Party Prediction for Twitter

Kellin Pelrine[1,2], Anne Imouza[1,3], Gabrielle Desrosiers-Brisebois[1,3], Sacha Lévy[1,2], Jacob-Junqi Tian[1,2], Zachary Yang[1,2], Aarash Feizi[1,2], André Blais[3], Jean-François Godbout[3], Reihaneh Rabbany[1,2]
[1]Mila - Quebec AI Institute
{kellin.pelrine,jacob-junqi.tian,reihaneh.rabbany}@mila.quebec
[2]School of Computer Science, McGill University
{sacha.levy,zachary.yang,aarash.feizi}@mail.mcgill.ca
[3]Département de Science Politique, Université de Montréal
{anne.imouza,gabrielle.desrosiers-brisebois,andré.blais,jean-francois.godbout}@umontreal.ca

ABSTRACT

A large number of studies have attempted to determine how social media affects partisan preferences by relying on predictive models for inferring political affiliation. This task is often performed based on the content generated by the users (e.g., tweet texts), the relations they have (e.g., who they follow), and their activities and interactions (e.g., which tweets they like). We provide a comprehensive survey and an empirical comparison of these current practices, in order to compare their signal strength and performance in predicting the party affiliation of users based on their online activities on Twitter. We also propose our own three approaches, which are competitive with or outperform state-of-the-art methods, and let the practitioner select from a wide range of data types that all give strong performance. Finally, we conduct extensive experiments on different aspects of these methods, which can provide insights for both applied and methodological research.

KEYWORDS

social network, political party, graph, NLP, evaluation

1 INTRODUCTION

Knowing the political orientation of social media users is critical to many research areas. It is fundamental to polling and predicting election outcomes. It is a first step in measuring political conflicts in online societies [13, 19], and necessary for many strategies to mitigate polarization's harms [3, 47, 51, 54]. It is a significant factor in the spread of misinformation [31, 40], and understanding its impact on social media usage is of paramount importance in a context where democratic norms are declining.

Increasing partisan animus, or what scholars call affective polarization, has been confirmed not only in the United States [49], but also in other countries, such as Canada [25], the United Kingdom [22], and France [8]. For the most part, these studies have concluded that partisan tensions have dramatically increased in recent years by looking at different survey questions to capture the levels of dislike and distrust between people who identify with opposing parties [24]. Several explanations have been put forward to account for this trend, most notably partisan sorting or economic inequalities [16, 17, 37], but increasingly researchers are identifying social media as one of the root cause of this phenomenon [4, 21, 27, 30]. To the extent that online communities influence the level of partisan polarization within the electorate, it becomes critical to understand how political party identification can affect the behavior of social network users. Indeed, before we can determine with certainty whether social media influences the levels of partisan polarization in the broader public, we need to be able to identify the partisanship of users and their ties to political parties.

However, despite its importance, there is no single definitive method or procedure for predicting a user's political party affiliation from online activities. This paper aims to take a step in that direction, including but not limited to, a survey of existing methods, extensive experiments evaluating different approaches, and new methods that deliver state-of-the-art performance with more coverage and more easily obtainable data.

Party prediction for social media users is well-studied (e.g., [39]), and approaches have been developed based on many types of data, such as text, followership, and interactions (e.g., retweets). But each of these methods is usually evaluated on a unique dataset, without any attempts to compare its overall performance with other existing approaches or baselines. The collection process and resultant dataset difficulty vary widely, with different types of users (e.g., politicians vs. general public), diverse filters, and often entirely different time periods and topics. Thus, there are few comparisons within each paper, and evaluations of the different approaches are generally inaccurate due to the diverse data sources.

To solve this problem, we first survey the literature, paying particular attention to not only the method and its aggregate performance numbers, but also the characteristics of the data used. We summarize the key findings in Tables 1 and 14, and elaborate in detail in Section 2.

Next, we collect data from approximately 15,000 Twitter users who discuss American politics during the time period directly before and after the 2020 US election. We use these users to test the performance of seven different methods, including three approaches of our own that cover all types of interactions and text. In this way we confirm quantitatively the major impact of different datasets and how challenging the task of comparing existing models is. Moreover, our study provides the missing thorough comparison necessary to evaluate the performance of these models for the first time.

We also do extensive ablation and other experiments validating different components of our approaches. We confirm that our methods deliver strong performance and can do so from a wide variety of data types. Our experiments can also help future researchers better understand all the different factors involved in this task, such as which data types are most informative and applicable, how to structure interaction data, and differences between the public and

politicians. We hope that in this way our experiments will lead to improved methods and measures of online partisan conflict.

In summary, our main contributions are:

- To show, through both an extensive survey and quantitative experiments, that party prediction results in the literature are extremely challenging to compare. This can result in suboptimal or even unstable foundations for downstream research.
- To fill this hole in the literature by taking strong methods from this survey and adding our own data and approaches.
- To determine their level of performance. Our three new approaches are all competitive with or outperform the state-of-the-art methods identified from the literature. Together with our encompassing experiments, they open up new options and data types for applied practitioners and new insights for future methodology research.

2 BACKGROUND AND RELATED WORK

There is a lack of consensus in the literature as to what factors explain the heightened levels of polarization in the United States [8, 18]. This is especially important today, as democratic norms appear to be eroding across many Western democracies [15, 32]. Several explanations have been put forward to account for this trend, such as the choice media [45], partisan sorting [55], economic inequalities [52], and demographic changes [7]. For many, however, the responsibility for this change rests with social media usage, especially since the COVID-19 pandemic, which saw many of the more traditional political activities move online. This conclusion is perhaps surprising since there is no clear evidence in the literature that social media has an impact on political polarization, either within or outside the digital world [24]. One thing is certain, however, in order to determine if social media outlets have an effect on partisan conflicts, we need to have a reliable measure of online polarization. If users are not polarizing, then there is a much lower chance that social networks will influence the broader levels of partisan conflicts observed in the general public. But if they are, we can begin to focus on determining whether polarization is spreading offline and develop tools to promote more civil exchanges within those communities.

There is surprisingly little consensus among scholars on the right approach to classify users according to their partisan affiliation, let alone measuring polarization [11]. In a comprehensive review, Tucker et al. [49] highlighted different gaps in the literature to understand the relationship between social media usage and political polarization. However, the authors failed to mention the problems associated with classifying partisanship and measuring partisan conflicts [30]. This is surprising, as there does not seem to be a widely-accepted, scalable, and easily implementable method to perform these tasks. Without such an approach, how can we determine if polarization is increasing or decreasing across social networks?

Most techniques for measuring online polarization begin by classifying users according to their partisan affiliations. In the United States, for example, users can either be supporters of the Democratic or the Republican parties; in some cases, users can also be classified according to their ideology, as liberals or conservatives. If the measured distance (or opposition) between these two group is high, than polarization is said to be elevated, and vice-versa.

But how accurate are these different classification approaches? Since the goal of this paper is to develop a standard measure of party affiliation from online activities, comparing our own approach to other competing techniques is a necessary step to validate its performance. For the purpose of this study, we reviewed 20 papers and identified eleven approaches that offer a unique method to classify users according to their partisan affiliation. We have selected these approaches because they represent a broad range of estimation techniques and because they offer strong performances to predict the partisan affiliation of social network users, with the most promising of these methods tested against different gold standards to establish their accuracy.

Each classification technique is based on Twitter data and uses different features to assess the partisan affiliation of users. Table 1 summarizes the main approaches and explains their limits. Table 14 reports similar information, but for the other remaining papers. We selected the models for inclusion in Table 1 based on their higher level of prediction accuracy (as reported by the authors). Even so, the accuracies range from 66% to 97%, meaning that there is a wide range of classification error, depending on which approach is used.

The table is divided into nine columns. The first one represents accuracy (1), which indicated how well the party predictions perform. We also report how each of the approaches works by indicating what features are used in the classification tasks. These are: (2) media outlets; (3) network activities; (4) network relations; or (5) content.[1] For instance, some approaches use media outlets shared by the users to infer partisan affiliation [2, 36, 46, 48]. Others rely on the structure of networks to infer partisan leanings [6, 12, 20, 42, 57], while some rely instead on other types of network activities, such as retweets and mentions [2, 13, 20, 36, 42, 46, 57]. Finally, several approaches focus on the content of messages, by either looking at the text or the hashtags used to represent a partisan side, like #Democrats or #Republican [12, 13, 42, 46]. Note that most of these approaches combine more than one feature to make their predictions. Only Barberá et al. [6] relies exclusively on a single feature in their party prediction model.

The next column (6) indicates if the classification task was performed within a subsample of "politicians" or the general "public"; that is, accounts from well-known politicians and parties (see for example [6, 20, 46, 57]), or from a broader set of Twitter users. We also added a column (7) related to the test size (i.e., the number of users an approach was evaluated on) to validate the accuracy of the method which varies from around 500 [46] to approximately 40,000 [6, 36]. Finally, the last two columns indicate whether the code used to make those predictions is publicly available (8), and our own evaluation of the difficulty level of the prediction task and data which the method was tested on (9).

To determine this overall level of difficulty of the classification tasks, we proceeded in four steps. First, we examined whether the

[1] The media outlets approach infers the user's ideology based on the political leaning of the media shared in online messages, while the network activities approach relies on the retweets and mentions systems of users. The network relation approach considers the fellowship of users to determine their partisanship. In other words, party affiliation is predicted based on who they follow. Lastly, the content approach includes the political leaning based on words and terms such as hashtags and keywords to indicate users' ideology.

