From Retweets to Follows: Facilitating Graph Construction in Online Social Networks Through Machine Learning

Anahit Sargsyan[1 [0000−0001−8142−8591]] and Jürgen Pfeffer[1 [0000−0002−1677−150X]]

School of Social Sciences and Technology, Technical University of Munich, Munich, Germany {anahit.sargsyan,juergen.pfeffer}@tum.de

Abstract. Online social networks (OSNs), such as Twitter and Facebook, enable users to create, share, and interact with diverse content, thereby producing intricate pathways for information propagation. This flow, which can be modeled through graphs that capture Follower/Following relationships and various interactions such as retweets and mentions, can offer valuable insights into the dynamics of online social behavior and information sharing. While the Follower/Following networks are important for modeling user characteristics and behaviors, their construction can prove expensive in terms of both time and resources. More importantly, in some OSNs, partial or full restrictions have been posed on the access to users' Follower/Following information, effectively rendering the regular construction process of Following graphs intractable. In this paper, we explore the viability of extracting users' Following connections from their Retweet/Mention networks through predictive models. Taking Twitter as a case study, we train and contrast the performance of five different models, including classical Machine Learning (ML) methods as well as a recently developed Deep Learning (DL) approach, on two different datasets. The difference in prediction results across the models and datasets is traced and analyzed. Lastly, we round up the contributions by providing a carefully curated Twitter dataset compiled from over 9,000 individuals' timelines, encapsulating their retweets, followers, and following networks. Taken together, the results and findings featured herein can aid in paving the way for improved understanding and modeling of online social networks.

Keywords: Online Social Networks · Follow Graph · Link Prediction · Machine Learning · Egocentric Networks.

1 Introduction

Online social platforms, such as Twitter (currently X) and Facebook, allow users to form Follower/Following connections, generate content on their timelines, and interact with content generated by other users, e.g., by liking, mentioning, sharing, commenting, and retweeting. This enables the content to spread from one user to another via different pathways. Each of these interactions can then be

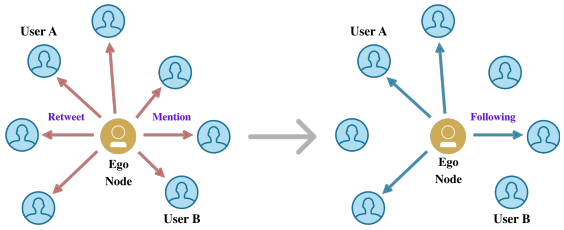

Fig. 1: An abstract illustration of the studied problem of inferring the ego user's Following connections (on the right) from their Retweet/Mention Network (on the left).

used to build graphs that model relationships between users (e.g., the graphs portrayed in Fig. 1) or between a user and content. Studying these graphs could help model and predict the evolution of OSNs from both social and information point of view. The established models, in turn, can be leveraged to improve the current systems and develop new applications in OSNs, assist advertisers and marketers in designing more effective campaigns, and enrich user behavior analysis.

Among various OSNs, Twitter stands out for its fast-paced nature, massive user base, and unique format of communication through short, public messages. With millions of active contributors ($\approx$ 40 M.) and hundreds of millions of tweets ($\approx$ 375 M.) posted daily [15], Twitter has become a valuable asset for researchers and analysts seeking to understand social dynamics, information dissemination, and user behavior in online spheres. To this end, prior works have extensively analyzed the structural and topological properties of Follow, Retweet, and Mention graphs on Twitter, revealing interesting insights into connections/correlations between them. In [13], Myers et al. examined the Twitter Follow graph's topological properties, concluding that while the Follower/Following relationship is primarily associated with *information consumption*, it often indicates connections rooted in *social ties*. Further investigations into the Twitter ecosystem have revealed that, as compared to the Follow graph, the Retweet graph can effectively capture *genuine interest and trust connections* among users [4]. Complementing the above two studies, Amati et al. [2] explored the Twitter Mention graph and provided a qualitative and quantitative comparison between the three graphs. As transpired from the analysis, the Mention graph tends to better capture the information spreading on the network and, perhaps more interestingly, can provide a quantitative assessment of the *actual strength* of a Follow relationship.

