Twitter Influence and Verification

David Atwood, Mark Raj, Justin Shultz
Vanderbilt University

Abstract

For our final project, we wanted to analyze and predict Twitter influence. Some of the motivation that led us to come up with this idea was that Twitter is an essential hub for political discourse, the spread of information, and creating social connections, and it is an extremely useful platform for studying and analyzing social networks. Twitter has come under fire over issues of free speech, which users are promoted by Twitter and, most recently, monetizing the "verified" token with Twitter Blue. We wanted to study the factors that are important to "influence" on Twitter and make predictions for "influential" users.

Introduction

Problem & Context

The problem at hand is that we have little understanding of the isolated effects of verification status. We do not know if it influences credibility or algorithmic recommendations and Twitter does not provide any information on how the verification status is used in their platform. This is incredibly important to understand, especially now, because of the recent monetization of Twitter verifications.

It was recently proven that Twitter Blue has made it easier than ever to impersonate influencers and relevant members within the Twitter world. For example, a Washington Post reporter was able to pay the eight-dollar subscription and create a verified account posing as Senator Ed Markey (Fowler 2022). In the weeks immediately after the rollout of Twitter Blue, fake accounts became so rampant that the service was blocked for newly created accounts. The blue checkmark's status as a symbol of trust and authenticity has been called into question ever since it became purchasable. For this reason, it could be assumed that the verification token holds little importance in the algorithm that determines influence over a Twitter network. However, we will prove that this may not be the case.

We also wanted to analyze the role that verification plays in a Twitter community. More specifically, we wanted to see if verification holds greater influence over certain communities than others. Thus, we decided to build multiple Twitter networks centered around users with seemingly different backgrounds and communities. The communities we chose to examine were centered around LA Galaxy soccer player Dejan Jovelic, data scientist Steve Hedden, and politician Sheila Cherfilus-McCormick respectively. We also created an artificial network that we based on online data we gathered about Twitter verified users and their average metrics.

Timeline

In order to successfully collect the necessary data from Twitter so that we could build a model to predict verified users, we broke down the project into weekly tasks. We began with scanning the web for potential datasets we could use to build a large network that we could use for training and testing. However, we could not find any Twitter datasets that included a user's verified status, so we decided to use the Twitter API to scrape data directly from the website. Due to the Twitter API's time constraints on how many users we could scrape at a time, we designated a large portion of the week to scrape the data of each user in the network. Once we gathered the necessary data, we needed to build the networks. Each network consisted of an edge list where each node contained attributes about that user. We then used those networks to train and test our model.

Data & Methodology

Scraping Twitter

For a representative sample of real-world social networks, we wanted to use current data from Twitter. Twitter data is most easily accessible with their API, which is available publicly upon approval after a short application review process - this gives users consumer and authentication tokens that we used for the entirety of the project. The API includes extensive documentation on the functions provided

which can be found here[1]. An important limitation of their API is the inclusion of rate limits, which are 1 request per minute to pull a user's following information (i.e. 1 user per minute) and 300 per 15 minutes to pull a profile's information (verification, location, translation services, etc).

The base algorithm used to pull the initial network is based on Steve Hedden's article (Hedden 2021) to visualize any user's network of followers. The TweePy Python package used is able to handle the API Authentication as well as sleeping a process to satisfy the rate limits, making the scraping process far more efficient. The initial algorithm takes a "source" user's Twitter User ID (accessible via the API with a user's screen name) and pulls each of their followers' User ID's into a data frame. These User IDs will become the node IDs of the final graph and are unique to each user. A modification we made was to make this a directed graph with an edge from node i to node j indicating node i follows node j. Thus, this initial data pull results in a single central node with many directed edges inward toward it.

The second part of the algorithm will use the API to iterate through each of this initial ring of users and pull <= 5000 of their followers' User IDs. Thus, we create a network of followers-of-followers from the initial source node who also, presumably, have connections amongst themselves to create a snapshot of a community. One variation we made was to pull <= 500 (rather than 5000) of the followers-of-followers due to API constraints. This allowed us to have more influential source users without sacrificing information on the outer ring of users.

Our main contribution to scraping is the inclusion of node profile data. The API offers a get_user_info method that returns a parsable User object. With this data, we iterated through each node of the graph in NetworkX and called the API to return user information for that ID. This user object was parsed and the relevant attributes were added as NetworkX node attributes to the graph. Relevant attributes included information that could plausibly be a factor in influence (translation, followers) and only removing factors that are internal to Twitter (like profile pictures). This resulted in node tuples in the format below with (id, attributes). The rate limits for get_user_info are far less restrictive, allowing a graph of ~3000 nodes to be populated with information within 30 minutes.