Table 1: Survey methods to predict party affiliation. Here we report for each paper: Accuracy (1), whether they used Media outlets (2), Network Activities (retweets and mentions) (3), Network Relation (followership) (4), Content (words and hashtags) (5), as well as if they consider Public, Elite or both (6), their test size in terms of number of users on which the ideology is inferred (7), if their code is available (8) and finally the level of difficulty (9), as explained more in the text.

Papers	Acc.	Media	Activity	Relation	Content	Type	Size	Code	Difficulty
Conover et al.[2011]	94.9%		✓		✓	Public	1,000		Medium
Barberá[2015]	78%			✓		Both	42,008	✓	Hard
Rheault and Musulan[2021]	90.9%	✓	✓		✓	Politicians	505	✓	Easy
Luceri et al.[2019]	89%	✓	✓			Public	38K		Medium
Colleoni et al.[2014]	79%			✓	✓	Public	10,551		Medium
Pennacchiotti and Popescu[2011]	88.9%		✓	✓	✓	Public	10,338		Medium
Stefanov et al.[2020]	82.6%	✓	✓		✓	Public	806		Medium
Gu et al.[2016]	66.3%		✓	✓		Both	1,200		Medium
Badawy et al.[2018]	91%	✓	✓			Public	29K		Medium
Xiao et al.[2020]	96%		✓	✓		Both	20,811	✓	Hard
Preoţiuc-Pietro et al.[2017]	97.2%			✓	✓	Public	13,651	✓	Medium

models were tested on a general sample of Twitter users or if they focus politician accounts instead. Second, we looked at the number of filters applied to the data before the classification task. Almost all of the models are tested on a sample of tweets related to political topics (i.e., datasets selected from policy-related keywords or hashtags). However, we also consider the application of additional filters (e.g., selection on polarizing keywords, user activity levels, or user location) as a factor that could potentially increase the difficulty level. Third, we examined the data collection approach by considering the number of features included in the classification task; methods that integrate several features are assumed to be more complex in their implementation. Fourth, we considered how each approach was tested, such as on the type of data and on the sample size of users and tweets. Here, the larger the sample or test size, the more complex the classification task. From this, we have labelled all approaches as either "easy", "medium", or "hard".

Models tested exclusively on politicians are assigned an "easy" difficulty level since these types of users are generally easier to classify [11]. Only one paper falls into this category [46]. A "medium" level of difficulty implies that the method applies to a broader set of users from the general public, and contains more complex features [2, 12, 13, 20, 36, 42, 48]. An example here would be Stefanov et al. [48] who classify general public users based on the ideological leanings of the news articles they share on Twitter. For the classification task to be hard, the test size must be large and validation must be done on both the public and an elite group of politicians. This category also implies that the method used is sophisticated, either because of the test sample size or the features used. Two papers fall into this category [6, 57]. For example, Barberá et al. [6] compute the ideological position of users based on who they follow on Twitter, and use an item-response model to determine users ideology.

Table 1 confirms that the most successful approaches use network activities to make their predictions, the only exceptions here is Gu et al. [20] with a a 66.3% accuracy score. On average, the different approaches use between two and three features to make their prediction. However, having more features does not necessarily imply greater accuracy. Only four of the eleven techniques obtain a classification success rate greater than 90%; one does an "easy" category task [46], two medium [2, 13, 44], and one hard [57].

In this study, four state-of-the-art approaches are used as benchmarks to compare the performance of our own classification approach. These are Barberá [5], Preoţiuc-Pietro et al. [44], Liu et al. [34], and Xiao et al. [57]. We selected these papers for several reasons, namely, the performance levels, the availability of the code, and the implementation and compatibility of the method. We also selected these approaches as baselines because they rely on a different features and data types, which can then be compared to our own method. Note, however, that these levels of accuracy are determined according to different validation tests, depending on the type of model or data used in a paper, which makes comparison difficult. This explains why one of the main objectives of this study

is to compare these models under the same test conditions. We describe below each of these approaches in greater details.

The first approach, from Xiao et al. [57], relies on learning embeddings from sparsely-labeled heterogeneous graph data to predict the users' party affiliation. It simultaneously learns the links structures and relationships between likes, friends and follows, replies, mentions, and retweets. This sophisticated model achieves one of the highest performance level of all the papers reviewed with 96%. It is also the only approach that achieves above 90% on a "hard" category task.

The second approach, from Barberá et al. [6], is one of the most commonly used measure to estimate the levels of partisanship of users at the individual level [10, 26, 43]. It relies exclusively on the structure of networks in a latent space model to estimate their ideology. Because of its simplicity, this approach, which is based on network relations, is easier to implement and can be used to estimate the party affiliation of several million users. The overall accuracy of 78% obtained when classifying a set of approximately 42,000 users whose party registration records were available is not as high as other methods that combine more than one type of features. But this is a hard dataset because there are few restrictions on the users.

The third approach, from Preoţiuc-Pietro et al. [44], focuses on the linguistic features of Twitter messages. The authors first identify groups of ideologically loaded political words in order to infer users' party positions from the text contained in their tweets. The authors then use a linear regression algorithm to determine the party identification. Because Preoţiuc-Pietro et al. [44] test their model on different types of users, the general accuracy of the model varies greatly, ranging from 62.5% to 97.2%. The model performs best when predictions are based on text content, and when the data is limited to users who follow politicians' account (either Democrats or Republicans).

Finally, the fourth approach by Liu et al. [34] also uses a content-based model to predict users' ideologies. However, their model relies on a pretrained language model which uses news articles reported by media with known partisan biases. This pretrained model is applicable to various datasets, ranging from Twitter and Youtube users, to Congressional speeches and news articles. Their Twitter user classification accuracy is low, under 50%, so they are not included in Table 1. However, this is likely in large part because they complexified their model by classifying Twitter users using a 3-way classification (left, right, and center) instead of a binary split between the left and the right or Democrats and Republicans.[2] Still, they are able to reach accuracy levels above 85% when predicting the ideological labels (left vs. right) of newspaper articles based on the political words they employ. Therefore, we hypothesized that this model would have strong performance on a two-way Twitter classification task (and as shown later, this is borne out in our experiments).

In order to determine which of these four approaches obtains the highest performance, we present in this paper our own partisan classification models that combine network activities, relations, and content features on test size datasets of approximately 2,000 general public Twitter accounts, 900 politician accounts, and 300 users matched to voter records. The performance of our models will be assessed by testing and comparing them with the four models discussed above on the same set of users.

3 PROPOSED METHOD

3.1 Data

We curated 2 main datasets, summarized in Table 2 and discussed in further detail below. In this table, the users are the authors of the posts, and the activity columns represent the total occurrences of their respective activity within these posts. For example, if the numbers of posts and retweets were the same, it would imply that every post was a retweet. The relations (friends and followers) were collected separately from the posts.

Mass public data. We first collected around 1% of real-time tweets using Twitter's streaming API, that included one of the following US election related keywords: [JoeBiden, DonaldTrump, Biden, Trump, vote, election, 2020Elections, Elections2020, PresidentElectJoe, MAGA, BidenHarris2020, Election2020]. This created a dataset (not shown in table) with approximately 350 million tweets and 20 million users. Out of these, we sample 20 thousand users from those with keywords indicating their party affiliation in their profile description: [conservative, gop, republican, trump, liberal, progressive, democrat, biden].[3]

This initial dataset had limitations: the timespan covered was not as extensive as desired, particularly lacking the days surrounding the January 6^{th} Capitol Attack. And due to the 1% collection process it could miss many of a user's tweets. Therefore, using a combination of the normal and academic Twitter APIs, we retroactively (late 2021) retrieved all of the 20K users' tweets. Twitter does not allow access to tweets of deleted, protected, or suspended users. Thus, we were able to successfully retrieve the tweets of about 15k users.

In April 2022, during the process of retrieving "Likes" data for all Public users, we also collected data on the cause of missing accounts. Combined with our profile labels (see section below on classification), we examine the distribution of these missing users in Table 3.[4] We see that a strong majority are Republicans. We hypothesize that many of these users left around the January 6^{th} capitol attack, either voluntarily or during the subsequent wave of suspensions that included the account of Donald Trump.

Politicians Data. We collected all tweets, retweets and replies from 995 elite accounts linked to the public and personal Twitter accounts of the United States representatives (433), senators (99),

[2]The model used by Rheault and Musulan [46] also classifies users according to a multi-way classification. Since they are analyzing the 2019 Canadian federal election, the model classifies users according to their party affiliation (Liberal, Conservative, New Democatic Party, Green, or Pepople's Party of Canada) based on a test-size dataset of 505 politician accounts.

[3]Due to the database sampling procedure, approximately 1000 of the sampled users had instead stemmed versions of these keywords (e.g. "progress" instead of "progressive"). Due to the relatively small percentage, and the fact that all of our evaluation data was human labeled, this should not impact the conclusions. Nonetheless, we plan to correct this in an updated version of this paper.

[4]We omit from the table four users whose retrieval failed for other miscellaneous reasons, e.g. Twitter API crash. We also omit 67 users whose profile label was corrupted due to a data storage error. These errors have no correlation with the dimensions examined in the table, and the users are too few to impact any conclusions drawn from it.

