Knowledge-Based Systems 254 (2022) 109659

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

HeDAN: Heterogeneous diffusion attention network for popularity prediction of online content

Xueqi Jia [a,b], Jiaxing Shang [a,b,*], Dajiang Liu [a,b], Haidong Zhang [c,d], Wancheng Ni [c,d,**]

[a] College of Computer Science, Chongqing University, Chongqing 400044, China
[b] Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University, Chongqing 400044, China
[c] Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[d] University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Received 29 March 2022
Received in revised form 5 August 2022
Accepted 6 August 2022
Available online 12 August 2022

Keywords:
Information popularity prediction
Graph neural network
Hierarchical attention
Social network analysis
Predictive factors

ABSTRACT

Popularity prediction of online content over social media platforms is a valuable and challenging issue, the core of which lies in how to capture predictive factors from available data. However, existing studies either treat each cascade independently, which neglects the correlation among different cascades, or lack a comprehensive consideration of user behavioral proximity and preference with respect to different messages. Motivated by the above observation, this article proposes a graph neural network-based framework named HeDAN (heterogeneous diffusion attention network), which comprehensively considers various factors affecting information diffusion to provide more accurate prediction results. Specifically, we first construct a heterogeneous diffusion graph with two types of nodes (*user* and *message*) and three types of relations (*friendship*, *interaction*, and *interest*). Among them, *friendship* reflects the strength of social relationships between users, *interaction* reflects the behavioral proximity between users, and *interest* reflects user preference for information. Next, a graph neural network model with a hierarchical attention mechanism is proposed to learn from these relations. Specifically, at the node level, we utilize the graph attention network to learn the subgraph structure and generate the representations of users and messages under each specific relationship. At the semantic level, we distinguish the importance of different nodes in different relations via the multi-head self-attention mechanism and fuse them into the final prediction representation. Extensive experimental results on three real diffusion datasets show the superior performance of HeDAN over the state-of-the-art baselines.

1. Introduction

Currently, social media platforms have greatly promoted the generation and dissemination of online content, as a result, they have intensified the competition among different messages for users' attention. Among the hot topics related to social media analysis and mining, online content popularity prediction is a key issue. From the perspective of marketers, predicting the popularity of online content can help them discover potential best-selling products in advance and take proactive actions in their decision-making, which drives downstream applications such as viral marketing [1,2] and social recommendation [3]. From the perspective of platform managers, predicting the popularity of information can help them anticipate and limit the spread of illegal or harmful content, which supports downstream applications such as fake news detection [4–7] and rumor blocking [8,9]. Therefore, the theoretical and practical values of information popularity prediction have been widely recognized by both academia and industry. However, since real-world social media platforms are highly open and user behaviors exhibit strong uncertainty and variability, accurately predicting the popularity of online content has become a challenging issue.

Formally speaking, the problem of information popularity prediction refers to predicting the future popularity of a message given the early cascade information within a specific observation time, and the information popularity prediction problem over social media platforms refers to further considering the effect of social relations on information diffusion and regarding all users on social media platforms as the diffusion domain. In recent years, a great amount of effort has been devoted to the information popularity prediction problem, which can be specifically divided into three groups of methods: (1) *feature engineering-based approaches* [10–13], (2) *generative models* [14–17], and (3) *deep*

* Corresponding author at: College of Computer Science, Chongqing University, Chongqing 400044, China.
** Corresponding author at: Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.
E-mail addresses: shangjx@cqu.edu.cn (J. Shang), wancheng.ni@ia.ac.cn (W. Ni).

https://doi.org/10.1016/j.knosys.2022.109659
0950-7051/

learning-based methods. Deep learning-based methods can be further divided into *sequential representation-based methods* [18–21] and *graph representation-based methods* [22–25].

Early deep learning methods for popularity prediction were mainly based on sequence modeling, which modeled information diffusion as a sequence of forwarding behaviors in temporal order. However, these methods were deficient in capturing structural information from the diffusion paths. Therefore, more information popularity prediction methods [22,26,27] based on graph representation learning have emerged in recent years. For example, Chen et al. [26] proposed a recurrent cascade convolutional network (CasCN) which modeled the cascade graph as a series of sequential sub-cascades and adopted a dynamic multi-directional graph convolutional network to learn the structural information of sub-cascades. Such methods [22,26] mainly focus on the local graph of participating users without considering the underlying social networks. However, some studies [28,29] have pointed out that users' information from diffusion behavior and social networks shows a strong correlation. Therefore, considering the influence of social networks would be highly beneficial when modeling information diffusion. Taking the social structure into consideration, Cao et al. [30] applied two graph neural networks to model the evolution of a cascade on the global social network, thus effectively capturing the cascade effect in propagation, but this method only focused on one cascade sample at a time and could not directly exploit the correlation among samples. Therefore, the simultaneous consideration of all cascade samples can help to learn the interaction intimacy between users from their historical forwarding behaviors, which is helpful for information diffusion modeling. Among the methods [27,31] that co-process all cascades, Feng et al. [27] proposed two higher-order graphs with cascades as nodes, which were constructed based on the similarity between cascades, and learned the higher-order graphs by random walks and semi-supervised language models so that cascades with similar structure and content had closer representations. The idea of directly establishing the relationship between messages in that method is worth considering. Furthermore, messages attracting the same group of users are more likely to have similar popularity in the future, which means that establishing the direct links between messages and users can reflect the users' preferences for different messages, thereby helping to predict the popularity of messages. However, all the above studies were relatively scattered since they only considered partial factors affecting information diffusion, while a comprehensive consideration is still lacking. Therefore, this paper aims to comprehensively consider the role of social influence, interaction intimacy among users, and user preference for messages on information diffusion to effectively capture predictive factors for more accurate information popularity prediction.

