EDGE-UP: Enhanced Dynamic GNN Ensemble for Unfollow Prediction in Online Social Networks

Soheila Farokhi[1], Arash Azizian Foumani[1], Xiaojun Qi[1], Tyler Derr[2], and Hamid Karimi[1]

[1] Department of Computer Science, Utah State University
{soheila.farokhi, arash.foumani, xiaojun.qi, hamid.karimi}@usu.edu
[2] Computer Science Department, Vanderbilt University
tyler.derr@vanderbilt.edu

Abstract. In the complex landscape of online social networks, predicting unfollow events is challenging due to data sparsity, class imbalance, and the dynamic nature of user interactions. This paper presents **EDGE-UP**, an Enhanced Dynamic Graph Neural Network (GNN) Ensemble model adeptly designed to overcome these challenges in unfollow prediction. EDGE-UP leverages a large-scale, longitudinal Twitter dataset featuring 58 weekly snapshots across 118,890 users to capture the evolving social dynamics. It minimizes the need for extensive feature engineering by utilizing GNNs for spatial encoding and LSTMs for capturing temporal dynamics, addressing data sparsity and class imbalance through ensemble learning and negative sampling strategies. Our experiments demonstrate EDGE-UP's superior performance in accurately predicting unfollow events, setting a new standard in social network analysis, and offering versatile applicability across different platforms. The code and data are available here: https://github.com/DSAatUSU/edge-up.

Keywords: Unfollow Prediction · Dynamic Graph Neural Networks · Social Networks · Twitter (X)

1 Introduction

Online social networks, notably Twitter (X), are pivotal in the digital realm. They enable communication and interaction while serving as a substantial source of behavioral insights across various fields [8,27,6,7,14,1,9,24,10]. The dynamics of follower-followee relationships are vital for comprehending user behavior, and predicting unfollow events emerges as a critical challenge in this landscape. Figure 1 illustrates the evolution of social ties over time in a network, showcasing how users' actions of following (creating ties) and unfollowing (breaking ties) progressively modify the network's structure.

In this paper, we delve into the follower-followee relationships on online social networks, aiming to predict instances where users are likely to unfollow one another. The motivation behind this research stems from the need to have an **accurate** model that can analyze and **predict** the evolution of ties in a social network. Such a model comes with several benefits. Firstly, understanding the dynamics of forming and dissolving friendships on social media is crucial [26,17]. Secondly, analyzing how network structure influences tie breaks in online social networks is significant [13]. Thirdly, gaining insights into the evolution and decay of online relationships helps understand digital social dynamics [23]. Lastly, developing predictive models for follower loss, which incorporate factors like tweet content and engagement, is beneficial for proactive relationship management on social media platforms [18]. These reasons underscore the importance of unfollow prediction in comprehending and managing online social interactions, enhancing user experience, and informing social media strategies. However, unfollow prediction faces several challenges, reflecting the complex nature of social networks and human behavior. Some of the key challenges include:

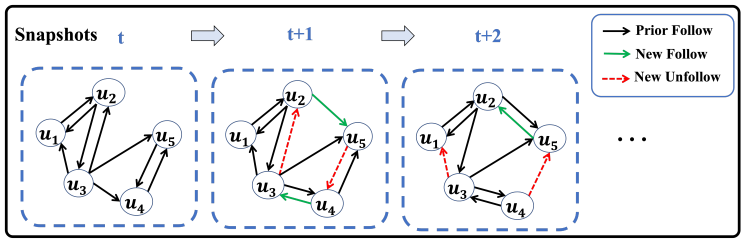


Fig. 1: Example of the evolution of ties in a social network as users follow or unfollow each other.

- **Lack of Large-scale and Longitudinal Data:** The absence of large-scale, longitudinal datasets that capture the dynamism of online social networks limits the effectiveness of unfollow prediction models in accurately understanding and forecasting user behavior over time. This limitation leads to models that fail to capture the complexity of network dynamics and user interactions, resulting in reduced predictive accuracy and challenges in developing robust, generalizable models. Unlike follow prediction (a.k.a link prediction), unfollow prediction remains largely unexplored [30], primarily due to the scarcity of high-quality data.
- **Dynamic Nature of Social Networks:** Social networks are dynamic, with user interactions and preferences evolving over time. Predicting unfollows requires models to adapt to changing patterns and user behaviors, making it challenging to build robust and accurate models.
- **Data Sparsity and Class Imbalance:** Social network data is often sparse, and there is a significant class imbalance between unfollow incidents compared to follow ones– See Section 3. This can lead to biased models that struggle to generalize well, as the majority class (users who do not unfollow) may dominate the training data.
- **Hand-crafted Specific Features:** Prior research on unfollow behavior prediction in social networks has predominantly utilized platform-specific manually crafted features [30,18], which are resource-intensive and limit the *generalizability* of the models. Additionally, the vast amount of social media data complicates identifying features that accurately reflect user behavior and engagement. Thus, there is a need for a universally applicable model that transcends the constraints of specific platforms and the nature of user-generated content.

To address the challenges in predicting unfollow events, we first created a comprehensive and dynamic Twitter dataset. We then developed a new model named EDGE-UP, which adeptly integrates temporal dynamics with network structure, employing techniques like Graph Neural Networks (GNNs) and Long Short-Term Memory (LSTM) networks to capture users' evolving interactions and preferences. We address unfollow prediction challenges in the following ways:

