Forecasting Retweet Count During Elections Using Graph Convolution Neural Networks

Raghavendran Vijayan[1] and George Mohler[2]

***Abstract*— A retweet refers to sharing a tweet posted by another user on Twitter and is a primary way information spreads on the Twitter network. Political parties use Twitter extensively as a part of their campaign to promote their presence, announce their propaganda, and at times debating with opponents. In this work we consider the problem of early prediction of the final retweet count using information from the network during the first several minutes after a post is made. Such predictions are useful for ranking and promoting posts and also can be used in combination with fake news detection. From a machine learning perspective, the task can be viewed as a regression problem. We introduce a novel graph convolution neural network for forecasting retweet count that combines network level features through graph convolution layers as well as tweet level features at a higher dense layer in the network. We first will provide an overview of the graph convolution network architecture and then perform several experiments on Twitter data collected during presidential elections in South Africa (2014) and Kenya (2013). We show that the model outperforms baseline models including a feed-forward neural network and the popular point process based model SEISMIC.**

I. INTRODUCTION

Social media (Twitter, Facebook, etc.) is playing an increasingly important role in elections around the world [1] driven in part by the fact that many people receive their news from social media. Recent research has shown that fake news can dominate the truth online [2] and in the U.S. 2016 election new research indicates that fake news may have contributed to Donald Trump's victory [3]. Machine learning methods have been introduced for the purpose of fake news mitigation on social media [4], however the effectiveness of such methods will rely on how well emerging Twitter cascades can be ranked for virality to prioritize interventions.

In this research work we focus on the following problem. Given an observation of the first several minutes of a political retweet sequence cascade during a pre-election period, predict the final retweet count (or the count after some final time) of the cascade. For example, in the 2014 South African election there were 466 retweets for the tweet sequence contained in Figure 1 posted by a verified user who is a radio jockey and philanthropist in South Africa. Each point in Figure 1 represents a retweet. The retweet cascade has accumulated 23 retweets (marked in red color) in the first ten minutes and an additional 22 retweets (marked in blue color) in the second ten minutes. We wish to construct a predictive model that takes into account features corresponding to these 45 retweets, and the actors who retweeted them, and estimates the final number of retweets marked in cyan color.

Forecasting the final retweet count given an initial cascade is the focus of several recent research studies. These studies have taken into account the role and behavior of actors [5], [6], predictive features from the images attached to the tweets [7], network properties [8], and the relationship between the author's profile and success of the tweet [9]. Some researchers [10], [11], [12], [11] have treated the problem from a classification point of view by classifying if the post is going to be viral or assigning the post to categories of popularity. In [13], the factors motivating users to retweet are investigated. Statistical models [14], [15], [16] that do not require feature engineering and can systematically take into account how cascades propagate on networks have been constructed using point process models and survival theory.

[1]Raghavendran Vijayan, Department of Computer and Information Science, IUPUI

[2]George Mohler, Associate Professor, Department of Computer and Information Science, IUPUI

Fig. 1. (Left) A sample tweet from the 2014 South Africa Election. (Right) Resulting retweet Cascade.

In this paper we propose a novel method for retweet prediction during elections that draws upon the advantages of both the machine learning regression framework, that allows for election related features to be crafted, and models such as point processes that are good at incorporating network and diffusion effects. In particular, we introduce a deep learning framework where we use graph convolution layers [17], [18] that can learn non-linear diffusion features from the social network coupled with a dense layer containing tweet level, election specific features that is introduced at a higher level of the neural network. In Section II we discuss the architecture of our neural network. In Section III we provide an overview of the election related features we use in the model. In Section IV we give results for several experiments conducted on Twitter data from the 2013 Kenya election and 2014 South Africa election. We show that the graph convolution neural network outperforms SEISMIC and other baseline models in terms of ranking populars retweet sequences by up to 15%.

