Modeling Personalized Retweeting Behaviors for Multi-Stage Cascade Popularity Prediction

Mingyang Zhou , **Yanjie Lin** , **Gang Liu*** , **Zuwen Li** , **Hao Liao** and **Rui Mao**
Guangdong Province Key Laboratory of Popular High Performance Computers, College of Computer Science and Software Engineering, Shenzhen University, China
{zmy@,linyanjie2021@email.,gliu@,mao@}szu.edu.cn, 952585157@qq.com, jamesliao520@gmail.com

Abstract

Predicting the size of message cascades is critical in various applications, such as online advertising and early detection of rumors. However, most existing deep learning approaches rely on cascade observation, which hinders accurate cascade prediction before message posting. Besides, these approaches overlook personalized retweeting behaviors that reflect users' inclination to retweeting specific types of information. In this study, we propose a universal cascade prediction framework, namely **Cas**cade prediction regarding **M**ultiple **S**tage (CasMS), that effectively predicts cascade popularity across message generation stage as well as short-term and long-term stages. Unlike previous methods, our approach not only captures users' personalized retweeting behaviors but also incorporates temporal cascade features. We perform the experiments in datasets collected ourselves as well as public datasets. The results show that our method significantly surpasses existing approaches in predicting the cascade during the message generation stage and different time periods in the cascade dynamics.

1 Introduction

In the realm of online social platforms such as Twitter, Sina Weibo, and Facebook, users generate and share various types of information with their friends/fans. Certain information messages have the potential to follow social connections and spread rapidly to a large number of users. This cascade phenomenon has been extensively utilized in viral marketing [Leskovec *et al.*, 2007; Robles *et al.*, 2020], recommendations [Wu *et al.*, 2019; Wu *et al.*, 2020] and rumor detection [Bian *et al.*, 2020]. Predicting the cascade popularity is a key problem in these applications. Although deep learning-based methods have been employed to address this challenge, existing approaches primarily focus on analyzing the temporal cascade paths while disregarding personalized retweeting behaviors [Li *et al.*, 2017; Cao *et al.*, 2017; Chen *et al.*, 2019]. In reality, different users possess distinct preferences when retweeting or sharing different messages, which significantly impacts the accuracy of information cascade prediction. Besides, in the message generation stage, users also want to evaluate the potential popularity and optimize the message contents based on the evaluation feedback. Hence, how to effectively predict the popularity in the message generation stage and the cascade dynamics is a permanent problem in social network analysis.

The cascade process relies on both the diffusion model of messages and social connections, making diffusion model-based approaches a natural choice. To enhance diffusion model-based approaches, some studies utilize the Hawkes point process to model message cascades [Zhao *et al.*, 2015; Mishra *et al.*, 2016]. Inspired by traditional machine learning techniques, feature regression is employed to forecast the message cascades, where the features can be extracted from message contents, social connections, and even the prior domain knowledge [Cheng *et al.*, 2014; Shulman *et al.*, 2016; Gao *et al.*, 2019]. In recent years, deep learning-based methods have been introduced to identify complex features in social graphs and temporal cascade snapshots [Cao *et al.*, 2017; Chen *et al.*, 2019; Lu *et al.*, 2023]. However, most existing deep learning-based methods overlook the significance of personalized retweeting behaviors regarding some specific kinds of messages, which are crucial for accurate cascade prediction [Xu *et al.*, 2021; Sun *et al.*, 2023a].

The challenges associated with cascade prediction can be categorized into three classes: (1) Extracting complex features from multimodal data poses a significant challenge. Users generate and share diverse types of data such as social graphs, tweet text, images and videos. Effectively extracting features from this multimodal data is a challenge.(2) Accurately characterizing the diffusion model is a problem in cascade prediction. Recent deep learning-based methods overlook the underlying model of the diffusion process and have small prediction accuracy when the observed data is insufficient. (3) Extracting the temporal features of cascades is a challenge. The dynamics of cascades are influenced by both the static graph structure and the temporal cascade. How to combine different features is a problem.

In this paper, we present a universal framework, called **Cas**cade regarding **M**ultiple **S**tage (CasMS), to forecast the size of message cascades. Our proposed CasMS first leverages the retweeting history of users to extract their behavioral patterns. By incorporating a Graph Convolutional Net-

*Corresponding author.

work (GCN) module [Kipf and Welling, 2016], our approach effectively captures both static graph features and temporal cascade features, enabling simultaneous cascade prediction in the message generation stage and in long-term time period. The primary contributions of CasMS are as follows:

- *Characterizing personalized retweeting behavior*: We explicitly capture users' preferences in retweeting specific types of messages, in contrast to existing methods that overlook the importance of personalization.
- *Cascade-driven graph convolutional network*: We introduce a graph convolutional network that combines the diffusion model and personalized retweeting behaviors. This integration facilitates the joint description of the diffusion model and the users' personalization.
- *Real-world dataset experiments*: To validate our framework, we conduct extensive experiments using datasets collected ourselves as well as publicly available datasets. The results unequivocally demonstrate that CasMS outperforms state-of-the-art baselines.

2 Related Work

The task touches on the diffusion model of cascade, feature engineering, and deep learning-based cascade prediction.

Cascade model & prediction: Typical cascade models, such as the independent cascade model and the linear threshold model, are commonly used to describe the process of information propagation [Chen *et al.*, 2022; Ling *et al.*, 2022b; Ling *et al.*, 2022a]. These models rely on prior knowledge and involve calculating parameters based on cascade snapshots. By evaluating these parameters, it becomes possible to predict based on the diffusion model. However, it is worth noting that the current models primarily focus on predicting the outbreaks of information that are characterized by the thresholds of graphs [Shulman *et al.*, 2016; Xia *et al.*, 2021].