- **Introducing a Large-Scale and Longitudinal Data:** We constructed a large-scale Twitter dataset with *58 weekly snapshots* of interactions among 118,890 Twitter users. This dataset provides a broad and detailed view of social network dynamics over an extended period. This long-term perspective is crucial for understanding the nuanced patterns and trends in unfollow behavior, often missed in smaller or shorter-term studies [31,13].
- **Minimizing Feature Engineering:** The Spatial Encoder in EDGE-UP uses GNNs to create simplified representations of user nodes. This allows the model to identify essential features from the social network structure automatically, eliminating the need for manual feature selection. EDGE-UP effectively captures user tie dissolution behavior by *merely focusing on the network's core patterns*, improving its prediction ability.
- **Adapting to Network Changes:** The Temporal Encoder in EDGE-UP is designed to keep up with users' changing interactions and preferences over time. By examining the sequence of network snapshots and their changes using an introduced concept of *lookback* period, EDGE-UP stays in tune with evolving patterns and behaviors, increasing its accuracy.
- **Addressing Data Sparsity and Class Imbalance:** EDGE-UP uses ensemble learning to tackle the problem of sparse data and the imbalance between the number of retained ties and dissolved ties. Additionally, we implement a robust negative sampling strategy to counter the imbalance between the frequencies of retained and dissolved ties in the dataset. By aggregating predictions from multiple models, this approach facilitates a more balanced perspective, enhancing the model's ability to generalize effectively across diverse data scenarios.

Our study contributes to the growing body of literature on social network analysis in the following ways:

- We construct a large-scale temporal dataset on Twitter with 58 weekly snapshots of the follow network among 118,890 users. To our knowledge, this dataset is the most comprehensive in terms of time span in this field, poised to significantly advance research in analyzing and predicting broken ties in online social networks.
- We propose EDGE-UP, a novel model for accurately predicting unfollow events in social networks. Our model incorporates a unique approach to capture the temporal evolution of user relationships, offering new insights into how social ties change over time, and addresses key challenges such as data sparsity, class imbalance, and the need for extensive feature engineering, setting a precedent for future research in this field. To the best of our knowledge, this is the first model to perform end-to-end unfollow prediction using only network data.
- We perform extensive experiments and show the superiority of EDGE-UP to the baseline methods and its high performance and reliability in unfollow prediction on Twitter (X).

Remark. While EDGE-UP has been tested on our Twitter (X) dataset, its application is not confined to a single platform. Its design allows easy adaptation to data from various social media platforms, provided such data is accessible. This versatility stems from its design, which is not dependent on the unique characteristics of any single platform's content.

2 Related Work

2.1 Unfollow Analysis and Prediction

Although unfollow prediction in social networks is gaining importance, research in this area remains relatively sparse. Next, we will examine some key studies focusing on analyzing and predicting unfollow behavior, often called broken tie analysis.

In a pioneering research effort, [16] conducted a study on 1.2 million Korean-speaking users on Twitter for 51 days to analyze unfollow dynamics. They identified a few significant factors influencing unfollow events, including overlap, duration, and reciprocity of the relationships, as well as followee's informativeness based on users' tweets and relationships. In a subsequent study, [17] examined two snapshots of follow networks of the same users 10 months apart and identified 12 structural and actional properties that affect the decision to unfollow. Later, [31] employed actor-oriented modeling (SIENA) to investigate how reciprocity, embeddedness, status, homophily, and informativeness influence the dissolution of ties. Their study on closely-knit user groups on Twitter showed that relational properties such as reciprocity and embeddedness significantly affect tie breaks. At the same time, the effect of informativeness and homophily based on common interest is not deemed significant. [13] investigated the impact of network structure alone on unfollow behavior and identified some structural properties that show significant effects on tie breaks. [23] examined the impact of several factors (e.g., age, gender, personality traits) that sociological studies have linked to friendship dissolution in real-world scenarios within the context of Facebook. [18] analyzed the content of posts from Twitter users experiencing consistent follower loss to extract various behavioral features that are subsequently utilized for the early detection of follower loss.

While these approaches provide valuable insights, they necessitate substantial feature engineering and cannot capture the interaction between spatial and temporal attributes within a social network. Therefore, [30] proposed UMHI for unfollow prediction in a real-world Weibo dataset, which captures users' spatial attributes (e.g., their social role) through a network structure encoder and infers their temporal attributes through their posted content and unfollow history. This study is closest to our work. Nonetheless, there are several drawbacks to this model: 1) it lacks the capability for end-to-end training; 2) it relies on users' posted content, which could be challenging or expensive to gather and may not significantly influence unfollow behavior in certain Twitter groups, as demonstrated by [31]; 3) it does not consider the temporal evolution of users' local neighborhood.

Unlike traditional unfollow prediction methods, our approach focuses on harnessing spatio-temporal data **solely** from the user follow relationship network. This strategy eliminates the dependency on platform-specific, content-related, and often complicated hand-crafted features with very low generalizability. In particular, by leveraging a combination of graph neural networks and recurrent neural networks, we efficiently encode spatio-temporal information, offering a compelling end-to-end solution for unfollow prediction.

2.2 Follow (Link) Prediction

In contrast to unfollow prediction in social networks, follow prediction or link prediction has garnered considerable attention, especially in graph machine learning. As our approach draws on the research and breakthroughs in link prediction, including those methods used as baselines, we will briefly overview some of these pertinent methodologies.

Authors in [22] proposed DeepWalk, a random walk-based method for learning latent network representations for nodes in a network that effectively captures social relations in a continuous vector space. Node2vec was introduced in [3], which employs flexible, biased random walks to encode a graph's structure, balancing local and global network properties. node2vec has shown great performance in encoding graphs from many different domains [2,32]. LINE (Large-scale Information Network Embedding) [25] is designed for embedding extensive networks into low-dimensional spaces. It effectively captures both first-order (direct connections) and second-order (neighborhood similarity) relationships within the network. These methods focus on learning structural node embeddings in an unsupervised manner.

However, a significant advancement occurred with the introduction of Graph Convolutional Networks (GCN) by [12], combining node features and graph structure in a neural network approach, shifting towards more supervised learning in graph analysis. Authors in [12,11] discovered that the GCN-based approach consistently outperforms DeepWalk in link prediction. Building upon these developments, the Graph Attention Network (GAT) and GraphSAGE have further expanded the capabilities of graph machine learning. GAT [28], leverages attention mechanisms to weigh the importance of nodes' neighbors, allowing for more nuanced feature aggregation from a node's local neighborhood. On the other hand, GraphSAGE [4] innovates by using a sampling-based approach to generate node embeddings efficiently. It aggregates features from a node's local neighborhood while allowing for inductive learning, enabling the model to generalize to unseen nodes.