Dataset	Posts Collected				Activity				Relations			
	Start	End	Posts	Users	Hashtag	Mention	Retweet	Quote	Friends	Uniq. Friends	Followers	Uniq. Followers
Public	2020-10-01	2021-02-28	6,771,120	15,042	1,630,132	10,538,082	4,676,725	447,455	17,393,800	4,335,983	12,207,636	4,947,396
Politicians	2020-08-01	2021-01-17	156,562	995	162,121	212,074	83,920	54,630	1,724,222	863,509	33,424,367	9,342,089

Table 2: Statistics on the collected datasets, including the posts, the activity extracted from them, and the relations of the users.

Party	Republican	Democrat
Suspended	1,824	439
Deleted	1,857	543
Private	186	267

Table 3: Distribution of users whose data could not be retrieved retroactively. The majority are Republican, possibly users who left around January 2021, i.e., the wave of suspensions that included Donald Trump.

as well as vice presidential and presidential candidates (8) using Twitter's Search API.[5] We call this the Politicians dataset.

We further collect relation and/or like data on the users in our two datasets, as explained below and summarized in Table 2.

Relations. We collected friends (accounts a given user follows) and followers for all accounts available at collection time. For Public, followers were retrieved during May 2022 and friends between mid July and mid August 2022. For Politicians, both friends and followers were retrieved during January 2021. Because some users have an extreme numbers of friends or especially followers that is not feasible to retrieve, we cap the total retrieved per user at 5,000 friends and followers for normal users. For politicians we cap followers at 100,000 and retrieve friends fully (maximum friends a politician has in our data: 135,389).

Likes. We collected tweets liked by users in our Public dataset in April 2022. We capped retrieval at 1000 maximum per user, leading to 6,014,206 total likes.

3.1.1 Training and Test Sets. In order to train and evaluate our models, we begin by classifying users according to their party affiliation and ideology based on the description they provide on their *user profile*. This will create a training set with noisy labels ($\sim$ 90% accurate) that we will use to train the classifiers introduced in Section 3.2, which are based on activity and/or relations.

First, for each dataset, we classify users as "Republican", "Democrat" or "unknown" based on identifiers in their profile description. For "Republican" we use: [conservative, gop, republican, trump]. For "Democrat" we use: [liberal, progressive, democrat, biden]. We label users as "Republican" ("Democrat") if the description contains at least one of the Republican (Democrat) identifiers and does not include any of the Democrat (Republican) identifiers. The rest of the users remain as "unknown." Note here that we combine concepts related to both the ideology and the partisanship to label Democrat and Republican users [34].

This is a "weak" classification because user keywords may not match their actual party affiliation or ideology. For example, instead of a president name indicating support, they could say "I hate Trump" or "I hate Biden." In order to validate the overall performance of these labels, we asked two political science MA students (co-authors of the paper) to classify, on the basis of the same information (that is, the description provided in the user profile) 60 users from the politicians dataset and 1000 general public Twitter users from each party. This "strong" classification either confirms the weak labels, or indicates the presence of a coding error. Note that while in most cases an incorrect weak Democrat label indicates that the user is in fact a Republican (or vice versa), a small number of these users can also be independent or apolitical. After comparing the weak with the "strong" labels, we found that users in the Politicians dataset are generally more politically involved and hence the simple keyword search is very accurate. However, for other users, the accuracy was lower, with only around 70% of the weak labels matching the strong (manually coded) labels.

Therefore, we used the strong labels to train a classifier to generate more accurate labels. We randomly split the strong-labeled data 75% into training set and 25% into test set. We also add into the training set 525 Parler[6] users that were labeled with the same process above. With this data we finetuned a RoBERTa-large [33] model (a pretrained language model; we provide a more detailed overview in Section 3.2.2), to predict the party each user is closest to from their profile description.We report the results in Table 4.

Dataset	Counts		Accuracy	
	Rep.	Dem.	Rep.	Dem.
Politicians	102	65	98.0%	98.5%
Public	6,375	8,375	87.0%	90.5%

Table 4: Number of users with explicit party/ideological keywords in their profile description (on the left). Accuracy of our profile label classifier based on manually labeled sample of public users and actual politician parties (on the right).

We then use this profile classifier to make a prediction on the remaining users in our sample. The result is labels that are still "weak," but much more accurate than the simple keywords matching. We use these profile classifier labels to help train the approaches discussed below on large number of users when manual labelling is not feasible. After training the profile classifier, we set aside our 2000 total "strong" human-labeled users for evaluation to make sure comparison of these approaches is reliable.

We illustrate the process of building from the small set of manually labeled users to increasingly general users in Figure 1. We next discuss how we go from users with profile classifier labels to more general users.

3.2 Methods

To deliver good performance on all data types, we provide three methods, which we reference based on their core component: GCN

[5]Some Members of Congress have more than one social medial account (e.g., one personal and one official account). In this case, we collected information for all of the relevant accounts.

[6]A social networking site similar to Twitter but more associated with far-right users.

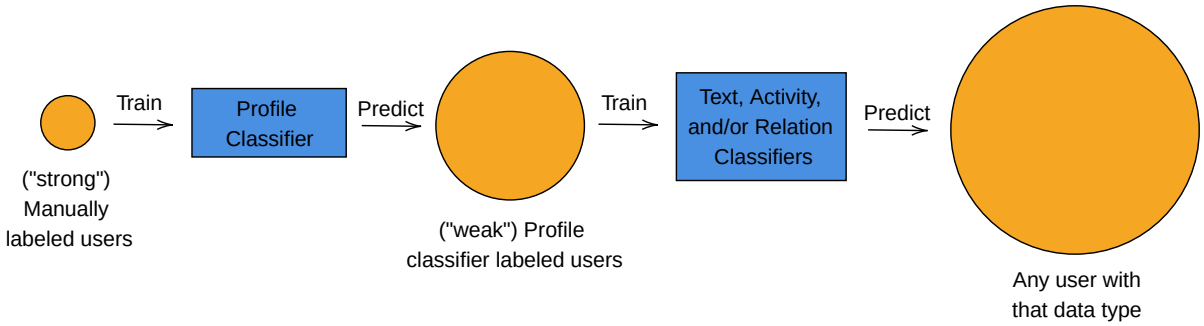

Figure 1: Starting from a small set of manually labeled users, we use the profile classifier to expand to a larger set of users (our Public dataset), and then use that to build classifiers that can classify any user with text, activity, and/or relations.

[29] (a simple graph neural network model), Label Propagation [58] (a non-neural graph method based on "propagating" label information to neighboring nodes in the graph), and RoBERTa [33] (a pretrained, transformer-based language model).

3.2.1 *Preprocessing.*

GCN. We structure the relationships between users as graphs, where the nodes are users and edges represent interactions between them. We consider two perspectives on which interactions link people together:

- Direct links. For example, person i mentions or retweets person j. This corresponds to the graph's adjacency matrix A, a (0,1)-square matrix, where $A_{ij} = 1$ if i links to j.
- Projected (indirect) links. For example, persons i and j both mention the same person k, even though i and j may not mention each other directly. Formally, this corresponds to the projected adjacency matrix $A' = A^T A$.

Direct links are intuitive, however, we find empirically (Table 9) that projected links are more informative for our GCN approach. Therefore, aside from the experiment that explicitly tests this, all our GCN models use the projected graph.

Next, in order to get accurate predictions and measurements, some amount of user activity is needed—if a user is not connected to anyone else in the network, no graph model will give a meaningful prediction. Furthermore, to compare different approaches equitably, we need a set of users that are active in all the particular ways that the different methods require. For instance, to compare [5] (which is based on follow relations) with [44] (which is based on tweet text), we need users that both follow others and post tweets.

We therefore apply the following filters unless otherwise noted. First, for each individual interaction type, we train only on the top 50% of most active users for that relation according to raw counts. This filter is not applied to test data. Second, we require that all users in the test set have all the interaction types needed for every method we consider. This filter is not applied to the training data.

Label Propagation. This method also works on graph data. But here we find the reverse of GCN: the direct graph performs better than the projected one. So in this case we always use the direct graph, again excepting the experiment explicitly investigating this. Otherwise the preprocessing is the same as for GCN.

RoBERTa. This is a contextualized text-based model. In order to provide maximum context, we concatenated each user's tweets into chunks following the order in which they were posted. Since the positional encoding scheme of the model limits the length of input to a fixed number of tokens, we start a new chunk each time the previous chunk goes above the length limit of the model. Consequently, these chunks correspond to sequences of text similar in length to a paragraph, though since they are composed of multiple tweets connected only by a shared user and sequential post time, they may or may not represent a single thought. Any tweet that would have too many tokens to fit into a single chunk is truncated and placed into a chunk of its own. As a result, no tweet is split between chunks.

Note we also use this same preprocessing for the POLITICS model described in the previous section [34], which has a similar architecture.

3.2.2 *Core Model.*

GCN. Our next step here is to construct one embedding per user and interaction type—i.e., a learned vector representing each user's realized interactions of that type. We use the titular GCN [29]. We generally use a single-layer version that is semi-supervised, with an unsupervised link prediction task and a supervised node classification task using profile classifier labels on training nodes. In one experiment (Table 9 we test a two-layer version. In two experiments (Tables 8 and 11) we use a fully unsupervised version.