To this end, we propose a graph neural network-based framework named **HeDAN** (**He**terogeneous **D**iffusion **A**ttention **N**etwork), which utilizes a hierarchical attention mechanism to directly learn representations for both users and messages to provide more accurate popularity prediction. Specifically, we first construct a heterogeneous diffusion graph with two types of nodes (*user* and *message*) and three types of relations (*friendship*, *interaction*, and *interest*). Among them, *friendship* refers to the underlying follower–followee relationships among users on social media platforms, which reflects the influence of users themselves from the perspective of social friendship. *Interaction* refers to the historical forwarding behaviors between users, which reflects the proximity among users from the perspective of user behaviors. *Interest* refers to the direct interactions between messages and users, which reflects the attractiveness of messages to users from the perspective of user preference. We combine the above three types of relations to form a heterogeneous diffusion graph. Next, we propose a graph neural network model with a hierarchical attention mechanism to learn from this heterogeneous diffusion graph. Specifically, at the node level, we leverage graph attention layers to learn the structure of relational subgraphs and characterize the mutual importance of nodes in terms of relations. Then, at the semantic level, we utilize a multi-head self-attention mechanism to distinguish the influence of different relationships and users on information diffusion, and finally fuse various influences to obtain the final representation for popularity prediction.

We validated our proposed method on three real-world datasets and compared it with state-of-the-art methods. Experimental results demonstrated the superior performance of our proposed model over the state-of-the-art. Our main contributions and advantages are as follows:

(1) We consider the correlation among different cascades and creatively construct a heterogeneous diffusion graph that contains friendship relationships, interaction relationships and interest relationships between users and messages to make information diffusion modeling more comprehensive.
(2) We propose a graph neural network model with a hierarchical attention mechanism to learn from the heterogeneous diffusion graph, where the node-level attention mechanism learns the graph structure under each relation, while the semantic-level attention mechanism learns the effect of different relations for more accurate popularity prediction.
(3) Experimental results on the information popularity prediction task over three realistic datasets demonstrate the effectiveness of our proposed model, where the overall prediction errors are relatively reduced by 10% on the Weibo dataset. Further dimensionality reduction and visualization experiments show the potential interpretability of the proposed model.

The rest of this paper is organized as follows: Section 2 reviews the related research on information diffusion prediction and graph representation. Section 3 gives the definitions of the heterogeneous diffusion graph and the popularity prediction problem. In Section 4, we illustrate our proposed model, and in Section 5, we introduce the experimental evaluation framework. The experimental results are presented in Section 6. Finally, we conclude the paper and discuss some future work in Section 7.

2. Related work

Many scholars have devoted significant research efforts to the problem of information diffusion prediction, which can be divided into macro-level (cascade level) tasks and micro-level (user level) tasks on the basis of prediction targets. The development of those methods thus far can be divided into three categories: (1) *Feature engineering-based approaches* mostly focus on identifying and designing complicated hand-crafted features, such as structural features [11,32], content features [10,33], temporal features [5,12], etc. However, the performances of these methods largely rely on the hand-crafted features which require extensive domain knowledge, making them difficult to generalize to new domains. (2) *Generative approaches* [14,15, 34,35] regard the change in information popularity over time as a dynamic time series model and develop a macroscopic distribution or random process to fit the model based on various strong assumptions. However, these methods usually provide less desirable predictive performance due to the strong model assumption. (3) *Deep learning-based methods* utilize various deep learning models to capture predictive factors from the input data. With the success of deep learning, the latest studies generally apply end-to-end deep representation learning models to automatically learn information diffusion representations for

prediction. Due to the time series and graph structural characteristics of cascaded data, deep learning-based methods mostly focus on sequential representation-based methods and graph representation-based methods. Therefore, we will first review related works focusing on sequential representation-based methods and graph representation-based methods. Furthermore, the information diffusion prediction task over social platforms involves the techniques of heterogeneous graph representation learning and dynamic graph representation learning, so we will further discuss the state-of-the-art related works.

2.1. Sequential representation-based methods

Sequential representation-based methods usually regard information cascades as dynamic time series and apply recurrent neural network (RNN) and its variants [36,37] to learn and model the diffusion process.

