Information Sciences 622 (2023) 1109–1127

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Hierarchical attention neural network for information cascade prediction

Chu Zhong [a], Fei Xiong [a,*], Shirui Pan [b], Liang Wang [c], Xi Xiong [d]

[a] School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
[b] School of Information and Communication Technology, Griffith University, Queensland 4215, Australia
[c] School of Computer Science, Northwestern Polytechnical University, Xi'an 10072, China
[d] School of Cybersecurity, Chengdu University of Information Technology, Chengdu 610225, China

ARTICLE INFO

Article history:
Received 18 April 2022
Received in revised form 27 November 2022
Accepted 29 November 2022
Available online 8 December 2022

Keywords:
Cascade prediction
Neural network
Attention mechanism
User influence
Community redundancy

ABSTRACT

Online social networking platforms have drastically facilitated the phenomenon of information cascades, making cascade prediction an important task for both researchers and practitioners. In cascade propagation paths, influential users can often attract more attention. Moreover, the community structure formed by users with similar interests creates information redundancy, thereby restricting the propagation process. These two features greatly affect the growth of the cascades. This paper investigates the incremental prediction of cascades during a future time period based on the evolution of cascades in early stages and proposes a neural network framework with hierarchical attention mechanisms, named Hierarchical Attention Cascade Neural Network (CasHAN). This network has a node-level attention mechanism based on user influence and a sequence-level attention mechanism based on community redundancy. User influence considers the user's own attributes and the influence feedback provided by neighbors, while community redundancy measures information redundancy characteristics that limit cascade propagation. Through hierarchical attention mechanisms, the effective combination of these two features improves the accuracy of cascade increment prediction. Extensive experiments on real-world datasets demonstrate that our approach outperforms other state-of-the-art prediction methods.

1. Introduction

Social networking platforms, such as Facebook, Twitter, YouTube, Weibo, and WeChat, have now become an indispensable part of daily life, making the economy of attention a central component of this era. Only a small fraction of the vast amount of information generated daily on these various platforms receive much attention, owing to people's limited attention. As a result, the task of predicting the future growth trend of information cascades is essential, which can help users alleviate the problem of information overload and provide support for platform operators to carry out advertising marketing [20] and content recommendation [39].

The trajectory and structure of information diffusion, as well as participants in information dissemination, form so-called information cascades [48]. The information cascade prediction problem is often regarded as a classification or regression problem. The classification task aims to predict whether the size of a cascade will reach a certain threshold in the future,

* Corresponding author.
E-mail addresses: 20120178@bjtu.edu.cn (C. Zhong), xiongf@bjtu.edu.cn (F. Xiong), shirui.pan@monash.edu (S. Pan).

https://doi.org/10.1016/j.ins.2022.11.163
0020-0255/© 2022 Elsevier Inc. All rights reserved.

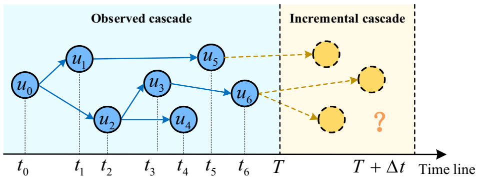

Fig. 1. Example of cascade increment prediction problem. u_0 is the initiator of the information cascade.

while the regression task attempts to predict the exact value of the future cascade. In this paper, we focus on the latter, i.e., predicting the exact increment sizes of information cascades in the future, as shown in Fig. 1. However, owing to the openness of social platforms and various factors that influence users' sharing behaviors, to accurately predict information cascades is considerably challenging.

Persistent efforts have been made towards improving the accuracy and robustness of information cascade prediction over the past decade. Previous studies have typically used generative models inspired by epidemiology [36] or machine learning models that incorporate various features [7,17] to make predictions. Subsequently, information cascade prediction has successfully benefited from deep learning on graphs [5,7,10,16,34]. Nevertheless, two critical contributory factors, individual user influence and community redundancy, are often overlooked during cascade prediction based on graph representation learning.

User influence cannot be ignored in cascade prediction because influential users often tend to attract more attention, leading to a larger range of information cascades. Simply using the number of followers of a user to measure user influence [4,46] may not be sufficiently accurate, especially because some of these followers may be fake accounts created specifically to expand the number of followers. In addition to the number of followers, the number of followers of these followers and their interactions in the surrounding local network are also worthy of attention. Moreover, at the cascade level, users from the same community often publish or spread similar information, which creates information redundancy and affects the cascade diffusion. Therefore, the features of community redundancy must be learned in each cascade path, which benefits the representations of cascade graphs and improves the cascade growth prediction.

To solve these problems, we propose a neural network framework with two-layer attention mechanisms for cascade prediction based on user influence and community redundancy, named Hierarchical Attention Cascade Neural Network (CasHAN), as shown in Fig. 2. Specifically, the random walk sampling is used to convert the complex graph structure into a sequence structure that is easier to handle. The sequence is then encoded by a bidirectional Gated Recurrent Unit (GRU) [28] to obtain node representations. A node-level attention mechanism based on user influence performs a weighted aggregation of node representations to obtain sequence representations. Subsequently, a cascade graph representation is obtained by the weighted aggregation of sequence representations through a sequence-level attention mechanism based on community redundancy. Finally, the following fully-connected layer produces a prediction output.

The main contributions of this work can be summarized as fourfold:

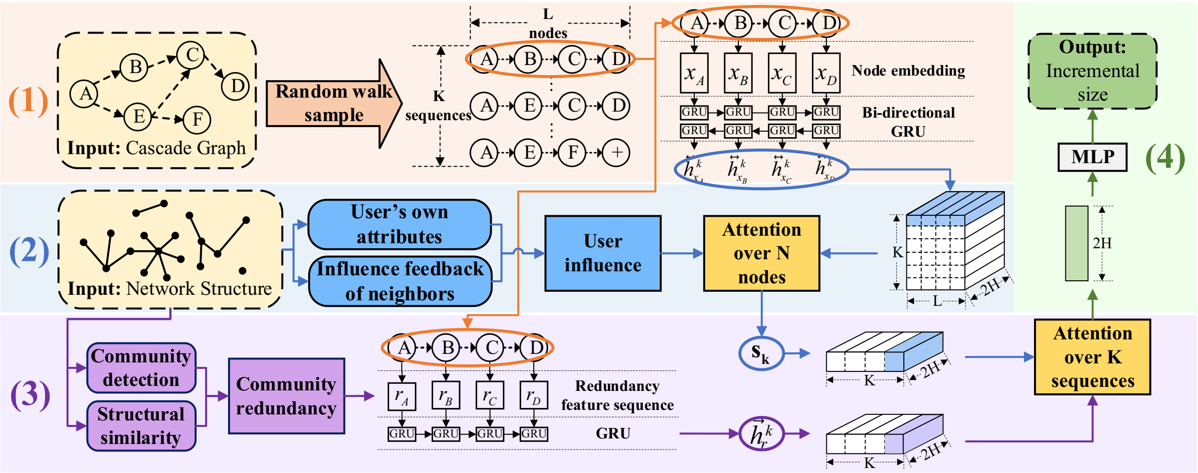

Fig. 2. Overall framework of the proposed CasHAN model. The model comprises four key components: (1) sequence sampling and encoding, (2) node-level attention mechanism based on user influence, (3) sequence-level attention mechanism based on community redundancy, and (4) output module.

- ***Measurement of user influence:*** Based on the number of nodes and topological connectivity of local networks around the nodes, a method to measure user influence is presented, which considers the nodes' own attributes and the influence feedback of their neighbors, rather than simply considering the node degree.
- ***Consideration of community redundancy:*** We define and quantitatively characterize community redundancy based on the effective size of the network to represent the information redundancy features carried by users from the same community in cascade paths.
- ***Establishment of neural network prediction framework:*** We propose a framework of hierarchical attention neural networks with two-layer attention mechanisms for cascade prediction (CasHAN). The node-level attention mechanism distinguishes the different contributions of users to cascade propagation in a sequence based on the different influences of users. Meanwhile, the sequence-level attention mechanism is based on the different community redundancies of sequences and aggregates sequence representations.
- ***Experiments on real-world datasets:*** Extensive experiments on two real-world datasets demonstrate the effectiveness and superiority of our proposed model, compared with several competitive baselines.

The rest of this paper is organized as follows. We briefly review related work in Section 2. Section 3 introduces the problem definition and elaborates on our proposed model CasHAN. Section 4 presents the experimental results and analysis, and Section 5 summarizes the paper and outlines directions for future work.

2. Related work

This section reviews relevant studies on both cascade prediction and attention mechanisms.

2.1. Cascade prediction

Existing work on the prediction of information cascade propagation can be classified into three groups: random point process-based methods, feature engineering-based methods and deep learning-based methods.

2.1.1. Random point process-based methods

The basic idea of such methods is to regard the process of information dissemination as a type of arrival process of user forwarding and sharing of behavioral events. Shen et al. [30] proposed a probabilistic model based on a reinforced Poisson process to describe the citation cascade process in paper networks. Lu et al. [23] proposed a generative framework with a potential user interest layer based on the Poisson process to learn and capture collective user behaviors in cascade systems. Moreover, early research by Sreenivasan et al. [31] found that the extent of user attention impacts the probability of a cascade becoming viral, and showed that the range of the cascade increases with longer attention spans. Li et al. [18] studied and enhanced a cascade prediction model based on a Hawkes point process on the WeChat platform by incorporating visual reasoning. They utilized learning rules summarized by visual reasoning to predict future growth. Kong et al. [15] conducted information cascade modeling of self-excited point processes based on generalized epidemic models, and they established the relation between generalized stochastic Susceptible-Infected-Recovered (SIR) models and self-excited point processes in finite populations (HawkesN). However, the prediction performance of such methods is limited because of various strong assumptions made during the modeling process.

2.1.2. Feature engineering-based methods

These methods typically focus on designing and extracting various hand-crafted features for specific platforms or datasets, such as user features [17,38], content features [8], structural features [35], and temporal features [37]. The core of these methods lies in feature selection and data processing. To address the problem of feature selection, Alweshah et al. [1] chose the monarch butterfly optimization algorithm, which was implemented with a wrapper feature selection method that uses the k-nearest neighbor classifier. Srivastava et al. [32] applied data augmentation in training of deep neural networks to solve the problem of data scarcity during data processing. Xiao et al. [40] devised a time-sensitivity-based predictive model using time and neighbor user features. Similarly, Yang et al. [43] also mined contextual information and user characteristics to assess meme popularity. Carta et al. [6] defined a binary classification task to determine whether the popularity of future posts will increase compared with the average popularity of past posts, and predicted the popularity of future posts on the Instagram platform based on a method of gradient enhancement, incorporating time features, user features and semantic content features of post titles. Purba et al. [25] found that image quality, posting time, image type and user history have a great influence on the prediction of cascades. However, the performance of these methods depends heavily on the extracted features, which are difficult to design and measure uniformly.