3 Dataset

To effectively predict unfollow events, it is essential to have access to a large, dynamic dataset of detailed user interactions. Therefore, we chose to gather data from Twitter (X). We gained access to Twitter's API through their Academic Research program and set up the necessary software and hardware to store data efficiently. This setup allowed us to capture weekly snapshots of social connections and content from 118,890 users over a year, beginning on June 18, 2018.

Table 1: Dataset statistics

Property	**Average Value**
# of nodes/users per week	118,890
# of edges/ties per week	2,841,814.09
Density	0.0002
# of new follows in weeks 2 to 58	9,648.88
# of new unfollows in weeks 2 to 58	4,095.17
Ratio of unfollows to # of edges	0.0014
Maximum # of followers	4,015.19
Minimum # of followers	1
Maximum # of followees	3,261.95
Minimum # of followees	1

We started with one author's Twitter account and expanded our user base using a breadth-first search approach [19], initially reaching around 130,000 users. After removing users who changed their privacy settings or deactivated their accounts during our data collection, we were left with 118,890 users. Our data, spanning from June 18, 2018, to July 22, 2019, includes 58 weekly snapshots that detail the follow relationships among these users (see Table 1 for statistics). It is important to highlight that Twitter's API does not provide direct information about unfollow events and when they occur. Therefore, to identify such events, we had to repeatedly collect data on the same users over an extended timeframe. As shown in Table 1, the number of unfollows compared to new follows is significantly smaller.

4 Problem Statement

Suppose we have observed the follow relationships among a subset of users in an online social network across T consecutive time-steps. In this context, a follow relationship is a directed link from a follower to a followee, and each user might be associated with a set of features. At any given time t, we can represent the state of the social network with a directed graph $G_t = (V, E_t)$, where V is the set of users that remains the same in all time steps, and E_t comprises directed edges reflecting the follow relationships at time t.

Unfollow: At time-step t, an edge (u_i, u_j) is labeled as *unfollow* if $(u_i, u_j) \in E_t$ and $(u_i, u_j) \notin E_{t+1}$.

Our goal is to develop a model $f(\cdot)$ that, given any sequence of k graph snapshots from time t to time $(t+k-1)$, predicts the likelihood of an unfollow event occurring between any two users u and v in the subsequent time-step $(t+k)$. The parameter k represents a designated *lookback* period, a crucial time window that determines the number of historical snapshots used for each prediction, optimized through hyper-parameter tuning. The model's predictive function can be mathematically represented as $\hat{y}_{(u,v)} = f(G_t,...,G_{t+k-1})$, where $\hat{y}_{(u,v)}$ represents the predicted probability that user u will unfollow user v at time $(t+k)$.

5 The Proposed Method (EDGE-UP)

We propose a novel model for unfollow prediction in online social networks, which relies solely on the network's inherent relationship graph and temporal dynamics. Our approach involves four key components: 1) A Spatial Encoder employing Graph Neural Networks (GNNs) to derive compact, low-dimensional user node representations, capturing the structural intricacies of the social network; 2) A Temporal Encoder utilizing Long Short-Term Memory (LSTM) networks, to track the evolving local neighborhood structure surrounding user pairs before an unfollow event; 3) A Multi-Layer Perceptron (MLP) tasked with generating the final predictions for each pair of users; and 4) An Ensemble Learning strategy that enhances overall predictive accuracy by synergizing multiple models. This comprehensive methodology is designed to optimize unfollow prediction by effectively integrating both spatial and temporal aspects of social networks. Figure 2 shows the overall architecture of our model. In the following subsections, we provide a detailed explanation of each component within our model.

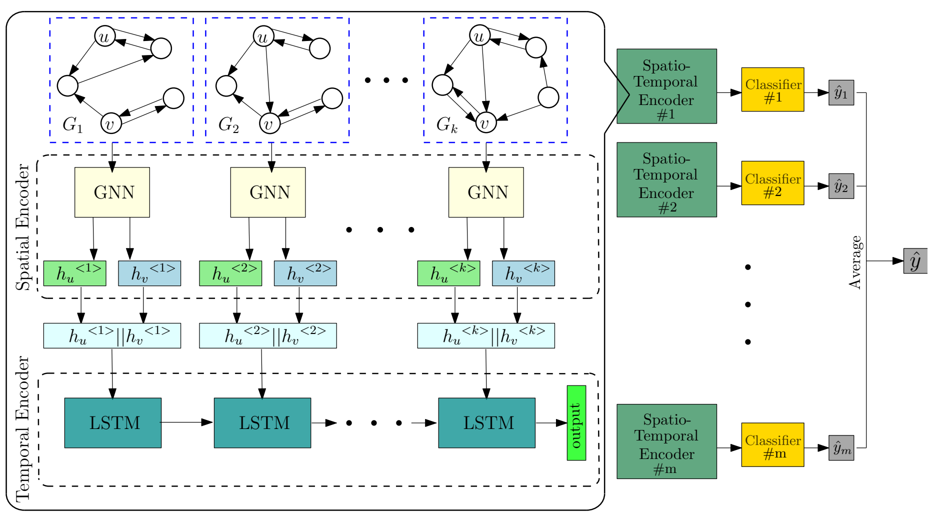


Fig. 2: Overview of EDGE-UP with m ensemble members.

5.1 Spatial Encoder

As highlighted earlier, one of the critical challenges in unfollow prediction within social networks is feature engineering. Earlier research has primarily focused on using hand-crafted features, which can be either costly or impossible to extract [30,18]. We propose using Graph Neural Networks (GNNs) to address this challenge. GNNs are a class of neural network models designed to process and analyze graph-structured data. We believe they are particularly effective in predicting unfollow actions in online social networks due to their proficiency in processing the complex and dynamic relationships inherent in these networks. They can adeptly capture local and global interaction patterns among users, making them well-suited for analyzing large-scale social media data, leading to more accurate and insightful predictions in online social interactions.