For micro-level studies, CYAN-RNN [38] developed an attention RNN to model the interdependence in cascade graphs to fill the gap between cascade structure and traditional sequence modeling. Topo-LSTM [18] extended the standard LSTM model from the topology perspective and learned the information diffusion path to generate a topology-aware node embedding. DAN [39] utilized a diffusion attention module to capture the implicit diffusion dependencies among users from cascades and generated user embeddings and further combined the time decay effect to fuse user embeddings in the cascade. DeepDiffuse [20] utilized representation learning and attention mechanisms to learn from the infection timestamps and generated predictions about when and who will be infected in the next step based on observed cascade data in the past. SNIDSA [19] utilized structure attention to explore the structural characteristics of users' connection graph and added the information to the sequential information diffusion model. To alleviate the long-term dependency issue of sequential models, NDM [21] developed a microscope cascade prediction model based on self-attention and convolutional neural networks. On the basis of traditional RNN, SIDDA [40] incorporated the idea of disentangled representation learning and employed a sequential attention module and a disentangled attention module to better aggregate the history information and disentangle the latent factors to achieve a better prediction performance.

For macro-level studies, DeepCas [41] and DeepHawkes [34] were first proposed to employ a recurrent neural network (RNN) for encoding cascade sequences into feature vectors instead of hand-crafted features. Specifically, DeepCas [41] utilized random walks to sample the cascade graphs to obtain node sequences as the input of the bidirectional gated recurrent unit (Bi-GRU). DeepHawkes [34] merged three crucial concepts of the Hawkes process, i.e., user influence, self-exciting mechanism, and time decay effect, with RNN to make the modeling process more interpretable. FOREST [42] utilized reinforcement learning to incorporate the information of macroscopic diffusion size into the RNN-based microscopic diffusion model by addressing the non-differentiable problem.

Most sequential representation-based methods aim to model how historical diffusion sequences affect future diffusion trends. However, those methods ignored the graph structural information of cascades, leading to insufficient modeling of information diffusion. Moreover, sequential representation-based methods cannot effectively utilize the underlying social network information.

2.2. Graph representation-based methods

With the development of graph neural networks (GNNs) [43], graph representation-based information diffusion studies [22,26,30,44,45] have received increasing attention in recent years.

For micro-level studies, DeepInf [22] evaluated the social influence of the central user by predicting the user's state (active or inactive) based on the given r-ego network and neighbors' states. Specifically, the r-ego network was generated by DeepWalk [46] and the ego-network structure was extracted by a graph convolutional neural network (GCN) [47] or graph attention network (GAT) [48]. DiffuseGNN [44] improved the DeepInf model and applied it to the WeChat platform for "Wow" behavior prediction. It applied BFS to generate an r-ego network, propagated features in a trainable modulated spectral domain to filter noise and expanded the hierarchical graph representation model to learn subgraph representations by clustering nodes. Both of the above models learned the graph representations on the ego network of the target node, while DyHGCN [45] learned the graph representations on the global network and then extracted the representation vector of the target node and inserted it into the time-aware attention layer. It captured changes in user preferences by modeling the evolution of the dynamic diffusion global graph, thereby predicting user behavior more accurately. HDD [23] exploited metapaths in the diffusion graph and learned heterogeneous network representations by GNN.

For macro-level studies, CasCN [26] sampled a cascade graph as a series of sequential sub-cascades and adopted a dynamic multi-directional GCN to learn the structural information of cascades. DMT-LIC [49] proposed a deep multi-task learning framework with a shared-representation layer, which applied a multi-layer graph attention network with multi-head attention to model the cascade graph and employed a bidirectional LSTM to sequentially model the diffusion process. DeepCon+Str [27] designed a framework to learn the low-dimensional representation of each cascade graph by constructing the content and structural proximity-based high-order graph where each node refers to each cascade. VaCas [25] utilized a variational encoder to simulate the uncertainty of user behavior based on the graph representation method for learning cascade graphs. CoupledGNN [30] leveraged two specifically designed GNNs, one for node states and the other for influence spread, to model the cascading effect.

However, these graph representation-based methods mainly focus on the current diffusion sequence and ignore the diffusion dynamics of other messages at the same time, which cannot capture global forwarding relations. Both social relations and forwarding interactions have important impacts on the modeling of the diffusion process. In contrast to the partial or limited consideration of influencing factors in the above studies, we aim to comprehensively consider various factors that affect the diffusion process to model information diffusion more accurately.

2.3. Heterogeneous graph representation learning

The representation learning for heterogeneous graphs aims to learn the multiple interactions between different types of objects to capture their impact on the downstream tasks.

The R-GCN [50] splits a heterogeneous graph into multiple subgraphs by building an independent adjacency matrix for each type of edge and utilizes a GCN to learn the structure of each subgraph. HAN [51] extended GAT [48] to heterogeneous networks through a metapath-based neighbor discovery strategy and hierarchical attention mechanism. These studies [50,51] disassembled heterogeneous graphs into multiple homogeneous graphs and bipartite graphs based on meta-paths or other methods. For highly multi-relational data, due to the high dependence and cost of constructing metapaths, RSHN [52] utilized the line graph to convert

the edges in the original graph into nodes in the new graph, so that the co-occurrence probability of homogeneous relationships could be learned from the structure of the line graph, and the implicit relationship between the relationships could be learned automatically without manually defining the metapath. However, some issues would be raised in real application scenarios. For example, the line graph generated by a graph containing users with large in-and-out degrees usually becomes a complete graph containing a large number of nodes, which may result in erroneous information after operations such as random walks. For faster convergence and better scalability, FAME [53] efficiently mapped different information (network topology, various node features, and relationships) into a latent space. It first decomposes the multiplex heterogeneous network into homogeneous and bipartite subnetworks, and then utilizes the spectral transformation module to automatically aggregate the semantic-level decoupled subnetworks. However, this method expressed the importance of the corresponding subnetwork as the weighting coefficient of the adjacency matrix in the spectral graph transformation, which cannot model the connections between different subnetworks. HM2 [54] implemented the relative valuation of firms using heterogeneous multimodal graph neural networks. It is worth noting that this method implements adaptive attention aggregation multimodal embedding to handle unknown situations, but it ignores the graph structural information.