Being the most natural and intuitive, Follower/Following networks have attracted considerable interest in the literature. Besides influencing the information flow within OSNs, these networks can reveal valuable insights into community structures, influence dynamics, and user preferences. In fact, the graphs constructed from Follower/Following connections have been utilized for enhancing the prediction of users' political preferences [6], fake news identification [18], early prediction of virality of tweets during incidents [21], to name a few.

Despite their utility and potential, however, Follow networks demand *significantly more time and effort* to be constructed in comparison to Retweet or Mention networks. Particularly, unlike the latter two, which can be built without relying on API access, the compilation of Follow networks involves recursive calls, and the number of users returned in each response is typically restricted. Furthermore, in some OSNs, access to users' Follower/Following connections has been artificially *limited or disabled completely*. Case in point, currently in Twitter, in addition to restricted and paywalled access to APIs, limitations have also been imposed on viewing the Follower/Following connections.

In response to these obstacles, and in an effort to facilitate the construction of large-scale Follow networks in OSNs, this paper seeks to answer the question whether information from users' retweets and mentions can be leveraged to accurately predict their Follow connections. Taking Twitter as a case study, we train and evaluate the performance of five different predictive models on two different datasets. The difference in prediction results across the models and datasets is traced and analyzed. In summary, the present work complements and advances the existing research as elaborated below.

First, we propose to extract (infer) users' Following network from their Retweet/Mention network (i.e., a combination of Retweet and Mention graphs) through predictive models. We cast this problem both as a binary classification task and as an edge classification problem within a graph. Five different predictive models are evaluated, and their performance comparison is provided. The results demonstrate that an ensemble tree-based classifier can achieve an average accuracy and an F1-score of nearly 90% for certain network types, yet its performance may significantly vary depending on the degree of the input and output networks. Specifically, the findings suggest a positive correlation between the prediction accuracy and the degree of Retweet/Mention network, whereas with respect to the degree of the Following network the opposite trend is observed.

Second, we provide a Twitter dataset compiled from around 9,000 individuals' timelines, including their retweets, followers, and following networks. The *dataset*, *trained models*, and the complete source code to reproduce the conducted analysis can be accessed online at http://bit.ly/3y5zaRg.

The roadmap of this paper is as follows. In Sec. 2, we briefly survey the related literature on link prediction methods in social networks. Sec. 3 provides a formal description of the problem and lays out the details of the two considered datasets. In Sec. 4, we overview the employed candidate predictive models, and in Sec. 5 report the results of their performance comparison. Lastly, Sec. 6 concludes the paper with a discussion and an outlook on future work.

2 Related Work

In what follows, we briefly survey the existing studies on link prediction methods, whereas Table 1 provides an overall comparison between the present work and the literature reviewed. For a more exhaustive review on link prediction in social networks, we refer the reader to [10, 12].

Prediction of connections in OSNs has been widely studied within the framework of the classical link prediction problem (LPP) [11]. In LPP, given the current snapshot of a social network, the objective is to predict the missing or future links. On the other hand, here, the goal is to infer a different type of link from the current graph (i.e., extracting the Following relationships from a Retweet/Mention graph). This problem can be modeled as an LPP on a heterogeneous graph provided one has partial information regarding Follow connections, whereas in the current setting, we assume a complete absence of the latter information.

The domain of link prediction in social networks has garnered a rich arsenal of techniques and approaches, ranging from topological metric-based analyses to Machine Learning models. Early research, such as the work by Liben-Nowell et al. [11], explored classical topological metrics, including common neighbors, Jaccard's coefficient, and Adamic-Adar for link prediction in co-authorship networks. Following this direction, Ahmad et al. [1] proposed a link prediction algorithm named Common Neighbor and Centrality-based Parameterized algorithm, which can suggest the formation of new links in complex networks. Hours et al. [8] analyzed a large Twitter dataset aiming to predict mention links among users by combining contextual and local structural features.