II. Graph Convolution Neural Network Model

Our goal is to predict how many retweets will be in a cascade based on observed retweets during the first T minutes after the original post. Let A be the adjacency matrix of the social network and let $\vec{x}$ be a node-level feature vector. For example, if $\vec{x}$ is an indicator variable for who retweeted before time T then $A \cdot \vec{x}$ will be a vector representing the number of followers of those who posted in the first T minutes. Similarly, $A^k\vec{x}$ can be used to construct features capturing network diffusion out to k steps on the network from each of the users who posted in the first T minutes. While these sort of features could be engineered, we propose that graph convolution neural networks (GCN) are well-suited for this task.

The overall architecture of our model is given in Figure 2. Two types of features are introduced at different points in the network. Node-level features are introduced early on through graph convolution layers that capture network-diffusion effects (we describe specific features in Section III). In particular, a GCN layer [18] is given by,

$$X^{l+1} = f(X^l, A) = \sigma(AX^lW^l) \tag{1}$$

where X^l denotes the node-level features of the lth layer, W^l represent the weights of each feature vector, A is the network adjacency matrix, and σ represents a nonlinear activation function.

The primary difference between a feed forward neural network and the graph convolution neural network is that the former operates at a tweet instance level, whereas the GCN operates on nodes (actors). During election periods, it is common to see tweeters that support the same party sharing similar type of tweets. Hence there is a significant advantage to capturing the connections between actors that are represented as nodes in the network.

We note that point process based models [16] are also suitable for capturing information diffusion on networks. A Hawkes process model may

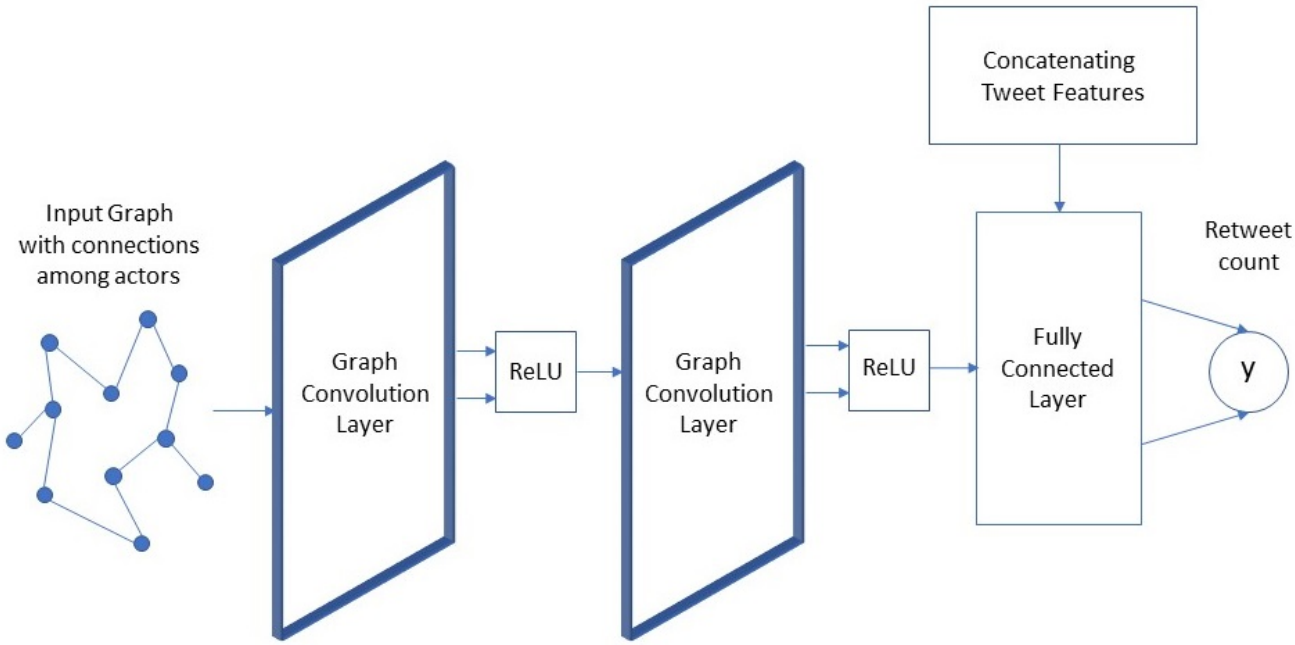

Fig. 2. Architecture of the Graph Convolution Neural network Model

also incorporate the adjacency matrix, where the intensity of retweets $\lambda_i(t)$ at node i is determined by,