Feature-based prediction: This type of approach utilizes various features for cascade size prediction, including information content, user characteristics, social graphs, and temporal cascade features [Cheng *et al.*, 2014; Szabo and Huberman, 2010]. These features are then fed into a discriminative machine learning algorithm to facilitate prediction. Therefore, the extraction and integration of features play a pivotal role in the cascade prediction. Tsur et al. [Tsur and Rappoport, 2012] demonstrated the informativeness of user features from early adopters as predictors, while Pinto et al. [Pinto *et al.*, 2013] emphasized the substantial impact of temporal features. It is important to note that the marginal gain diminishes as more diverse features are combined. Inspired by this, the models of Poisson process and Hawkes process have been incorporated into cascade prediction [Shen *et al.*, 2014], where only a limited set of features are utilized. In the Hawkes process, content features are used to evaluate the content virality, temporal cascade features are used to evaluate the memory decay, and the social graph features are used to evaluate user influence. Later on, various methods have been proposed to extract and combine different features [Gao *et al.*, 2019].

Deep learning-based prediction: Deep learning methods have demonstrated remarkable achievements in natural language processing and computer vision. Leveraging their ability to automatically extract and fuse diverse features, as well as perform end-to-end tasks, deep learning has also been applied to cascade prediction. One notable instance is DeepCas [Li *et al.*, 2017] that is the pioneering use of deep learning as a cascade predictor. Subsequently, several traditional methods have been adapted to achieve end-to-end prediction through the integration of deep learning techniques. For instance, Deephawkes [Cao *et al.*, 2017] utilizes deep learning to extract features from the node sequences of the cascade and determine the decay parameters of the Hawkes process. Chen et al. [Chen *et al.*, 2019] employ graph convolutional neural networks to extract temporal features from the cascade. Sun et al. [Sun *et al.*, 2023b] introduce Transformer to capture both the spatio-temporal features and users' characteristics.

Our work shares similarities with the cascade prediction approach proposed in references [Chen *et al.*, 2019; Lu *et al.*, 2023]. In these studies, the authors employ GCN to capture temporal characteristics within the cascade snapshots. In contrast, our approach first learns the retweeting patterns of users. We incorporate users' retweeting features, along with the users' states throughout the cascade process, into our graph convolutional network. Unlike previous works that roughly extract temporal features, our novel GCN aims to describe personalized behavior in the implicit diffusion models, implying better interpretation. More importantly, our method could predict the popularity not only in the short-term and long-term periods but also in the message generation stage.

3 Preliminaries

3.1 Motivation

Existing popularity prediction approaches in the field of social media suffer from the following problems:

- *How to predict the popularity during the message generation stage?* While large language models(LLMs), such as GPT [Brown *et al.*, 2020], PaLM [Chowdhery *et al.*, 2023] and LLaMA [Touvron *et al.*, 2023], and multimodal software assist in generating high-quality messages, they do not guarantee popularity. To address this issue, the prediction of popularity during the message generation stage necessitates consideration of users' social position, personalized preferences, and the spreading model. However, existing methods mostly rely on the early cascade paths, which are not applicable during the message generation stage.
- *How to perform long-term prediction of the message cascade?* Most existing works either employ Hawkes processes or graph neural networks that are suitable for smoothly spreading processes. However, these methods do not explicitly account for the underlying cascade models, resulting in poor performance in predicting burst cascades for popular messages.
- *How to incorporate users' personalized retweeting preferences and cascade models in the popularity prediction models?* In social networks, each user possesses

a personalized preference for retweeting specific types of messages(depending on message content), which in turn affects cascade models and popularity prediction. While personalized behaviors in social networks have been extensively studied, incorporating them into popularity prediction remains an open question.

To tackle the above problems, we put forward an innovative framework for predicting popularity that effectively integrates users' personalized behaviors with cascade models. By bridging the gap between micro-level user behaviors and macro-level cascade processes, our approach enables accurate popularity prediction across message generation, short-term, and long-term periods.

3.2 Problem Formulation

Let $G = (V, E)$ be a static social network, where V denotes the the set of users and $E \subseteq V \times V$ denotes the set of edges. Suppose we have M messages, denoted by $M = \{m^i, i \in [1, M]\}$. For each message m^i, we use a cascade $C^i = \{(u_j^i, v_j^i, t_j^i)\}$ to record the diffusion process of message m^i, where the triple (u_j^i, v_j^i, t_j^i) corresponds to the $j-$th retweet, meaning that user v_j^i retweets message m^i from user u_j^i , and the time elapsed between the original post and the $j-$th retweet is t_j^i. The popularity R_t^i of the message m^i up to time t is defined as the number of retweets, i.e., $R_t^i = |\{(u_j^i, v_j^i, t_j^i)|t_j^i \leq t\}|$. The problem of popularity prediction in this paper is then formulated as follows:

Problem definition(Popularity prediction): Given a message m^i, the observation time window $[0, t]$ and the prediction time Δt, the objective is to predict the increment of the size of cascade C^i until time $t + \Delta t$, which is denoted as $\Delta R_t^i = R_{t+\Delta t}^i - R_t^i$.

Prediction task in message generation stage: In the message generation stage, the corresponding cascade set C^i is empty. Therefore, our approach relies solely on the content of a message, the static social graph, and the cascade model to make predictions, in contrast to previous works that are based on early cascade paths. Hence, it is essential to carefully extract and fuse the features into the prediction model.

Prediction task in short-term and long-term periods: In the problem definition, the parameter t determines the earliness of the prediction, while Δt represents the time period for the prediction. Specifically, when Δt is small, we focus on short-term popularity prediction. Conversely, when Δt is larger, we shift our attention to long-term prediction.

In the prediction task, we only consider the popularity in static social networks. But in reality, the social network evolves over time. This exclusion of network structure evolution helps to avoid any undesired effects that may arise from new edges influencing the information cascades.