While topological metrics can offer a simplistic yet insightful way to predict links, there has been a shift towards machine learning-based methods. One subdirection in this line of research employs network representation learning (NRL) based algorithms for link prediction, such as DeepWalk [14], Node2vec [7], and embedGAN [9]. Zhang et al. [22] proposed a Graph Neural Network (GNN) based link prediction framework, referred to as SEAL, that can jointly learn from three types of features (subgraphs, embeddings, and attributes). Praznik et al. [17] applied the SEAL framework to three sets of Twitter data to predict future hashtag co-occurrences. Focusing on follower connections, Behera et al. [3] proposed a link prediction model based on XGBoost classifier. The authors included features such as Katz centrality, Page Ranking for the nodes, and profile-related features. Leveraging social cognitive theories, Toprak et al. [20] proposed supervised and unsupervised social circle-aware link prediction methods for egocentric graphs. The method has been validated on two Twitter datasets comprising a community of video gamers and generic users against several benchmarks, including SEAL and Node2vec.

Table 1: A comparative summary of present work versus prior studies on link prediction in social networks.

	Objective	**Network Type**	**Graph Structure**	**Approach**	**Application Scope**
Liben-Nowel et al. [11]	Link prediction	Evolving	General	Structural similarity based	Social networks
Ahmad et al. [1]	Link prediction	Evolving	General	Structural similarity based	Social networks
Perozzi et al. [14]	NRL	Static	General	Random walks	Social networks
Grover et al. [7]	NRL	Static	General	Second-order random walks	General networks
Jin et al. [9]	Link prediction	Static	General	Generative Adversarial Network based	General networks
Hours et al. [8]	Link prediction	Evolving	General	Structural and contextual similarity based	OSN
Zhang et al. [22]	Link prediction	Evolving	General	GNN with network representation embeddings	General networks
Behera et al. [3]	Follower link prediction	Evolving	General	XGBoost	Social networks
Toprak et al. [20]	Link recommendation	Evolving	Egocentric	Social circle-aware and structural similarity based	OSN
Present work	Following link prediction	Static	Egocentric	XGBoost	OSN

3 Problem Statement and Datasets

Recall that given a user's Retweet/Mention network, the problem at hand seeks to extract the corresponding Following connections. More formally, let $G_{\mathrm{RT}} = (V, E)$ be a directed parameterized graph denoting the given Retweet/Mention network of an ego user, with V denoting the set of all users (i.e., the ego user and those whom the ego user mentioned or retweeted) and E denoting the connecting directed edges. Note that in our case $|V| = |E|+1$ as the graph G_{RT} is egocentric. Every vertex $v \in V$ is associated with a set f^v of n *user-level features* $f^v \triangleq \{f_1^v, f_2^v, ..., f_n^v\}$ and every edge $e \in E$ is characterized by a set k^e of m interaction related features $k^e \triangleq \{k_1^e, k_2^e, ..., k_m^e\}$ which are detailed in the paragraphs to follow. The objective is to infer a direct mapping $G_{\mathrm{RT}} \rightarrow G_{\mathrm{F}} = (V, \tilde{E})$, where G_{F} defines the corresponding network of Following relationships of the ego user and $\tilde{E} \subseteq E$.

Alternatively, as previously noted, one can formulate the above problem as a binary classification task where the input is represented by a real-valued vector formed by concatenating the user-level, edge-level, and graph-level features. To formalize, let $X \in \mathbb{R}^{2n+m+3}$ denote the input vector constructed by concatenating f^v of the ego-user with that of the alter, along with the corresponding edge-level features k^e as well as the following three additional graph-level features: (1) *ego_ degree*: Degree of G_{RT}; (2) *avg_ tweet_ impression_ count*: Average impression count of ego-user's tweets within G_{RT}; (3) *avg_ alter_ tweet_ count*: Average number of tweets of alters within G_{RT}.

With the above notation, the problem then translates into learning a mapping of the form $X \rightarrow Y \in \{0, 1\}$, where the prediction target Y denotes the absence or presence of a Following connection. With this framing, the problem can be tackled by standard ML classification methods, which are discussed in Sec. 4.

3.1 Datasets

As previously noted, to provide a more comprehensive comparison of the selected predictive models (explained in Sec. 4), we test their performance on two different datasets, TwiBot–22 [5] and our manually compiled data corpus, termed CodeSwitchNet. The key difference between these two datasets lies in the approaches to data collection and the intended application, as elaborated below.

