes represent normal users and green nodes represent troll users. From figure 5 (c), we can see that only 9 comments on common posts exist for 77 trolls. From this we infer that trolls are sparsely connected between themselves. Also, average node degree of the network drops by 0.1 when we drop troll users. This is because troll users have 7,886 edges with users and 9 edges between themselves. Hence we surmise that the trolls have no notable impact over the overall network properties.

4.3 Post to Author network

Figure 6 represents a post to author network where a post and an author are connected if the author comments on a post or makes the post. The weight of the edge is the number of comments made by the user on the same post. Green nodes indicate troll users, and pink nodes indicate posts. The network is based on the troll dataset containing the posts and comments data of all the 944 trolls for 2015 to 2018 and the new trolls. This network shows that only half of the total trolls have made one or more comments. We also observe that trolls are sparsely connected among themselves, and hence it bolsters our inference that the trolls have a negligible impact over the network properties.

Typically, a Post to author graph has fewer nodes than the author to author graph because, for each comment on a post, only one link is formed between the author of the comment and the post node, whereas in the author to author graph, edges are created between the author of the current comment and all the authors who commented on this post. Also, the number of nodes increases in a post to post graph as we create additional nodes for posts. The difference in edge definition also affects the average clustering coefficient of these two networks. In the author to author network, all the authors who comment on a post are connected with each other. This results in a very high clustering, as we see in figures 5 (a) and 5 (b). Whereas in a post to post network, connections are only possible between a post and an author. Hence, authors that comment on the same post(neighbors of post node) are never connected between themselves, and all the posts a specific author commented on(neighbors of author nodes) are also never connected between themselves. This lack of connections between neighboring nodes results in a very low average clustering coefficient.

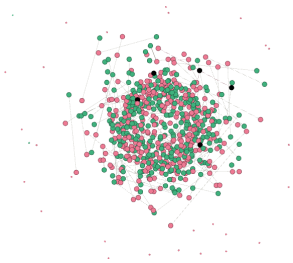
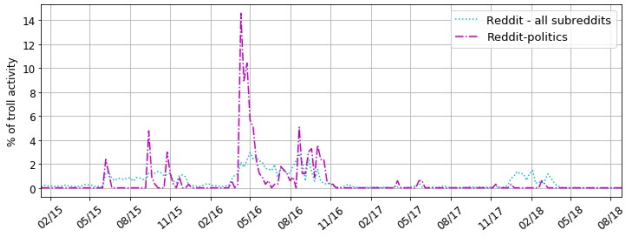


Figure 6. Post to User network with trolls in green and posts in pink, nodes=643, edges=306, clustering coefficient= 0, Avg. path length=1, Avg. degree=0.952

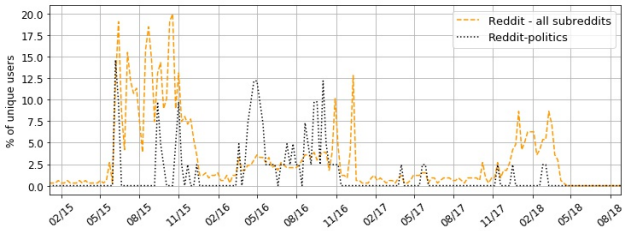
4.4 Temporal Analysis

To observe the time-varying patterns of the Russian trolls and the new trolls (77 trolls from the September 2018 dataset), we plot the troll activity (normalized volume of Reddit posts and comments) against various time frames from 2015 to 2018. For the temporal analysis, we have considered the top 20 subreddits affected by the trolls, as discussed in section 3.1. For instance, in Fig. 7a, we plotted the weekly troll activity between 2015 and 2018. Here we observe big spikes in the overall troll activity on weeks close to the 2016 and 2018 US elections. However, this figure does not tell us anything about the amount of participation done by each troll. To visualize the weekly participation of unique trolls, we plot Fig. 7b. Based on both Fig 7a and 7b, we surmise that the participation of uniques trolls are proportional during 02/2016-11/2016 and 01/2018-05/2018 (close to US elections) and irregular otherwise.

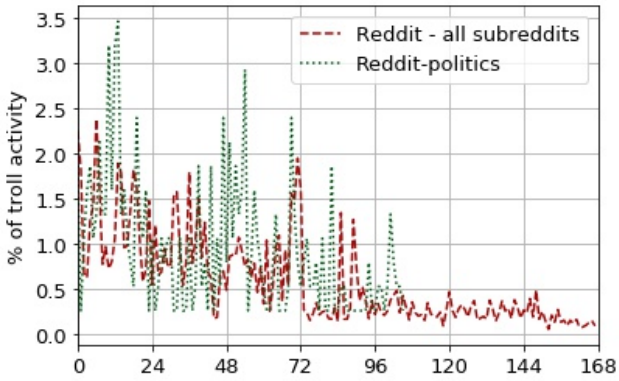
We have gone deeper to see the weekly and the daily traits of the trolls. In Fig. 7c, we plot the troll activity against the hours of the week, here we observe that the trolls are more active on the initial days of the week and less active on weekends. This hints towards the possibility that these trolls work for some agency supported by the Russian government. Next, we plot the troll activity as a function of the hours of a day. In Fig. 7d, we observe that overall the trolls are mostly active during the first half of the day, and the trolls in the r/politics subreddit are active in some parts in both halves.