2.1.3. Deep learning-based methods

Prediction methods based on deep learning avoid the complicated process of hand-crafting features and are favored by contemporary researchers [19,21,29,33,42,45]. Methods based on a deep neural network are typically end-to-end models that do not require any machine learning stages, and can optimize the model and deal with the problem of data imbalance through a specially designed loss function [11]. Without loss of generality, Recurrent Neural Networks (RNN) are often used to analyze user behavior in social networks or media and for sequence modeling [3]. The first deep learning-based cascade prediction model, DeepCas, was proposed by Li et al. [16] as an end-to-end popularity predictor that lay the foundation for subsequent deep learning-based work. Cao et al. [4] proposed an innovative DeepHawkes model by combining a random point process model with a deep learning framework, bridging the gap between the predictability and interpretability of information cascades. Liao et al. [19] proposed a framework for the deep fusion of temporal processes and content features, and achieved a dynamic fusion of multiple factors through an attention mechanism for information cascade prediction. Cao et al. [5] applied a coupled graph neural network to model the iterative interaction between user activation states and influence propagation for cascade prediction. Xu et al. [42] presented a novel framework for cascade prediction on social media with multimodal features, including visual, text, user, and temporal-spatial information. Nevertheless, user influence and community redundancy are often not deeply considered despite the improved prediction performance of such methods.

2.2. Attention mechanism

Originally applied to machine translation, attention mechanisms have become an important component in neural network architectures and have been applied in numerous applications for natural language processing, statistical learning, speech, and computer vision. Attention mechanisms can be intuitively explained by human visual mechanisms. For example, our visual system tends to focus on parts of information in an image that aids decision-making and ignores irrelevant information [41]. Similarly, while training neural networks, an attention mechanism allows the model to learn the alignment between different parts of the input, capture important information, and discard redundant information [26,44,47]. Zhou et al. [49] utilized an attention mechanism to aggregate information from diverse kinds of neighbors and calculated the importance of different neighbors. Bielski et al. [2] proposed a model with a self-attention mechanism to weigh the relative importance of frames in the time domain to predict the popularity of videos. In this paper, we propose innovative hierarchical attention mechanisms to effectively combine user influence and community redundancy for cascade prediction.

3. Method

This section first gives the formal definitions of the relevant problems studied in this paper, and then introduces our proposed method step-by-step.

3.1. Problem Definition

Basic definitions are as follows.

Definition 1 Network Structure. Let $G = (V, E)$ be the underlying structure of a social network, where V is the set of all users (i.e., node set), and $E \subset V \times V$ denotes the set of all relationships between users (i.e., edge set).

Definition 2 Cascade Graph. A cascade graph represents the diffusion of a piece of information over a period of time. Assuming that we have C cascades, the observed cascade graph $c \in C$ within observation time window T is denoted by $g_c^T = \left(V_c^T, E_c^T\right)$, where V_c^T is a subset of nodes in V and $E_c^T = E \cap \left(V_c^T \times V_c^T\right)$ is the set of edges. A node $u \in V_c^T$ represents a cascade participant (e.g., an author in a paper citation network or a user in a social network) and an edge $(u, v) \in E_c^T$ represents an interaction between two nodes u and v (e.g., citation or reposting).

Definition 3 Cascade Prediction. In this paper, we define the cascade prediction problem as learning a model that can predict the incremental cascade size $\Delta R_c^T = \left|V_c^{T+\Delta t}\right| - \left|V_c^T\right|$, given a set of cascades C within observation time window $t_0 = [0, T]$ and time interval Δt, as shown in Fig. 1. In other words, the cascade prediction is to find a function f(•) which maps g_c^T to ΔR_c^T.

3.2. Proposed model

Our model, CasHAN, as shown in Fig. 2, takes the cascade graph g_c^T and the network structure G as input, and outputs the incremental size of the cascade ΔR_c^T that needs to be predicted. The framework of CasHAN consists of four key components: (1) **Sequence sampling and encoding –** transforming the cascade graph into node sequences by random walk sampling and applying a bidirectional GRU to model the sequences after embedding the nodes into a vector space; (2) **Node-level attention mechanism based on user influence –** considering the user's own attributes and the influence feedback of neighbors to measure user influence, generating node-level attention weights based on user influence, and then obtaining sequence representations after weighted aggregation; (3) **Sequence-level attention mechanism based on community redundancy –** learning the path redundancy feature using the last hidden state of GRU after community detection as well as community redundancy calculation, and then utilizing a sequence-level attention mechanism to obtain the cascade graph representation. (4) **Output module –** feeding the representation of the cascade graph into a fully connected layer to implement the cascade prediction.

After an overview of CasHAN, we now focus on the details of the respective sub-sections.

3.3. Sequence sampling and encoding

Given a cascade graph g_c^T, CasHAN first converts it into a tractable sequence structure by means of random walk sampling. In this way, the initial representation of g_c^T is generated using a set of node sequences. This idea of serialized representations of graphs is essentially an analogy. If we create an analogy between the entire cascade graph and a document, the sampled path sequences will be analogous to the sentences in the document, with each user node in the path resembling a word in the sentence.

Empirically, we apply a fixed number of sequences K and sequence length L (i.e., the number of nodes in each sequence) for random walk sampling. Briefly speaking, for each sampling process, we select the starting node with a probability according to the following formula:

$$p(v) = \frac{d_c(v) + \varepsilon}{\sum_{u \in V_c}(d_c(u) + \varepsilon)} \tag{1}$$

where $d_c(v)$ is the out-degree of node v in the network structure G, ε is a smoother, and V_c is the set of nodes in cascade graph g_c^T. Similarly, after the starting node is sampled, the sampling probability of its neighbor node is:

$$p(v \in N_c(s)|s) = \frac{d_c(v) + \varepsilon}{\sum_{u \in N_c(s)}(d_c(u) + \varepsilon)} \tag{2}$$

where $N_c(s)$ denotes the set of out-degree neighbors of node s in g_c^T. The sampling process for one sequence will be stopped when the sequence length reaches a predefined parameter L, or when we reach a node with no outgoing neighbors. In this instance, a sequence with a length less than L will be filled with a special node "+" until the length reaches L. The entire sampling process continues until K sequences are sampled.

So far, for each cascade graph, we have obtained K paths with the length L to use as the initial representation of the cascade graph. Due to the superiority of a sequential neural network in sequence modeling, we applied this neural network to learn the representation of sequences.

First, the node embedding must be performed. Specifically, each node in a sequence is represented as a one-hot vector, $\boldsymbol{q} \in \mathbb{R}^{|V|}$, where $|V|$ is the total number of nodes in G. In order to avoid the problem of occupying memory space for sparse matrices, we convert each one-hot vector into a low-dimensional dense vector $\boldsymbol{x} = W_e\boldsymbol{q}$, where $W_e \in \mathbb{R}^{H \times |V|}$ is an embedding matrix learned during the training process, and H is an adjustable dimension of the embedding, thus, we have $\boldsymbol{x} \in \mathbb{R}^H$.

Subsequently, the dense vectors are fed into the GRU to generate hidden representations and capture the flow of information. These sampled sequences can be regarded as the paths of information propagation in the cascade graph so that the sequential neural network can be used to model the information flow. More concretely, we apply a bidirectional GRU, a specific type of RNN, to encode the sampled sequences. The gating mechanism of GRU is as follows:

update gate:$u_n = \sigma(W_u x_n + U_u h_{n-1} + b_u)$
reset gate:$r_n = \sigma(W_r x_n + U_r h_{n-1} + b_r)$
candidate state:$\widetilde{h}_n = \tanh(W_h x_n + U_h(r_n \odot h_{n-1}) + b_h)$

$$\text{hidden state}: h_n = u_n \odot \widetilde{h}_n + (1 - u_n) \odot h_{n-1} \tag{3}$$

where $\odot$ represents the element-wise product, $W_u,W_r,W_h,U_u,U_r,U_h \in \mathbb{R}^{H \times H}$ are parameter matrices, and $b_u,b_r,b_h \in \mathbb{R}^H$ are bias parameters in the GRU. Each iterative calculation process of Eq. (3) can be abbreviated as $\boldsymbol{h}_{\boldsymbol{x_n}} = GRU(\boldsymbol{x_n}, \boldsymbol{h}_{\boldsymbol{x_{n-1}}})$.

Note that we are using a bidirectional GRU, which reads sequences from left to right (i.e., forward GRU) as well as from right to left (i.e., backward GRU). The forward GRU makes the sequence representation continuously enriched by the subsequent nodes in the sequence, and the gating mechanism determines the type of information that will be updated, which, to an extent, simulates the information flow process in information diffusion. The backward GRU allows nodes earlier in the sequence to know which nodes will be affected when the cascaded information passes from them. The representation of the n-th node in the k-th sequence, $\overleftrightarrow{\boldsymbol{h}}_{\boldsymbol{x_n}}^{\boldsymbol{k}} \in \mathbb{R}^{2H}$, is computed by the following formulas:

$$\overleftrightarrow{\boldsymbol{h}}_{\boldsymbol{x_n}}^{\boldsymbol{k}} = \vec{\boldsymbol{h}}_{\boldsymbol{x_n}}^{\boldsymbol{k}} \oplus \overleftarrow{\boldsymbol{h}}_{\boldsymbol{x_n}}^{\boldsymbol{k}}$$

$$\vec{\boldsymbol{h}}_{\boldsymbol{x_n}}^{\boldsymbol{k}} = GRU_f\left(\boldsymbol{x_n}, \vec{\boldsymbol{h}}_{\boldsymbol{x_{n-1}}}^{\boldsymbol{k}}\right)$$

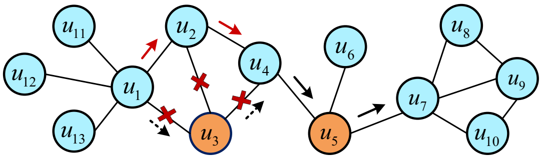

Fig. 3. Illustration of user influence.

$$\overleftarrow{\boldsymbol{h}}^{\boldsymbol{k}}_{\boldsymbol{x_n}} = GRU_b\left(\boldsymbol{x_n}, \overleftarrow{\boldsymbol{h}}^{\boldsymbol{k}}_{\boldsymbol{x_{n-1}}}\right) \tag{4}$$

where $\oplus$ denotes the concatenation operation, and $\vec{\boldsymbol{h}}^{\boldsymbol{k}}_{\boldsymbol{x_n}}$ and $\overleftarrow{\boldsymbol{h}}^{\boldsymbol{k}}_{\boldsymbol{x_n}}$ are forward and backward hidden vectors, respectively. As displayed in Fig. 2, the k-th sequence with length L can be represented as $\left[\overleftrightarrow{\boldsymbol{h}}^{\boldsymbol{k}}_{\boldsymbol{x_1}}, \overleftrightarrow{\mathbf{h}}^{\mathbf{k}}_{\mathbf{x_2}}, \ldots, \overleftrightarrow{\mathbf{h}}^{\mathbf{k}}_{\mathbf{x_n}}, \ldots, \overleftrightarrow{\mathbf{h}}^{\mathbf{k}}_{\mathbf{x_L}}\right]$. Then, we integrate the hidden representations of nodes in the sequence using the node-level attention mechanism presented in the following subsection, which is similar to assembling "words" into "sentences".