TwiBot–22: TwiBot-22 dataset [5] was collected primarily for evaluating and enhancing bot detection algorithms on Twitter. The data collection was conducted in two stages, from January 2022 to March 2022. A rich collection of 1,000,000 user profiles was collected, including bots and genuine users, along with their associated tweets and various relationships, such as followers, following, likes, and retweets. Though these relationships are present, they are not exhaustive, offering a representative rather than a comprehensive view.

CodeSwitchNet: Arabizi (transliterated Arabic) seed words were used to collect tweets from 2020 to 2023 using Twitter's Academic API v2 [16]. From over

Table 2: Statistics of the two datasets.

	Number of egos	*Number of nodes*	*Number of edges*	*Median* G_{RT} **degree**	*Median* G_F **degree**
CodeSwitchNet	9,155	573,075	2,169,388	205	23
TwiBot–22	7,849	350,500	1,337,517	142	40

1,000,000 users that were in the dataset, we randomly selected 10,000 users who used code–switching and, more precisely, used a mixture of Arabic, English, French, and Arabizi to collect their timelines, Follower, and Following networks. Out of the selected users, 9,155 were public and accessible at the time of data collection which took place from May 2023 to June 2023. Due to the constraints imposed by Twitter API, up to 3,000 latest tweets were collected from each user's timeline, based on which the corresponding Retweet/Mention networks were constructed.

3.2 Data Preprocessing

For data consistency, the users flagged as bots were removed from Twibot-22, whereas for CodeSwitchNet, no user filtering steps were taken in this regard. Table 2 provides further details regarding the two datasets after preprocessing.

As seen from Table 2, the average size of ego networks varies considerably among the two datasets, which stems from the nature of data collection for each. Specifically, CodeSwitchNet encapsulates the complete network of the users at the time of data collection, as opposed to TwiBot–22, which captures only a subset of the connections. Another major difference between these two datasets is related to the median degree of ego users' Following graphs (i.e., G_{F}), which in TwiBot–22 is nearly double the value in CodeSwitchNet. In contrast, the median degree of Retweet/Mention graphs (i.e., G_{RT}) is higher in CodeSwitchNet than in TwiBot–22. These potentially indicate the presence of different Following patterns in these two datasets.

Table 3: Node and edge level features.

User-level	**Edge-level**
f_1^v: Account age in years	k_1^e: Avg. no. of retweets per interaction
f_2^v: Followers to Following ratio	k_2^e: Avg. no. of replies per interaction
f_3^v: Total no. of tweets	k_3^e: Avg. no. of quotes per interaction
f_4^v: Account is verified	k_4^e: Avg. no. of likes per interaction
f_5^v: G_{RT} degree	k_5^e: Avg. no. of impressions per interaction
f_6^v: No. of lists the ego has been added to	k_6^e: No. of tweets with the alter mentioned
	k_7^e: No. of times ego retweeted alter
	k_8^e: Avg. no. of tweets by ego in G_{RT}

To create the networks for the users, we used their timelines to extract each Twitter handle appearing in a tweet. Each occurrence of a handle can be either a retweet or a mention, hence the graph naming. We further extracted features describing both the profile of a user and the interaction between ego and alter. Specifically, six user level features $(f_1^v, f_2^v, ..., f_6^v)$ and eight edge-level features $(k_1^e, k_2^e, ..., k_8^e)$ were considered as listed in Table 3. The features k_1^e, k_2^e, k_3^e, k_4^e and k_5^e were not available in TwiBot–22, therefore the predictive models were trained on each dataset separately.

4 Predictive Models

The pool of candidate approaches selected for evaluation involves two categories of models, which are discussed below. For each model, the same split of data, 80%-20%, was used for training and testing, ensuring that all ego networks were present in both sets. The results reported in Sec. 5 are based on the test set.

Statistical ML Models: From this category, we consider both standard classification models, such as Logistic Regression, distance-based method, such as k-Nearest Neighbors (kNN), and two ensemble tree-based models, namely Random Forest (RF) and XGBoost. (kNN) was trained with $k = 11$ with the distance metric set to Euclidean. For RF, the number of estimators was kept at the default value of 100, while the maximum depth was adjusted to 50. For XGBoost, the number of estimators was increased to 500, maximum depth and learning rate were set to 75 and 0.01, respectively.