In contrast, based on the characteristics of information diffusion over social platforms, our method proposes a hierarchical framework to obtain the heterogeneous diffusion graph representation, which focuses on the graph structure and the associations between users with graph attention layers at the node level, as well as the integration of multiple users and multiple relationships with a self-attention mechanism at the semantic level.

2.4. Dynamic graph representation learning

Since information diffusion over social media platforms evolves over time and has varying dynamics, the latest dynamic graph representation learning methods are further discussed.

Dyngraph2vec [55] utilized a deep architecture composed of dense and recurrent layers to learn the node embeddings of the dynamic graph and visualized community shifts in network evolution to verify its effectiveness in capturing the dynamic evolution of the network from the temporal dimension. However, this method ignored the utilization of graph structures, which is insufficient for understanding the relationship between users of information diffusion. Furthermore, the heavy utilization of recurrent layers will impose a high computational overhead when learning long-term sequence graph representations. MTSN [56] captured temporal dynamics through a temporal shift operation in a dynamic attribute network, where the temporal shift operation simulated a one-dimensional convolution operation over the representations across different historical time snapshots along with the time dimension. Specifically, it first utilized the shift operation to continuously update the snapshot-specific node embeddings in the time window, then integrated multiple snapshots by the multiplication-accumulate operation, and finally generated the fused node representations within the time window. This method captured the dynamic changes of the network, but it is worth noting that the weight of the multiplication-accumulate operation in one-dimensional convolution was fixed for each position, which could not adaptively handle the differences in network dynamics, since information diffusion in the real world usually exhibits high heterogeneity. To parallelize the processing of graph sequences and adapt to unknown node attributes and graph structures, Co-EvoGNN [57] proposed an S-stack temporal self-attention architecture to adaptively learn the changes of dynamic graphs. After learning the structural information through the static GNN method, it mapped each snapshot at each moment to a specific matrix to synthesize multiple snapshots to obtain the final network representation. The mapping of snapshots at each moment can be performed in parallel to reduce the time complexity. However, their method performed fusion of relations only through simple addition, while our proposed framework utilizes multi-head self-attention to fuse nodes with different representations and relations in a more targeted manner.

In summary, compared with the existing dynamic graph representation learning-based methods, the self-attention mechanism in our method is more efficient and general in handling the representation of each node in information propagation.

3. Preliminaries

3.1. Heterogeneous diffusion graph

To allow our graph neural network model to capture various predictive factors that affect information diffusion, we first define and construct a heterogeneous diffusion graph that contains two types of nodes (*user* and *message*) and three types of relations (*friendship*, *interaction*, and *interest*). Fig. 1 shows how to extract the corresponding relations from the cascade graphs (Fig. 1(a)) and the global social graph (Fig. 1(b)) to form a heterogeneous diffusion graph (Fig. 1(c)).

Definition 1 (*Cascade Graph [26]*)**.** Given a cascade C_i representing the diffusion paths of a specific message m_i, a cascade graph is defined as a directed graph $\mathcal{G}_{C_i} = (\mathcal{V}_{C_i}, \mathcal{E}_{C_i})$, where node set $\mathcal{V}_{C_i}$ includes all participants of cascade C_i, and $\mathcal{E}_{C_i} = \{(v_j, u_j)|j = 1, \ldots, |\mathcal{E}_{C_i}|\}$ is the set of edges where each directed edge (v_j, u_j) indicates that user u_j reposted the message from v_j.

Example. The cascade graph corresponding to C_3 in Fig. 1(a) is denoted as $\mathcal{G}_{C_3} = (\mathcal{V}_{C_3}, \mathcal{E}_{C_3})$, with node set $\mathcal{V}_{C_3} = \{u_4, u_5, u_6, u_8, u_9, u_{12}\}$, and edge set $\mathcal{E}_{C_3} = \{(u_5, u_4), (u_5, u_9), (u_5, u_6), (u_4, u_{12}), (u_6, u_8)\}$.

Definition 2 (*Global Social Graph [58]*)**.** The global social graph is defined as a directed graph $\mathcal{G}_S = (\mathcal{V}_S, \mathcal{E}_S)$, where $\mathcal{V}_S$ is the set of all user nodes, and $\mathcal{E}_S \in \mathcal{V}_S \times \mathcal{V}_S$ is the set of edges representing the underlying follower–followee relationships, where each directed edge $(v, u) \in \mathcal{E}_S$ indicates that user v follows user u.