3.3 Challenges

The cascade prediction is actually to find a function f that maps (G, m^i, C^i) to ΔR_t^i, $f : (G, m^i, C^i) \rightarrow \Delta R_t^i$. There are three challenges to calculate the function f: (1) In the message generation stage, most computer-aided software uses the network buzzwords as the prompt of large language (or multimodal) models to generate potential popular messages. However, predicting the popularity based solely on the network buzzwords fails to consider the message content and users' social influence. Additionally, we recognize that different types of messages may require different approaches to measure social influence. For instance, a sports-related tweet from a famous athlete may have a higher impact compared to a traffic-related tweet from the same user. Integrating these features effectively remains a challenge. (2) In the long-term prediction, though existing frameworks apply deep node/network embedding and temporal feature extraction methods to predict the popularity, they fail to consider the diversity of diffusion models for different message types, hindering the accuracy of long-term prediction. Furthermore, traditional prediction models may experience a significant performance decline when the topic of the message changes on the Internet (concept drift). (3) The retweeting behavior of users heavily relies on the content of the messages in the cascade model. Extracting behavior preference features from sparse data and incorporating them into the corresponding prediction framework remain significant challenges.

4 Proposed CasMS Framework

In this section, we propose a novel and comprehensive framework CasMS to predict popularity, which tackles the aforementioned challenges simultaneously. The CasMS framework, as depicted in Figure 1, consists of four essential components: (1) **Modeling Personalized Retweeting Behaviors**: We dedicate this component to extracting the feature of personalized retweeting behaviors. (2) **Learning Cascade Snapshot Feature**: This component integrates the personalized retweeting behaviors with message diffusion models to extract the cascade snapshot feature. (3) **Learning Multi-Resolution Temporal Feature**: Leveraging a temporal convolutional network, this component facilitates the learning of the temporal feature of the message cascade. (4) **Joint Prediction Module**: This component takes the temporal cascade features as input and uses multi-layer perceptrons (MLPs) to predict popularity. In the subsequent sections, we provide detailed explanations of each component.

4.1 Learning Personalized Retweeting Behavior

To extract the personalized retweeting behavior feature, we need the features of messages and the users' social influence.

Message feature extraction: The potential popularity of a message is determined by its contents. Additionally, users tend to retweet messages that revolve around specific topics reflecting his/her interests. To extract the features of message contents, we have employed Bidirectional Encoder Representations from Transformers (BERT) [Devlin *et al.*, 2018], an esteemed semantic model in the natural language processing field. While alternative more complex pre-trained models like XLNet [Yang *et al.*, 2019] and UniLM [Dong *et al.*, 2019] also exist for extracting message features, their marginal gains in our specific task are minor. Therefore, BERT is sufficient for the feature extraction task. In our experiments, we utilized the RoBERTa-wwm-ext-large [1] (an

[1]https://github.com/ymcui/Chinese-BERT-wwm

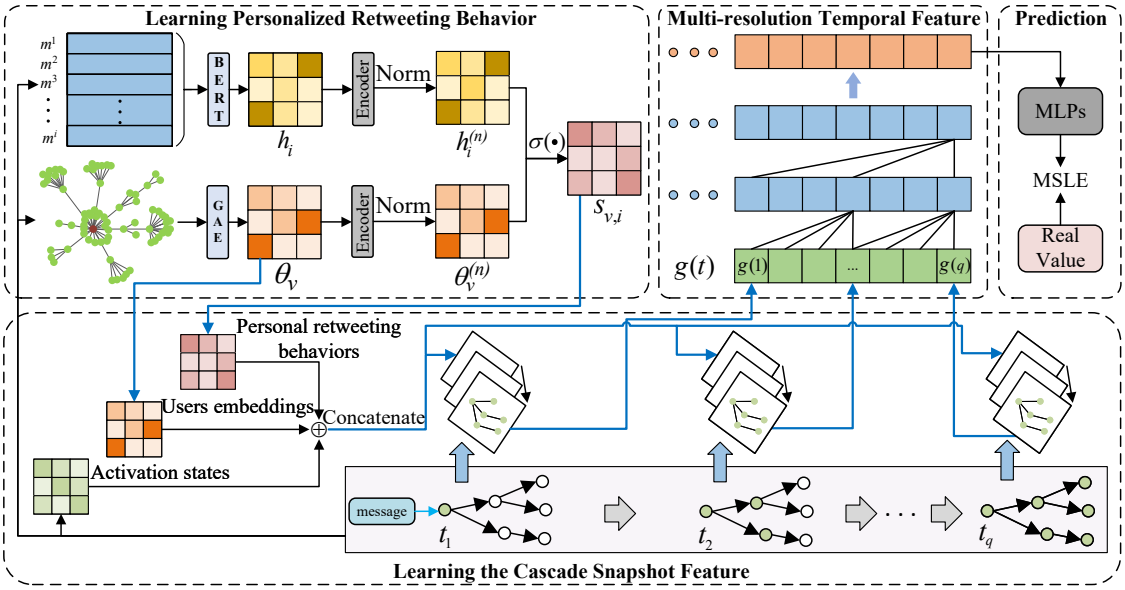

Figure 1: The framework of the proposed CasMS method. It consists of a personalized retweeting behavior module, a cascade snapshot module, a multi-resolution module, and a prediction module to predict future popularity.

open enhanced version of BERT) as the input module, from which we extracted the feature $\mathbf{h}_i$ of message m^i as

$$\mathbf{h}_i = BERT(m^i). \tag{1}$$

User feature extraction: Users' characteristics can be derived from profile tags, such as age, gender, and interests, as well as the social network structure. In our dataset, we only have access to the network structure. Therefore, we utilize this structural feature alone to represent users' characteristics.