This model supports using node features as well as the graph structure. We conducted preliminary experiments on using text embeddings for this, but found it did not improve performance. This parallels [57], which likewise found that node features did not help their GNN-based model for party prediction. We instead use a uniform random vector with dimension 100 to initialize the embeddings.

We train for 1000 epochs with the Adam optimizer [28] with PyTorch default parameters (learning rate = 1e-3).

Label Propagation. This approach [58] "propagates" labels between connected nodes (users). It is a more classical, non-neural method, and rather than an embedding, it directly produces a prediction. It has two parameters: the number of iterations of the propagation process, and the rate α at which new label information replaces the old one.

We use the semi-supervised version of this algorithm, i.e. seeding train nodes with profile classifier labels. We first tested it on projected graphs, where we found performance decreases consistently and monotonically with more iterations irrespective of α value (specifically, tested iterations in [1,2,3,4,5,6,8,10,15,20,25], with fully tested alpha values [0.1,0.5,0.9] and partially tested [0.03,0.97]). When using a single iteration, this method corresponds to taking the majority vote of the neighbors' labels, and changing α has no effect.

We later found that performance is better on the direct graph, specifically with two iterations and $\alpha = 0.5$. With 1 iteration on this graph setup performance is poor, while with more iterations performance remains constant (accounting for margin of error) or decreases. The direct graph takes much longer to run, so due to time/computation constraints we were not able to test more values of α in this setting. We plan to revisit this in future work. In this paper, except where stated explicitly, we report results from the best performing version with two iterations and $\alpha = 0.5$.

RoBERTa. The core here [33] is a language model based on BERT [14], with changes to the pretraining process to improve performance. It uses a transformer [50] architecture and is pretrained on 160GB of text. We examine two versions, RoBERTa-base (125M parameters) and RoBERTa-large (355M parameters).

These models are pretrained without a sequence classification task. Consequently, despite RoBERTa models having a [CLS] token like BERT and many similar language models, the representation of this token is not pretrained and thus unsuitable for direct use in downstream tasks. In one of our experiments (Table 10 we tested the model without the finetuning needed to properly learn this token. Therefore, to improve comparability, in all experiments with RoBERTa instead of [CLS] we use the mean of all individual word embeddings from the final output layer.

To further improve the prediction accuracy, we fine-tuned these models with a tweet chunk classification task. First, we label each chunk in the train set according to the profile classifier label of the corresponding user. We add a fully-connected dense layer (the "classification head") to each model. We then finetuned each model end-to-end for 1 epoch using the Optax [1] implementation of the adamw optimizer [35] with learning rate 1e-5 and default weight decay strength 1e-4.[7]

Final Prediction.

GCN. Given user embeddings, we use a random forest model to make a final prediction. This setup facilitates combinations of embeddings from different data types. In particular, we first train separate, independent GCNs on each individual input type, producing one embedding per user per GCN. We can then mix and match data types as desired by concatenating the embeddings of the same user from different GCNs, before passing the combined embedding to the random forest. In this way, we compare each data type individually and different combinations of them over multiple experiments.

For the random forest, we use the scikit-learn [41] implementation with default hyperparameter settings. We train it using a train set of users with profile classifier labels, and report test results on a (fully separate) set of users using manual labels.

Label Propagation. There are no additional steps for this model; it makes a prediction directly.

RoBERTa. To get a prediction for each user, instead of a prediction on an individual tweet chunk, we first predict a label for each of a user's chunks. Then we take the majority vote, producing a single prediction.

3.3 Baselines

As noted in the related work section, we compare our approaches with 4 state-of-the-art methods from the literature.These were selected based on performance, taking into account difficulty of the data they tested on. Here we briefly summarize some of their key points and provide the details on how we implemented them:

- **Barberá** [5, 6] This item response model assigns scores to users based on who they follow. We use the Tweetscores[8] implementation, which compares a user against "elite" (politicians and media) users with pre-trained scores. We classify any user with score greater (resp. less) than 0 as Republican (resp. Democrat), which both matches intuition and gives the best performance empirically (see Appendix C).
- **TIMME** [57] This approach is based on a variation of a GCN, but the architecture is adapted to learn from multiple data types simultaneously and end-to-end. This contrasts with our GCN approach which trains an independent GCN per data type, and combines them in a separate final stage. Therefore, this design hopes to learn more complex interactions between the different data types, at the cost of being much more computationally intensive and potentially more difficult to train. Unfortunately, even when using high-end AI-specialized hardware (such as RTX8000 GPUs) and the original authors' code,[9] we found the computational burden is severe. We discuss the implementations and results further in the experiments section.
- **Preoţiuc-Pietro** [44] This is a bag-of-words style approach with a custom, specialized vocabulary of 352 politics-related words. For each user we concatenate all their tweets into one document and calculate the counts of these words. Following the original paper, we then use this feature vector as input to a logistic regression that classifies each user. We implement the logistic regression through scikit-learn [41] with default hyperparameters.

[7] When evaluated on the profile classifier labels, this setup leads to 94.4% ± 0.4% test accuracy on tweet chunks for RoBERTa-base and 94.5% ± 0.2% for RoBERTa-large. Note these accuracies for tweet chunk classification are only on weak labels, and are correlated with but not directly comparable to user classification—we report on the latter in the experiments section.

[8] https://github.com/pablobarbera/twitter_ideology

[9] https://github.com/PatriciaXiao/TIMME

- **POLITICS** [34] This model is based on RoBERTa-base, but adds political domain adaptation through training on news articles with ideology labels. Unlike RoBERTa, the [CLS] token that provides a representation of a full sequence is trained and the authors use it for their downstream tasks. So we follow this, passing the final-layer embedding of this token through a single fully-connected one-layer classification head. With this architecture, we fine-tune the model with the same setup as we use for RoBERTa-base. We also report performance of an untuned version where we obtain user embeddings from averaging the pretraining-only tweet chunk [CLS] embeddings, and classifying users using the final prediction part of our GCN approach.

3.4 Computation

Most experiments were done using RTX8000 GPUs. A number of text-based experiments were run on v3-8 TPU VMs from Google's TPU Research Cloud.[10] Graph-based models were implemented using DGL [53] and language models using HuggingFace [56] in JAX [9]. To run all models of the main comparison experiment (Table 5), excluding TIMME (which is very slow; please see details in experiments section), the roughly estimated time using 10 RTX8000 is one week.

4 EXPERIMENTS

Overall Comparison. In Table 5 we compare different versions of our approaches with the four benchmarks from the literature. For a high fidelity comparison, the test data is the same across every approach. This requires filtering for users which have all 6 interaction types considered, plus follow an elite user for the Tweetscores implementation of [5]. In total 687 of our manually labeled users meet this criteria. We split these users 40-20-40 train-validation-test. Currently we do not use this training set of manually labeled users—for supervised training we only use other users and the profile classifier labels —but we maintain it for forward-compatibility with future experiments. We repeat the random splitting 10 times, along with the rest of the model training process, and report the mean and standard deviation.

In the results, we see three main conclusions:

- **Testing different approaches on the same users, like here, is necessary for clear comparisons.** For example, if judging only by the number reported in their original paper, Barberá [5, 6] would perform over 8 percentage points worse than here. Depending on which result is being considered, none of which are close to the findings reported here, Preoţiuc-Pietro et al. [44] would perform anywhere from over 10 points better to nearly 30 points worse. But the performance of these two approaches turns out to be similar.
- **Our relatively simple approaches deliver state-of-the-art performance.** The best performance by an existing state-of-the-art model comes in at rank 12.
- **One can achieve strong performance with many different types of data.** Note that from rank 1 to 15 there is less than one percentage point difference between all of the methods used. Furthermore, there are top-15 methods in every category, meaning that one can get a strong prediction regardless what category of data one has access to. Conversely, if deciding what data to collect, one can choose an efficient, scalable option; for example, retrieving retweets may be much more efficient than retrieving friends.

[10] https://sites.research.google/trc/

Table 5: Classification accuracy of different methods. Rank is shaded from blue (best) to orange (worst).