Definition 3 (*Heterogeneous Diffusion Graph*)**.** Given the global social graph $\mathcal{G}_S$, a set of messages $\mathcal{M} = \{m_1, m_2, \ldots, m_d\}$ and the corresponding set of cascade graphs $\mathbf{G}_C = \{\mathcal{G}_{C_1}, \mathcal{G}_{C_2}, \ldots, \mathcal{G}_{C_d}\}$, the heterogeneous diffusion graph is defined as $\mathcal{G}_H = (\mathcal{V}_H, \mathcal{E}_H)$, where node set $\mathcal{V}_H = \mathcal{M} \cup \mathcal{V}_S \cup \mathcal{V}_{C_1} \cup \mathcal{V}_{C_2} \cup \cdots \cup \mathcal{V}_{C_d}$ is the set of all message nodes and user nodes. Each user node is associated with two states, active or inactive. If the user has participated in one of the messages, then it is active, otherwise it is inactive. And the edge set $\mathcal{E}_H = \mathcal{E}_{F(u)} \cup \mathcal{E}_{I(u)} \cup \mathcal{E}_{I(m)}$ contains three subsets $\mathcal{E}_{F(u)}$, $\mathcal{E}_{I(u)}$ and $\mathcal{E}_{I(m)}$. Among them, $\mathcal{E}_{F(u)} = \mathcal{E}_S$ is the *friendship* edge set, $\mathcal{E}_{I(u)} = \mathcal{E}_{C_1} \cup \mathcal{E}_{C_2} \cup \cdots \cup \mathcal{E}_{C_d}$ is the *interaction* edge set, and $\mathcal{E}_{I(m)} = \{(u_j, m_i)|u_j \in \mathcal{V}_{C_i}, m_i \in \mathcal{M}\}$ is the *interest* edge set.

Example. As shown in Fig. 1, the *interest* edge set $\mathcal{E}_{I(m)}$ corresponding to m_1 in Fig. 1(c) is $\{(u_1, m_1), (u_2, m_1), (u_3, m_1)\}$, where an edge (u, m) indicates that user u has is interested message m.

As shown in Fig. 1(b), assuming that message m_1 is a training sample and message m_2 is a test sample, it is observed that messages m_1 and m_2 are reposted by the same users u_1 and u_2,

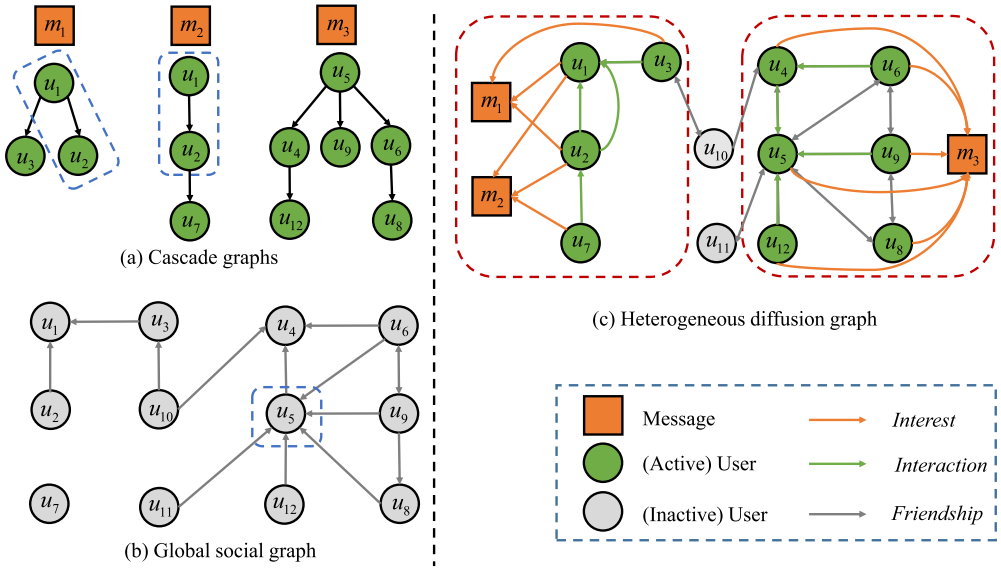


Fig. 1. An example of the heterogeneous diffusion graph. (a) An example of cascade graphs of messages m_1, m_2, and m_3 (marked as yellow squares). The edges denote that the user (marked as green circles) reposted a message from another user at a certain timestamp. (b) The global social graph consisting of follower–followee relationships between users. (c) The constructed heterogeneous diffusion graph, which includes two types of nodes (*user* and *message*) and three types of edges (*friendship*, *interaction*, and *interest*). The green circles represent active user nodes, while the gray circles represent inactive user nodes.

so it can be inferred that the above two messages have relatively similar cascading effects and will show similar popularity in the future. The heterogeneous graph we proposed (as shown in Fig. 1(c)) visually indicates that the distance between node m_1 and node m_2 is relatively close. This example demonstrates the effect of directly modeling the relationship between users and messages on prediction. In addition, since most active users with message m_3 are those with high influence on the social network, such as u_4 and u_5, message m_3 may generate a high popularity. With our proposed heterogeneous graph structure, these impacting factors related to information popularity can be effectively captured.

Through the above definitions, we construct the heterogeneous diffusion graph with three types of relationships, i.e., social friendship, user interaction, and user interest, which is beneficial to a more comprehensive modeling of information diffusion and thus yields better prediction performance.