To obtain the node embeddings, we adopt the Skip-gram-based model [Mikolov *et al.*, 2013]. Let us consider a network $G = (V, E)$, where V represents the set of nodes and E represents the set of edges. The objective function [Perozzi *et al.*, 2014; Grover and Leskovec, 2016] optimized by the Skip-gram model with negative sampling is as follows:

$$\mathcal{L}_{embed} = max_{\Theta} \sum_{(u,v)\in E} \log \sigma(\theta_u^T \cdot \theta_v) + kE_{v'\sim P}[\log \sigma(\theta_u^T \cdot \theta_{v'})], \tag{2}$$

where $\sigma(x) = 1/(1 + exp(-x))$ denotes the sigmoid function, k signifies the negative sampling rate and P indicates the negative sampling distribution (We use the uniform distribution in the experiments). The objective of the above equation is to learn a d-dimensional representation $\theta_v \in \mathbb{R}^d$ for each user v. We employ a $|V| \times d$ matrix Θ to represent the social influence representations of all users.

In our framework, the social representation is initially computed based on Eq. 2 and serves as input for the next module.

Characterizing personalized retweeting behavior: The retweeting behavior of users is influenced by both textual features of messages and user features derived from network structures. In order to quantify the likelihood of a user retweeting a message, we employ the dot product between the message feature and the user feature. However, since the dimensions of the message feature and user feature differ, it is necessary to align the two dimensions. We employ a neural network module to first compress each feature to the same dimension - namely, $\mathbf{h}_i$ for the message feature and θ_v for the user feature. This compression is achieved using multilayer perceptron modules in Eq. 3, parameterized by ϕ and ψ, both of which yield an output dimension of 64. After compressing, the normalized features undergo dot product computation in Eq. 4, followed by a sigmoid function $\sigma(\cdot)$ to ensure similarity values in the range of $(0, 1)$. The resulting output $s_{v,i}$ signifies the probability that user v will retweet message i.

$$\mathbf{h}_i^{(c)} = Encoder_{\phi}(\mathbf{h}_i), \quad \theta_v^{(c)} = Encoder_{\psi}(\theta_v), \tag{3}$$

$$\mathbf{h}_i^{(n)} = Norm(\mathbf{h}_i^{(c)}), \quad \theta_v^{(n)} = Norm(\theta_v^{(c)}), \tag{4}$$

$$s_{v,i} = \sigma(\theta_v^{(n)} \cdot \mathbf{h}_i^{(n)}). \tag{5}$$

The parameters in the module of users' personalized retweeting behavior are trained together with the popularity prediction task.

4.2 Learning the Cascade Snapshot Feature

The cascade snapshot of a message m^i records the cascade sequence of activated nodes until time t, i.e., $\{(u_j^i, v_j^i, t_j^i) | t_j^i \leq t\}$. In the cascade model, each user has personalized retweeting behavior and is influenced by his/her neighbors.

In this module, the GCN [Kipf and Welling, 2016] is utilized to describe the cascade model and extract snapshot features. Let $a_v = 1(0)$ represents whether node v is activated(inactivated). The input attribute of a node v is the concatenation of node (user) embedding feature, activation state a_v, personalized retweeting behavior $s_{v,i}$, and other hidden parameters (denoted by $\eta \in \mathbb{R}^{r\times 1}$),

$$\mathbf{x}_{v,0} = [\theta_v^T, \eta^T, a_v, s_{v,i}], \tag{6}$$

where $\mathbf{x}_{v,0}$ has dimension $(d + r + 2) \times 1$. The feature matrix X_0 of all nodes is constructed, where each row represents a user's attribute. In the GCN, the features are aggregated from the local neighborhood and combined as,

$$o_l = \mathcal{A}^{(l)}(X_{l-1}; W_l), \quad X_l = \mathcal{C}^{(l)}(o_l; \mathbf{b}_l), \forall 1 \leq l \leq K, \tag{7}$$

where X_l represents features in different layers and we set $X_l (l \neq 0)$ the same dimension with X_0, $\mathcal{A}^{(l)}(\cdot)$ and $\mathcal{C}^{(l)}(\cdot)$ are the aggregation function and combine function parameterized by W_l and $\mathbf{b}_l$ respectively, $K = 5$ in the experiment. The elements $s_{v,i}$ denote the likelihood of a user towards a message, while η encodes the unknown features in the snapshot. GCN has been widely applied in various tasks, including information diffusion estimation [Chamberlain *et al.*, 2021; Xia *et al.*, 2021; Ko *et al.*, 2020], graph source localization [Wang *et al.*, 2022; Ling *et al.*, 2022b], and graph analogical reasoning [Ling *et al.*, 2022a]. In the module, GCN is leveraged to characterize the underlying diffusion model and extract the snapshot features.

Notably, the element a_v in X_l represents the probability of user v being activated. Let $X_l(a_v)$ represent the element a_v in X_l. To consider the activation probability increment caused by neighbors, an additional variable $a_{v,l}$ is introduced for each node v and each layer l. The activation probability $a_{v,l}$ is calculated as the sum of $a_{v,l-1}$ and $X_l(a_v)$, constrained to be no greater than 1,

$$a_{v,l} = min\{a_{v,l-1} + X_l(a_v), 1\}, l = 1, 2, ..., K. \tag{8}$$

In fact, $X_l(a_v)$ represents the probability increment of user v.

Theorem 1 (Monotonicity of activation probability elements $a_{v,l}$). For any GCN-based diffusion model, the activation probability elements $a_{v,l}$ are monotonic and non-decreasing for all nodes and all layers, assuming that $\mathbb{C}^{(l)}(\cdot)$ in Eq. 8 is a non-negative function.