$$\lambda_i(t) = \mu_i + \sum_j a_{ij} g(t - t_j). \tag{2}$$

Here tweets occur at some baseline rate and are also triggered by a post by friend j at time t_j in the network. The kernel g decays to zero to model the fact that the probability of retweeting decays over time.

However, models such as those in Equation 2 do not incorporate nonlinear effects and also there is information in the tweet itself that we would like to capture as a feature, for example the political party, topic, sentiment, etc. For this purpose we introduce a dense layer after the GCN layers where tweet level features are concatenated. The dense layer is also necessary because we are not predicting at the node level (which is the common application of GCNs) and are instead predicting the final retweet count over the whole network.

The specific architecture we employ is three hidden layers, two GCN layers and a fully connected layer. Because we aren't privy to the actual followers network, we use a proxy based on historical retweet sequences where actors have jointly posted. The weighted adjacency matrix A is constructed so that a_{ij} is proportional to the number of times i and j have retweeted together in the historical dataset.

III. Election Specific Feature Engineering

We extract election-specific tweet level features as follows. We have first identified the top keywords belonging to the political parties in each country using manual labeling. We provide an example of these keywords and the associated party in Table I. The brackets denote the party assigned to each keyword. ANC, DA, EFF refers to the African National Congress, Democratic Alliance, and Economic Freedom Fighters respectively. We then use keyword matching on each tweet to identify to which political party the tweet belongs. If the keyword/hashtag is irrelevant to elections or inconclusive to classify, we assign the keyword into an "unknown" category. Next we compute a polarity score for each tweet, assigning the tweet to positive, neutral, or negative.

In Figure 3 we provide an example tweet gathered during South Africa Election. The party supported in the tweet is identified through the keywords anc and whyivoteanc. The polarity score

TABLE I

TOP 10 RENDING KEYWORDS IN SOUTH AFRICAN ELECTIONS

Hashtag	User Mention	Word
ayisafani(ANC)	helenzille(DA)	da(DA)
siyanqoba(Unknown)	lindimazibuko(DA)	anc(ANC)
ivoteda(DA)	julius sello malema(EFF)	helenzille(DA)
nkandla(ANC)	mmusi maimane(DA)	zuma(ANC)
zuma(ANC)	myanc_(ANC)	maimaneam(DA)
iecmustanswer(Unknown)	agangsa(Agang)	malema(EFF)
togetherforchange(DA)	jacob g. zuma(ANC)	ayisafani(ANC)
wecanwin(ANC)	iecsouthafrica(Unknown)	amp(Unknown)
voteda(DA)	mamphela ramphele(Agang)	elections2014(Unknown)
20yrsdemoc(Unknown)	whyivoteanc(ANC)	sabc(Unknown)

is positive therefore we classify that the tweet belongs to an author who supports ANC.

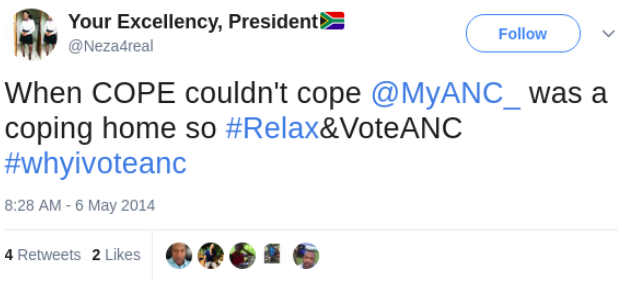

Fig. 3. A tweet posted during South Africa Presidential Elections

We also use tweet content and meta data to create features. The number of mentions, hashtags, and media items contained in the tweet history are included as features, along with the number of unique words contained in the original tweet content. To estimate acceleration or deceleration of the cascade we also include the the number of retweets in the 1st 10 minutes and the number of retweets in 2nd 10 minutes after the original tweet gets posted.