In a GNN, several graph convolution layers are employed, resembling a perceptron, but with an additional step for neighborhood aggregation. Each layer l of a GNN model involves three key components: message computation, message aggregation, and non-linearity. In the message computation step, each node creates a message that will be sent to other nodes. Mathematically, we can write: ${m_u}^{(l)} = MSG^{(l)}({h_u}^{(l-1)})$.

Next, in the message aggregation step, each node aggregates the messages from its neighboring nodes with the message from the node itself. Finally, a non-linearity is applied to compute the node representation. For node v, we can write

$$ {h_v}^{(l)} = \sigma(CONCAT(AGG^{(l)}(\{{m_u}^{(l)}, u \in N(v)\}), {m_v}^{(l)})) \tag{1} $$

where $N(v)$ is the set of neighbors of node v. After L layers of computation, the final embedding for node v is a d-dimensional vector ${h_v}^{(L)}$.

Researchers have introduced powerful GNN models in the past years by exploring possible choices for MSG and AGG functions. One such model is GraphSAGE, which has shown strong performance in various tasks involving graph-structured data [4]. GraphSAGE generalizes the aggregation step by allowing multiple aggregation functions, such as the max-pooling aggregator. In this approach, each neighbor's vector is transformed using a fully connected neural network, which constitutes the MSG function. So, we have:

$$ {m_u}^{(l)} = MLP({h_u}^{(l-1)}), \tag{2} $$

where MLP is an arbitrarily deep multi-layer perceptron.

The aggregation step has two stages. First, an element-wise max-pooling operation is applied to aggregate the messages from neighboring nodes. For node v, we can write:

$$ {m_{(N(v))}}^{(l)} = Max(\{{m_u}^{(l-1)}, \forall u \in N(v)\}). \tag{3} $$

In the next stage, to further aggregate over the node itself, a GraphSAGE layer follows this equation:

$$ {h_v}^{(l)} = \sigma(W^{(l)} \cdot CONCAT({m_{(N(v))}}^{(l)}, {m_v}^{(l)})). \tag{4} $$

To predict unfollows, we need to learn representations for pairs of nodes in the graph. We achieve this simply by concatenating the embeddings of such nodes to construct a representation for the pair. Therefore, for $p = p(u,v)$, which is the pair of nodes u and v: $h_p = [{h_u}^L || {h_v}^L]$, where $||$ is the symbol for concatenation operation and is equivalent to $CONCAT$. In this paper, we use GraphSAGE layers as described above to encode graph snapshots that represent the structure of a social network at different time steps. We experimented with other aggregators, but max-pooling produced the best results.

5.2 Temporal Encoder

We hypothesize that incorporating the temporal evolution of a node's local neighborhood can enhance the prediction of future unfollow events. To validate this hypothesis, creating *sub-sequences* of consecutive graph snapshots from our dataset is essential. To this end, firstly, we focus on examining the network's structure within a specific time window preceding an unfollow event, which we refer to as the *lookback* period. Then, we employ a sliding window technique to generate data samples that track the development of local interactions around nodes before unfollow events. Given a series of graph snapshots $\{G_1,G_2,...,G_T\}$, employing the sliding window technique with a lookback period of k and stride of 1 results in $T-k$ graph sequences. Each of these sequences contains k consecutive graphs and is utilized to predict unfollow events in the following time-step. Figure 3 illustrates the application of this technique to our dataset, using a stride of 1 and a lookback period of 2. In Section 6, we demonstrate that the proposed *lookback* approach is very practical in unfollow prediction, improving the performance significantly.

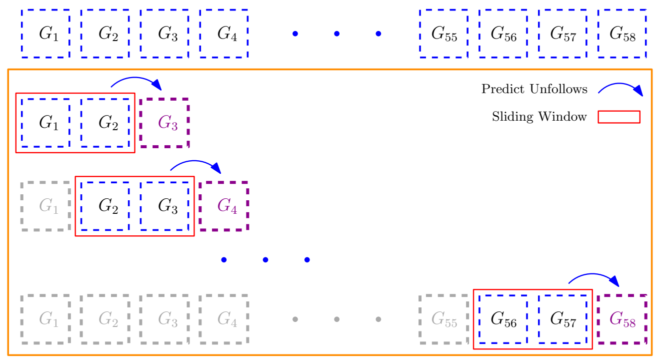


Fig. 3: Sliding window technique applied to our dataset with lookback set to 2.

In this paper, we employ Long Short-Term Memory (LSTM) networks to derive latent representations that capture the temporal evolution of local neighborhoods surrounding node pairs within each sequence of k graphs. LSTM networks [5] are a type of recurrent neural network designed for modeling and capturing dependencies in sequential data. LSTMs have wide application in tasks such as natural language processing, time series analysis, and speech recognition due to their ability to capture temporal dependencies effectively.

Suppose node u unfollows node v at time t. To be able to predict this unfollow event, we observe the evolution of the local neighborhood around nodes u and v from time $(t-k)$ to time $(t-1)$. To this end, we first obtain the embedding of this pair of nodes at each time-step using a GNN. Then we input this sequence of embeddings $[{h_p}^{<t-k>},...,{h_p}^{<t-1>}]$ to an LSTM network. The LSTM learns a low-dimensional representation for the pair after time k has passed, which can be used to predict whether node u will unfollow node v or not.

5.3 Classifier

A Multi-layer Perceptron (MLP) is a type of neural network architecture characterized by multiple layers of interconnected nodes, including an input layer, one or more hidden layers, and an output layer [15]. In an MLP, each node (or neuron) in a layer is connected to every node in the subsequent layer, forming a densely connected network. This architecture allows

MLPs to learn complex relationships in data, making them suitable for a wide range of tasks, including classification and regression.