3.4. Node-Level attention mechanism based on user influence

For a sentence, the importance of each word in the phrase is different. Similarly, the importance of each user in the dissemination path of information is diverse, which is attributed to the influence of the users themselves. For example, in the reposting behaviors of a social network, users with great influence tend to have a larger number of reposts, that is, a larger-scale information cascade will be formed. Hence, it is vital to consider user influence in cascade prediction.

The most simple and intuitive manner of measuring user influence is to make use of the node degree, i.e., the number of user followers or friends, but in practice, there may be a problem with 'fake followers'. Fake followers are often 'mechanical' users who lack real followers or social activities, and their accounts are created to increase the number of followers of other users, making those users appear to have greater influence. As a result, using the node degree without mitigation leads to an inaccurate measurement of user influence.

Therefore, we consider the node's own attributes and the influence feedback of neighbor nodes to measure the user's influence, based on the node degree and the topological connectivity of the local network around the node. The combination of these two parts can jointly reflect the influence of users and avoid the situation created by fake users. The topological connectivity between nodes in a local network is usually measured by the local clustering coefficient.

It is worth noting that the local clustering coefficient of a node itself usually plays a negative role during diffusion. That is to say, when a node's direct neighbor nodes are highly connected to each other, the node's own influence will decrease during propagation, because even without this node, information can be propagated among its surrounding neighbors. As shown in Fig. 3, u_3 and u_5 have the same sum of first- and second-order neighbors, but the connectivity between u_3's first-order neighbors u_1, u_2 and u_4 is higher than that between u_5's first-order neighbors u_4, u_6 and u_7. When u_3 is missing, the information can still spread from left to right normally. On the contrary, if u_5 is missing, the information dissemination chain on the right half will be broken. Therefore, in Fig. 3, the influence of u_5 should be greater than that of u_3 because of the negative effect of the local clustering coefficient of node u_5 itself.

Consequently, we use $\exp(c_n)$ to reflect the negative effect of the local clustering coefficient of node n itself. At the same time, considering that the more neighbors the node itself has, the more likely there are fake followers, we use $\ln(|N(n)|)$ to characterize this effect of first-order neighbors on user influence. Moreover, the clustering coefficient of neighbors reflects the property of a node's indirect neighbors, which belongs to the influence feedback of the neighbors on the node. Unlike direct neighbors, high connectivity between indirect neighbors can lead to a wider range of information dissemination and avoid fake followers, thus enhancing the influence of the users themselves. Accordingly, the influence I_n of user n in the network structure can be calculated by the following equations:

$$I_n = \ln(|N(n)|) \cdot \exp(c_n) \cdot (\beta \cdot f_{num} + (1 - \beta) \cdot f_{con}) \tag{5}$$

$$f_{num} = \sum_{w \in N(n)} \frac{|N(w)|}{\sqrt{\sum_{i \in N(n)} |N(i)|^2}} \tag{6}$$

$$f_{con} = \sum_{w \in N(n)} \frac{c_w}{\sqrt{\sum_{i \in N(n)} c_i^2}} \tag{7}$$

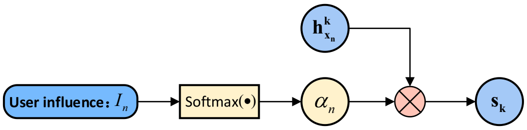

Fig. 4. Node-level attention mechanism based on user influence.

where $N(n)$ is the set of nearest neighbors (i.e., one-hop neighbors) of node n, c_n is the clustering coefficient of node n, and β is an influence balance factor with a value between 0 and 1. Additionally, if $|N(n)|$ equals 0, indicating that the user n does not have any neighbors, then their influence is considered to be 0.

In Eq. (5), $\ln(|N(n)|) \cdot e^{-c_n}$ represents the property of the node itself, and $(\beta \cdot f_{num} + (1-\beta) \cdot f_{con})$ represents the influence feedback of its neighbors. f_{num} stands for the contribution of the number of second-order neighbor nodes, and f_{con} indicates the contribution of the connectivity between second-order neighbor nodes. For each neighbor node $w \in N(n)$, we consider its degree and clustering coefficient, and use a standardized function $u(x) = x/\sqrt{\sum x^2}$ to process them. Here, standardization aims to eliminate the impact of data with different properties and solve the problem of the measurement results being dominated by a certain attribute due to the difference between the value ranges of the two attributes, so that indicators of different properties can correctly reflect the comprehensive results of different contributions.

The local clustering coefficient of a node measures how closely the nearest neighbors of a node form a group, and its basic calculation formula is as follows:

$$c_n = \frac{|e_{jk} : j, k \in N(n), e_{jk} \in E|}{|N(n)|(|N(n)| - 1)/2} \tag{8}$$

where E denotes the set of edges between nodes, the numerator represents the number of actual edges between the nearest neighbor nodes, and the denominator is the maximum number of possible edges between $|N(n)|$ nearest neighbors.

Concerning research analyses of user influence, the paper [24] shows that a model based on percolation is theoretically able to find essential spreaders. The percolation model simulates the spread of information. It simulates the percolation of the network and searches for the best set of seed nodes in the network, but this method cannot provide the appropriate influence value for each node in the network. Moreover, for the problem of influence measurement based on the static network topology, we are exploring concerns when calculating the influence size of each node in the cascade according to various indicators, which is not the influence maximization problem based on a dynamic information propagation model studied in the paper [24]. In addition, we have conducted experiments based on the idea of Morone et al. [24] with poor results, which reconfirmed the superiority of our influence measurement metric. We aim to propose a reasonable user influence measurement metric based on the existing network topology such that the weight of user-level attention can be generated. Therefore, a series of theoretical methods represented by that paper are not suitable for our current research.

Through the representation of user influence, we use the *softmax* function to generate the user's attention weight in the path:

$$\alpha_n = soft\max(u_n) = \frac{\exp(I_n)}{\sum_n^N \exp(I_n)} \tag{9}$$

Then, through weighted aggregation, the representation of the k-th cascade path $\boldsymbol{s_k} \in \mathbb{R}^{2H}$ is as follows:

$$\boldsymbol{s_k} = \sum_{n=1}^{L} \alpha_n \overset{\leftrightarrow}{\boldsymbol{h}}{}^{\boldsymbol{k}}_{\boldsymbol{x_n}} \tag{10}$$

Eq. (10) is the attention mechanism over L nodes, which is a node-level attention mechanism based on user influence. Fig. 4 illustrates the entire process.

3.5. Sequence-Level attention mechanism based on community redundancy

This subsection utilizes the proposed concept of community redundancy to implement the sequence-level attention mechanism to represent the graph, stemming from the fact that the opinions of neighbors from the same social group are similar and thus redundant. Here, a social group typically refers to a community structure formed by people in a social network who often gather together because of common interests. Users in a community structure connect closely to each other and interact frequently. Accordingly, on each spreading path of the information, such features of information redundancy carried by users will reduce the information propagation range, thus we again learn this effect of the propagation path through a sequential neural network. Learning the redundancy features of each propagation path of information is conducive to setting the corresponding weights to each path in the attention mechanism. To introduce the definition of community redundancy, we first present the following lemma.

Lemma: In a community structure, users' neighbors connect more closely to each other, which increases information redundancy, reduces the effective size of the network, and reduces the range of cascade diffusion.

Proof. According to the information dynamics theory and the homogeneous mixing hypothesis, the information dissemination density ρ of an infected node is:

$$\rho = \frac{n_S}{N} = \frac{N - n_I}{N} = 1 - \frac{n_I}{N} \tag{11}$$

where n_S and n_I are the numbers of susceptible and infected nodes in a node's neighbors, respectively. N is the total number of all neighbors of the node.

This form of dissemination can be explained with the help of Fig. 5. u_1,u_2,u_3, and u_4 are in the same community structure and are neighbors to each other. Suppose u_1 is infected and transmits information to u_2, u_3 and u_4 at time t. Then at time $t+1$, u_4 will choose its neighbors to continue spreading information. However, since u_2 and u_3 are common neighbors of u_1 and u_4, among the neighbors of u_4, we note that u_1, u_2 and u_3, have all already received such information, that is, they are all infected. Therefore, the effective range of information dissemination from u_4 will be reduced, that is to say, nodes that have been infected need to be removed. Actually, the part that needs to be eliminated in the information dissemination is equivalent to information redundancy.