We include user related features such as whether the user is verified and the number of followers. User features defined at the node level and input into the GCN layer are non-zero only if the user participated in the tweet in the first 20 minutes after the original post. Table II lists the three types of features that we extract from the dataset.

IV. EXPERIMENTS

A. Dataset

We evaluate our model by testing it on data from Twitter collected during presidential elections in Kenya (2013) and South Africa (2014). Subject-matter experts who studied the elections were first asked to generate semi-structured descriptions of each election. These descriptions included collections of keywords, phrases, and individuals for which we can search in a social media dataset. Textual expansion was then performed by querying an undirected Twitter sample for content that matched the expert-generated keywords. Keywords with a strong co-occurrence connection with the original query but are rare in the subject-matter expert list were then added to the original query. After expanding expert queries, we searched Twitter's full historical archive to acquire a large dataset of Tweets for each election using Gnip's native support for textual, social, spatial, and temporal queries to search for relevant Tweets posted from within the target country, all of which are restricted to 60 days prior and 30 days following the election date in each country.

We then restricted to retweet cascades during the elections results in 22,572 sequence for South Africa and 140,521 retweet sequences for Kenya. We further pruned the dataset to retweet sequences having reshare count 10 or more. Table III summarizes the number of retweet sequences.

B. Baseline Algorithms

We compare our approach to both statistical and machine learning baseline models:

- SEISMIC [16]: A self-exciting Hawkes point process model that predicts retweet count.
- Linear Regression using tweet-level features.
- Feed Forward Neural Network using tweet-level features. A network with two hidden

TABLE II

DIFFERENT TYPES OF FEATURES

Feature Type	Feature	Type
Tweet Related Features	# Mentions	Numeric
	# Unique Words	Numeric
	# Hashtags	Numeric
	# Media Items	Numeric
	# Retweets in 1st 10 minutes	Numeric
	# Retweets in 2nd 10 minutes	Numeric
User Related Features	# Followers	Numeric
	# Friends	Numeric
	Is Verified	Binary
Sentiment Related Features	Polarity of the tweet	Categorical
	Party supported in the tweet	Categorical

TABLE III

RESULTS OF DATASET PRUNING

Dataset	(#) before pruning	(#) after pruning
Kenya	140521	9579
South Africa	22572	1677

layers (for comparison with two GCN layers) and with ReLU activation function.

For linear regression, we use the scikit-learn module available in Python. The neural network models (feedforward and GCN) are implemented in Tensorflow and training was conducted using ADAM optimizer with a learning rate of .001. The SEISMIC model is implemented as a package in R.

C. Evaluation Metrics

Given the first 20 minutes of a retweet cascade for tweet i, our goal is to predict the final retweet count y_i (cutoff to 1 month following the original post). For fake news mitigation, being able to flag the most viral tweets is of importance and therefore we use precision in the top k tweets as a metric (i.e. what % of tweets predicted to be in the top k actually were in the top k). We also use Mean Absolute Error and the r^2 score:

- Mean Absolute Error (MAE) is the average of the absolute difference between predicted and original values. It is defined as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|(y_i - \hat{y}_i)|$$

- The r^2 score also referred to as the coefficient of determination is a statistical measure of how close the predicted values are mapped to the regression line. It is defined as follows:

$$r^2 \ = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \mu)^2}$$

- Precision@k identifies how many of the the predicted popular(TP) tweets are actually popular(TP, FP). It is defined as follows:

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

D. Experimental Results

We use a 70/30 train/test split to evaluate all models. First, we conducted an experiment to investigate how the different components of the GCN, in particular the GC-layers and the tweet-level layer, affect the performance of the model. Table IV shows the MAE value of the GCN model with different network component combinations.