We use an MLP with one hidden layer to predict whether a user will unfollow another user, given the final hidden state from the LSTM network. Suppose for node pair $p(u,v)$, the final hidden state of the LSTM is a_p. Then, the output of MLP is computed as follows:

$$h = ReLU(W_{in} \cdot a_p + b_h), \tag{5}$$
$$\hat{y}_{(u,v)} = \sigma(W_{out} \cdot h + b_o). \tag{6}$$

In Equation 5, $ReLU$ is the Rectified Linear Unit activation function, W_{in} is the weight matrix for the input layer to the hidden layer, b_h is the bias for the hidden layer, and h is the final hidden layer vector. In Equation 6, W_{out} is the weight matrix for the hidden layer to the output layer, b_o is the bias for the output layer, and $\hat{y}_{(u,v)}$ is the prediction for node pair $p(u,v)$ after applying sigmoid activation function.

5.4 Average Ensemble

As previously discussed, the scarcity of data in unfollow events poses challenges for unfollow prediction models, particularly their ability to generalize effectively to new and unencountered data. To mitigate this issue, we employ the strategy of ensemble learning. Ensemble learning is a machine learning technique where multiple models are combined to improve the overall performance and robustness of a predictive model [20]. Instead of relying on a single model, ensemble methods leverage the strengths of multiple models to achieve better accuracy, generalization, and reliability. In this work, we used ensemble averaging, a basic form of ensemble learning, where we trained multiple models on the same training set and averaged their predictions on the test set to achieve more accurate results.

Suppose f is a model that, given a set of graphs $G_t,...,G_{t+k-1}$, predicts whether a user will unfollow another user in the next time-step $t+k$, i.e. $\hat{y}_{(u,v)} = f(G_t,...,G_{t+k-1})$. Then ensemble averaging works as follows: First, we train m such models (also called ensemble members) independently; secondly, we apply the models on the test graphs separately and average their results. Suppose in test time, the goal is to predict whether user u unfollows user v in time $t'+k$, given graphs of k previous time-steps. Mathematically, we can write:

$$\hat{y}_{(u,v)} = \frac{1}{m}\sum_{i=1}^{m} f_i(G_{t'},...,G_{t'+k-1}) = \frac{1}{m}\sum_{i=1}^{m}\hat{y}_{(u,v),i} \tag{7}$$

where $\hat{y}_{(u,v)}$ is the final prediction for node pair $p(u,v)$ in time $t'+k$.

6 Experiments

In this section, we provide an in-depth look at our extensive experiments. Initially, in Section 6.1, we outline our experimental setup. Next, in Section 6.2, we introduce the baseline models used in our experiments. Section 6.3 presents the findings of our experiments, highlighting the comparison between EDGE-UP and the baseline models. Finally, we conclude with a comprehensive ablation study (component analysis) of our proposed model in Section 6.4.

Data Split We divided the graph snapshots into training, validation, and testing subsets to evaluate our model comprehensively. Specifically, we utilized the initial 49 graphs along with their corresponding labels for the training set. In total, these labels include 207,542 unfollow events. For validation, we employed the subsequent 3 graphs and their labels, which include 10,804 unfollow events. Lastly, the final 5 graphs, covering weeks 53 to 57, and their associated labels with 15,079 unfollow events were designated for the testing set. This strategic partitioning is designed to rigorously assess the EDGE-UP model's capability to generalize and perform effectively on unseen data. We initialize each user's node features with a 50-dimensional random vector to eliminate the need for manual feature specification.

6.1 Experimental Settings

Label Generation To train our model for unfollow prediction, we implemented a structured approach for label generation using sequential graph snapshots. Labels were generated for each snapshot by analyzing changes in the graph structure over consecutive weeks. Specifically, positive unfollow samples were identified by locating edges present in a given snapshot but absent in the next week– See Section 4. We performed a negative sampling strategy to generate negative labels, where the head nodes of positive samples are altered, as depicted in Figure 4. As a result, for every positive unfollow sample, we created one negative sample. The overarching objective of this methodology is to effectively train the model to distinguish between followers that will be retained and those that are likely to unfollow. Incorporating this labeling strategy, we successfully generated both positive and negative labels for a series of 57 graph snapshots spanning from week 1 to week 57. This comprehensive dataset ensures a balanced and robust foundation for modeling unfollow prediction as a binary classification problem. Utilizing these labels, we use binary cross-entropy loss function to train the model.

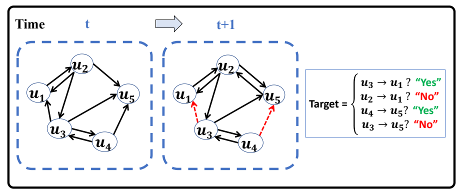


Fig. 4: This example showcases our approach in generating negative labels through negative sampling for the graph of week t after identifying positive labels.

Implementation We used PyTorch [21] and DGL [29] libraries for implementation. Each simulation was run for 200 epochs with early stopping based on validation loss. The learning rate was set to 0.0005, determined by hyper-parameter tuning, with a decaying rate of 0.9 every 50 steps. Based on hyperparameter tuning, we set the lookback value to 2 and the number of ensemble members to 60. The experiments were conducted on a system with an AMD EPYC 7513 CPU, 4 NVIDIA RTX A4000 GPUs, and 1 TB of RAM.

Evaluation Metrics We evaluated the performance of unfollow prediction models using accuracy, precision, recall, and area under the curve (AUC) metrics. We ran each experiment 5 times and reported the mean and standard deviation of results for each metric.