Therefore, in a community structure, users connect more closely and have more common neighbors, which introduces information redundancy, thereby reducing the range of cascade diffusion. Consequently, for node i in a weighted undirected graph $M = (V_m, E_m)$, we define the effective size ES_m of the network as follows:

$$l_{jk} = \frac{p_{jk}}{\max\limits_{q\in V_m} p_{jq}} \tag{12}$$

$$\mu_{ik} = \frac{p_{ik} + \tau}{\sum_{j\in V_m left\{i\}} (p_{ij} + \tau)} \tag{13}$$

$$\begin{aligned} ES_m &= \sum_{j\in V_m left\{i\}} \left(1 - \sum_{k\in V_m left\{j,i\}} \mu_{ik} l_{jk}\right) \\ &= |V_m| - \sum_{j\in V_m left\{i\}} \sum_{k\in V_m left\{j,i\}} \mu_{ik} l_{jk} \\ &= |V_m| - \sum_{j\in V_m left\{i\}} \sum_{k\in V_m left\{j,i\}} \left(\frac{p_{ik} + \tau}{\sum_{j\in V_m left\{i\}} (p_{ij} + \tau)} \left(\frac{p_{jk}}{\max_{q\in V_m} p_{jq}} \right) \right) \\ &\leqslant |V_m| \end{aligned} \tag{14}$$

Then, the community redundancy r_i of node i can be defined as:

$$\begin{aligned} r_i &= \sum_{j\in V_m left\{i\}} \sum_{k\in V_m left\{j,i\}} \mu_{ik} l_{jk} \\ &= \sum_{j\in V_m left\{i\}} \sum_{k\in V_m left\{j,i\}} \left(\frac{p_{ik} + \tau}{\sum_{j\in V_m left\{i\}} (p_{ij} + \tau)} \left(\frac{p_{jk}}{\max_{q\in V_m} p_{jq}} \right) \right) \end{aligned} \tag{15}$$

Explanation. V_m is the set of nodes in community M where node i exists. For each node pair$\{(j,k)|j,k \in V_m\}$, we compute a weight p_{jk} to indicate the similarity between them. Note that node similarity can indicate the quantitative property of a node's common neighbors. In other words, even if two nodes are not connected directly, they can share information through common neighbors. Therefore, such structural equivalence can be reflected by node similarity. Here, we use SimRank [13] that considers node connectivity to calculate the similarity score associated with each node pair in the community. To achieve run-time efficiency, we use the similarity scores obtained after the first iteration, which are essentially equal to a normalized version of a co-citation. Given $\{p_{jk}\}$, the relative similarity l_{jk} can be obtained by Eq. (12) for standardization. If $\max\limits_{q\in V_m} p_{jq}$ equals 0, i.e., p_{jk} equals 0 for all k, then we consider the value of the relative similarity $p_{jk}/\max\limits_{q\in V_m} p_{jq}$ to equal 0, indi-

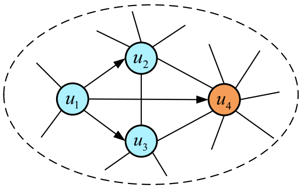

Fig. 5. Schematic illustration of information dissemination.

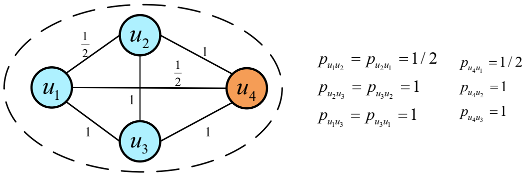

Fig. 6. Example of information redundancy.

cating that no information is shared with any node k to cause redundancy. Subsequently, in order to sum up the influence of all nodes in community M on node i, we further define μ_{ik} as shown in Eq. (13) that represents the contribution portion of node k's influence on node i. τ is an additive smoothing parameter, which was set to 10^{-8} in our experiments. Besides, since there is no ground-truth for community labels, community structure is computed by InfoMap [27], which is a widely used community detection method. Finally, we get the effective size of the network shown in Eq. (14) by subtracting the weighted sum of the relative similarity of all nodes in the community to node i from the total size (i.e., the number of nodes) of the network. Meanwhile, from Eq. (14), $\sum_{j\in V_m left\{i\}}\sum_{k\in V_m left\{j,i\}}\mu_{ik}l_{jk}$ can be interpreted as the redundancy for node i, as shown in Eq. (15).

Example. The similarity scores for each node pair in a community structure shown in Fig. 6 are listed aside. Consider node u_4 as an example. Evidently, $l_{u_1u_2} = l_{u_2u_1} = 1/2$, $l_{u_1u_3} = l_{u_3u_1} = l_{u_2u_3} = l_{u_3u_2} = 1$, and $\mu_{u_4u_1} = (p_{u_4u_1} + \tau)/((p_{u_4u_1} + \tau) + (p_{u_4u_2} + \tau) + (p_{u_4u_3} + \tau)) = 1/3$, similarly, $\mu_{u_4u_2} = \mu_{u_4u_3} = 1/3$. Hence, $r_{u_4} = \left(\mu_{u_4u_1}h_{u_2u_1} + \mu_{u_4u_3}h_{u_2u_3}\right) + \left(\mu_{u_4u_2}h_{u_1u_2} + \mu_{u_4} u_3h_{u_1u_3}\right) + \left(\mu_{u_4u_1}h_{u_3u_1} + \mu_{u_4u_2}h_{u_3u_2}\right) = ((1/3)\times(1/2) + 1/3) + ((1/3)\times(1/2) + 1/3) + (1/3 + 1/3) = 5/3$, i.e., the redundancy carried by u_4 is $5/3$. Moreover, $ES = 3 - 5/3 = 4/3$. Because of information redundancy, the effective size of the network for node u_4 changes from 3 (a total of three nodes can be propagated) to $4/3$, that is, the range of cascade propagation is reduced.

Here, it is worth mentioning about the community structure that the paper [22] shows that clusters of nodes with the same interests (not necessarily communities defined by modularity) are essential for cascade formation. However, the datasets used in our experiments contain an explicit network topology, and the paper [22] indicates that the performance of a community detection algorithm that directly uses topology information is better than using node clusters to indirectly infer community structure. Thus, we directly utilized the community detection algorithm to obtain the community structure instead of considering the node clusters. Moreover, this study focuses on the effect of information redundancy introduced by a community structure on the cascade, rather than community structure inference. In other words, what matters is the impact of the community that already exists, not how the community is formed. Accordingly, using the topology structure directly for community detection is a superior choice, which ensures accuracy and reduces complexity. The research direction of node clusters can be further explored in future work without an explicit network topology.

The above analysis shows that each node carries a certain amount of community redundancy, and this redundancy accumulates with the extension of the path, which greatly affects the process of information dissemination. Thus, for the k-th cascade path, we can extract the associated community redundancy feature of each node in the path and obtain the corresponding redundancy feature sequence $R_k = \{r_1, \ldots, r_n, \ldots, r_L\}$. Then, as shown in Fig. 2, we feed the redundancy feature sequence R_k into the GRU, where the hidden state of sequence k can be updated by

$$\overrightarrow{h}^k_r = GRU\left(r_n, \overrightarrow{h}^k_{r-1}\right) \tag{16}$$

Taking the hidden state of the last node in each sequence as the redundancy feature of the sequence, for K sequences, we obtain K redundancy features, denoted as $R = \left[\overrightarrow{h}^1_r, \ldots \overrightarrow{h}^K_r\right]$.

From Eq. (10), we obtain the representation $\boldsymbol{s_k}$ of the k-th path and then assign the corresponding attention weight to each path according to the redundancy feature $\overrightarrow{h}^k_r$ (hereinafter $\boldsymbol{h^k_r}$). The weight η_k is formalized as

$$\eta_k = \frac{\exp\left(\omega\left(\boldsymbol{s_k}, \boldsymbol{h^k_r}\right)\right)}{\sum_{k=1}^K \exp\left(\omega\left(\boldsymbol{s_k}, \boldsymbol{h^k_r}\right)\right)} \tag{17}$$

In this manner, η_k is the attention to the hidden state representation of the k-th sequence in the cascade graph, and $\omega\left(\boldsymbol{s_k}, \boldsymbol{h^k_r}\right)$ is obtained by the following function

$$\omega\left(\boldsymbol{s_k}, \boldsymbol{h^k_r}\right) = A_k \tanh\left(W_k\boldsymbol{s_k} + U_k\boldsymbol{h^k_r}\right) \tag{18}$$

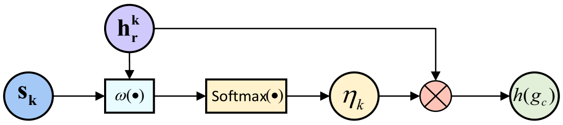

Fig. 7. Sequence-level attention mechanism based on community redundancy.

Table 1
Dataset statistics.

Datasets		Aminer	Sina Weibo	Twitter
# cascades	train	3920	4350	3442
	val	820	540	430
	test	860	550	432
# nodes in G		102,202	395,097	442,158

where $A_k \in \mathbb{R}^{1\times 2H}$,$W_k$, and $U_k \in \mathbb{R}^{2H\times 2H}$ are parameter matrices. So far, we can obtain a representation of cascade graph g_c via the attention mechanism as displayed in Fig. 7. The final representation $h(g_c) \in \mathbb{R}^{2H}$ is as follows:

$$h(g_c) = \sum_{k=1}^{K} \eta_k \boldsymbol{s_k} \tag{19}$$

3.6. Output module

Ultimately, the output module consisting of a multi-layer perceptron (MLP) receives the cascade graph representation $h(g_c)$ as input and outputs the prediction of the cascade increment size:

$$\Delta R_c^T = MLP(h(g_c)) \tag{20}$$

This fully-connected layer does not introduce too much model complexity and while ensuring the ability of nonlinear modeling.

The objective function to be minimized is defined as

$$L = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 + \lambda \sum_{\theta\in \mathrm{P}} \| \theta \|_2 \tag{21}$$

where P is the set of parameters and λ is the $L2$ regularization coefficient. $L2$ regularization is added to the objective function to avoid overfitting and accelerate the convergence progress. Note that we predict a scaled version of the actual cascade incremental size, i.e., $y_i = \log_2\left(\triangle R_i^T + 1\right)$, following the practice of DeepCas [16].

4. Experiments

This section presents the comprehensive experiments conducted on real-world datasets to validate the performance of the proposed CasHAN model and analyzes the experimental results to understand the role of each module in CasHAN for cascade prediction.

4.1. Datasets

The proposed model CasHAN is applied on three different cascade prediction scenarios to evaluate its performance. The first scenario predicts citation cascades of scientific papers, the second predicts reposting cascades on Sina Weibo, which is a microblog social network in China, and the third predicts spreading cascades of tweets on Twitter. Table 1 lists the statistics of the datasets.