Proof: The graph convolutional network can be represented as the iteration $\mathcal{A}^{(1)} \circ (\mathcal{C}^{(1)} \circ \mathcal{A}^{(2)} \circ \mathcal{C}^{(2)} ... \circ \mathcal{A}^{(K)} \circ \mathcal{C}^{(K)})$. Since $\mathcal{C}^{(l)}(\cdot) \forall l = 1, 2, .., k$ are non-negative, $X_l = \mathcal{C}^{(l)}(o_l; \mathbf{b}_l) \geq 0$ and $X_l(a_v) \geq 0$. Given that $y = min\{x, 1\}$ is a non-decreasing function, we can conclude that $a_{v,l}$ is monotonic and non-decreasing.□

It is worth noting that many neural units, such as sigmoid, ReLU, satisfy the non-decreasing and non-negative properties of $\mathbb{C}^{(l)}(\cdot)$ in Theorem 1. The cascade snapshot feature module takes cascade snapshots, users' features, and personalized retweeting behavior as inputs, and extracts the snapshot features using the stacked GCN. The output of the module is the concatenation of nodes' feature in the last layer X_l of GCN and the potential activation probability vector of the last layer $a_l = [a_{1,l}, ..., a_{v,l}, ..., a_{|V|,l}]$.

Remark: The cascade snapshot feature module differs from the classical GCN in three perspectives: (1) Users' personalized retweeting behavior is explicitly included in the input. (2) The input user features are pretrained from the network embedding module, as opposed to being optimized by the final task in prior works. (3) The activation probability $a_{v,l}$ is ensured to be monotonic and non-decreasing by utilizing non-negative neural units.

4.3 Learning Multi-resolution Temporal Feature

The temporal feature of the cascade dynamics is crucial for predicting popularity. In this module, we employ a Temporal Convolutional Network (TCN) [Lea *et al.*, 2017] to extract multi-resolution temporal features. Unlike previous researches that mainly used LSTM [Hochreiter and Schmidhuber, 1997] for processing time series data, recent studies have demonstrated that TCN outperforms LSTM. This is because TCN, based on the traditional convolutional neural network, possesses a longer effective memory than LSTM. As a result, TCN is capable of capturing more distant and comprehensive information in time series data.

To handle the temporal cascade data, we first split the cascade records into q time-interval uniform snapshots. For each snapshot, denoted by $\mathbf{g}(t) = [Flatten(X_l), a_l]$, we calculate the snapshot feature. Here, $\mathbf{g}(t)$ represents the concatenation of the last layer of Graph Convolutional Network (GCN) and the activation probability vector a_l. The resulting number q of snapshot features construct the time series input of the TCN, namely, $\mathbf{g}(1), \mathbf{g}(2), ..., \mathbf{g}(t), ..., \mathbf{g}(q)$.

Next, we use the classical dilated causal TCN on the time series input. Let $\mathbf{f} = (f_1, f_2, ..., f_z)$ be the convolution kernel, and d be the expansion factor of the dilation convolution. The dilated causal convolution is given by the equation

$$F(t) = (g *_d \mathbf{f})(t) = \sum_{i=1}^{z} f_i \cdot g(t - d \cdot i). \tag{9}$$

4.4 Joint Prediction Module

Joint feature extraction: The proposed framework incorporates three key components: message content features, users' social influence, and a multi-resolution temporal cascade. Firstly, the users' social influence feature is computed based on social network embedding using Eq. 2. Importantly, this social influence feature is shared among all cascades and is only calculated once within the framework. Secondly, given a set of temporal cascade records, we extract the content feature using the message feature module. Additionally, we leverage Eqs. 3–5 to determine users' personalized retweeting behavior in relation to the message. Thirdly, we split the cascade records into uniform snapshots and extract the multi-resolution temporal feature jointly using the cascade snapshot module and the multi-resolution temporal module. The output of the multi-resolution temporal module is fed into the final prediction module to predict the popularity.

Prediction module: We employ the multi-layer perceptrons (MLPs) as our prediction module. The MLPs take the output from the multi-resolution temporal module as input and consists of two hidden layers. Its output has only one neural unit activated by ReLU function and corresponds to the predicted popularity increment, denoted as $\widehat{\Delta R_t^i}$. We adopt the Mean Squared Logarithmic Error (MSLE) as the loss function, which is defined as follows:

$$\mathcal{L}_{pred} = \frac{1}{N} \sum_{i=1}^{N} \left(\log_2 (\Delta R_t^i) - \log_2 (\widehat{\Delta R_t^i}) \right)^2 + \lambda \mathcal{L}_{reg}, \tag{10}$$

where $\mathcal{L}_{reg}$ is the regularization term for the parameters.

Regarding prediction in message generation stage: In the message generation stage, we assume that a user v generates a message m^i and aims to evaluate the popularity prior to posting it. Within our framework, we set that the cascade record consists of only one element, denoted as C^i, specifically as $C^i = \{(v_0^i, v_0^i, t_0^i = 0)\}$. To ensure consistency in our analysis, we divide the cascade records into q time-interval uniform snapshots. In this process, we designate the first $q-1$ snapshots as empty, while allowing the last snapshot to contain one element from C^i. By implementing these precise settings, our framework is capable of predicting the potential popularity during the message generation stage.

Regarding prediction in short-term and long-term periods: To predict the popularity across different time periods, distinct sets of model parameters are trained under different Δt. The experiment involves training three categories of model parameters specifically for short-term, medium-term, and long-term time predictions, respectively.

5 Experiment

5.1 Datasets

We use four real-world datasets in the experiments, including three public datasets and a private dataset collected by ourselves: (1) **Twitter** [Weng *et al.*, 2013] contains the tweets published between Mar 24 and Apr 25, 2012 on Twitter and their retweets during this period. (2) **Weibo2016** [Cao *et al.*, 2017] was collected on Sina Weibo in China on July 1, 2016. Every cascade in this dataset represents the cascade process of a post. (3) **APS** [2] is the citation relationships before 2017 on American Physical Society (APS). The cascade in this dataset represents citation behavior. (4) **Weibo2022** is a Sina Weibo dataset we collected ourselves between December 1, 2022 and June 1, 2023. For all datasets, we randomly sample 70% as training data, 15% for validation, and the rest for testing. Table 1 shows the statistics of the datasets.