(1434574860.0, {'follower_count': 364, 'verified': False, 'name': 'jazmin', 'screen_name': 'JazminTrjll', 'protected': False, 'friends_count': 449, 'listed_count': 1, 'favourites_count': 27261, 'geo_enabled': True, 'statuses_count': 19217, 'contributors_enabled': False, 'is_translator': False, 'is_translation_enabled': False, 'profile_use_background_image': True, 'has_extended_profile': True, 'default_profile': True, 'default_profile_image': False})

Node attributes included from each profile

With these algorithmic steps completed, we conducted data cleaning before having our final networks. This consisted of two parts. First, the follower-of-followers graph was filtered using the k-core NetworkX algorithm to remove users with a degree of less than 10. These nodes were unlikely to be meaningful users of the network and would be difficult to separate distinct communities from one another if they were not well connected. This went from a graph of ~300,000 nodes to ~3,000 nodes. Secondly, the get_user_info API call would occasionally return errors (ex. mismatched IDs) that resulted in blank nodes. We dropped all nodes that didn't have information and any attributes that were 'None' to allow for cleaner modeling in the network learning. This reduced the ~3,000 nodes to ~2,000 nodes that were fully connected and had all available information.

Data Pipeline
1. **Twitter API** Data lives in Twitter's internal storage
2. **CSV Edgelist** Following-follower relationships stored as edges
3. **NetworkX Graph** Graph constructed from edges and Twitter User ID's
4. **GML** Used to capture node attributes and edges in a transferrable file format
5. **PyTorch Tensor** Used for GCN with PyTorchG
6. **Gephi** format required to make Gephi visualizations

Artificial Network

After our in-class discussion of random network algorithms, we wanted to devise a randomized graph generator algorithm that mimics a Twitter community. We split up the network into four classes based on follower count ranges. Each group of follower ranges had different probabilities so lower follower counts between 10 and 1000 followers were the most likely and larger follower counts were much less likely. We determined from a google search that the average follower count was 707 followers (Ahlgren 2022), the percentage of verified users on social media is

[1] https://developer.twitter.com/en/docs/twitter-api

between 0.05 percent, and the average number of followers that a verified user has is 125,000 with a median of 9000 (Kamps 2019). Therefore, we gave users with higher follower counts a higher probability of being verified and connected to other users. Our model was based on averages, so it was not completely accurate, but we could use this network to test the validity of our model.

Source Node Selection

With the algorithm described above, we were tasked with deciding on representative nodes. These nodes would, ideally, represent different communities on Twitter (niche interests, geographically concentrated, or source of truth) and different influence types (interest-based, professionals, or word-of-mouth)

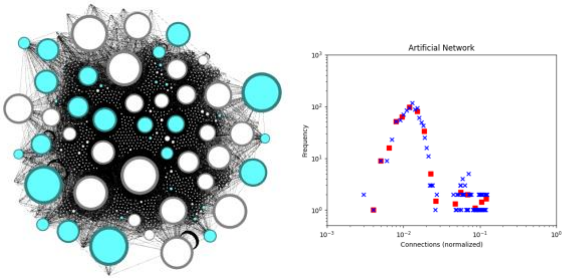
With these parameters and constraints on the following size of a source node (<1,500 followers to ensure ~2 days of scraping), three representative profiles were chosen. The first was Steve Hedden[2], a Data Scientist based in Denver, Colorado. We expect this community to be highly connected among data scientists with many reciprocal relationships. Steve may not be the most influential person in this network if there is another data scientist with a larger following and it will be interesting to see if this affects later analysis.

The second was Dejan Joveljić[3], a Serbian soccer player at LA Galaxy - this allows us to explore what sports influence looks like, a popular use of Twitter for many. We expect Dejan to be a less reciprocal graph with many people following Dejan directly. We also hope to find some geographic diversity with a community of local fans in LA and fellow compatriots from Serbia. In terms of influence, we expect there to be significant profiles in the network with competing influence to Dejan, such as fellow popular teammates and the LA Galaxy main account.