Aminer. This dataset released by ArnetMiner focuses on a scientific paper citation network and is publicly available on the website.[1] It is widely used in cascade prediction work such as the research in [16,21]. We construct the global network structure G by using citation data between 1992 and 2002. Specifically, we draw an edge between nodes A and B if B has ever cited A's paper (that is, A's paper is a reference of B's paper). Thus, the citation cascade for a given paper includes all authors who have written or cited that paper. After arranging the dataset chronologically, papers published between 2003 and 2007 were grouped into the training set. Papers published in 2008 and 2009 were used as the validation and test sets, respectively. Note that nodes

[1] https://www.aminer.cn/citation.

involved in the above citation data are all in G. The observation time length T and prediction time interval Δt were set to 1 year and 1, 2, 3 years, respectively.

Sina Weibo. This dataset was collected and provided by Cao et al. [4] and is publicly available on the website.[2] It contains Weibo messages published on June 1, 2016 with more than 10 retweets, as well as the retweets of these messages within 24 h after publication. In addition to user id, the forwarding information includes retweet paths. We consider cascades with a published time between 8:00 and 18:00 and where the number of retweets does not exceed 1000. These cascades are used to generate the global network structure G. Next, the dataset was split into training, validation, and test sets in chronological order of publication. Specifically, the first 80 % are used as the training set, the middle 10 % as the validation set and the remaining 10 % as the test set. Observation time length T and prediction time interval Δt were set to 1 h and 1, 2, 3 h, respectively.

Twitter. This dataset, provided by Hodas et al. [12], contains URL links (messages) posted on Twitter in October 2010 and the corresponding participating users as well as the participation time. It is publicly available on the website.[3] Each URL is interpreted as a cascade of information that spreads between users. At the same time, the dataset provides the structural information of the following relationship network among users. Similarly, we consider messages with the number of retweets greater than 10 and less than 1000, and use the first 80 % of the retweets in the order of publication time as the training set, and the remaining 20 % are equally divided into the validation and test sets. The observation time length T and prediction time interval Δt were set to 1 h and 1, 2, 3 h, respectively.

4.2. Comparison methods

For comparison with the proposed model, we evaluate the following methods on the datasets.

Features-linear. As demonstrated by previous studies [9,17,35] structural features, user features, temporal features and content features are all commonly used and effective features in cascade prediction. In this paper, we primarily extract user features and structural features, since our proposed model CasHAN considers these two aspects. The extracted handcrafted features include: the mean and 90th percentile of the local degrees of each node in a cascade graph, diameter, transitivity and average clustering coefficient of the cascade graph, number of leaf nodes, all nodes and triangles in a cascade graph, edge density in a cascade graph, and publisher's id as well as the global degree in a global graph. These features are combined to represent the cascade graph and are fed into a linear regression with L2 regularization.

DeepCas [16]. This is the first end-to-end deep learning method for cascade prediction in academia. It mainly uses the structure of the cascade graph and node identity information to make predictions. Specifically, it also uses a recurrent neural network to model cascaded sequences, and attention mechanisms for the weighted aggregation of the sequence representations, but its attention mechanisms are all based on mathematical assumptions rather than considering actual situations.

DeepHawkes [4]. Based on the Hawkes process [46], the DeepHawkes model was proposed, which introduces user representations, path modeling based on an RNN, and a non-parametric time decay function into the modeling of a point process intensity function. It combines deep learning methods with generative models and bridges the gap between the prediction and understanding of information cascades.

CasCN [7]. The idea of this state-of-the-art approach is similar to that of DeepHawkes, which is based on a self-exciting mechanism for model construction. The main difference between CasCN and DeepHawkes is that CasCN uses a Graph Convolutional Network (GCN) [14] to extract the structural information of the cascade graph in the process of generating the vector representation of the cascade graph, and samples the cascaded graph into a sequence of subgraphs rather than propagation paths.

Coupled-GNN [5]. This state-of-the-art deep learning method represents a global propagation graph for cascade prediction. Specifically, it models the iterative interplay between node activation states and the spread of influence through coupled graph neural networks. The structural representation of the final message is obtained from the active states of all users representing the pooling aggregation.

4.3. Experimental setup

We adopt the Mean Square Error (MSE) and the Root Mean Square Percentage Error (RMSPE) to evaluate the predictive performance of the model, both of which are widely used regression evaluation metrics in the field of cascade prediction. Denote y_i and $\hat{y}_i$ as the ground truth value and the predictive value of the i-th cascade, respectively. Then, the calculation formulas for the MSE and RMSPE are as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{22}$$

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\hat{y}_i - y_i}{y_i}\right)^2} \tag{23}$$

where n is the total number of cascades. The lower the MSE and RMSPE, the better the prediction performance.

[2] https://github.com/CaoQi92/DeepHawkes.

[3] https://www.isi.edu/~lerman/downloads/twitter/twitter2010.html.

For model parameters, we choose λ from $\left\{10^{-8}, 5\times10^{-8}, \cdots, 0.01, 0.05, 0.1, 0.5, 1\right\}$, the number of units in the hidden layer of each GRU from $\{16, 32, 64\}$, and the learning rate for user embedding and other variables from $\{0.0005, 0.001, 0.005, 0.01, 0.05\}$. Following the settings of DeepCas, the embedding size of users is set to 50, the number of fully connected layers is 2, and the hidden layer dimensions are 32 and 16, respectively. The value of the smoother ε is 0.01. Empirically, the number of sequences K and the sequence length N are equal to 200 and 10, respectively. Moreover, the batch size for each iteration is set to 32, and the training process stops immediately as long as the loss of the validation set does not decrease for 10 consecutive iterations.

4.4. Overall performance

The overall prediction performances of all competing methods on the three different datasets is displayed in Fig. 8 and Fig. 9. Our proposed model CasHAN outperforms all other baseline methods with a significant drop in the MSE as well as RMSPE on all datasets. For example, on the Aminer dataset, when the prediction time interval Δt is 1 year, the prediction error in terms of the MSE of CasHAN is 2.043, while the other five methods increase by 0.089, 0.153, 0.233, 0.376 and

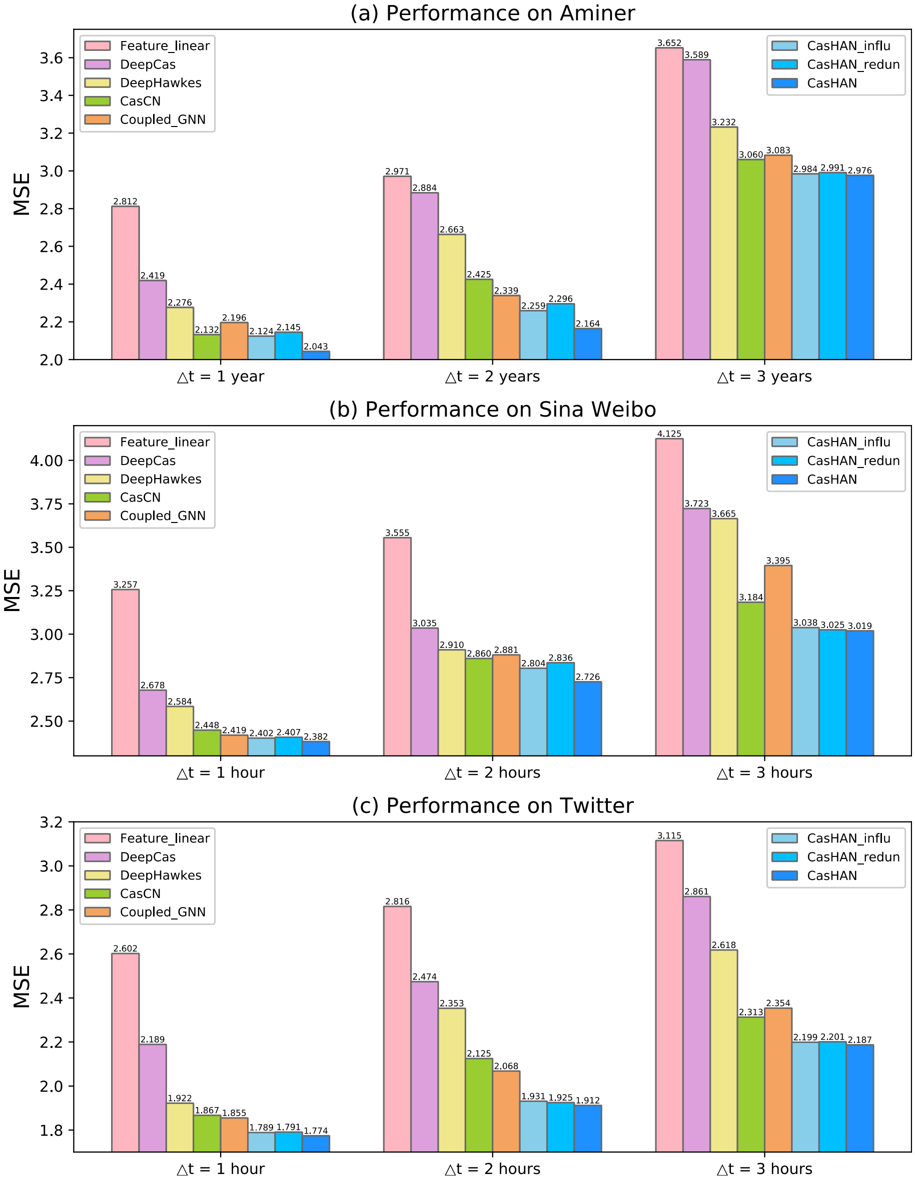

Fig. 8. Overall prediction performance on the MSE.