5.2 Baselines

We compare our approach with 10 baseline methods: (1) **Feature-Linear** is a feature-based approach that takes the features of users and messages as input and uses the linear classifier to predict the popularity. (2) **Feature-Deep** is also a feature-based approach. It uses the multilayer perceptron to predict the popularity, with neural layers 64-16-1. (3) **DeepCas** [Li *et al.*, 2017] is the first end-to-end model for the problem. (4) **DeepHawkes** [Cao *et al.*, 2017] integrates deep learning and point processes to predict cascades. (5) **CasCN** [Chen *et al.*, 2019] treats each cascade as a graph sequence and uses the GCN and LSTM to learn cascade representations. (6) **TempCas** [Tang *et al.*, 2021] designs a sequence model to learn macroscopic temporal patterns on the cascade graph. (7) **CasFlow** [Xu *et al.*, 2021] combines the users' representations and cascade features based on GRU and VAE to get representations of cascades. (8) **TCAN** [Sun *et al.*, 2023a] integrates the explicit time embedding and attention mechanism to fully learn the representation of cascade graphs and cascade sequences. (9) **CasTformer** [Sun *et al.*, 2023b] utilizes the transformer mechanism to capture diverse cascade prediction. (10) **CTCP** [Lu *et al.*, 2023] uses a continuous-time graph learning method for cascade prediction.

[2]https://journals.aps.org/datasets

#Datasets	#Users	#Cascades	#Retweets
Twitter	578,913	88,440	7,998,380
Weibo2016	6,738,040	119,313	15,311,973
APS	616,316	207,685	3,304,400
Weibo2022	2,171,833	746,826	8,435,052

Table 1: Statistics of the Datasets

5.3 Evaluation Metrics

We choose three widely used evaluation metrics: Mean Squared Logarithmic Error (MSLE), Mean Absolute Percentage Error (MAPE) and R-Squared (R^2) [Sun *et al.*, 2023b].

5.4 Experimental Settings

In the temporal feature module, we utilize two convolutional layers with expansion factors $d = \{2, 3\}$ respectively to extract multi-resolution temporal features. In the prediction module, we utilize the traditional $l2$ norm as the regularization function $\mathcal{L}_{reg}$ and set $\lambda = 0.001$. In short-term prediction, the time Δt are 4 days for Twitter, 3 hours for Weibo2016, and 3 years for APS. As for long-term prediction, the time Δt are 32 days for Twitter, 24 hours for Weibo2016, and 20 years for APS. The code is available on https://anonymous.4open.science/r/CasMS-B8F7.

5.5 Performance Comparison

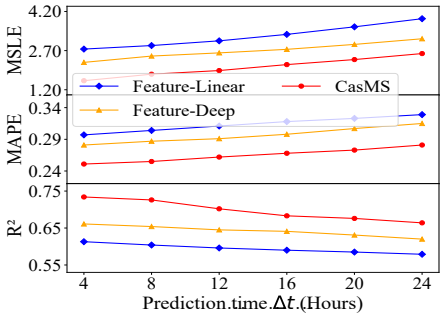

Figure 2: Prediction accuracy in message generation stage on the dataset of Weibo2022.

Prediction in the message generation stage: In the stage of message generation, the observation cascade is empty, hence the prediction can only rely on the message content feature and social graph feature. Among the baseline methods, only Feature-Linear and Feature-Deep are applicable in this scenario. Figure 2 shows the predictive performance of our proposed approach, compared with Feature-Linear and Feature-Deep methods, in the dataset *Weibo2022*. In fig. 2, our method demonstrates a significant improvement over the baseline methods in the message generation stage. By capturing users' personalized retweeting behavior, our approach

Dataset	Twitter						Weibo2016						APS					
Time t	1 Day			2 Days			0.5 Hour			1 Hour			3 Years			5 Years		
Metrics	MSLE	MAPE	R^2	MSLE	MAPE	R^2	MSLE	MAPE	R^2	MSLE	MAPE	R^2	MSLE	MAPE	R^2	MSLE	MAPE	R^2
Feature-Linear	7.615	0.672	0.376	5.778	0.641	0.405	3.517	0.297	0.372	3.325	0.283	0.402	1.583	0.254	0.356	1.496	0.223	0.381
Feature-Deep	6.983	0.632	0.397	5.366	0.613	0.426	2.928	0.289	0.411	2.746	0.277	0.438	1.511	0.231	0.387	1.466	0.209	0.406
DeepCas[2017]	11.587	0.691	0.081	9.283	0.665	0.095	4.142	0.301	0.242	3.829	0.288	0.261	1.966	0.289	0.072	1.785	0.256	0.089
DeepHawkes[2017]	6.045	0.605	0.529	4.295	0.581	0.574	2.819	0.295	0.482	2.632	0.279	0.512	1.388	0.246	0.405	1.213	0.211	0.421
CasCN[2019]	6.520	0.661	0.418	5.187	0.631	0.450	2.635	0.306	0.426	2.522	0.295	0.453	1.684	0.257	0.348	1.421	0.219	0.373
TempCas[2021]	4.071	0.606	0.579	3.977	0.577	0.610	2.583	0.312	0.541	2.497	0.289	0.575	1.534	0.262	0.327	1.396	0.223	0.353
CasFlow[2021]	3.956	0.589	0.586	3.794	0.559	0.627	2.232	0.297	0.567	2.160	0.282	0.598	1.483	0.255	0.375	1.253	0.217	0.401
TCAN[2023]	3.671	0.574	0.607	3.529	0.537	0.638	2.152	0.289	0.584	2.007	0.276	0.603	1.365	0.244	0.412	1.121	0.206	0.427
CasTformer[2023]	3.517	0.536	0.616	3.401	0.515	0.649	2.115	0.276	0.613	1.956	0.264	0.641	1.434	0.254	0.423	1.227	0.218	0.451
CTCP[2023]	3.474	0.529	0.621	3.387	0.506	0.655	2.009	0.285	0.639	1.800	0.269	0.664	1.313	0.247	0.429	1.112	0.211	0.469
CasMS (ours)	**3.171**	**0.512**	**0.655**	**3.044**	**0.484**	**0.683**	**1.821**	**0.264**	**0.684**	**1.606**	**0.246**	**0.734**	**1.159**	**0.235**	**0.477**	**0.971**	**0.197**	**0.502**
Improvement	**8.7%** ↑	**3.2%** ↑	**5.4%** ↑	**10.1%** ↑	**4.4%** ↑	**4.2%** ↑	**9.4%** ↑	**4.3%** ↑	**7.1%** ↑	**10.8%** ↑	**6.9%** ↑	**10.5%** ↑	**11.8%** ↑	**3.9%** ↑	**11.1%** ↑	**12.7%** ↑	**4.6%** ↑	**7.1%** ↑