TABLE IV

COMPONENT COMBINATIONS AND PERFORMANCE OF THE GCN MODEL

Dataset	Only Actor features	Only Tweet features	Both Tweet and Actor features
Kenya	12.47	11.43	**7.34**
South Africa	13.20	14.26	**10.18**

To check how well GCN performs at ranking viral tweets, we classified tweets into popular and non-popular tweets (with 10% as the threshold). In V we present precision@k values where k is chosen to contain the top 10% of tweets in terms of

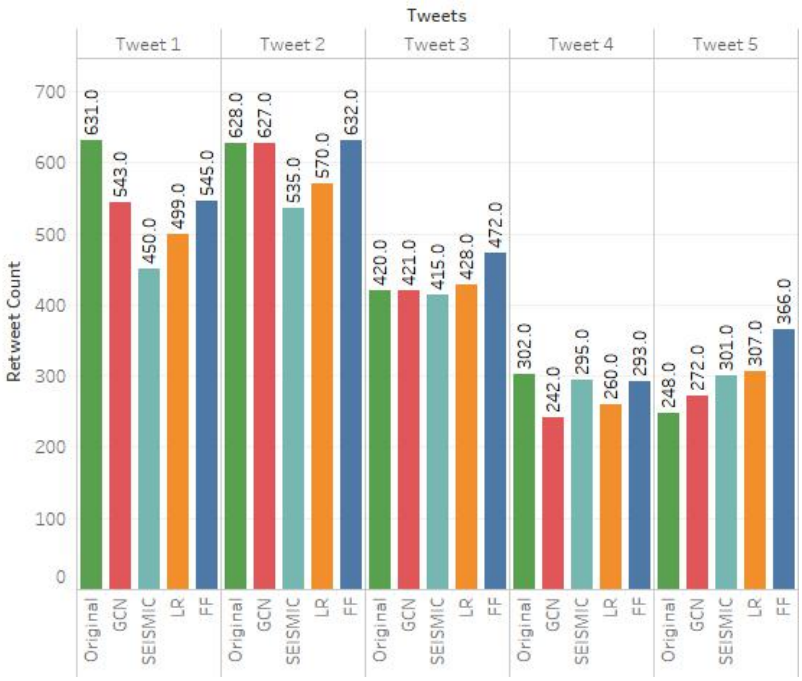

Fig. 4. Top five trending tweets in Kenyan Presidential Elections - Comparison between original retweet count and the predicted values from all the models

retweet count. We find that the GCN significantly improves upon SEISMIC and the other baseline models in terms of its ability to rank top viral tweets.

TABLE V

PRECISION SCORES FOR THE POPULAR TWEETS

Model	Kenya Elections	South Africa Elections
SEISMIC	0.66	0.66
Linear Regression	0.7	0.53
Feed Forward Neural Network	0.46	0.48
Graph Convolution Neural Network	**0.77**	**0.72**

In Table VI we the mean absolute error observed on all models when tested on the two data sets. In South Africa GCN is second to SEISMIC, however outperforms the other methods for Kenyan elections.

Table VII shows how well all models fit to the regression line (r^2) with 72-73 percent of variance explained by the GCN. Here GCN is 1st for South Africa, though it is 3rd for Kenya. However, we believe precision is the most important metric for flagging viral tweets during the election.