Graph-based DL Approach: Directed Graph Convolutional Network (DGCN) [19] was selected, which is tailored to leverage both first-order and second-order proximity information between nodes. Given our focus on direct connections, a single layer of convolution was used, since it was sufficient to capture the relevant information efficiently. Furthermore, to adapt the DGCN for edge classification, we concatenated the embeddings of adjacent nodes with edge-specific features and then applied fully connected layers followed by Softmax activation to derive predictions for edges. The model was trained with a learning rate of 0.001 and weight decay of 0.005. The number of epochs was set to 500 with an early stopping condition.

Baseline Model: As an additional reference, we employ a naive classifier, referred to as Baseline. We parameterize this model by a scalar θ, which measures the number of retweets/mentions. Two values for θ are considered: (i) Baseline $[\theta > 0]$: If the ego user retweeted/mentioned the alter, then a Following connection is predicted; (ii) Baseline $[\theta >$ Avg.$]$: Following connection is predicted if the number of retweets/mentions is greater than the average number of retweets/mentions within the same ego network.

5 Prediction of Following Relationships

In this section, we first evaluate and compare the predictive performance of the selected candidate methods. Then, we scrutinize the performance of the best-

performing model, revealing the most important features and investigating the factors affecting the accuracy of predictions.

5.1 Comparative Analysis

Table 4 reports the performance of the models with respect to accuracy, precision, recall, and F1-score. As the performance of the naive Baseline [$\theta > 0$] indicates, interaction alone (i.e., retweeting or mentioning) does not serve as an accurate predictor of a following connection achieving F1-score of only 47% on TwiBot–22 dataset, and an even lower score on CodeSwitchNet. Due to the threshold value of 0, this method classifies all the edges as Following connections (i.e., the positive class), hence the Recall score of 1. When the threshold value is with respect to the average number of interactions within the ego network, the overall performance of this baseline model slightly improves. This suggests that while the frequency of retweets/mentions is an informative feature, it lacks precision.

Relative to the Baseline, the other models exhibited significantly improved accuracy. Yet, as can be seen from Table. 4, their performance varies notably across the two datasets. Specifically, XGBoost achieves 74% on TwiBot–22, compared to 88% on CodeSwitchNet. The difference of around 14% is consistently observed in other models. In some models the difference is larger, for example in DGCN. This disparity can be attributed to the difference in the statistics and characteristics of the datasets (see Table 2). In the following subsection we investigate this observation more closely.

Compared to the statistical ML models, DGCN demonstrated an overall inferior performance. On TwiBot–22, DGCN attained an F1-score of only 57%, closely approaching the score of the baseline model. Note that a similar behavior was observed in [20]. Though convolutional graph neural networks can offer powerful prediction capabilities, these were not originally designed for one-layer ego-centric graphs. Considering this, the low performance on the current datasets at hand was expected.

Table 4: Prediction results for each candidate method broken down by datasets. The scores represent the weighted average of the two classes, and the best ones are highlighted in bold.

	Twibot-22				**CodeSwitchNet**			
Method	*Accuracy*	*Precision*	*Recall*	*F1-score*	*Accuracy*	*Precision*	*Recall*	*F1-score*
Baseline [$\theta > 0$]	0.30	0.30	**1.00**	0.47	0.18	0.18	**1.00**	0.31
Baseline [$\theta >$ Avg.]	0.69	0.49	0.37	0.42	0.76	0.35	0.38	0.37
Logistic Regression	0.74	**0.74**	0.74	0.68	0.82	0.78	0.82	0.76
kNN	0.64	0.62	0.64	0.63	0.83	0.80	0.83	0.81
Random Forest	0.75	**0.74**	0.76	**0.74**	**0.89**	**0.88**	0.89	**0.88**
XGBoost	**0.76**	**0.74**	0.76	**0.74**	**0.89**	**0.88**	0.89	**0.88**
DGCN	0.70	0.48	0.70	0.57	0.79	0.71	0.79	0.74

Among the considered models, ensemble tree-based models, XGBoost and Random Forest, achieved the highest performance metrics, closely following each other. Both yielding an F1-score of 74% on TwiBot–22 and 88% on CodeSwitchNet respectively. In the following section the variation is investigated more closely.