Table 2: Prediction accuracy on datasets of Twitter, Weibo2016, APS. Smaller MSLE and smaller MAPE are better, and larger R^2 is better.

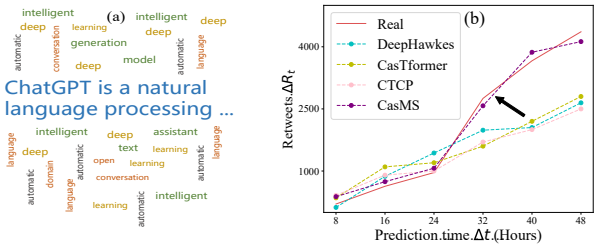

Figure 3: The cascade evolution and prediction of an example message from Weibo2022. (a) The cloud of example messages. (b) The size evolution of an example message in the panel a. Only our method could precisely predict the sharp surge at $\Delta t \in [24, 32]$.

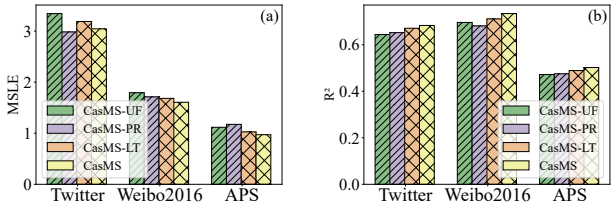

Figure 4: Ablation study on Twitter, Weibo2016, and APS.

excels at predicting the popularity of different messages during the message generation stage.

Prediction in the short-term and long-term periods: Table 2 presents the prediction comparison of our method against 10 baseline methods. Table 2 clearly indicate that our method consistently outperforms the existing approaches with error bars less than 1%. Our method achieves an average improvement of over 10% for the MSLE metric. This notable improvement can be attributed to the characterization of personalized retweeting behavior and the incorporation of multi-resolution temporal feature extraction in our model. Notably, our method outperforms all other methods across various observation time windows and prediction time periods. Considering the consistent agreement of the results for other prediction time periods Δt with those presented in Table 2, we omit them due to space limitations.

A case study: We chose one popular message from Weibo2022 with burst cascades and predicted its popularity at different prediction time periods in fig. 3. In fig. 3, the number of real retweets (colored red line) experiences a sharp surge during the time interval $[24, 32]$. Traditional approaches, which predict a smooth increase in popularity, fail to capture this sudden burst. In contrast, our method intriguingly anticipates such a burst during the aforementioned period, which indicates the effectiveness of our model.

Ablation study: We examine the predictive performance of CasMS by comparing it with various modifications on Twitter, Weibo2016, and APS. Here, we explore the impact of submodules on the overall prediction performance: (1) **CasMS-UF** removes user feature extraction. (2) **CasMS-PR** removes personalized retweeting behavior module. (3) **CasMS-LT** replaces the TCN of CasMS with LSTM.

From fig. 4, we observe that: (1) Based on *CasMS-UF*, the user feature module provides a comprehensive depiction of users, thereby improving predictive performance. (2) Based on *CasMS-PR*, users' personalized retweeting behaviors allow us to decipher their preferences for different kinds of messages. (3) Based on *CasMS-LT*, TCN demonstrates a substantial advantage over LSTM in effectively capturing cascade temporal features.

6 Conclusion

In this study, we studied the problem of cascade prediction in the message generation phase, as well as predictions for different time periods. Our methodology initially captures users' personalized retweeting behavior. We combine GCN and TCN to generate the multi-resolution features. Specifically, GCN extracts cascade features in social graphs, while TCN captures temporal features. The proposed method's effectiveness is demonstrated through extensive experiments conducted on four real-world datasets.

Our present model primarily accommodates static graphs and purely textual communications. We plan to generalize the model to dynamic, evolving graphs, as well as messages enriched with images and a broader spectrum of multimedia content. and future research.

Ethical Statement

There are no ethical issues.

Acknowledgments

The authors acknowledge financial support from the Shenzhen Fundamental Research-General Project (Grant Nos. 20220811155803001, JCYJ20190808162601658), National Natural Science Foundation of China (Grant Nos. 62276171, 62002233, 61972145), Guangdong Basic and Applied Basic Research Foundation (Grant Nos. 2024A1515011938 and 2020B1515120028), Guangdong Peral River Recruitment Program of Talents (Grant Nos. 2019ZT08X603), and Swiftlet Fund Fintech funding, Gang Liu is the corresponding author.

References

[Bian *et al.*, 2020] Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 549–556, 2020.

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[Cao *et al.*, 2017] Qi Cao, Huawei Shen, Keting Cen, Wentao Ouyang, and Xueqi Cheng. Deephawkes: Bridging the gap between prediction and understanding of information cascades. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1149–1158, 2017.

[Chamberlain *et al.*, 2021] Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In *International Conference on Machine Learning*, pages 1407–1418. PMLR, 2021.