3.2. Problem definition

The traditional popularity prediction problem aims to predict the final participants, attention, or influence of a message, given its initial diffusion dynamics [59]. Considering that any user from the underlying social network has the possibility to disseminate the message, we define all users on the social media platform as the diffusion domain. In this paper, the information popularity prediction problem is transformed into a regression problem for message nodes in the proposed heterogeneous diffusion graph, which will be trained in a semi-supervised manner.

Definition 4 (*Information Popularity Prediction Problem [60]*)**.** Given a message m_i and the corresponding cascade graph $\mathcal{G}_{C_i}$ within the observation window $(0, T_i]$, the information popularity prediction problem is formulated as a regression problem that aims at predicting the increased cascade size of message m_i, i.e., $\Delta S_i^{\infty} = |\mathcal{V}_i^{\infty}| - |\mathcal{V}_i^{T_i}|$.

4. Heterogeneous diffusion attention network framework

In this section, we present the framework of our HeDAN model, as illustrated in Fig. 2. On the whole, HeDAN consists of four major components: (a) the heterogeneous diffusion graph construction module, which extracts the *interaction* relations among users and user-message *interest* interactions from information cascades and then combines them with social relationships to construct a heterogeneous diffusion graph; (b) the node-level attention module, which utilizes graph attention networks to learn the graph structure under each specific relational subgraph, thereby generating node embeddings representing specific relationships; (c) the semantic-level attention module, which utilizes a multi-head self-attention mechanism to distinguish the importance of different types of nodes under different relationships and fuses them into the final representation vector; and (d) the prediction module, which transforms the final representation vector into the predicted popularity value via a multi-layer perceptron (MLP).

4.1. Heterogeneous diffusion graph construction module

As shown in Fig. 2(a), the heterogeneous diffusion graph has two types of nodes (each user is associated with two states) and three types of edges. For the node types, the *M* nodes (yellow nodes) represent the message nodes, the *AU* nodes (green nodes) represent the user nodes that have been activated within the observation time window, and the *U* nodes (gray nodes) represent other user nodes in the social network. For the edge types, *interest* edges (yellow edges) go from *AU* nodes to *M* nodes which means that the *AU* users are interested in information *M* within the observation time window. *Interaction* edges (green edges) go from *AU* nodes to other *AU* nodes, indicating the forwarding relationship within the observation time window. *Friendship* edges (gray edges) go from the *U* nodes to the *AU* nodes or from the *U* nodes to other *U* nodes, indicating the underlying follower–followee social relationships.

4.2. Node-level attention module

The heterogeneous diffusion graph contains three types of relationships, and the correlations between nodes are different under different relationships. The purpose of this module is to model nonlinear associations between nodes and generate the node

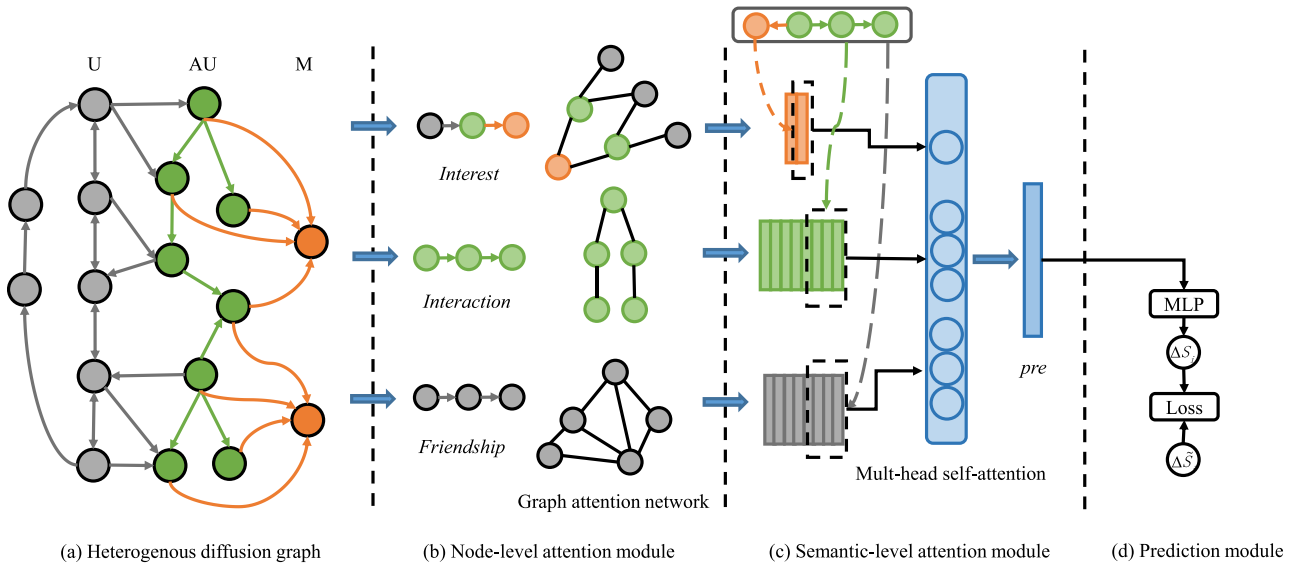


Fig. 2. The framework of HeDAN.