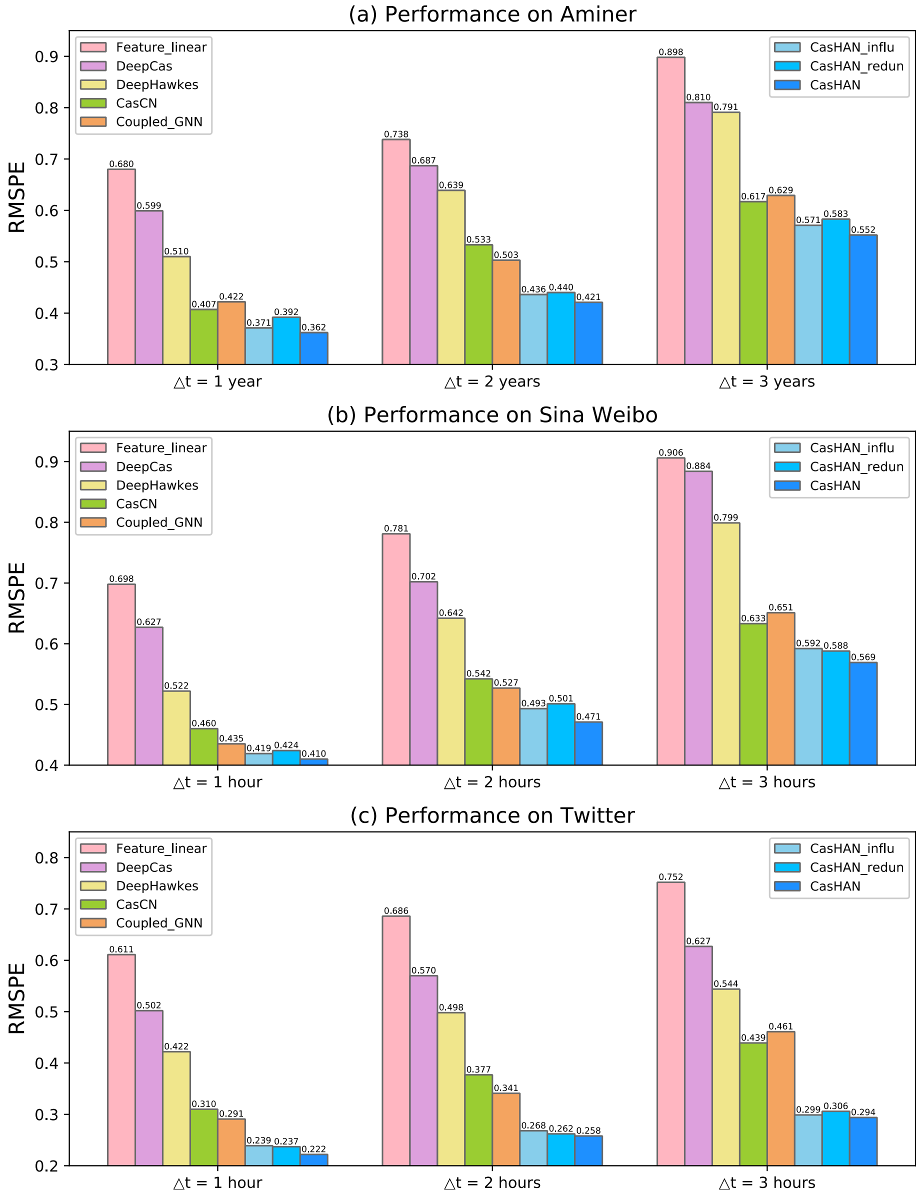

Fig. 9. Overall prediction performance on the RMSPE.

0.769, respectively. Meanwhile, the RMSPE was 0.362, while the other five increased by 0.045, 0.06, 0.148, 0.237, and 0.318, respectively.

Among the five methods, the Feature-linear model tends to be the least satisfactory, because the prediction performance of this model always needs to rely on hand-crafted features, which are difficult to design and require considerable prior knowledge. It cannot automatically learn the corresponding effective features, so its prediction performance is limited.

DeepCas introduces an end-to-end deep learning approach to cascade prediction, and such an innovative attempt provides the model performance with a great advantage over the Feature-linear model. This improvement is attributed to the deep learning method being able to automatically learn feature representations and avoids the tedious feature extraction process. Nonetheless, it is still worse than other deep learning-based methods due to its assumptions regarding node and sequence aggregation. DeepHawkes integrates a self-exciting point process based on this, which simultaneously improves the prediction performance and increases the interpretability of the deep learning model. However, since the model represents the intensity function under the assumption of a self-exciting point process, the risk of an incorrect model selection remains.

As for the two state-of-the-art approaches, CasCN and Coupled-GNN, their prediction errors are relatively close on the three datasets, and are lower than those of the above three methods. Compared with DeepHawkes, CasCN uses a GCN to extract the structural information of the cascade graph, and the prediction performance is improved. Coupled-GNN models the cascade effect through a graph neural network, and also reduces the prediction error. Nevertheless, the prediction errors of CasCN and Coupled-GNN are still at least 6 % and 3 % higher than those of our model on the MSE metric, and at least 4 % and 2 % higher on the RMSPE metric, respectively. Compared with Coupled-GNN, which learns complex global propagation graphs, the effective structural representations of local cascade graphs in our model provide a larger contribution to cascade prediction. Moreover, CasCN is still based on a self-exciting process and lacks an attention mechanism, which limits its prediction performance.

CasHAN comprehensively considers the effects of user influence and community redundancy, and generates two-level attention mechanisms to learn the representations of cascade graphs. The results shown in Fig. 8 and Fig. 9 prove the effectiveness of the proposed model. In addition, our model performs relatively well when using different prediction time intervals Δt. The larger the time interval Δt, the higher the MSE and RMSPE, and the difficult in making good cascade predictions increases. It is always more challenging to make long-term predictions than short-term predictions, which is in line with intuitive cognition. After all, the information cascade may grow and change significantly over time owing to various external factors.

Comparing the three different scenarios of cascade prediction, the prediction errors on the Sina Weibo dataset are generally larger than those on the other two datasets. One possible explanation is that, compared with the Aminer dataset, Sina Weibo is a social platform with a huge amount of traffic and can be unstable and unpredictable even in a short period of time, making the prediction of the future increment of cascades more complicated. However, the experiments on Twitter, which is also a social platform, show a better predictive performance from the model. This is because the Twitter dataset provides a longer average cascade length (evidenced by fewer cascades with more nodes), leading to lower prediction errors by increasing the amount of known data that can be observed.

4.5. Ablation study

To demonstrate the effectiveness of the two main components of CasHAN, we propose two variants, denoted as **CasHAN-influ** and **CasHAN-redun**.

CasHAN-influ includes the component of a node-level attention mechanism based on user influence without distinguishing differences in sequence redundancy features, so each sequence contributes equally to cascade propagation and the representation of a cascade graph. Therefore, we consider the method of mean pooling to directly and simply aggregate each sequence representation into a graph representation.

CasHAN-redun removes the node-level attention mechanism based on user influence features, and uses only the sequence-level attention mechanism based on community redundancy features. We simply assume a multinomial distribution over nodes following the work of DeepCas [16] without considering user influence.

Fig. 8 and Fig. 9 show the prediction performance of these two simplified variants of CasHAN. Both the MSE and RMSPE of CasHAN-influ and CasHAN-redun are lower than those of the baseline models on different datasets, except for the case in which CasHAN scores higher than CasCN by 0.013 in terms of the MSE when the prediction interval is 1 year on the Aminer dataset. Furthermore, the prediction error of the variants with either component removed increases compared with that of CasHAN. This indicates that these two critical components of CasHAN are indispensable, and both contribute to cascade prediction.

4.6. Analysis of parameter sensitivity

4.6.1. Observation time length *T*

The length of the observation time determines the amount of known cascade information, which largely affects the prediction results. Thus, we explore the effect of variable T on the prediction performance of CasHAN and its variants. On the Sina Weibo and Twitter datasets, with a fixed prediction time interval Δt of 1 h, we vary T from 1 h, 2 h to 3 h. On the Aminer dataset, the prediction time interval Δt is fixed at 1 year and T varies from 1 year, 2 years to 3 years. The results are shown in Table 2.

Table 2
Prediction performance with different observation time *T*.

T	Aminer		Sina Weibo		Twitter	
	MSE	RMSPE	MSE	RMSPE	MSE	RMSPE
1 year/hour	2.043	0.362	2.382	0.410	1.774	0.222
2 years/hours	1.925	0.349	2.258	0.396	1.552	0.201
3 years/hours	1.802	0.317	1.973	0.362	1.224	0.178

Table 2 shows that when the observation time length T increases, the MSE and RMSPE values decrease. This proves the intuitive fact that a longer observation time makes more information of a cascade available, and the prediction becomes easier. That is, the more information that is known, the more helpful that predicting unknown information should be. Furthermore, the metrics change more obviously on the Twitter dataset, which is also due to the longer average cascade length provided by the Twitter dataset, leading to more available information for prediction, thereby improving the prediction performance.

4.6.2. Influence balance factor β

As shown in Eq. (5), the influence feedback of neighbor nodes in the user influence measurement contains two parts of information: the quantity information of second-order neighbors and topological connectivity information between second-order neighbors. The influence balance factor β determines the contribution of the two parts. To explore its impact, we conduct the following experiments. On the Aminer dataset, we set T = 1 year and Δt = 1 year, and on the Sina Weibo and the Twitter datasets, we set T = 1 h and Δt = 1 h. Then, we observe the effect of changing β from 0 to 1 on the outcome. Fig. 10 shows the results.

Fig. 10 shows that when influence balance factor β is 0.5, 0.6, and 0.4 on the Aminer, Sina Weibo and Twitter datasets, respectively, the corresponding MSE and RMSPE are both at their lowest. An excessive increase or decrease in the proportion of one of the components will have a negative impact on the prediction performance of the model, especially when one of the components is completely removed, the prediction error of the model increases significantly. This shows that the quantity information of the second-order neighbors of a node and the topological connectivity among second-order neighbors are effective and almost equally important in the neighbor influence feedback on user influence measurement.

4.6.3 L2. Regularization coefficient λ

The $L2$ regularization coefficient λ often plays an important role in model training. Fig. 11 shows the results of different λ on the three datasets. If λ is too small or large, the prediction error increases. If λ is too small, although more conducive to model fitting, overfitting becomes easier. On the contrary, if λ is too large, the model is less likely to overfitting, but the deviation of the model increases significantly. On the Twitter dataset, the most appropriate regularization coefficient λ is 1×10^{-5}, which is smaller than that on the other two datasets, indicating that the data on the Twitter dataset are more sufficient and have more relationships between users, making it less likely to overfitting.

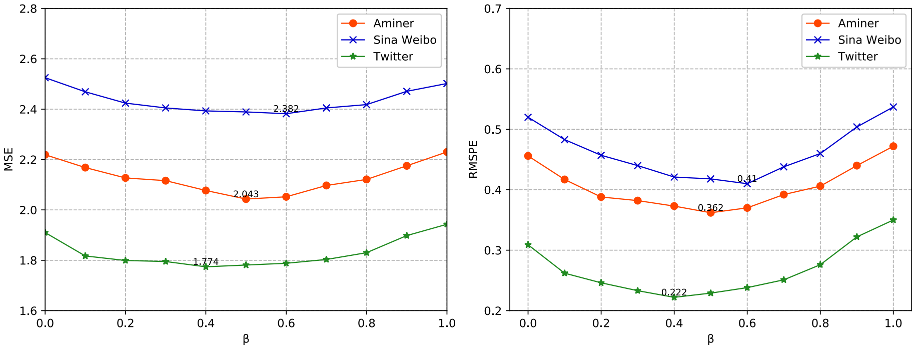

Fig. 10. Effect of parameter β.