5.2 Performance Scrutiny of XGBoost

To identify the most influential factors that affected the performance of XGBoost, we extracted the feature importance scores, which are depicted in Tab. 6. The two most prominent features concern the number of retweets and mentions. These observations are consistent with the findings in [2], which suggest that Mentions can quantify the actual strength of Follow connections, thus providing some measure of validation. Additionally, the degree of the ego is also relatively important.

Next, in order to shed light on the performance disparity of XGBoost across the two datasets, we partition the ego-users in TwiBot–22 and CodeSwitchNet, and analyze the accuracy of predictions for each category separately. First, we group the ego-users by the degree of their Retweet/Mention networks (i.e. G_{RT}) into 3 classes with divisions at the 25^{th}, 50^{th}, and 75^{th} percentiles. Both in CodeSwitchNet and TwiBot–22, we observe that the performance accuracy of XGBoost increases as the size of ego users' G_{RT} increases. This suggests that extracting Following connections from a larger Retweet/Mention networks is comparably easier.

Second, we cluster the ego-users by the degree of their Following networks (i.e., G_{F}) into 3 classes with divisions at the 25^{th}, 50^{th}, and 75^{th} percentiles. In contrast to the above pattern, the accuracy of XGBoost seems to be negatively correlated with the size of the network. In both CodeSwitchNet and TwiBot–22, the performance decreases as the size of the Following networks increases. This implies that prediction of Following connections is relatively easier when ego-user follows only a few users whom they mention or retweet.

The above findings indicate that the size of Retweet/Mention networks can play a significant role, and the choice of the predictive model depends largely on the dataset. While for datasets like CodeSwitchNet, with large G_{RT} networks and small G_{F}, ensemble tree models can achieve reasonably high accuracy, for other datasets more advanced predictive models and additional features should be

Table 5: Accuracy broken down by degree category of G_{RT} and G_{F}. Degree categories are calculated based on quantiles for both datasets. The numbers in parentheses indicate the mean and standard deviation.

	G_{RT}			G_{F}		
	Low degree (102.28 ± 140.05)	***Medium degree*** (209.70 ± 164.21)	***High degree*** (303.45 ± 144.95)	***Low degree*** (27.00 ± 20.40)	***Medium degree*** (176.04 ± 66.45)	***High degree*** (441.29 ± 129.41)
CodeSwitchNet	0.75	0.85	0.90	0.88	0.85	0.79
TwiBot–22	0.68	0.73	0.77	0.76	0.73	0.68

Table 6: Top 5 Feature importance scores extracted from the XGBoost model.

Feature	mentioned_count	retweeted_count	ego_account_age	tweet_retweet_count	ego_follower_following_ratio
Importance	0.439	0.168	0.117	0.044	0.035

considered. For example, to incorporate features that can capture the interaction nuances between ego-users and alters, the topic and sentiment of the tweets.

6 Concluding Remarks

In this study, we explored the problem of predicting users' Following networks on Twitter from their Retweet/Mention graph. Our approach circumvents the traditional challenges associated with the construction of following networks, particularly on platforms like Twitter that have inherent data access constraints. As demonstrated through evaluations on two different datasets, an ensemble tree-based approach supplied with a moderate number of features can achieve sufficiently accurate results in predicting the Following connections for certain category of ego-networks, thereby reinforcing the potential of this research direction. However, we observe significantly degraded performance for cases when ego-users' Retweet/Mention network is small or when the target Following network is large, which calls for the development of more advanced prediction models.