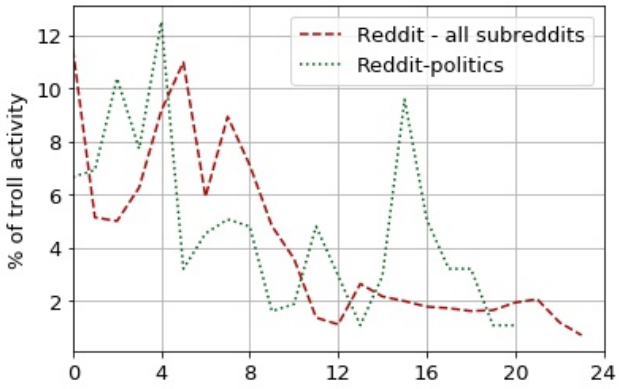
a. Unique troll activity as a function of month



b. Unique troll activity as a function of month.



c. Troll activity as a function of hours of day.



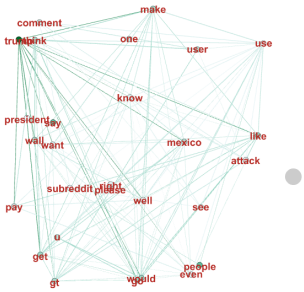
d. Troll activity as a function of hours of day.

Fig 7: Temporal analysis of trolls

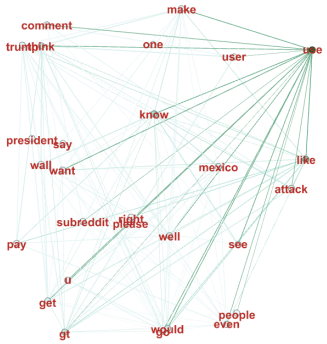
4.5 Word Frequency Analysis

We used Word2Vec [2] to analyze word representations and plot word graphs based on cosine similarity. Word2Vec takes a large corpus of text and generated vector representations in given dimension for each word. Word vectors generated using Word2Vec capture contextual information about the word and its usage in the input corpus. For this project, we use all the troll comments as input. We then extract the 100 dimensional representation of words in this corpus. To obtain word- similarity plot in figure 8, we consider words that occur more than 1300 times and edges

are formed between two words if their cosine similarity is greater than 0.6. For figure 8.a, we rank the words according to their word frequency in the word corpus. We can see that the trump campaign related words have good cosine similarity and are also frequently used by trolls. In figure 8.b, we rank the words based on their betweenness centrality. We see that many regularly used words have a high betweenness centrality.



a. The words were ranked by word-count and we observe words supporting Trump's campaign were used very often by trolls.



b. The words were ranked by their betweenness centrality in the graph.

Fig.8 Troll word usage analysis

5. Conclusion and Future work

Social media has a tremendous influence on how people perceive political news, and honest discourse will become increasingly difficult with disingenuous interference by trolls. This work identifies and studies political trolls in the Reddit platform; specifically, we have studied the Russian trolls and their influence on the large political networks. We have developed a classifier to identify trolls for Reddit data. Our experiments have identified new trolls, and the analysis has revealed three important properties of the trolls. We found that 1) trolls do not affect the network properties, 2) the behavior of trolls vary over time, and 3) the trolls have shown support for the Trump campaign. As future work, we plan to build a real-time suspected troll detection system to be used by human moderators. We also plan to analyze the trolls across other platforms such as Twitter and Facebook. We worked together on developing the random forest classifier and performing the network analysis. Arka has worked on data parsing, data preprocessing, and word frequency analysis. Sunny has contributed towards temporal analysis and drafting the posters, slides, and reports. Broadly through this project, we have learned the critical aspects of implementing machine learning and network science for large scale social media networks.

References

[1] S. Hurtado, P. Ray, and R. Marculescu. Bot detection in reddit political discussion. page 30–35, 2019.

[2] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. page 3111–3119, 2013.

[3] M. Orabi, D. Mouheb, Z. A. Aghbari, and I. Kamel. Detection of bots in social media: A systematic review. *Information Processing Management*, 57(4), July 2020.

[4] B. Punturo. Predicting russian trolls using reddit comments. 2019.

[5] H. Weller and J. Woo. Identifying russian trolls on reddit with deep learning and bert word embeddings. 2019.

[6] S. Zannettou, T. Caulfield, W. Setzer, M. Sirivianos, G. Stringhini, and J. Blackburn. Who let the trolls out? towards understanding state-sponsored trolls. page 353–362, 2019.