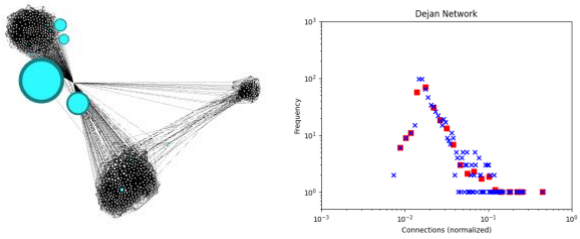
Finally, we wanted to include a politician due to the increasing role Twitter plays in American politics and recent changes to free speech policy at Twitter. We chose Representative Sheila Cherfilus-McCormick[4] of Florida's 20th Congressional district, who was recently elected to the US House of Representatives as a Democrat in a special election. We expected the community to be more connected geographically around the Congresswoman's home district. We also expect her to be among the most influential nodes in the graph since this community is likely more built around her than a general interest in politics.

Network Statistics & Visualizations

Upon completing the Twitter API portion and having our graphs in NetworkX, we sought to test these hypotheses and understand the nuances in each network to have context for later comparisons of results. First, we made degree distribution plots to confirm if the real-world networks exhibit the linear log-log relationship we discussed in class and compare this to the artificial network. Secondly, we wrote our networks to GML files so that they could be read into Gephi, a network visualization tool for larger networks. With Gephi, we were able to isolate the verified users as a way of visualizing their significance within the network.

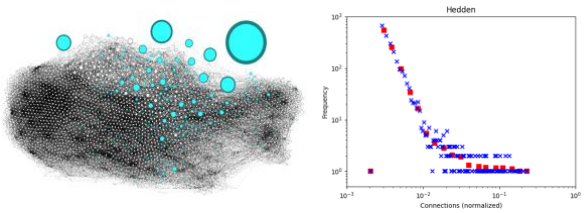


Artificial Network (n = 1000, k =20.34, Avg. Path Length=3.50, Avg. Clustering=0.05, Largest Component=99.1%)

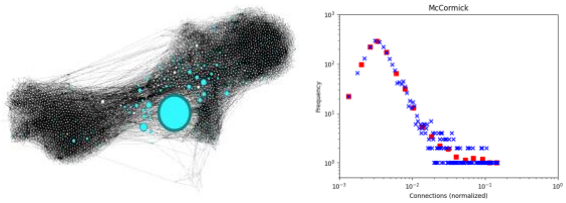


Dejan Network (n = 685, k = 21.19, Avg. Path Length=0.27, Avg. Clustering=0.16, Largest Component=7.45%)

[2] https://twitter.com/stevehedden
[3] https://twitter.com/JoveljicDejan
[4] https://twitter.com/CongresswomanSC



Hedden Network (n = 2446, k = 19.19, Avg. Path Length=0.29, Avg. Clustering=0.25, Largest Component=9.36%)



McCormick Network (n = 2251, k = 17.49, Avg. Path Length=0.46, Avg. Clustering=0.09, Largest Component=7.73%)

There are some interesting and promising indicators we noticed from these visualizations and basic network statistics. First, we can see the clear difference between real and artificial networks in the degree distribution plots we constructed. The artificial network has 3 distinct classes of nodes, with low-influence users being more frequent with lower degrees and high-influence users being rarer (the size of the right-most bump is very small) and with high degrees. This is compared to the 3 Twitter networks, which each have a cascading plot and, as expected, some central hubs with a very high degree. Another interesting trend among the real networks is that McCormick and Dejan networks have a tail of very low degree nodes that increase in frequency until a critical value and then begin a negative relationship with degree while the Hedden network has a negative relationship the entire time. We suspect these are very low-usage Twitter users who follow these influencers, who all have an "average" profile of just a friend in your network. Hedden, on the other hand, is likely professionals and other data scientists, where everyone has the propensity to post findings or news, making it a clearer influence relationship.

The Gephi visualizations add some very critical insights that confirm some of the hypotheses we began with. First, we do see a rough relationship between being verified (node is blue) and having a larger following (node is large in size). Indeed, in the three Twitter networks, the nodes after a certain following size are almost all verified and any of the smaller nodes are guaranteed to not be verified.