representations under each relation type. As shown in Fig. 2(b), this module generates three subgraphs from the original graph in line with the three types of relationships and then utilizes the graph attention layer to capture the importance between nodes and generate the representations of nodes on the subgraphs in parallel. The detailed process of this module is as follows:

4.2.1. Node feature transformation

Following the works [51,61,62] on heterogeneous graph representation learning, and considering that the feature spaces of message nodes and user nodes are different, we use transformation matrices to project both kinds of nodes into the same feature space. The projection process can be expressed as follows:

$$\mathbf{h}_i'^{(u)} = \mathbf{M}^{(u)} \cdot \mathbf{h}_i^{(u)}, \tag{1}$$

$$\mathbf{h}_j'^{(m)} = \mathbf{M}^{(m)} \cdot \mathbf{h}_j^{(m)}, \tag{2}$$

where $\mathbf{h}_i^{(u)} \in \mathbb{R}^{d_u}$ and $\mathbf{h}'^{(u)}_i \in \mathbb{R}^{d'}$ are the original and projected features of the user node i, $\mathbf{h}_j^{(m)} \in \mathbb{R}^{d_m}$ and $\mathbf{h}'^{(m)}_j \in \mathbb{R}^{d'}$ are the original and projected features of the message node j, $\mathbf{M}^{(u)} \in \mathbb{R}^{d' \times d_u}$ and $\mathbf{M}^{(m)} \in \mathbb{R}^{d' \times d_m}$ are the transformation matrices of user and message nodes respectively.

4.2.2. Friendship subgraph attention layer

We utilize the friendship subgraph attention layer to capture the friendship importance among users and obtain user representations based on friendship relations. The friendship subgraph $\mathcal{G}_{F(u)}$ is a bidirectional homogeneous subgraph generated by the edge-set $\mathcal{E}_{F(u)}$. $\mathcal{G}_{F(u)}$ is bidirectional because each social user plays the two roles of sender and receiver in information diffusion. For example, if there is a following relationship between node B and node A, edge (A, B) indicates that A is the sender and B is the receiver, while edge (B, A) indicates that A is the receiver and B is the sender. Furthermore, we adopt the graph attention layer to learn the importance $e_{ij}^{F(u)}$ on the subgraph $\mathcal{G}_{F(u)}$, which measures how sender j would contribute to receiver i on friendship. It can be formulated as follows:

$$e_{ij}^{F(u)} = \text{LeakyReLU}(\mathbf{w}_{F(u)}^T \cdot \left[\mathbf{h}_i'^{(u)} \parallel \mathbf{h}_j'^{(u)}\right]), \tag{3}$$

where $\mathbf{w}_{F(u)} \in \mathbb{R}^{2d'}$ is the parameterized attention vector for subgraph $\mathcal{G}_{F(u)}$ and '$\parallel$' denotes the concatenate operation. Therefore, edge (A, B) and edge (B, A) will correspond to different weight values, i.e., $e_{ij}^{F(u)} \neq e_{ji}^{F(u)}$.

Then we apply softmax function to obtain the normalized weight coefficients $\alpha_{ij}^{F(u)}$, which can be formulated as follows:

$$\alpha_{ij}^{F(u)} = softmax_j(e_{ij}^{F(u)}) = \frac{\exp(e_{ij}^{F(u)})}{\sum_{k \in \mathcal{G}_i^{F(u)}} \exp(e_{ik}^{F(u)})}, \tag{4}$$

where $\mathcal{G}_i^{F(u)}$ is the first-order in-degree neighborhood of user i. For users with a large number of followers, due to its large in-degree value, the influence of each follower is lower on average. For users with few followers but who are active in their own communities, the influence of their neighbors' connections is higher on average.

Finally, the embedding of node i in subgraph $\mathcal{G}_{F(u)}$ can be aggregated by the neighbors' projected features with the corresponding coefficients as follows:

$$\mathbf{z}_i^{F(u)} = \sigma\Big(\sum_{j \in \mathcal{G}_i^{F(u)}} \alpha_{ij}^{F(u)} \cdot \mathbf{h}_j'^{(u)}\Big), \tag{5}$$

where $\mathbf{z}_i^{F(u)}$ is the output of node i for subgraph $\mathcal{G}_{F(u)}$, and $\sigma(\cdot)$ is the activation function.

4.2.3. Interaction subgraph attention layer

We utilize the interaction subgraph attention layer to capture the interaction intimacy among activated users and obtain activated user representations based on interaction relations. Similar to $\mathcal{G}_{F(u)}$, we generate the interaction subgraph $\mathcal{G}_{I(u)}$ by the edge-set $\mathcal{E}_{I(u)}$, which includes the forwarding relationship among activated users. We process the generated subgraph as a directed homogeneous graph $\mathcal{G}_{I(u)}$ and employ the graph attention layer to learn interaction attention and generate interaction-based user representations on $\mathcal{G}_{I(u)}$. Similar to that in the friendship subgraph, the calculation formulas involved are as follows:

$$e_{ij}^{I(u)} = \text{LeakyReLU}(\mathbf{w}_{I(u)}^T \cdot \left[\mathbf{h}_i'^{(u)} \parallel \mathbf{h}_j'^{(u)}\right]), \tag{6}$$

$$\alpha_{ij}^{I(u)} = softmax_j(e_{ij}^{I(u)}) = \frac{\exp(e_{ij}^{I(u)})}{\sum_{k \in \mathcal{G}_i^{I(u)}} \exp(e_{ik}^{I(u)})}, \tag{7}$$

$$\mathbf{z}_i^{I(u)} = \sigma\Big(\sum_{j \in \mathcal{G}_i^{I(u)}} \alpha_{ij}^{I(u)} \cdot \mathbf{h}_j'^{(u)}\Big). \tag{8}$$