Method Type	Method Name	Accuracy	Rank
Existing State-of-the-Art	Barberá [5, 6]	86.5 ± 1.1	19
	POLITICS Untuned [34]	81.4 ± 2.8	27
	POLITICS Finetuned [34]	89.2 ± 2.7	12
	Preoţiuc-Pietro et al. [44]	85.9 ± 1.5	23
	Timme [57]	< 88	17
	TIMME-Hierarchical [57]	< 87	18
Text	RoBERTa-base	89.2 ± 2.7	12
	RoBERTa-large	88.9 ± 2.9	16
Activity	Label Prop. Retweet	89.8 ± 1.3	3
	Label Prop. Mention	85.2 ± 1.9	24
	Label Prop. Quote	88.3 ± 1.6	19
	Label Prop. Hashtag	86.3 ± 1.5	22
	GCN Retweet (RT)	89.2 ± 1.3	12
	GCN Mention (@)	80.4 ± 1.5	28
	GCN Quote (QT)	84.1 ± 1.6	25
	GCN Hashtag (#)	81.9 ± 2.1	26
	GCN RT+QT	89.5 ± 1.5	7
	GCN All Post Activity (RT+QT+@+#)	89.3 ± 1.5	11
Relation	Label Prop. Friend	89.5 ± 1.2	7
	Label Prop. Follow	89.4 ± 1.3	10
	GCN Friend	89.5 ± 1.5	7
	GCN Follow	86.5 ± 1.4	19
	GCN All Relations (Friend+Follow)	89.1 ± 1.6	15
Combined	GCN All	89.8 ± 1.5	3
	GCN All but Follow	90.0 ± 1.4	1
	GCN RT+QT+Friend	89.7 ± 1.5	5
	GCN RT+Friend	90.0 ± 1.4	1
	GCN QT+Friend	89.7 ± 1.4	5

Besides those main conclusions, we also note the following. First, we implemented all runs of TIMME using 100GB RAM and an RTX8000 GPU (which has 48GB VRAM), and the code released by the authors. However, we found both versions TIMME and TIMME-Hierarchical were very unstable and difficult to run due to out of memory errors. This is caused by the architecture that requires training on all input data types simultaneously, in combination with the size of our datasets. Furthermore, when not crashing outright, it takes over 25 minutes to run 1 epoch (our GCN approach, for comparison, runs over 100 in the same time). We therefore report here on partial results from a limited number of epochs, maximum 40, across learning rates 0.005 to 0.05 (the TIMME authors reported they found 0.01 to be optimal, so we tested values around that one). Out of the top 10 TIMME results according to validation accuracy, none reached 88% test accuracy, while out of the top 10 for TIMME-Hierarchical, none reached 87%. This is some indication that the additional computational cost may not provide good value. However, if this model could be run for more epochs, it might deliver stronger results. Therefore, we plan to further test and attempt to optimize this model in a future version of this paper.

Next, we found that when fine-tuning RoBERTa-large, one of the 10 runs diverged completely. We exclude the divergent run, reporting accuracy and standard deviation for RoBERTa-large over 9 runs/splits. Nonetheless, despite being a larger model, performance

is slightly worse for this approach than RoBERTa-base. We hypothesize, and preliminary experiments support, that the learning rate chosen (1e-5) is on the edge of being too large for this type of model, implementation, and data. This could lead to both the divergent run and the lower than expected performance. Alternatively, the larger model might be overfitting the training data. In future work we plan to conduct a full hyperparameter search and investigation for this model.

We see that finetuning significantly improves POLITICS, however, it only reaches performance equal to RoBERTa-base. This could be due to the finetuning process fully "overwriting" the original political domain adaptation. We hypothesized that a lower learning rate alongside other alternative hyperparameter choices could help the fine-tuned POLITICS exceed the performance of RoBERTa-base without the domain adaptation.

Finally, we also note that 5 of the 6 best performing methods, including the best one, are Combined methods. This suggests there is information captured in the activity but not the relations, and vice versa. However, these methods by definition require more data and computation than single feature approaches, and the improvement in accuracy is not large. So in many cases a carefully-chosen single feature method may be more practical. In particular, Retweet—using label propagation —seems particularly effective, closely followed by Friend. If choosing a combination, these two also work well together.

Users without all relations. In this experiment we remove the requirement that users reported on have *all* the types of relations and activity that we consider. Instead we test our approach with each interaction type on every user that has that type of interaction. Note that this means the set of users tested varies between the interaction types. Furthermore, the set of users in each case is strictly larger than other experiments and includes users with less diverse activities. Therefore, this is a more general and harder task.

When evaluating combinations of interaction types here we use the GCN embedding where available; otherwise we treat the embedding as all zeros. So for example, if we are considering Retweet plus Quote (RT+QT in the table), then if both types of activity are available we use both. While if only one is available we concatenate that one with a vector of zeroes for the missing one, which tells the final classification model that quote is not available.

Results are shown in 6. The standard deviation reported for our approaches here is the result of re-running the model itself 10 times, as in the previous experiment, but here because we examine all possible test users the test set does not change within these 10 runs. Existing state-of-the-art models we run once. In addition to accuracy, we report the user coverage, which shows the percentage of users that have the data needed to run the method.

Unsurprisingly, performance almost always decreases compared with exclusively considering users with all interaction types simultaneously. Nonetheless, we see our methods still provide solid performance. Using any available interaction type gives 100% coverage with over 85% accuracy, while our method obtains even higher accuracy on users with retweets or friends.

Likes. In Table 7 we compare the performance of the Like relation with Retweet and Friend. In the first column, "Accuracy on All Relations Users," we show performance in a setting nearly equivalent to Table 5. The only difference is that Like has 6 less test set users (under 1% less) and its train-test splits are re-randomized. In the next two columns, we show the performance and applicability percentage equivalent to Table 6.

We see that Like gives arguably passable performance but is outperformed by both Retweet and Friend. Its coverage of users is comparable—a bit more than Retweet, but less than Friend. It is possible that the cap of 1000 likes per user, combined with the later date of retrieval for Like vs. Retweet, hurts its performance. However, considering Friend (and Follow) can be retrieved from the Twitter API 1000 at a time[11] while likes only 20 at a time[12], and tweets including retweets can be retrieved even faster still, we do not currently recommend using likes with this approach.

Supervision. We next examine the role of supervision in the GCN. We compare our main GCN model, which does both unsupervised link prediction and supervised party prediction with training data labeled by the profile classifier, with a fully unsupervised version doing link prediction alone. Note that in both cases, the final classification model (random forest) is supervised.

Results are shown in Table 8. We see that although the margin is not huge, the semisupervised version performs consistently better.

Table 6: Accuracy on users without all relations.

Relation	Accuracy	User Coverage (%)
Barberá [5]	86.5	90.7
Preoţiuc-Pietro et al. [44]	81.5	100.0
Retweet	85.4 ± 1.1	90.0
Mention	74.8 ± 1.9	90.2
Quote	80.3 ± 1.2	77.9
Hashtag	79.6 ± 1.2	73.3
Friend	85.5 ± 0.9	93.7
Follow	82.4 ± 1.6	82.4
Any Available	85.3 ± 1.2	100.0
Any But Follow	85.8 ± 1.1	99.9
RT+QT+Friend	86.2 ± 1.0	99.7
RT+Friend	86.1 ± 0.9	99.7
QT+Friend	85.1 ± 0.7	98.7
RT+QT	85.4 ± 1.1	90.2
Activity Only (RT+QT+@+#)	83.4 ± 1.3	97.8
Relation Only (Friend+Follow)	84.7 ± 1.2	94.7

Table 7: Evaluating the Like data type. Its performance is adequate but not better than other data that is easier and more common to collect. The first accuracy column is comparable to Table 5 and the second to Table 6. The Retweet and Friend numbers are reprinted from there. We separate this experiment due to small differences in the data.

Relation	Accuracy on All Relations Users	Accuracy on Any Possible Users	Users w/ Activity Type(s) (%)
Retweet	89.2 ± 1.3	85.4 ± 1.1	90.0
Friend	89.5 ± 1.5	85.5 ± 0.9	93.7
Like	85.2 ± 1.7	79.2 ± 1.4	93.0

[11] https://developer.twitter.com/en/docs/twitter-api/users/follows/introduction
[12] https://developer.twitter.com/en/docs/twitter-api/v1/tweets/post-and-engage/api-reference/get-favorites-list

Table 8: Comparing semisupervised vs. unsupervised GCN. Semisupervised performs better.

	Retweet	Mention	Quote	Hashtag	Friend	Follow	All
Semisupervised	89.2 ± 1.3	80.4 ± 1.5	84.1 ± 1.6	81.9 ± 2.1	89.5 ± 1.5	86.5 ± 1.4	89.9 ± 1.6
Unsupervised	88.5 ± 1.8	77.0 ± 2.2	82.8 ± 2.0	80.7 ± 1.2	86.5 ± 2.3	87.0 ± 1.6	89.5 ± 1.3

Table 9: Comparing projected vs. direct graphs. Projection improves GCN performance. But it hurts label propagation performance, at least if the number of iterations is tuned for it.

	Retweet	Mention	Quote	Hashtag	Friend	Follow	All
Projected - GCN 1 Layer	89.2 ± 1.3	80.4 ± 1.5	84.1 ± 1.6	81.9 ± 2.1	89.5 ± 1.5	86.5 ± 1.4	89.9 ± 1.6
Direct - GCN 1 Layer	85.4 ± 1.4	71.7 ± 2.4	76.8 ± 1.7	77.9 ± 2.2	83.2 ± 1.9	70.3 ± 1.7	88.0 ± 1.6
Direct - GCN 2 Layers	81.1 ± 1.4	66.5 ± 2.8	69.9 ± 3.2	75.3 ± 1.9	77.8 ± 1.6	65.0 ± 2.3	80.5 ± 1.6
Projected - Label Prop. 1 Iteration	89.7 ± 1.4	63.6 ± 1.8	78.8 ± 1.5	81.5 ± 1.9	69.0 ± 2.6	89.3 ± 1.6	NA
Direct - Label Prop. 1 Iteration	72.2 ± 2.7	63.5 ± 2.0	65.2 ± 2.4	62.3 ± 2.3	82.7 ± 1.4	83.4 ± 1.4	NA
Direct - Label Prop. 2 Iterations	89.8 ± 1.3	85.2 ± 1.9	88.3 ± 1.6	86.3 ± 1.5	89.5 ± 1.2	89.4 ± 1.3	NA

There is one exception, the Follow relation, but there the performance is equal up to margin of error. This indicates the supervision helps to produce more informative embeddings from the GCN.

Graph Projection. In this experiment, we examine the effect of using projected (indirect) graphs vs. the original (direct) ones. We show results in Table 9 for both label propagation and GCN approaches.