[Chen *et al.*, 2019] Xueqin Chen, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Fengli Zhang. Information diffusion prediction via recurrent cascades convolution. In *2019 IEEE 35th international conference on data engineering (ICDE)*, pages 770–781. IEEE, 2019.

[Chen *et al.*, 2022] Wei Chen, Carlos Castillo, and Laks VS Lakshmanan. *Information and influence propagation in social networks*. Springer Nature, 2022.

[Cheng *et al.*, 2014] Justin Cheng, Lada Adamic, P Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, pages 925–936, 2014.

[Chowdhery *et al.*, 2023] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Dong *et al.*, 2019] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32, 2019.

[Gao *et al.*, 2019] Xiaofeng Gao, Zhenhao Cao, Sha Li, Bin Yao, Guihai Chen, and Shaojie Tang. Taxonomy and evaluation for microblog popularity prediction. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(2):1–40, 2019.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

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

[Ko *et al.*, 2020] Jihoon Ko, Kyuhan Lee, Kijung Shin, and Noseong Park. Monstor: an inductive approach for estimating and maximizing influence over unseen networks. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 204–211. IEEE, 2020.

[Lea *et al.*, 2017] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[Leskovec *et al.*, 2007] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5–es, 2007.

[Li *et al.*, 2017] Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. Deepcas: An end-to-end predictor of information cascades. In *Proceedings of the 26th international conference on World Wide Web*, pages 577–586, 2017.

[Ling *et al.*, 2022a] Chen Ling, Tanmoy Chowdhury, Junji Jiang, Junxiang Wang, Xuchao Zhang, Haifeng Chen, and Liang Zhao. Deepgar: Deep graph learning for analogical reasoning. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 1065–1070. IEEE, 2022.

[Ling *et al.*, 2022b] Chen Ling, Junji Jiang, Junxiang Wang, and Zhao Liang. Source localization of graph diffusion via variational autoencoders for graph inverse problems.

In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 1010–1020, 2022.

[Lu *et al.*, 2023] Xiaodong Lu, Shuo Ji, Le Yu, Leilei Sun, Bowen Du, and Tongyu Zhu. Continuous-time graph learning for cascade popularity prediction. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 2224–2232. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Mishra *et al.*, 2016] Swapnil Mishra, Marian-Andrei Rizoiu, and Lexing Xie. Feature driven and point process approaches for popularity prediction. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 1069–1078, 2016.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[Pinto *et al.*, 2013] Henrique Pinto, Jussara M Almeida, and Marcos A Gonçalves. Using early view patterns to predict the popularity of youtube videos. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 365–374, 2013.

[Robles *et al.*, 2020] Juan Francisco Robles, Manuel Chica, and Oscar Cordon. Evolutionary multiobjective optimization to target social network influentials in viral marketing. *Expert systems with applications*, 147:113183, 2020.

[Shen *et al.*, 2014] Huawei Shen, Dashun Wang, Chaoming Song, and Albert-László Barabási. Modeling and predicting popularity dynamics via reinforced poisson processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[Shulman *et al.*, 2016] Benjamin Shulman, Amit Sharma, and Dan Cosley. Predictability of popularity: Gaps between prediction and understanding. In *Proceedings of the international AAAI conference on web and social media*, volume 10, pages 348–357, 2016.

[Sun *et al.*, 2023a] Xigang Sun, Jingya Zhou, Ling Liu, and Wenqi Wei. Explicit time embedding based cascade attention network for information popularity prediction. *Information Processing & Management*, 60(3):103278, 2023.

[Sun *et al.*, 2023b] Xigang Sun, Jingya Zhou, Ling Liu, and Zhen Wu. Castformer: A novel cascade transformer towards predicting information diffusion. *Information Sciences*, 648:119531, 2023.

[Szabo and Huberman, 2010] Gabor Szabo and Bernardo A Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88, 2010.

[Tang *et al.*, 2021] Xiangyun Tang, Dongliang Liao, Weijie Huang, Jin Xu, Liehuang Zhu, and Meng Shen. Fully exploiting cascade graphs for real-time forwarding prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 582–590, 2021.

[Touvron *et al.*, 2023] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[Tsur and Rappoport, 2012] Oren Tsur and Ari Rappoport. What's in a hashtag? content based prediction of the spread of ideas in microblogging communities. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 643–652, 2012.

[Wang *et al.*, 2022] Junxiang Wang, Junji Jiang, and Liang Zhao. An invertible graph diffusion neural network for source localization. In *Proceedings of the ACM Web Conference 2022*, pages 1058–1069, 2022.

[Weng *et al.*, 2013] Lilian Weng, Filippo Menczer, and Yong-Yeol Ahn. Virality prediction and community structure in social networks. *Scientific reports*, 3(1):1–6, 2013.

[Wu *et al.*, 2019] Qitian Wu, Yirui Gao, Xiaofeng Gao, Paul Weng, and Guihai Chen. Dual sequential prediction models linking sequential recommendation and information dissemination. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 447–457, 2019.

[Wu *et al.*, 2020] Zhiang Wu, Changsheng Li, Jie Cao, and Yong Ge. On scalability of association-rule-based recommendation: A unified distributed-computing framework. *ACM Transactions on the Web (TWEB)*, 14(3):1–21, 2020.

[Xia *et al.*, 2021] Wenwen Xia, Yuchen Li, Jun Wu, and Shenghong Li. Deepis: Susceptibility estimation on social networks. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 761–769, 2021.

[Xu *et al.*, 2021] Xovee Xu, Fan Zhou, Kunpeng Zhang, Siyuan Liu, and Goce Trajcevski. Casflow: Exploring hierarchical structures and propagation uncertainty for cascade prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[Yang *et al.*, 2019] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[Zhao *et al.*, 2015] Qingyuan Zhao, Murat A Erdogdu, Hera Y He, Anand Rajaraman, and Jure Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1513–1522, 2015.