Regarding communities, we can also see different communities form around our source nodes, which is promising for the initial impetus to understand how different Twitter communities operate. The Dejan network exhibits 3 clear communities that are disparate and unified only by Dejan himself - these could be his Serbian community, an LA community, and a smaller soccer/sports interest community. The Hedden graph, as predicted, is highly connected with one another with Hedden being among a class of influential and/or verified users in the center. We don't see much separation of communities in this professional network, which indicates a less disparate user behavior. Finally, the McCormick network exhibits two generally disparate communities that are unified by a set of verified, influential users in the center. These central users could be main political players in both of these communities, such as Florida local politicians and national senators and representatives. The outer communities are also very dense, meaning information diffusion in that community likely happens very rapidly.

Influence Analysis

Methods

It is known in the literature that network centrality is a measure of influence when discussing information diffusion (Ghosh 2012). Within the context of Twitter, a particular node's centrality score could be seen as a proxy for influence absent an official ranking from Twitter's API. With these two elements, we wanted to explore what are the profile statistics that are most important to centrality without taking into account network information as a way of blindly categorizing influential users. We then hope to learn the importance of different factors for influence that can be later used in the process of verifying different users. This will also confirm whether verification is an important factor in influence in general as a guided tool for Twitter as they navigate the question of monetization.

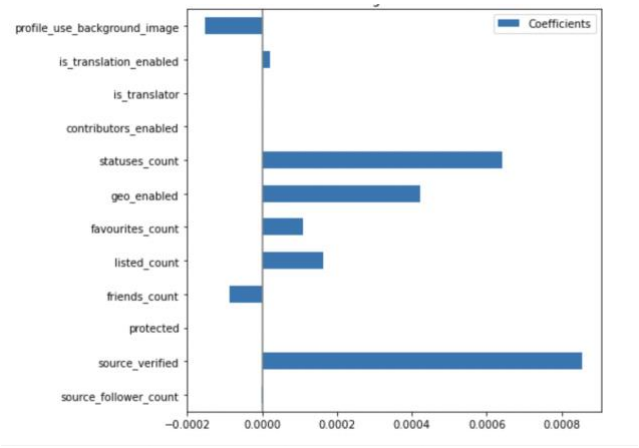
In terms of methodology, we constructed a data frame of all included node attributes from the API for each of the networks above. Each of these attributes would eventually become the features of a machine learning model. We then calculated the degree and eigenvector centrality of each node and appended those as a new column value for each of the rows in the data frame. These would eventually become the target or predicted values.

Using SciKit Learn, we then created two models to predict Degree and Eigenvector centrality: Simple Linear Regression and Stochastic Gradient Descent Regression (SGD). These were chosen for their relative simplicity and we suspected there would be overfitting issues with more advanced models. The simple regression model would also

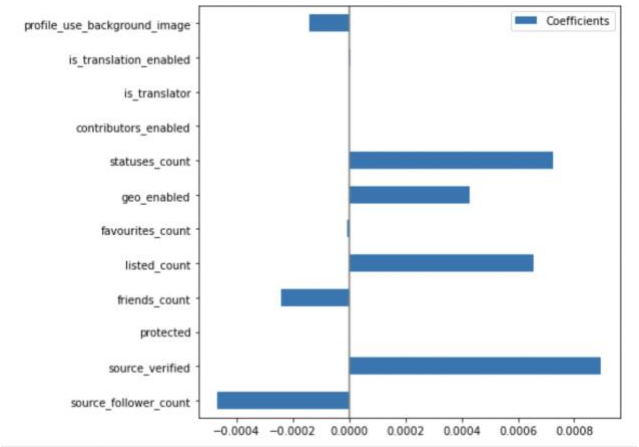
allow a baseline to compare against and we expect will not perform very well with prediction. Using SKLearn, we created a pipeline to normalize numerical data and drop categorical variables. We trained each model with an 80-20 training-test split with a k=5 cross-validation to ensure data quality. We calculated the accuracy for each of these models and calculated and plotted feature importance using the coefficients available via SKLearn.

Results

The influence analysis created some mixed results with some promising features. As expected, we saw low accuracies for predicting a node's centrality solely based on node information without any network features. The simple linear regression model had an average accuracy of 0.12 for degree centrality and effectively 0 for eigenvalue centrality. The SGD regression model had an accuracy of 0.30 for degree centrality and 0.10 for eigenvalue centrality.



Degree Centrality – SGD Model Variable Importance



Degree Centrality – Linear Model Variable Importance

The variable importance analysis for these bears some interesting results. The SGD and Linear models both place a high importance of verification on predicting the degree centrality of a node, which is promising for the verification analysis and ties a loose relationship between verification and influence. Each model also positively weights a node's activity with statues (which likely drive engagement) and having geo_enabled (to connect to users in the same area). The friends_count variable is weighted negatively, which is a measure of how many people the node follows back so users who have a high ratio of followers:following are more influential according to these models.