We see that GCN performance degrades when not projecting the graph. Adding another layer to the GCN, which in principle might help it use information from two-hop neighbors in a similar way to projection, turns out to be further detrimental. On the other hand, for label propagation, the best version we found is two iterations on the direct graph. The best projected graph version is one iteration, but it is clearly worse overall, while one iteration on the direct graph is worse still.

We discuss some of the factors involved and pros and cons of projecting the graph in Appendix B. Overall, this experiment shows that the choice of projecting or not can have a very large effect and is worth future investigation. While certainly not the last word here, we hope this experiment may help provide both practitioners and future researchers developing party prediction methods with more options (both direct and projected are worth considering) and insights.

Tweets vs. Tweet Chunks. In Table 10, we compare the technique of concatenating tweets to provide additional context, vs. embedding a single tweet at a time. We see that the former gives more accurate embeddings.

Note that this experiment was run prior to finetuning, using the final random forest part of the GCN model to make the prediction with the untuned RoBERTa embeddings as input. Therefore, the language model part of this process is only a single run (there is only one pretrained model available), and the variance reported is only from the random forest part. Thus this experiment also shows the improvement in the model due to finetuning: accuracy increases from 83.6% here to 89.2% in Table 5.

Politicians. We consider both learning from and predicting the party of politicians. Learning from politicians has a theoretical advantage because their parties are officially known, removing the need for any additional labeling process. Meanwhile, on the evaluation side, they provide an additional set of data with even more definitive labels than expert-labeled general public users.

Table 10: Comparing tweets vs. tweet chunks. The latter technique improves performance.

	Tweet	Tweet Chunk
RoBERTa-base - untuned	81.9 ± 2.3	83.6 ± 1.5

In this experiment we use the unsupervised GCN version of our model. This lets us test how training the final classification on Politicians alone will translate to performance on Public, with the GCN part of our model (and thus the embeddings) held constant.

We show results in Table 11. Here the first three columns consider the task of predicting party affiliation for the general public, and compare training on the public itself, politicians, or both. The remaining three columns examine the same training scenarios but evaluated on the politicians themselves.

Focusing first on the different interaction types (rows), we see that the best performance generally comes from combining different interaction types. This matches the other experiments, but the benefit on politicians is much higher than on the public. If not using a combined approach, the best performance for politicians comes from the Friend relation (trained on politicians). This suggests politicians are especially consistent in their friends—i.e., who they follow. Considering the strong partisan divisions in the US currently [24, 38], this agrees with the intuition that politicians may avoid following people from the opposite party in order to prevent an appearance of mixed loyalties. Meanwhile, again when trained on politicians, quote performs the worst. This could be due to politicians frequently using quotes not only to support fellow members of their party but also to rebut opponents, making this relation more challenging to learn from in this context.

Considering next the differences between datasets (columns) and results overall, performance on Politicians is generally and often significantly better than on Public, at least if the training data includes politicians. If only training on the public and predicting the politicians' parties, the performance is worse than predicting for the general public. Conversely, training on politicians and testing on the public gives consistently worse results than training on the public. These results match and reinforce the findings of [11]: politicians are easier to predict than the general public, and furthermore performance degrades when trying to train on one and predict the other, regardless which is trained on and which tested on.

This also again highlights the difficulties of comparing model performance when the users are different, and in turn the importance of testing different models on the same users like in Table 5. For example, [46] test their model on politicians. This is a legitimate way to evaluate performance, and their classification task is done on five different parties, which is a much more difficult challenge. But their 90.9% accuracy cannot be directly compared with approaches which report performance on the general public, or vice versa.

Users Matched to Voter Registration. We conducted preliminary experimentation on matching Twitter users to their voter registration, following [5]. We used publicly available voter records

Table 11: Learning from and predicting politician party affiliations. Politicians are easier to predict than the public, and training does not transfer well between the two.

Test Data		Public			Politicians		
Training Data		Public	Politicians	Both	Public	Politicians	Both
Activity	Retweet	89.8 ± 1.5	81.3 ± 2.0	90.1 ± 1.5	84.6 ± 0.7	92.9 ± 2.3	90.3 ± 3.2
	Mention	74.0 ± 1.7	59.5 ± 1.9	73.2 ± 1.9	59.1 ± 1.9	88.1 ± 1.7	81.9 ± 3.6
	Quote	83.8 ± 2.1	68.0 ± 2.8	84.2 ± 2.0	79.7 ± 0.9	81.3 ± 3.5	83.3 ± 2.5
	Hashtag	79.0 ± 1.0	70.4 ± 2.1	78.8 ± 1.4	71.9 ± 0.5	81.4 ± 3.8	76.2 ± 3.4
Relation	Friend	86.2 ± 1.4	72.5 ± 1.4	86.3 ± 1.2	83.1 ± 0.7	94.1 ± 2.4	89.1 ± 2.8
	Follow	87.5 ± 1.5	83.8 ± 1.5	87.6 ± 1.3	83.1 ± 0.7	93.6 ± 2.2	89.8 ± 2.0
All		90.3 ± 1.3	83.6 ± 1.7	90.3 ± 1.4	88.0 ± 1.4	97.4 ± 2.1	95.7 ± 2.1

Table 12: Classification accuracy of different methods on users with matched voter registration. Barberá performs well, but Label Prop. Mention even better, with Label Prop Retweet close behind.

Method Type	Method Name	Accuracy
Existing State-of-the-Art	Barberá [5, 6]	81.3 ± 2.2
Activity	Label Prop. Retweet	81.0 ± 2.3
	Label Prop. Mention	81.8 ± 2.3
	Label Prop. Quote	79.4 ± 1.7
	Label Prop. Hashtag	80.4 ± 1.5
	GCN Retweet (RT)	78.4 ± 2.5
	GCN Mention (@)	76.3 ± 3.0
	GCN Quote (QT)	77.8 ± 2.6
	GCN Hashtag (#)	74.7 ± 2.2
	GCN RT+QT	79.7 ± 2.2
	GCN All Post Activity (RT+QT+@+#)	79.8 ± 2.9

from Ohio, New York, Florida, Arkansas, North Carolina, and Washington DC. We searched the user profiles of our large dataset of 20 million users (the dataset noted in methodology section from which the Public users in the other experiments were sampled), and took exact matches on both name and county. This produced over 30k matched users. We describe this process in more detail in Appendix D.

For this preliminary experiment, we then sampled 500 matched users. We filter these 500 in a manner similar to the very first experiment (Table 5)—i.e., restricting to users with all the data types needed so that every method we examine in this experiment can be tested on the exact same set of users. This results in 280 users. We apply our approaches in the same way as before. Notably, we train them on our Public data; the matched users are only used for testing. We so far tested only one existing approach and our approaches based on activity but not on text or relations. In a future update, we plan to expand this experiment with more methods (and more users).

We report the results in Table 12. We see that the performance across all of the methods is somewhat lower than on our Public users. This is likely because these users are not filtered to have political keywords in their profile, and therefore may be less politically engaged on Twitter. Nonetheless, the patterns we see here are similar to the previous results. The existing approach of Barberá [5, 6] ranks relatively better, which may be because this data is more similar to the data it was developed for (since it was evaluated on similar matched users in the original study). But it still only ranks second behind our Label Propagation Mention approach, and is closely followed by Label Propagation Retweet.

5 FUTURE WORK

Besides some plans noted in earlier parts of the paper, there are a number of other ways we intend to expand this work. First, **further optimization and understanding of our approaches.** This will include extensive hyperparameter search, and testing different architectures for the GCN approach, such as different final classifiers (logistic regression, SVM, MLP), graph models (GAT, GIN), and ways of combining the different embeddings (instead of concatenation; e.g. max, average, weighted by activity).

Second, **more approaches.** Besides comparing with more models from the existing literature, we plan to test another approach for this task, Correct and Smooth [23], that combines label propagation with initial node embeddings (e.g., our text embeddings). We are also working on more sophisticated models for text.

Third, **more datasets and tasks.** We plan to test on a larger sample of users that have been matched to voter records to further validate the results. We also hope to make predictions for more complex labels, such as adding extreme vs. moderate in addition to democrat vs. republican, or looking at other countries with multiparty systems.

Finally, **practical packaging.** Our goal here is to facilitate not only clean replication but also provide an accessible tool for practitioners to use these approaches to make predictions for their own sets of users. This will hopefully lead to improved performance and higher fidelity conclusions on downstream tasks.

6 CONCLUSION

Although party prediction is a foundational part of many research projects on online polarization, in reviewing the literature, we found that (1) it was very challenging to compare the different methods to measure partisanship and that (2) they are often applied without thorough validation. To solve this problem, we first provided a survey of work on this task. This survey highlights not only the reported metrics, but also the data used, which varies widely and is a critical component of the evaluation for this task. Next, we selected state-of-the-art models from this survey and tested them on a consistent dataset we collected. Our results provide both quantitative evidence of the difficulty comparing approaches based on the literature, and the missing thorough comparison. We also contributed three simple approaches of our own, which we studied and

validated through extensive experiments, yielding insights along the way that can help further research in this area. We showed all three approaches are competitive with and often out-perform state-of-the-art methods, while opening up new data types and options for practitioners. In the future we hope to expand our work here into a comprehensive solution for party prediction to accurately measure political conflicts and polarization in different types of online communities.