6.2 Baseline Models

We experimented with three categories of baseline models:

Unsupervised Spatial Encoders We utilized a set of unsupervised shallow encoders to learn embeddings for all nodes within each graph snapshot. Next, we concatenated these embeddings for pairs of nodes to predict potential unfollow events in the subsequent time step. We experimented with various prediction methods, including Random Forest, Logistic Regression, Support Vector Machine (SVM), and XGBoost. We conducted thorough evaluations with these techniques and reported the most optimal results. The following is the list of these shallow encoders:

- **DeepWalk:** [22] utilized random walks to generate node sequences, applying language modeling techniques to learn representations that capture the graph's structural properties. We used the *karateclub*[3] package for implementation of DeepWalk.
- **node2vec:** [3] introduced a more flexible notion of a node's neighborhood. It employs biased random walks to balance between capturing local and global graph structures. We used the *nodevectors*[4] package for scalable implementation of node2vec.
- **LINE:** [25] designed LINE to efficiently preserve both first-order (direct connections) and second-order (neighborhood similarities) proximities in the embedding space in large-scale information networks. We used an existing TensorFlow implementation of LINE[5].

Supervised Spatial Encoders In another set of experiments, we employed three widely recognized graph neural network models to generate embeddings for graph snapshots. Subsequently, we concatenated these embeddings for node pairs, aiming to predict potential unfollow events in the next time step. An MLP was utilized to make predictions based on these final embeddings. However, it is important to note that these models do not capture the temporal evolution of users' local neighborhoods. We implemented these models using PyTorch and DGL libraries.

- **GCN:** GCNs [12] apply convolutional neural network principles to graph data, updating node representations by aggregating neighbor information, ideal for tasks like node classification.
- **GAT:** GATs [28] use attention mechanisms in graph neural networks to dynamically prioritize a node's neighbors, enhancing the learning of graph structures.
- **GraphSAGE:** GraphSAGE [4] inductively learns node embeddings by aggregating from neighbors and the node itself, enabling efficient embedding of unseen nodes in large graphs.

We also experimented with applying the average ensemble for each method as explained in Section 5. Number of ensemble members is set to 60 in all such experiments.

Supervised Spatio-Temporal Encoders In these experiments, we used the idea of sliding windows explained in Section 5 to evaluate the effectiveness of incorporating temporal information in the performance of previously mentioned graph neural network models. For all models, we used an LSTM, with a lookback set to 2, to capture the temporal evolution of embeddings. An MLP was utilized to make predictions based on the LSTM's final hidden state. Note that our proposed model, EDGE-UP, utilizes this category of encoders. To compare the effect of ensemble learning in each variant, we experimented with 60 ensemble members for each method and recorded the results.

6.3 Experimental Results

Table 2 presents the outcomes of our experiments, encompassing the three categories of baseline models alongside our EDGE-UP model. Based on these results, we make the following observations:

- Among the unsupervised spatial encoders, node2vec shows the best performance, indicating its effectiveness in capturing useful representations for unfollow prediction.

[3] https://github.com/benedekrozemberczki/karateclub
[4] https://github.com/VHRanger/nodevectors
[5] https://github.com/snowkylin/line

Table 2: Comparing the performance of baseline models and our proposed model. The **best** result in each column is displayed in bold, and the second-best result is underlined.

Category		Model	Accuracy	Precision	Recall	AUC
(1)	Unsupervised Spatial Encoders	LINE	0.51±0.00	0.51±0.00	0.51±0.00	0.50±0.00
		DeepWalk	0.51±0.00	0.52±0.00	0.49±0.00	0.51±0.00
		node2vec	0.55±0.00	0.58±0.00	0.51±0.00	0.57±0.00
(2)	Supervised Spatial Encoders	GCN	0.53±0.00	0.53±0.01	0.44±0.04	0.54±0.00
		GAT	0.53±0.01	0.54±0.02	0.29±0.04	0.53±0.02
		GraphSAGE	0.57±0.02	0.64±0.05	0.33±0.07	0.60±0.03
		GCN-Ensemble	0.53±0.00	0.53±0.00	0.47±0.00	0.54±0.00
		GAT-Ensemble	0.53±0.00	0.55±0.01	0.33±0.01	0.54±0.00
		GraphSAGE-Ensemble	0.61±0.00	0.69±0.00	0.40±0.01	0.64±0.00
(3)	Supervised Spatio-Temporal Encoders	GCN+LSTM	0.52±0.01	0.52±0.01	0.46±0.08	0.53±0.00
		GAT+LSTM	0.53±0.02	0.57±0.04	0.28±0.04	0.54±0.03
		GraphSAGE+LSTM	0.60±0.01	0.68±0.02	0.39±0.06	0.64±0.01
		GCN+LSTM Ensemble	0.68±0.01	0.78±0.01	0.49±0.01	0.76±0.01
		GAT+LSTM Ensemble	0.62±0.01	0.97±0.01	0.25±0.02	0.83±0.01
		EDGE-UP	**0.92±0.01**	**0.99±0.03**	**0.85±0.04**	**0.99±0.00**

- ❏ Supervised spatio-temporal encoders generally perform better than unsupervised spatial encoders as well as their spatial-only counterparts. This suggests that the use of supervised learning and addition of temporal information processing (via LSTM) enhances the model's predictive capabilities.
- ❏ Within the supervised spatial encoders, GraphSAGE shows the highest scores, especially in terms of accuracy and AUC, suggesting its superiority in embedding quality for this task. When spatio-temporal encoding is applied, GraphSAGE+LSTM achieves the best results among them, reinforcing the strength of GraphSAGE as a spatial encoder when combined with temporal encoding.
- ❏ EDGE-UP demonstrates markedly superior performance across all metrics (accuracy, precision, recall, and AUC), with scores significantly higher than the other models. This indicates the effectiveness of combining a spatial encoder (GraphSAGE), a temporal encoder (LSTM), and ensemble learning in our approach. This substantial lead also underscores the significant impact of ensemble learning in enhancing prediction accuracy and reliability.

6.4 Ablation Study

In this section, we present the results of our extensive experiments to analyze the effectiveness of different components of EDGE-UP.

Temporal Encoder To assess the effectiveness of the temporal encoder component, we devised an experiment with different lookback values. With a lookback of 1, the LSTM module is omitted, and node embeddings are directly concatenated for MLP-based prediction. We explored lookback values ranging from 2 to 4 to determine the optimal setting.