To confirm that the GCN is learning interesting network features, we also added an experiement where we use spectral embedding rather than the GC-layers in the neural network. In particular, we applied spectral decomposition to the adjacency matrix and used the top 10 components of the

TABLE VI

COMPARISON OF MEAN ABSOLUTE ERROR REPORTED BY ALL THE MODELS

Model	Kenya Elections	South Africa Elections
SEISMIC	9.31	**8.27**
Linear Regression	7.97	11.85
Feed Forward Neural Network	7.75	14.26
Graph Convolution Neural Network	**7.34**	10.18

TABLE VII

COMPARISON OF R2 SCORE REPORTED BY ALL THE MODELS

Model	Kenya Elections	South Africa Elections
SEISMIC	0.66	0.66
Linear Regression	**0.86**	0.59
Feed Forward Neural Network	**0.86**	0.32
Graph Convolution Neural Network	0.72	**0.73**

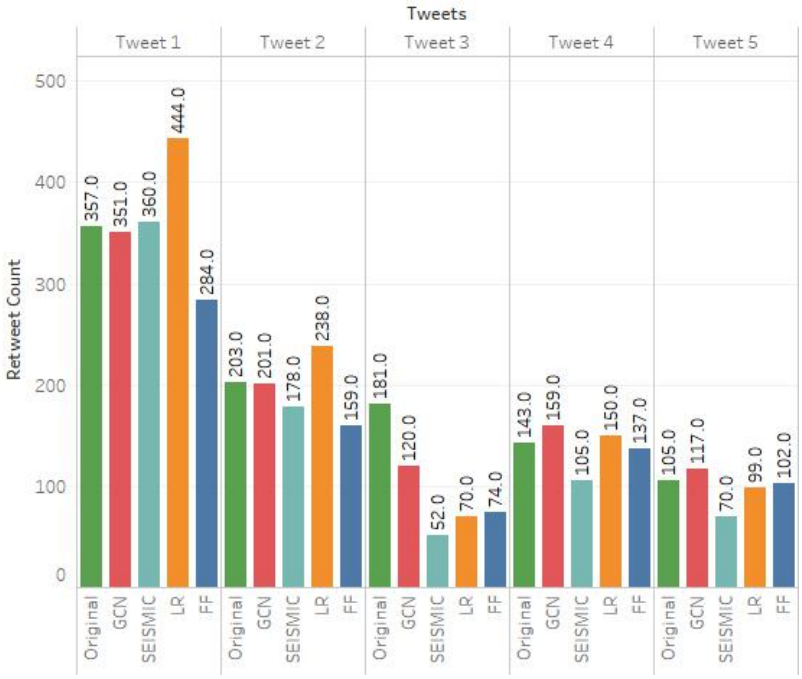

Fig. 5. Top five trending tweets in South African Presidential Elections - Comparison between original retweet count and the predicted values from all the models

spectral embedding as features. For each tweet sequence, we concatenate tweet related features with the embedding result for the adjacency matrix. Table VIII lists the MAE and r^2 score reported, which is significantly worse than the GCN.

TABLE VIII
PERFORMANCE EVALUATION ON FEED FORWARD NEURAL NETWORK MODEL WITH SPECTRAL EMBEDDING OF ACTORS ADJACENCY MATRIX

Dataset	Mean Absolute Error	R2 Value
Kenya	16.12	0.09
South Africa	10.82	0.38

Figures 4 and 5 compare the retweet count for the top five trending tweets in the Kenyan and South African elections dataset.

V. CONCLUSION

In this work, we introduced a graph convolution neural network model for early prediction of retweet count during elections. The three types of features extracted from the dataset were tweet related, user related, and sentiment related. We evaluated our model against a popular statistical model SEISMIC, feed forward neural networks, and a GCN without tweet-level information. For the top 10 percent trending tweets, our model predicts popularity with up to 15% better precision compared to compared to SEISMIC.

VI. FUTURE WORK

There are two areas where we are interested in taking this work further. Currently, the computations are handled on a CPU because the adjacency matrices do not fit in memory on a GPU. Constructing methods such that the operations AX^lW^l can be efficiently distributed across multiple GPUs or using an in-memory framework like Spark will be important to scale this type of method. Similar to the work in [19], [20], another research direction would be to extend recurrent neural point processes to the graph convolution framework to better handle the time dynamics of retweet cascades while still incorporating tweet level features and graph feature learning for which the GCN is designed.