ACKNOWLEDGMENTS

Paper presented at the *American Political Science Association* Meeting, September 16th, 2022. This project is supported by CIFAR through a CIFAR AI Catalyst Grant: Being Politic Smart in the Age of Misinformation. The first author is supported by a fellowship from IVADO. We thank Daniel Preoţiuc-Pietro and Pablo Barberá for providing additional information/data to help run their methods ([44] and [5, 6] respectively).

REFERENCES

[1] Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, John Quan, George Papamakarios, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Luyu Wang, Wojciech Stokowiec, and Fabio Viola. 2020. *The DeepMind JAX Ecosystem.* http://github.com/deepmind

[2] Adam Badawy, Emilio Ferrara, and Kristina Lerman. 2018. Analyzing the digital traces of political manipulation: The 2016 russian interference twitter campaign. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM).* IEEE, 258–265. https://arxiv.org/pdf/1802.04291.pdf

[3] Christopher A Bail, Lisa P Argyle, Taylor W Brown, John P Bumpus, Haohan Chen, MB Fallin Hunzaker, Jaemin Lee, Marcus Mann, Friedolin Merhout, and Alexander Volfovsky. 2018. Exposure to opposing views on social media can increase political polarization. *Proceedings of the National Academy of Sciences* 115, 37 (2018), 9216–9221.

[4] Antoine Banks, Ernesto Calvo, David Karol, and Shibley Telhami. 2021. # polarizedfeeds: Three experiments on polarization, framing, and social media. *The International Journal of Press/Politics* 26, 3 (2021), 609–634.

[5] Pablo Barberá. 2015. Birds of the same feather tweet together: Bayesian ideal point estimation using Twitter data. *Political analysis* 23, 1 (2015), 76–91.

[6] P. Barberá, J. Jost, J. Nagler, J. Tucker, and R. Bonneau. 2015. Tweeting From Left to Right. *Psychological Science* 26 (2015), 1531 – 1542.

[7] Levi Boxell. 2020. Demographic Change and Political Polarization in the United States. *Economics Letters* 192 (2020), 109187.

[8] Levi Boxell, Matthew Gentzkow, and Jesse M Shapiro. 2022. Cross-Country Trends in Affective Polarization. *The Review of Economics and Statistics* (2022), 1–60.

[9] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. *JAX: composable transformations of Python+NumPy programs.* http://github.com/google/jax

[10] William J Brady, Julian A Wills, John T Jost, Joshua A Tucker, and Jay J Van Bavel. 2017. Emotion shapes the diffusion of moralized content in social networks. *Proceedings of the National Academy of Sciences* 114, 28 (2017), 7313–7318.

[11] Raviv Cohen and Derek Ruths. 2013. Classifying political orientation on Twitter: It's not easy!. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 7. 91–99.

[12] Elanor Colleoni, Alessandro Rozza, and Adam Arvidsson. 2014. Echo chamber or public sphere? Predicting political orientation and measuring political homophily in Twitter using big data. *Journal of communication* 64, 2 (2014), 317–332.

[13] Michael D Conover, Bruno Gonçalves, Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer. 2011. Predicting the political alignment of twitter users. In *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing.* IEEE, 192–199.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[15] Larry Diamond. 2015. Facing Up to the Democratic Recession. *Journal of Democracy* 26, 1 (2015), 141–155.

[16] Morris P Fiorina. 2017. *Unstable mMjorities: Polarization, Party Sorting, and Political Stalemate.* Hoover Press.

[17] Noam Gidron, James Adams, and Will Horne. 2018. How Ideology, Economics and Institutions Shape Affective Polarization in Democratic Polities. In *Annual Conference of the American Political Science Association.*

[18] Noam Gidron, James Adams, and Will Horne. 2020. *American affective polarization in comparative perspective.* Cambridge University Press.

[19] Jon Green, Jared Edgerton, Daniel Naftel, Kelsey Shoub, and Skyler J Cranmer. 2020. Elusive consensus: Polarization in elite communication on the COVID-19 pandemic. *Science Advances* 6, 28 (2020), eabc2717.

[20] Yupeng Gu, Ting Chen, Yizhou Sun, and Bingyu Wang. 2016. Ideology detection for twitter users with heterogeneous types of links. *arXiv preprint arXiv:1612.08207* (2016).

[21] Raffael Heiss, Christian von Sikorski, and Jörg Matthes. 2019. Populist Twitter Posts in News Stories: Statement Recognition and the Polarizing Effects on Candidate Evaluation and Anti-Immigrant Attitudes. *Journalism Practice* 13, 6 (2019), 742–758.

[22] Sara B Hobolt, Thomas J Leeper, and James Tilley. 2021. Divided by the Vote: Affective Polarization in the Wake of the Brexit Referendum. *British Journal of Political Science* 51, 4 (2021), 1476–1493.

[23] Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin Benson. 2020. Combining Label Propagation and Simple Models out-performs Graph Neural Networks. In *International Conference on Learning Representations.*

[24] Shanto Iyengar, Yphtach Lelkes, Matthew Levendusky, Neil Malhotra, and Sean J Westwood. 2019. The origins and consequences of affective polarization in the United States. *Annual Review of Political Science* 22, 1 (2019), 129–146.

[25] Richard Johnston. 2019. Affective Polarization in the Canadian Party System, 1988–2015. In *Canadian Political Science Association Meetings, June*, Vol. 4.

[26] John T Jost, Pablo Barberá, Richard Bonneau, Melanie Langer, Megan Metzger, Jonathan Nagler, Joanna Sterling, and Joshua A Tucker. 2018. How social media facilitates political protest: Information, motivation, and social networks. *Political psychology* 39 (2018), 85–118.

[27] Yonghwan Kim and Youngju Kim. 2019. Incivility on Facebook and political polarization: The mediating role of seeking further comments and negative emotion. *Computers in Human Behavior* 99 (2019), 219–227.

[28] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[29] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [cs.LG]

[30] Emily Kubin and Christian von Sikorski. 2021. The role of (social) media in political polarization: a systematic review. *Annals of the International Communication Association* 45, 3 (2021), 188–206.

[31] M Asher Lawson and Hemant Kakkar. 2022. Of pandemics, politics, and personality: The role of conscientiousness and political ideology in the sharing of fake news. *Journal of Experimental Psychology: General* 151, 5 (2022), 1154.

[32] Steven Levitsky and Daniel Ziblatt. 2018. *How Democracies Die.* Broadway Books.

[33] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL]

[34] Yujian Liu, Xinliang Frederick Zhang, David Wegsman, Nick Beauchamp, and Lu Wang. 2022. POLITICS: Pretraining with Same-story Article Comparison for Ideology Prediction and Stance Detection. *arXiv preprint arXiv:2205.00619* (2022).

[35] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[36] Luca Luceri, Ashok Deb, Adam Badawy, and Emilio Ferrara. 2019. Red bots do it better: Comparative analysis of social bot partisan behavior. In *Companion proceedings of the 2019 World Wide Web conference.* 1007–1012.

[37] Lilliana Mason. 2015. "I Disrespectfully Agree": The Differential Effects of Partisan Sorting on Social and Issue Polarization. *American Journal of Political Science* 59, 1 (2015), 128–145.

[38] Nolan McCarty, Keith T Poole, and Howard Rosenthal. 2016. *Polarized America: The dance of ideology and unequal riches.* mit Press.

[39] Mohsen Mosleh, Cameron Martel, Dean Eckles, and David G Rand. 2021. Shared partisanship dramatically increases social tie formation in a Twitter field experiment. *Proceedings of the National Academy of Sciences* 118, 7 (2021), e2022761118.

[40] MATHIAS OSMUNDSEN, ALEXANDER BOR, PETER BJERREGAARD VAHLSTRUP, ANJA BECHMANN, and MICHAEL BANG PETERSEN. 2021. Partisan Polarization Is the Primary Psychological Motivation behind Political Fake News Sharing on Twitter. *American Political Science Review* 115, 3 (2021), 999–1015. https://doi.org/10.1017/S0003055421000290