The results, depicted in Figure 5, led to several key insights:

- ❏ The absence of the temporal encoder (lookback set to 1) significantly diminishes EDGE-UP's performance across accuracy, precision, recall, and AUC metrics, underscoring the crucial role of temporal information in the performance of unfollow prediction.

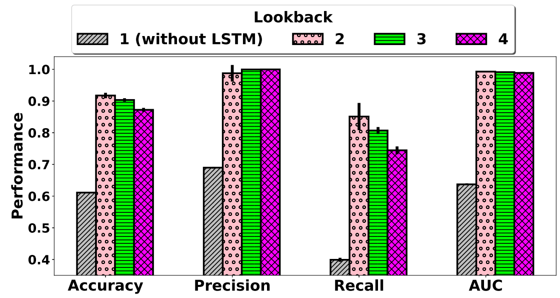


Fig. 5: Performance of EDGE-UP with different lookback values and associated error bars.

- ❏ A lookback of 2 yielded the highest accuracy, recall, and AUC, indicating that incorporating recent historical data of users' local neighborhoods enhances prediction efficacy.
- ❏ Increasing the lookback to 3 or 4 boosted precision but adversely affected other metrics. This suggests that while extended lookback enhances the reliability of positive predictions, it does not substantially benefit other aspects of prediction, possibly due to the diminishing relevance of more distant past information.

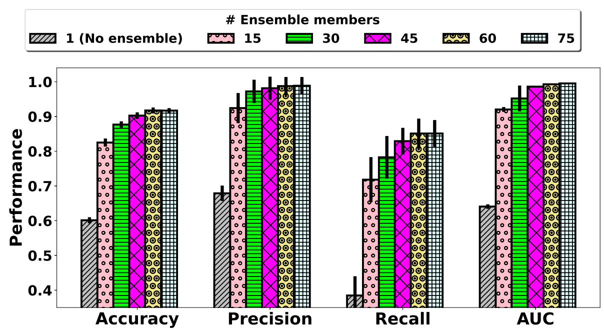
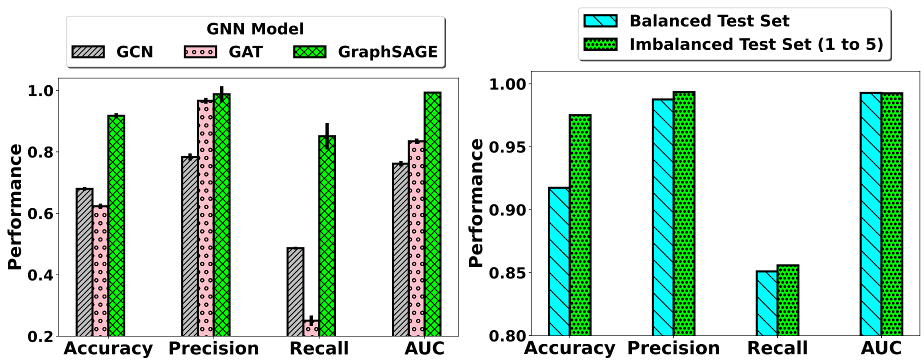


Fig. 6: Performance of EDGE-UP with different numbers of ensemble members and associated error bars.

Average Ensemble In a separate series of experiments, we evaluated the influence of employing ensemble learning within our model. This involved testing with varying quantities of ensemble members, where a single member equates to not implementing ensemble learning. Further, we explored ensemble sizes of [15, 30, 45, 60, 75] to determine the impact of increasing the number of ensemble members. The outcomes of our experiments on ensemble learning are presented in Figure 6, leading to the following key findings:

- ❏ Model performance improved consistently with the number of ensemble members, showing significant gains from a single member to 15 members.
- ❏ Beyond 60 members, the rate of performance improvement decreased, indicating diminishing returns.
- ❏ An ensemble of 45 members balanced performance gains and computational cost effectively.

Spatial Encoder Figure 7 (a) visualizes the effect of replacing GraphSAGE in EDGE-UP with GCN and GAT. These results were also reported in Table 2. From these results, we observe that GraphSAGE, with max-pooling aggregation used in EDGE-UP, yields the highest accuracy, precision, recall, and AUC. This emphasizes the effectiveness of using GraphSAGE convolution layers to encode graph snapshots.



(a) (b)

Fig. 7: Performance of EDGE-UP (a) with different GNN model members and associated error bars. (b) on a balanced test set versus an imbalanced test set.

Class Imbalance To confirm our model's capability to predict unfollow events in the imbalanced setting, we experimented with the same test set of 5 graphs as described in 6.1, but with 5 negative samples per each positive unfollow event. As a result, in this test set, we include 15,079 positive unfollows and 75,395 negative unfollows (retained ties). Figure 7 (b) demonstrates the results of this experiment, highlighting the effectiveness of our model in predicting unfollows in an imbalanced environment.

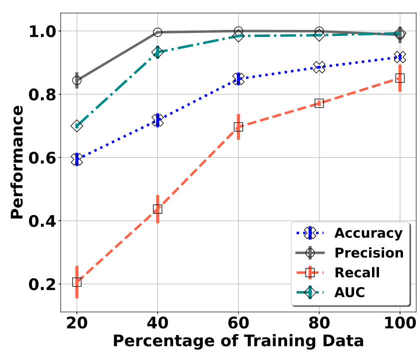


Fig. 8: Performance of EDGE-UP with different training sizes (with same test set) and associated error bars.

Training Set Size We conducted experiments with varying sizes of training sets to assess the performance of our model when faced with limited training data. The findings are illustrated in Figure 8. Despite the rarity of unfollow events in our dataset, our results demonstrate that EDGE-UP can attain an 85% accuracy in predicting unfollow events, even when trained with only 60% of the data (29 graphs).