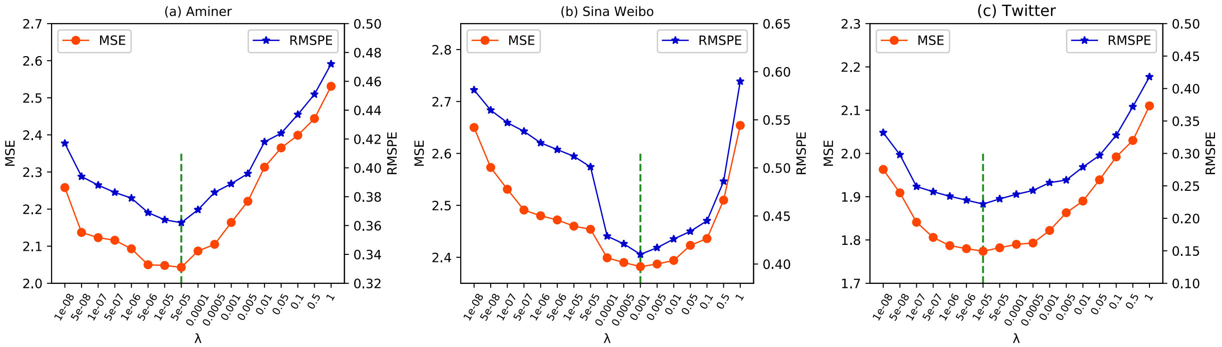

Fig. 11. Effect of $L2$ regularization coefficient λ.

Table 3
Overall prediction performance of extended experiments (Δt=4 years/h).

Dataset	Aminer		Sina Weibo		Twitter	
Metric	MSE	RMSPE	MSE	RMSPE	MSE	RMSPE
Feature_linear	4.086	0.944	4.358	0.982	3.589	0.790
DeepCas	3.834	0.883	3.929	0.920	3.294	0.660
DeepHawkes	3.642	0.852	3.824	0.863	3.125	0.593
CasCN	3.504	0.685	3.687	0.696	2.873	0.489
Coupled-GNN	3.519	0.701	3.702	0.715	2.880	0.492
CasHAN	**3.453**	**0.608**	**3.585**	**0.648**	**2.794**	**0.370**
CasHAN-influ	3.476	0.621	3.629	0.661	2.844	0.383
CasHAN-redun	3.462	0.617	3.614	0.658	2.820	0.378

4.6.4. Prediction time interval Δt.

To validate the prediction performance of our model more sufficiently, we extend the experiments on the prediction time interval Δt. In Section 4.4, we set different prediction intervals (1, 2, 3 years/hours) for a uniform observation time length T (1 year/hour). That is, we use 50 %, 33 %, and 25 % of the spread of a predicted cascade to make predictions for the remaining 50 %, 67 %, and 75 %, respectively. Following the principle of attempting to predict the growth of a cascade at an early stage, we consider further increasing the prediction interval Δt.

Specifically, the observation time length T is still set to 1 year/hour. The prediction time interval Δt is increased to 4 years/hours, which corresponds to training 20 % of the range of the known cascade to predict the remaining 80 % of the spread. Table 3 presents that. when the prediction time interval Δt increases more, the percentage of the range of a known cascade becomes smaller, and the prediction performance of all models decreases compared with that shown in Fig. 8 and Fig. 9. However, the prediction errors of CasHAN and its variants are consistently lower than those of other competitive baselines, with at least a 4 % reduction in the MSE and 3 % reduction in the RMSPE, which further demonstrates the superiority of the prediction performance of our model.

Additionally, we tried to further reduce the range of the training data by setting Δt to 9 h. It is worth stating that Δt could not be set to 9 years on the Aminer dataset because of the limitations of the available data. Although the prediction errors of our model can still be lower than those of the other baselines, we present the results and analysis of this part in the Appendix because of the poor performance of all models.

5. Conclusion

This paper investigated the problem of cascade prediction on online social networking platforms. In order to make the representations of cascade paths and graphs more reasonable and effective, and thereby improve the prediction accuracy, we considered the effects of user influence and community redundancy in cascade paths, and proposed an innovative hierarchical attention neural network model with two-layer attention mechanisms, named CasHAN. One attention mechanism focuses on the node level to obtain representations of cascade sequences based on user influence, and the other focuses on the sequence level based on community redundancy to aggregate sequence representations to obtain a graph representation. Experiments conducted on three scenarios validate that CasHAN significantly improves the prediction accuracy and outperforms state-of-the-art deep learning methods for cascade prediction.

In future research, we plan to consider additional factors that affect cascades, such as temporal and content features. In addition, future research can examine methods for handling the ever-changing dynamic network structure.

CRediT authorship contribution statement

Chu Zhong: Conceptualization, Methodology, Software, Visualization, Writing – original draft. **Fei Xiong:** Investigation, Conceptualization, Data curation, Writing – original draft. **Shirui Pan:** Conceptualization, Data curation. **Liang Wang:** Validation, Supervision. **Xi Xiong:** Validation, Writing – review & editing.

Data availability

Data will be made available on request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been supported by the Fundamental Research Funds for the Central Universities under Grant 2022JBMC005, the National Natural Science Foundation of China under Grant 61872033, the Beijing Nova Program from Beijing Municipal Science & Technology Commission under Grant Z201100006820015, and the National Key R&D Program of China under Grant 2018YFC0832304.

Appendix

When the prediction time interval Δt is set to 4 years/hours, this corresponds to training 20 % of the range of the known cascade to predict the unknown 80 % of the growth. When Δt is set to 9 h, only 10 % of the range of the known cascade is used to predict the remaining 90 % of the spread. It is worth stating that Δt could not be set to 9 years on the Aminer dataset because of the limitations of the available data. Table A and Table B show the experimental results on the MSE and RMSPE metrics, respectively. Compared with Fig. 8 and Fig. 9, the percentage of the range of a known cascade becomes smaller and the prediction performance of all models decreases as Δt increases. In particular, the prediction errors increase significantly when the percentage of available data is only 10 %. However, as shown in Table A and Table B, the prediction errors of CasHAN, and its variants are consistently lower than those of other competitive baselines, with at least a 4 % reduction in the MSE and 3 % reduction in the RMSPE, which again demonstrates the superiority of the prediction performance of our model.

Table A
Overall prediction performance of extended experiments on the MSE.

Dataset	Aminer	Sina Weibo		Twitter	
Δt	4 years	4 h	9 h	4 h	9 h
Feature_linear	4.086	4.358	6.121	3.589	5.018
DeepCas	3.834	3.929	5.437	3.294	4.688
DeepHawkes	3.642	3.824	5.365	3.125	4.501
CasCN	3.504	3.687	5.281	2.873	4.385
Coupled-GNN	3.519	3.702	5.279	2.880	4.360
CasHAN	**3.453**	**3.585**	**5.217**	**2.794**	**4.263**
CasHAN-influ	3.476	3.629	5.235	2.844	4.305
CasHAN-redun	3.462	3.614	5.226	2.820	4.287

Table B
Overall prediction performance of extended experiments on the RMSPE.

Dataset	Aminer	Sina Weibo		Twitter	
Δt	4 years	4 h	9 h	4 h	9 h
Feature_linear	0.944	0.982	1.013	0.790	0.923
DeepCas	0.883	0.920	0.998	0.660	0.804
DeepHawkes	0.852	0.863	0.925	0.593	0.775
CasCN	0.685	0.696	0.868	0.489	0.603
Coupled-GNN	0.701	0.715	0.872	0.492	0.621
CasHAN	**0.608**	**0.648**	**0.799**	**0.370**	**0.513**
CasHAN-influ	0.621	0.661	0.824	0.383	0.522
CasHAN-redun	0.617	0.658	0.813	0.378	0.527

References

[1] M. Alweshah, S.A. Khalaileh, B.B. Gupta, A. Almomani, A.I. Hammouri, M.A. Al-Betar, The monarch butterfly optimization algorithm for solving feature selection problems, Neural Comput. Appl. (2020), https://doi.org/10.1007/s00521-020-05210-0.

[2] A. Bielski, T. Trzcinski, Understanding multimodal popularity prediction of social media videos with self-attention, IEEE Access 6 (2018) 74277–74287, https://doi.org/10.1109/ACCESS.2018.2884831.

[3] H.A. Bouarara, Recurrent neural network (RNN) to analyse mental behaviour in social media, Int. J. Softw. Sci. Comput. Intell. 13 (2021) 1–11, https://doi.org/10.4018/IJSSCI.2021070101.

[4] Q. Cao, H. Shen, K. Cen, W. Ouyang, X. 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. Presented at the CIKM '17: ACM Conference on Information and Knowledge Management, ACM, Singapore Singapore, (2017) pp. 1149–1158. doi: 10.1145/3132847.3132973.

[5] Q. Cao, H. Shen, J. Gao, B. Wei, X. Cheng, Popularity prediction on social platforms with coupled graph neural networks, in: Proceedings of the 13th International Conference on Web Search and Data Mining. Presented at the WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, ACM, Houston TX USA, (2020) pp. 70–78. doi: 10.1145/3336191.3371834.