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[42] Marco Pennacchiotti and Ana-Maria Popescu. 2011. A machine learning approach to twitter user classification. In *Proceedings of the international AAAI conference on web and social media*, Vol. 5. 281–288.
[43] Gordon Pennycook, Ziv Epstein, Mohsen Mosleh, Antonio A Arechar, Dean Eckles, and David G Rand. 2021. Shifting attention to accuracy can reduce misinformation online. *Nature* 592, 7855 (2021), 590–595.
[44] Daniel Preoţiuc-Pietro, Ye Liu, Daniel Hopkins, and Lyle Ungar. 2017. Beyond binary labels: political ideology prediction of twitter users. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 729–740.
[45] Markus Prior. 2013. Media and Political Polarization. *Annual Review of Political Science* 16 (2013), 101–127.
[46] Ludovic Rheault and Andreea Musulan. 2021. Efficient detection of online communities and social bot activity during electoral campaigns. *Journal of Information Technology & Politics* 18, 3 (2021), 324–337.
[47] Kai Ruggeri, Bojana Većkalov, Lana Bojanić, Thomas L Andersen, Sarah Ashcroft-Jones, Nélida Ayacaxli, Paula Barea-Arroyo, Mari Louise Berge, Ludvig D Bjørndal, Aslı Bursalıoğlu, et al. 2021. The general fault in our fault lines. *Nature human behaviour* (2021), 1–11.
[48] Peter Stefanov, Kareem Darwish, Atanas Atanasov, and Preslav Nakov. 2020. Predicting the topical stance and political leaning of media using tweets. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 527–537.
[49] Joshua A Tucker, Andrew Guess, Pablo Barberá, Cristian Vaccari, Alexandra Siegel, Sergey Sanovich, Denis Stukal, and Brendan Nyhan. 2018. Social Media, Political Polarization, and Political Disinformation: A Review of the Scientific Literature. *Political Polarization, and Political Disinformation: A Review of the Scientific Literature* (2018).
[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
[51] Jan Voelkel, Michael Stagnaro, James Chu, Sophia Pink, and Joseph Mernyk. 2022. Megastudy Identifying Successful Interventions to Strengthen Americans' Democratic Attitudes. (2022).
[52] John Voorheis, Nolan McCarty, and Boris Shor. 2015. Unequal Incomes, Ideology and Gridlock: How Rising Inequality Increases Political Polarization. *Ideology and Gridlock: How Rising Inequality Increases Political Polarization (August 21, 2015)* (2015).
[53] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. 2019. Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks. *arXiv preprint arXiv:1909.01315* (2019).
[54] Benjamin R Warner and Astrid Villamil. 2017. A test of imagined contact as a means to improve cross-partisan feelings and reduce attribution of malevolence and acceptance of political violence. *Communication Monographs* 84, 4 (2017), 447–465.
[55] Steven W Webster and Alan I Abramowitz. 2017. The Ideological Foundations of Affective Polarization in the US Electorate. *American Politics Research* 45, 4 (2017), 621–647.
[56] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv* abs/1910.03771 (2019).
[57] Zhiping Xiao, Weiping Song, Haoyan Xu, Zhicheng Ren, and Yizhou Sun. 2020. TIMME: Twitter Ideology-detection via Multi-task Multi-relational Embedding. *arXiv preprint arXiv:2006.01321* (2020).
[58] Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. (2002).

Table 13: Users matched to their voter registration. Voters* and Users* respectively correspond to the number of unique voters in the records and unique Twitter users in our data.

State	Voters*	Users*	Matched	Democrat	Republican	Other
Ohio	7,771,590	4,913	1,431	320	193	917
New York	17,718,437	30,927	8,255	4,843	1,631	1,781
Florida	14,477,882	50,541	12,905	5,585	4,508	2,810
Arkansas	1,722,465	4,311	1,280	145	140	995
District of Columbia	510,026	17,661	2,538	1,929	153	456
North Carolina	8,004,814	20,761	6,050	2,450	1,655	1,945
Total			32,456	15,272	8,280	8,904

A APPENDIX A: ADDITIONAL LITERATURE SUMMARY

In Table 14 we provide information on papers with lower or not reported performance, below the threshold for inclusion in Table 1.

B APPENDIX B: ADDITIONAL DISCUSSION REGARDING PROJECTION

The projected graph generally has an advantage in computational efficiency compared to the direct one. The projected adjacency matrix has maximum possible row and column size equal to the total users of interest, while for the direct one the maximums possible are the total unique interactions of the type in question. This leads, for example, to nearly 1.3 million rows and columns in the direct Friend adjacency matrix, and over 1.5 trillion entries. Fortunately most of these are 0, so it is still manageable with sparse matrix tools. But it is significantly slower and necessitates the entire method using sparse matrices instead of just up to the projection step, making implementation more challenging.

On the other hand, with a hypothetical oracle method, the direct graph should perform at least as well as a projected one. This is because if one has the direct adjacency matrix A, one can compute $A^T A$ to get the projected one, but the reverse is not necessarily true. So the direct graph provides more information. However, these results suggest that in this context it can be difficult to leverage this information with a GCN model. Further experimentation with hyperparameters is needed to determine if it is simply difficult or in fact impossible, as well as experiments with other GNN models to determine if this holds for that class of models in general.

C APPENDIX C: TUNING CLASSIFICATION THRESHOLD FOR BARBERÁ

We show here the results of different classification thresholds for Barberá's model. This shows a single run in the setting of Table 6 (all users available), with threshold increments of 0.05 from -2 to 1. Anything user below the threshold is classified as Democrat and any user above as Republican. We see that the intuitive threshold of 0.0 performs best.

D APPENDIX D: MATCHING USERS WITH VOTER REGISTRATION RECORDS

We obtained the party affiliation of a unique set of users in each state by md5-hashing their names and county to construct a key identifier. Starting from our original dataset with over 20 million users, we obtained a set of 757,601 US-geolocated Twitter users

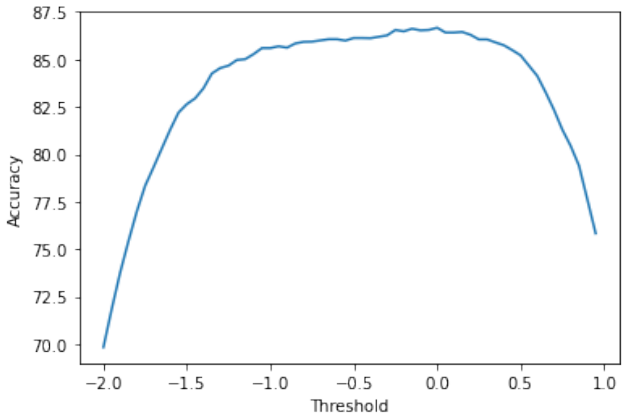

Figure 2: Tuning the threshold for classification with Barberá's approach. We see that accuracy is maximized at exactly 0.

based on their self-declared location in their profile. We then filtered this set for users in the five states plus DC from which we obtained voter party registration data. Finally, we matched the most recent records from the registration data to the unique Twitter users that matched both the county and either the first name and last name or the first, middle and last name. We pre-processed the user's name on Twitter to remove emojis. After matching, we removed users not affiliated with either one of the two major parties and users whose name matched with more than one record per county (indicating a non-unique match).

We report statistics on these users in Table 13.

Table 14: Appendix: Survey methods to predict ideology (with accuracy below 65% or without accuracy)

Papers	Media outlets	Network Activities (retweets and mentions)	Network Relation (fellowship)	Content (words and hashtags)	Accuracy	Dataset Difficulty: (how many users? any filter on users? type (politicians or their followers), type & amount of activity, keywords they have use?)	Code available?
Liu et al. (2022)				X	<50%	* 2,233,552 news articles analyzed. * 11 media outlets with a clear political leaning and popularity were crawled. * Crawling of the pages from these media from January 2000 to June 2021 from Common Crawl and Internet Archive. * The news has to be related to US politics. * Some media that dominate their model training were removed to have media ideology in their training that contribute equally.	YES
Pastor-Galindo (2020)				X	63%	* They classify 20,364 bot accounts. * They drop their sample to 20K, because these accounts have at least one tweet targeting one of the five political parties.	YES
Sinno et al. 2022	X			X	55%	* 1,749 news articles across nearly 3 decades (from 1947 till 1974) (political). These articles have to include politically relevant topics, coming from a center-left, central, or center-right ideology.	NO
Yang, Hui and Menczer (2020)	X			X	NA	* Data related to the 2018 U.S. midterm elections. From Oct. 5th to the end of 2018. Use a set of hashtags: 143 + state's Senate election hashtags. * The hashtag #ivoted was employed to identify potential voters on Twitter. * 60M tweets by 3.5M unique users.	YES
Chen (2015)			X	X	NA	*Politicians (73): senators in the 113th, and 112th congress that have a twitter account. *Public (103723): list of users following at least one of the senators with the Twitter REST API. They had to be active (more than 20 followers). * Use of the roll call data of 236 bills and voting records.	NO
Bright 2017		X		X	NA	* Politicians: list all official Twitter account names of major political parties and party leaders in all 28 EU member states. * Users (1,426,620 tweets): Tweets collection: from these accounts and tweets that mention them. Period: May 11th to June 10th (2016 ??).	NO
Gaisbauer et al. 2021		X		X	NA	* Tweets from a seed set of users (politicians: 270 users) and tweets mentioning one of the seed set of users (retweets, mentions, and replies). * Two political events considered: 1) Politicians and the public (364,626 tweets): Saxon state elections (25/07/2019 - 10/09/2019): candidates, state parties, leaders of fractions, local party organizations, members of the national and European parliament, Saxon correspondent, and media accounts list with Twitter accounts. * Users not included in the seed set but mentionning them at least oncea week were added. 2) Public (130,685 tweets): Police and citizen clash in the city of Leipzig on New Year's Eve (31/12/2019-19/01/2020): use of specific keywords (connewitz, antifa, polizei, polizist, le0101, etc).	NO
Garimella and Weber (2017)	X	X	X	X	NA	* Seed set of accounts of politicians (presidential/vice presidential candidates and their parties) and media outlets (npr,pbs,abc,cbsnews,nbcnews). From left and right. * Crawling of users' tweets that mention or retweet the seed set. Following users : 140M users. Retweeting users (from 50% of the users (679,000)): 2 billion tweets. * From 2009 to 2016.	YES
Kamiensk et al. (2022)		X	X	X	NA	* Crawl of hashtags and terms related to political events in Brazil. *Identification of influential users. After manually classifying their ideology, collect of the following relationship. * April 3 to November 27, 2020. * 33 events (33 datasets) chosen.	NO