VII. ACKNOWLEDGEMENTS

This work was supported in part by NSF grants SCC-1737585, SES-1343123, and ATD-1737996.

REFERENCES

[1] Marcel Broersma and Todd Graham. Social media as beat: Tweets as a news source during the 2010 british and dutch elections. *Journalism Practice*, 6(3):403–419, 2012.

[2] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[3] Richard Gunther, Paul Beck, and Erik Nisbet. Fake news may have contributed to trump's 2016 victory, 2018.

[4] Mehrdad Farajtabar, Jiachen Yang, Xiaojing Ye, Huan Xu, Rakshit Trivedi, Elias Khalil, Shuang Li, Le Song, and Hongyuan Zha. Fake news mitigation via point process based intervention. *arXiv preprint arXiv:1703.07823*, 2017.

[5] Eytan Bakshy, Brian Karrer, and Lada A. Adamic. Social influence and the diffusion of user-created content. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, EC '09, pages 325–334, New York, NY, USA, 2009. ACM.

[6] Andreas Kanavos, Isidoros Perikos, Pantelis Vikatos, Ioannis Hatzilygeroudis, Christos Makris, and Athanasios Tsakalidis. Modeling retweet diffusion using emotional content. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 101–110. Springer, 2014.

[7] Ethem F. Can, Hüseyin Oktay, and R. Manmatha. Predicting retweet count using visual cues. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 1481–1484, New York, NY, USA, 2013. ACM.

[8] Roja Bandari, Sitaram Asur, and Bernardo A. Huberman. The pulse of news in social media: Forecasting popularity. *CoRR*, abs/1202.0332, 2012.

[9] Bálint Daróczy, Róbert Pálovics, Vilmos Wieszner, Richárd Farkas, and András A Benczúr. Predicting user-specific temporal retweet count based on network and content information. In *INRA@ RecSys*, pages 6–13, 2015.

[10] Gang Liu, Chuan Shi, Qing Chen, Bin Wu, and Jiayin Qi. A two-phase model for retweet number prediction. In *International Conference on Web-Age Information Management*, pages 781–792. Springer, 2014.

[11] Liangjie Hong, Ovidiu Dan, and Brian D. Davison. Predicting popular messages in twitter. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 57–58, New York, NY, USA, 2011. ACM.

[12] Karthik Subbian, B Aditya Prakash, and Lada Adamic. Detecting large reshare cascades in social networks. In *Proceedings of the 26th International Conference on World Wide Web*, pages 597–605. International World Wide Web Conferences Steering Committee, 2017.

[13] Zi Yang, Jingyi Guo, Keke Cai, Jie Tang, Juanzi Li, Li Zhang, and Zhong Su. Understanding retweeting behaviors in social networks. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1633–1636, New York, NY, USA, 2010. ACM.

[14] Tauhid Zaman, Emily B. Fox, and Eric T. Bradlow. A bayesian approach for predicting the popularity of tweets. *CoRR*, abs/1304.6777, 2013.

[15] Manuel Gomez-Rodriguez, Jure Leskovec, and Bernhard Schölkopf. Modeling information propagation with survival theory. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–666–III–674. JMLR.org, 2013.

[16] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. SEISMIC: A self-exciting point process model for predicting tweet popularity. *CoRR*, abs/1506.02594, 2015.

[17] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, 2016.

[18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[19] Eric W. Fox, Martin B. Short, Frederic P. Schoenberg, Kathryn D. Coronges, and Andrea L. Bertozzi. Modeling e-mail networks and inferring leadership using self-exciting point processes. *Journal of the American Statistical Association*, 111(514):564–584, 2016.

[20] Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen M Chu. Modeling the intensity function of point process via recurrent neural networks. In *AAAI*, pages 1597–1603, 2017.