One promising avenue for future exploration would be integrating a richer, possibly multi-modal, set of features into our models. By intertwining Natural Language Processing, sentiment analysis, and further contextual information about users, can drastically enhance predictive models' expressiveness, allowing us to capture subtler nuances of user behaviors and preferences.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ahmad, I., Akhtar, M.U., Noor, S., Shahnaz, A.: Missing link prediction using common neighbor and centrality based parameterized algorithm. Scientific Reports **10**(1), 364 (2020)
2. Amati, G., Angelini, S., Bianchi, M., Fusco, G., Gambosi, G., Gaudino, G., Marcone, G., Rossi, G., Vocca, P.: Moving beyond the twitter follow graph. In: 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K). vol. 1, pp. 612–619. IEEE (2015)
3. Behera, D.K., Das, M., Swetanisha, S., Nayak, J., Vimal, S., Naik, B.: Follower link prediction using the xgboost classification model with multiple graph features. Wireless Personal Communications **127**(1), 695–714 (2022)
4. Bild, D.R., Liu, Y., Dick, R.P., Mao, Z.M., Wallach, D.S.: Aggregate characterization of user behavior in twitter and analysis of the retweet graph. ACM Transactions on Internet Technology (TOIT) **15**(1), 1–24 (2015)

5. Feng, S., Tan, Z., Wan, H., Wang, N., Chen, Z., Zhang, B., Zheng, Q., Zhang, W., Lei, Z., Yang, S., et al.: Twibot-22: Towards graph-based twitter bot detection. Advances in Neural Information Processing Systems **35**, 35254–35269 (2022)
6. Golbeck, J., Hansen, D.: A method for computing political preference among twitter followers. Social Networks **36**, 177–184 (2014)
7. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864 (2016)
8. Hours, H., Fleury, E., Karsai, M.: Link prediction in the twitter mention network: impacts of local structure and similarity of interest. In: 2016 IEEE 16th International Conference on Data Mining Workshops. pp. 454–461. IEEE (2016)
9. Jin, H., Xu, G., Cheng, K., Liu, J., Wu, Z.: A link prediction algorithm based on gan. Electronics **11**(13) (2022)
10. Li, T., Wu, Y.J., Levina, E., Zhu, J.: Link prediction for egocentrically sampled networks. Journal of Computational and Graphical Statistics **0**(0), 1–24 (2023)
11. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. journal of the association for information science and technology (2007). Google Scholar Google Scholar Digital Library Digital Library (2007)
12. Martínez, V., Berzal, F., Cubero, J.C.: A survey of link prediction in complex networks. ACM Comput. Surv. **49**(4) (dec 2016)
13. Myers, S.A., Sharma, A., Gupta, P., Lin, J.: Information network or social network? the structure of the twitter follow graph. In: Proceedings of the 23rd International Conference on World Wide Web. pp. 493–498 (2014)
14. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710 (2014)
15. Pfeffer, J., Matter, D., Jaidka, K., Varol, O., Mashhadi, A., Lasser, J., Assenmacher, D., Wu, S., Yang, D., Brantner, C., et al.: Just another day on twitter: A complete 24 hours of twitter data. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 17, pp. 1073–1081 (2023)
16. Pfeffer, J., Mooseder, A., Lasser, J., Hammer, L., Stritzel, O., Garcia, D.: This sample seems to be good enough! assessing coverage and temporal reliability of twitter's academic api. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 17, pp. 720–729 (2023)
17. Praznik, L., Qudar, M.M.A., Mendhe, C., Srivastava, G., Mago, V.: Analysis of Link Prediction Algorithms in Hashtag Graphs, pp. 221–245. Springer International Publishing, Cham (2021)
18. Su, T., Macdonald, C., Ounis, I.: Leveraging users' social network embeddings for fake news detection on twitter. arXiv preprint arXiv:2211.10672 (2022)
19. Tong, Z., Liang, Y., Sun, C., Rosenblum, D.S., Lim, A.: Directed graph convolutional network. arXiv preprint arXiv:2004.13970 (2020)
20. Toprak, M., Boldrini, C., Passarella, A., Conti, M.: Harnessing the power of ego network layers for link prediction in online social networks. IEEE Transactions on Computational Social Systems **10**(1), 48–60 (2023)
21. Upadhyaya, A., Chandra, J.: Spotting flares: The vital signs of the viral spread of tweets made during communal incidents. ACM Transactions on the Web **16**(4), 1–28 (2022)
22. Zhang, M., Chen, Y.: Link prediction based on graph neural networks. Advances in neural information processing systems **31** (2018)