[6] S. Carta, A.S. Podda, D.R. Recupero, R. Saia, G. Usai, Popularity prediction of instagram posts, Information 11 (2020) 453, https://doi.org/10.3390/info11090453.
[7] X. Chen, F. Zhou, K. Zhang, G. Trajcevski, T. Zhong, F. Zhang, Information diffusion prediction via recurrent cascades convolution, in: 2019 IEEE 35th International Conference on Data Engineering (ICDE). Presented at the 2019 IEEE 35th International Conference on Data Engineering (ICDE), IEEE, Macao, Macao (2019), pp. 770–781. doi: 10.1109/ICDE.2019.00074.
[8] X. Chen, X. Zhou, J. Chan, L. Chen, T. Sellis, Y. Zhang, Event popularity prediction using influential hashtags from social media, IEEE Trans. Knowl. Data Eng. 34 (2022) 4797–4811, https://doi.org/10.1109/TKDE.2020.3048428.
[9] J. Cheng, L.A. Adamic, P.A. Dow, J. Kleinberg, J. Leskovec, Can cascades be predicted? In: Proceedings of the 23rd international conference on World wide web - WWW '14 (2014) 925–936. doi: 10.1145/2566486.2567997.
[10] X. Feng, Prediction of information cascades via content and structure proximity preserved graph level embedding, Inf. Sci. 560 (2021) 424–440, https://doi.org/10.1016/j.ins.2020.12.074.
[11] M. Hammad, M.H. Alkinani, B.B. Gupta, A.A. Abd El-Latif, Myocardial Infarction detection based on deep neural network on imbalanced data, Multimedia Syst. (2021), https://doi.org/10.1007/s00530-020-00728-8.
[12] N.O. Hodas, K. Lerman, The simple rules of social contagion, Sci. Rep. 4 (2015) 4343, https://doi.org/10.1038/srep04343.
[13] G. Jeh, J. Widom, SimRank: A measure of structural-context similarity, in: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Presented at the KDD '02. New York, NY, USA (2002), 538–543. doi: 10.1145/775047.775126.
[14] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks. CoRR (2017), abs/1609.02907.
[15] Q. Kong, M.-A. Rizoiu, L. Xie, Modeling Information Cascades with Self-exciting Processes via Generalized Epidemic Models, in: Proceedings of the 13th International Conference on Web Search and Data Mining. Presented at the WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, ACM, Houston TX USA (2020), pp. 286–294. doi: 10.1145/3336191.3371821.
[16] C. Li, J. Ma, X. Guo, Q. Mei, DeepCas: an end-to-end predictor of information cascades, in: Proceedings of the 26th International Conference on World Wide Web. Presented at the WWW '17: 26th International World Wide Web Conference, International World Wide Web Conferences Steering Committee, Perth Australia (2017), pp. 577–586. doi: 10.1145/3038912.3052643.
[17] Q. Li, B. Hu, W. Xu, Y. Xiao, A group behavior prediction model based on sparse representation and complex message interactions, Inf. Sci. 601 (2022) 224–241, https://doi.org/10.1016/j.ins.2022.04.023.
[18] Q. Li, Z. Wu, L. Yi, K.S. N., H. Qu, X. Ma, WeSeer: visual analysis for better information cascade prediction of WeChat articles. IEEE Trans. Visual. Comput. Graphics 26 (2020) 1399–1412. .
[19] D. Liao, J. Xu, G. Li, W. Huang, W. Liu, J. Li, Popularity prediction on online articles with deep fusion of temporal process and content features, AAAI 33 (2019) 200–207, https://doi.org/10.1609/aaai.v33i01.3301200.
[20] L.F. Lin, Y.M. Li, An efficient approach to identify social disseminators for timely information diffusion, Inf. Sci. 544 (2021) 78–96, https://doi.org/10.1016/j.ins.2020.07.040.
[21] Y. Liu, Z. Bao, Z. Zhang, D. Tang, F. Xiong, Information cascades prediction with attention neural network, HCIS 10 (2020) 13, https://doi.org/10.1186/s13673-020-00218-w.
[22] X. Lu, B.K. Szymanski, Scalable prediction of global online media news virality, IEEE Trans. Comput. Soc. Syst. 5 (2018) 858–870, https://doi.org/10.1109/TCSS.2018.2857479.
[23] Y. Lu, L. Yu, T. Zhang, C. Zang, P. Cui, C. Song, W. Zhu, Collective human behavior in cascading system: discovery, modeling and applications, in: 2018 IEEE International Conference on Data Mining (ICDM). Presented at the 2018 IEEE International Conference on Data Mining (ICDM), IEEE, Singapore, 2018, pp. 297–306, https://doi.org/10.1109/ICDM.2018.00045.
[24] F. Morone, H.A. Makse, Influence maximization in complex networks through optimal percolation, Nature 524 (65–68) (2015) 2015, https://doi.org/10.1038/nature14604.
[25] K. Purba, D. Asirvatham, R. Murugesan, Instagram post popularity trend analysis and prediction using hashtag, image assessment, and user history features. Int. Arab J. Inf. Technol., 18(10) (2021). doi: 10.34028/iajit/18/1/10.
[26] P. Rodriguez, D. Velazquez, G. Cucurull, J.M. Gonfaus, F.X. Roca, J. Gonzalez, Pay attention to the activations: a modular attention mechanism for fine-grained image recognition, IEEE Trans. Multimedia 22 (2020) 502–514, https://doi.org/10.1109/TMM.2019.2928494.
[27] M. Rosvall, C. Bergstrom, Maps of random walks on complex networks reveal community structure. Proc. Natl. Acad. Sci. U. S. A. (2008).
[28] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (1997) 2673–2681, https://doi.org/10.1109/78.650093.
[29] J. Shang, S. Huang, D. Zhang, Z. Peng, D. Liu, Y. Li, L. Xu, RNe2Vec: information diffusion popularity prediction based on repost network embedding, Computing 103 (2021) 271–289, https://doi.org/10.1007/s00607-020-00858-x.
[30] H. Shen, D. Wang, C. Song, A. Barabási, Modeling and predicting popularity dynamics via reinforced poisson processes, in: Proc of the 28th AAAI Conference on Artificial Intelligence. AAAI (2014). arXiv:1401.0778 [physics].
[31] S. Sreenivasan, K.S. Chan, A. Swami, G. Korniss, B.K. Szymanski, Information cascades in feed-based networks of users with limited attention, IEEE Trans. Netw. Sci. Eng. 4 (2017) 120–128, https://doi.org/10.1109/TNSE.2016.2625807.
[32] A.M. Srivastava, P.A. Rotte, A. Jain, S. Prakash, Handling data scarcity through data augmentation in training of deep neural networks for 3D data processing, Int. J. Semant. Web Inf. Syst. (IJSWIS) 18 (1) (2022) 1–16, https://doi.org/10.4018/IJSWIS.297038.
[33] S. Tang, Q. Li, X. Ma, C. Gao, D. Wang, Y. Jiang, Q. Ma, A. Zhang, H. Chen, Knowledge-based temporal fusion network for interpretable online video popularity prediction, in: Proceedings of the ACM Web Conference 2022. Presented at the WWW'22: The ACM Web Conference 2022, ACM, Virtual Event, Lyon France, 2022, pp. 2879–2887, https://doi.org/10.1145/3485447.3511934.
[34] X. Tang, D. Liao, W. Huang, J. Xu, L. Zhu, M. Shen, Fully exploiting cascade graphs for real-time forwarding prediction, Proc. AAAI Conf. Artif. Intell. 35 (1) (2021) 582–590. https://ojs.aaai.org/index.php/AAAI/article/view/16137.
[35] X. Tian, L. Qiu, J. Zhang, User behavior prediction via heterogeneous information in social networks, Inf. Sci. 581 (2021) 637–654, https://doi.org/10.1016/j.ins.2021.10.018.
[36] H.T. Tu, T.T. Phan, K.P. Nguyen, Modeling information diffusion in social networks with ordinary linear differential equations, Inf. Sci. 593 (2022) 614–636, https://doi.org/10.1016/j.ins.2022.01.063.
[37] K. Wang, P. Wang, X. Chen, et al, A feature generalization framework for social media popularity prediction, MM '20: The 28th ACM International Conference on Multimedia, ACM, 2020.
[38] Y. Wang, J. Wang, H. Wang, R. Zhang, M. Li, Users' mobility enhances information diffusion in online social networks, Inf. Sci. 546 (2021) 329–348, https://doi.org/10.1016/j.ins.2020.07.061.
[39] Q. Wu, Y. Gao, X. Gao, P. Weng, G. 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. Presented at the KDD '19: The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM, Anchorage AK USA (2019) pp. 447–457. doi: 10.1145/3292500.3330959.
[40] C. Xiao, C. Liu, Y. Ma, Z. Li, X. Luo, Time sensitivity-based popularity prediction for online promotion on Twitter, Inf. Sci. 525 (2020) 82–92, https://doi.org/10.1016/j.ins.2020.03.056.
[41] K. Xu, L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R.S. Zemel, Y. Bengio, Show, attend and tell: neural image caption generation with visual attention, Comput. Sci. (2015) 2048–2057.
[42] K. Xu, Z. Lin, J. Zhao, P. Shi, W. Deng, H. Wang, Multimodal deep learning for social media popularity prediction with attention mechanism, in: Proceedings of the 28th ACM International Conference on Multimedia. Presented at the MM '20: The 28th ACM International Conference on Multimedia, ACM, Seattle WA USA (2020), pp. 4580–4584. doi: 10.1145/3394171.3416274.

[43] F. Yang, Y. Qiao, Y. Qi, J. Bo, X. Wang, BMP: A blockchain assisted meme prediction method through exploring contextual factors from social networks, Inf. Sci. 603 (2022) 262–288, https://doi.org/10.1016/j.ins.2022.04.039.

[44] W. Zhang, W. Wang, J. Wang, H. Zha, User-guided hierarchical attention network for multi-modal social image popularity prediction, in: Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18. Presented at the 2018 World Wide Web Conference, ACM Press, Lyon, France (2018) pp. 1277–1286. doi: 10.1145/3178876.3186026.

[45] Y. Zhang, J. Liu, B. Guo, Z. Wang, Y. Liang, Z. Yu, App popularity prediction by incorporating time-varying hierarchical interactions, IEEE Trans. Mob. Comput. 21 (2022) 14.

[46] Q. Zhao, M.A. Erdogdu, H.Y. He, A. Rajaraman, J. 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, 2015, pp. 1513–1522, https://doi.org/10.1145/2783258.2783401.

[47] J. Zheng, Z. Qin, S. Wang, D. Li, Attention-based explainable friend link prediction with heterogeneous context information, Inf. Sci. 597 (2022) 211–229, https://doi.org/10.1016/j.ins.2022.03.010.

[48] F. Zhou, X. Xu, G. Trajcevski, K. Zhang, A survey of information cascade analysis: models, predictions, and recent advances, ACM Comput. Surv. 54 (2022) 1–36, https://doi.org/10.1145/3433000.

[49] L. Zhou, J. Li, Z. Gu, J. Qiu, B.B. Gupta, Z. Tian, PANNER: POS-aware nested named entity recognition through heterogeneous graph neural network, IEEE Trans. Comput. Soc. Syst. 1–9 (2022), https://doi.org/10.1109/TCSS.2022.3159366.