GCN Verification Prediction

Methods

Graph convolutional networks (GCNs) have shown great promise in their applications to social networks, including issues of community detection, link prediction, and social influence (Zhang et al. 2019). GCNs have even been applied to Twitter data, as done by Vijayan et al. (2018) with predicting retweet counts. We built a GCN to predict verification. Broadly, we predict that because verification is now purchasable, verification status will be less tied to attributes like follower count or graph structure and potentially more difficult for the GCN to predict. We ran a linear regression on each of the three Twitter networks to evaluate the correlation between verification and the other attributes, as seen in the table below. Note that this does not include any of the graph features, only node attributes. Additionally, we limited the node data included in the model to the numerical attributes: followers, following (friends_count), number of public lists they appear on, favorites, and statuses. We maintained this selection of attributes in the GCN as well.

Network	*Dejan*	*Hedden*	*McCormick*
Coefficient of Determination	0.402	0.164	0.165

Coefficients of determination for linear regression on each network

All three coefficients are clearly low, suggesting that the relationship is not as simple as "followers = verification." This is in line with Twitter's official policy of giving verification status to "highly sought users" and not listing publicly a minimum follower requirement for

verification[5]. The absence of a clear relationship between user attributes alone and verification, plus the importance of verification in influence analysis from the previous section motivate our verification prediction GCN.

We designed three experiments to explore verification prediction across different communities on Twitter: (1) learn verification on a network and test on the same network, (2) learn verification on a network and test on a different network, and finally, (3) learn and test verification on all three networks. The models were tested and trained with random training and testing splits five times and the average accuracy along with standard deviation were used for model evaluation. We used a 60/20/20 training/validation/testing split for all experiments and our GCNs were constructed using PyTorch Geometric. The .gml file was converted into a NetworkX graph, which was then converted to a PyTorch Data object after cleaning the attributes. After data masking we conducted a hyperparameter grid search to find the optimal combination of layers, L2 regularization and learning rate. For the Hedden and McCormick models, we found 2 layers, an L2 of 0.0001, and a learning rate of 0.01 to be optimal. The Artificial and Dejan models had high accuracies (>94%) regardless of parameters so we elected to keep the hyperparameters consistent across all four models. Experiment 3 required concatenating the PyTorch data objects for each network. To the GCN in Experiment 3, the three networks might as well be three disconnected components of the same graph.

Results

Experiment 1 (Learn and test on same network)

Network model	*Avg. accuracy*	*Standard deviation*
Artificial	0.933	0.038
Dejan	0.966	0.047
Hedden	0.823	0.087
McCormick	0.813	0.078

Experiment 1 Model Evaluation Results

The Artificial and Dejan network models performed the best, with accuracies in the mid-nineties. The Hedden and McCormick models were still strong with accuracy in the low-80s, but also suffered from high standard deviations. The Hedden network especially was prone to the occasional trial with an accuracy of around 50. The standard deviation suggests we aren't overfitting our models, which bodes well for testing the models across networks.

Experiment 2 (Test on other two real networks)

	avg. accuracy (st. dev.)	*Testing Network*		
		Dejan	*Hedden*	*McCormick*
Training Network	*Dejan*		0.782 (0.047)	0.833 (0.096)
	Hedden	0.972 (0.016)		0.848 (0.031)
	McCormick	0.902 (0.160)	0.832 (0.075)	

Experiment 2 Model Evaluation Results

The Dejan model performed considerably worse than the other two testing across networks. Its accuracy on the Hedden and McCormick networks was slightly worse than the accuracy of the Hedden and McCormick models in Experiment 1. The Hedden and McCormick models, on the other hand, performed significantly better on foreign networks. The Hedden model had a higher accuracy on the other two networks than it did on its own network. The standard deviations across all models are noticeably larger than in Experiment 1. This is possibly due to small discrepancies in the training set having a magnified effect when tested on a completely unseen network. The differences across the models motivate our final experiment - learning and testing on all three networks.