7 Conclusion

In this paper, we constructed a large-scale longitudinal Twitter (X) dataset that facilitates research on dynamic online social networks. We then introduced EDGE-UP, an enhanced dynamic GNN ensemble method for unfollow prediction in online social networks. Our approach, which seamlessly integrates a spatial encoder, a temporal encoder, a classifier, and an ensemble learning strategy, represented a comprehensive solution to the complex problem of unfollow prediction. Our thorough experimentation showcased the efficiency of EDGE-UP in capturing spatial and temporal information within social networks, which significantly contributes to the accurate prediction of unfollow events.

Our research strictly followed Twitter's guidelines, using public data and ensuring privacy through anonymization and aggregated reporting. Despite acknowledging potential misuses like targeted advertising or social manipulation, we implemented strict ethical safeguards. We believe the benefits of our findings in understanding social interactions outweigh these ethical concerns.

The study's limitations include potential bias from its data collection method and the simplification of Twitter's dynamic social interactions into discrete events due to the reliance on network snapshots. Despite this, the large dataset size helps mitigate bias, and the focus on network structure over individual characteristics remains robust. However, there's a risk of overlooking the qualitative aspects of social relationships.

References

1. Brookhouse, A., Derr, T., Karimi, H., Bernard, H.R., Tang, J.: Road to the white house: Analyzing the relations between mainstream and social media during the us presidential primaries. In: Proceedings of the 32nd ACM Conference on Hypertext and Social Media. pp. 57–66 (2021)
2. Farokhi, S., Yaramal, A., Huang, J., Khan, M.F.A., Qi, X., Karimi, H.: Enhancing the performance of automated grade prediction in mooc using graph representation learning. In: 2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA). pp. 1–10. IEEE (2023)
3. 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)
4. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems **30** (2017)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
6. Karimi, H., Derr, T., Torphy, K.T., Frank, K.A., Tang, J.: Towards improving sample representativeness of teachers on online social media: A case study on pinterest. In: Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21. pp. 130–134. Springer International Publishing (2020)
7. Karimi, H., Torphy, K.T., Derr, T., Frank, K.A., Tang, J.: Characterizing teacher connections in online social media: A case study on pinterest. In: Proceedings of the seventh ACM conference on learning@ scale. pp. 249–252 (2020)
8. Karimi, H., VanDam, C., Ye, L., Tang, J.: End-to-end compromised account detection. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 314–321. IEEE (2018)
9. Kheiri, K., Karimi, H.: Sentimentgpt: Exploiting gpt for advanced sentiment analysis and its departure from current machine learning. arXiv preprint arXiv:2307.10234 (2023)
10. Kheiri, K., Khan, M.F.A., Derr, T., Karimi, H.: An analysis of the dynamics of ties on twitter. In: 2023 IEEE International Conference on Big Data (BigData). pp. 5809–5817. IEEE (2023)
11. Kipf, T., Welling, M.: Variational graph auto-encoders. arXiv preprint arXiv:1611.07308 (2016)
12. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
13. Kivran-Swaine, F., Govindan, P., Naaman, M.: The impact of network structure on breaking ties in online social networks: unfollowing on twitter. In: Proceedings of the SIGCHI conference on human factors in computing systems. pp. 1101–1104 (2011)

14. Knake, K.T., Karimi, H., Hu, S., Frank, K.A., Tang, J.: Educational research in the twenty-first century: Leveraging big data to explore teachers' professional behavior and educational resources accessed within pinterest. The Elementary School Journal **122**(1), 86–111 (2021)
15. Kruse, R., Mostaghim, S., Borgelt, C., Braune, C., Steinbrecher, M.: Multi-layer perceptrons. In: Computational intelligence: a methodological introduction, pp. 53–124. Springer (2022)
16. Kwak, H., Chun, H., Moon, S.: Fragile online relationship: a first look at unfollow dynamics in twitter. In: Proceedings of the SIGCHI conference on human factors in computing systems. pp. 1091–1100 (2011)
17. Kwak, H., Moon, S., Lee, W.: More of a receiver than a giver: why do people unfollow in twitter? In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 6, pp. 499–502 (2012)
18. Maity, S.K., Gajula, R., Mukherjee, A.: Why did they# unfollow me? early detection of follower loss on twitter. In: Proceedings of the 2018 ACM International Conference on Supporting Group Work. pp. 127–131 (2018)
19. Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. pp. 29–42 (2007)
20. Opitz, D., Maclin, R.: Popular ensemble methods: An empirical study. Journal of artificial intelligence research **11**, 169–198 (1999)
21. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32. pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
22. 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)
23. Quercia, D., Bodaghi, M., Crowcroft, J.: Loosing" friends" on facebook. In: Proceedings of the 4th Annual ACM Web Science Conference. pp. 251–254 (2012)
24. Solanki, S., Tsugawa, M.A., Karimi, H., et al.: Leveraging social media analytics in engineering education research. In: 2023 ASEE Annual Conference & Exposition (2023)
25. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web. pp. 1067–1077 (2015)
26. Tufekci, Z.: Who acquires friends through social media and why?"rich get richer" versus "seek and ye shall find". In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 4, pp. 170–177 (2010)
27. VanDam, C., Tan, P.N., Tang, J., Karimi, H.: Cadet: A multi-view learning framework for compromised account detection on twitter. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 471–478. IEEE (2018)
28. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
29. Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J., Zhang, Z.: Deep graph library: A graph-centric, highly-performant package for graph neural networks. arXiv preprint arXiv:1909.01315 (2019)
30. Wu, H., Hu, Z., Jia, J., Bu, Y., He, X., Chua, T.S.: Mining unfollow behavior in large-scale online social networks via spatial-temporal interaction. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 254–261 (2020)
31. Xu, B., Huang, Y., Kwak, H., Contractor, N.: Structures of broken ties: exploring unfollow behavior on twitter. In: Proceedings of the 2013 conference on Computer supported cooperative work. pp. 871–876 (2013)
32. Yaramala, A., Farokhi, S., Karimi, H.: Navigating the data-rich landscape of online learning: Insights and predictions from assistments (2024)