Experiment 3 (Learn and test on all three networks)

Avg. accuracy	0.870
Standard deviation	0.066

Experiment 3 Model Evaluation Results

Training on all three resulted in a consistent, accurate model. One of the five trials resulted in an accuracy of 0.752, but besides that the model was consistent in the upper-80s. The combined data was masked completely

[5] https://help.twitter.com/en/managing-your-account/about-twitter-verified-accounts

randomly with no controls on how many nodes were to be included in training from each graph, which could explain some of the variation between trials.

Conclusion

Discussion

In conclusion, we have shown that verification is predictable with good accuracy, but difficult without considering network structure. We showed there was little correlation between verification and other user attributes. Our cross-network GCN had a strong 0.87 accuracy. We also showed that verification is tied to influence on Twitter by using centrality as a proxy measure. The blue checkmark plays a unique role on Twitter as a symbol of authenticity that users can't get from follower counts and other profile information.

We also explored how verification varies community to community. First, we saw differences in the network structure of each of the three communities that we analyzed and visualized - the Serbian soccer player (Dejan), the data scientist (Hedden), and the house representative from Florida (McCormick). These distinct network features carried over into the verification prediction and the accuracy between models. Verification in the Dejan Network had a considerably higher correlation of determination than other networks, while still low, and it was easier for our GCNs to predict. Our GCNs performed worse overall on the Hedden and McCormick networks, but we found that training across communities improved performance. We showed that training on one community and testing on another provided a comparable model to training on the same community, albeit with more variability.

While we can't draw strong conclusions from our data about the effect of Twitter Blue on the reliability of the blue checkmark on Twitter today, we speculate that the verification badge remains an overall trustworthy indicator. We suspect that in the case that Twitter Blue checkmarks were widespread, verification would be much more difficult to predict.

Future Work

This project is just but a foray into a rich and highly promising realm of social network analysis. Twitter data is highly informative with its combination of network and node attributes and we hope to see this algorithm for scraping to set up a GCN used to continue exploration. We see three possible future directions for this work with some constraints on time and price. The first is pulling data from well known "influencers" on Twitter with an order of magnitude larger follower count (tens to hundreds of thousands). This could be achieved with paid access to the API, which increases rate limits. This would allow us to see if these results scale up for highly influential users or if there are new community dynamics that exist as these source nodes grow in size.

The second direction is one that was explored briefly which is distinguishing between "legacy" Twitter users and those who have paid for verification. There are some preliminary tools that scraps this information from the web but once this information is available in the API, it will be a very easy integration to our existing code. This would allow us to compare and contrast how a model that predicts "legacy" verification and one that predicts Twitter Blue verification differ in performance and feature importance, which could also be used to target users who are likely to purchase Twitter Blue.

The final direction is one that can be explored already. With our existing data, it would be interesting to uncover why verification is harder to predict in some communities rather than others - is this tied to relative connectedness of the graph, the interests of the users, or some external factors from our data? This could be revealed with selecting different representative source nodes for these same communities (e.g. a different data scientist or U.S. Representative) and seeing if these results hold or differ with a different slice of the community. Altogether, there are many interesting directions to move forward with this work and we hope to see them come to fruition.

References

Fowler, Geoffrey A. "Review | We Got Twitter 'Verified' in Minutes Posing as a Comedian and a Senator." *The Washington Post*, WP Company, 11 Nov. 2022, https://www.washingtonpost.com/technology/2022/11/11/twitter-blue-checkmark/.

Ghosh, R., & Lerman, K. (2012). Rethinking centrality: the role of dynamical processes in social network analysis. *arXiv preprint arXiv*:1209.4616.

Hedden, Steve. "How to Download and Visualize Your Twitter Network." *Medium*, Towards Data Science, 13 Feb. 2021, https://towardsdatascience.com/how-to-download-and-visualize-your-twitter-network-f009dbbf107b.

Kamps, Haje Jan. "Who Are Twitter's Verified Users?" Medium. Medium, October 13, 2019. https://haje.medium.com/who-are-twitter-s-verified-users-af976fc1b032.

Matt Ahlgren, WSR Team. "Top 40+ Twitter Statistics 2022: Stats, User Demographics & Facts." December 2, 2022. https://www.websiterating.com/research/twitter-statistics/.

Vijayan, Raghavendran and George O. Mohler. "Forecasting Retweet Count during Elections Using Graph Convolution Neural Networks." 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA) (2018): 256-262.

Zhang, S., Tong, H., Xu, J. *et al.* Graph convolutional networks: a comprehensive review. *Comput Soc Netw* 6, 11 (2019). https://doi.org/10.1186/s40649-019-0069-y