4.2.4. Interest subgraph attention layer

We utilize the interest subgraph attention layer to capture the user preferences for messages and to obtain message representations based on interest relations. When generating the interest subgraph, we consider two types of edges, i.e., the real connections between the active users and the message, and the virtual edges from other users and the message. A virtual edge means that if there is a reachable path of length 2 between an inactive user and a message, then a virtual edge is constructed where the inactive user is the source node and the message is the target node. Therefore, the interest subgraph $\mathcal{G}^{I(m)'}$ contains two directed bipartite subgraphs: one is $\mathcal{G}_{IA}$, whose edges directly connect active users to messages, and the other is $\mathcal{G}_{IB}$, whose edges connect inactive users who are 2-hop away from the corresponding messages. Furthermore, we train the graph attention network layer on $\mathcal{G}_{IA}$ and $\mathcal{G}_{IB}$, and finally obtain $\mathbf{z}_i^{I(m)}$. The formulas involved are as follows:

$$\alpha_{ij}^{IA} = \frac{\exp(\text{LeakyReLU}(\mathbf{w}_{IA}^T \left[\mathbf{h}_i'^{(m)} \parallel \mathbf{h}_j'^{(au)}\right]))}{\sum_{k\in\mathcal{G}_i^{IA}} \exp(\text{LeakyReLU}(\mathbf{w}_{IA}^T \left[\mathbf{h}_i'^{(m)} \parallel \mathbf{h}_k'^{(au)}\right]))}, \tag{9}$$

$$\alpha_{ij}^{IB} = \frac{\exp(\text{LeakyReLU}(\mathbf{w}_{IB}^T \left[\mathbf{h}_i'^{(m)} \parallel \mathbf{h}_j'^{(u)}\right]))}{\sum_{k\in\mathcal{G}_i^{IB}} \exp(\text{LeakyReLU}(\mathbf{w}_{IB}^T \left[\mathbf{h}_i'^{(m)} \parallel \mathbf{h}_k'^{(u)}\right]))}, \tag{10}$$

$$\mathbf{z}_i^{I(m)} = \sigma\Big(\sum_{j\in\mathcal{G}_i^{IA}} \alpha_{ij}^{IA} \cdot \mathbf{h}_j'^{(au)} + \sum_{k\in\mathcal{G}_i^{IB}} \alpha_{ik}^{IB} \cdot \mathbf{h}_k'^{(u)} + \mathbf{h}_i'^{(m)}\Big). \tag{11}$$

4.3. Semantic-level attention module

The goal of this module is to model the importance of different relationships on information diffusion to obtain a vector representation that integrates the effects of various impacting factors. Through the learning of each relational subgraph by the node attention module, we obtain the friendship-based user representations $\mathbf{Z}^{F(u)}$, the interaction-based user representations $\mathbf{Z}^{I(u)}$, and the interest-based message representations $\mathbf{Z}^{I(m)}$. Now, we apply semantic attention to distinguish the importance of each relationship and generate the final representation by fusing the above representations. The specific process is as follows:

Suppose the popularity of message m is to be predicted, and the active user list within the observation window is $[u_A, u_B, u_C]$. First, we query the message representation vector $\mathbf{v}^{I(m)}$ from matrix $\mathbf{Z}^{I(m)}$ according to the ID of message m. Next, we query the friendship-based user representation vector list $[\mathbf{v}_A^{F(u)}, \mathbf{v}_B^{F(u)}, \mathbf{v}_C^{F(u)}]$ from matrix $\mathbf{Z}^{F(u)}$ and the interaction-based user representation vector list $[\mathbf{v}_A^{I(u)}, \mathbf{v}_B^{I(u)}, \mathbf{v}_C^{I(u)}]$ from matrix $\mathbf{Z}^{I(u)}$ according to the user ID of the list $[u_A, u_B, u_C]$. Finally, we fuse the above vectors as $\tilde{\mathbf{V}} \in \mathbb{R}^{N\times d'}$ for semantic attention learning, where N represents the number of vectors in the list. The specific implementation of semantic attention adopts the following multi-head self-attention mechanism:

$$\textit{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \textit{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}, \tag{12}$$

$$h_i = \textit{Attention}(\tilde{\mathbf{V}}\mathbf{W}_i^{\mathbf{Q}}, \tilde{\mathbf{V}}\mathbf{W}_i^{\mathbf{K}}, \tilde{\mathbf{V}}\mathbf{W}_i^{\mathbf{V}}), \tag{13}$$

$$\mathbf{Y} = [h_1; h_2; \ldots; h_H]\mathbf{W}^{\mathbf{O}}, \tag{14}$$

$$\mathbf{pre} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{y}_n, \tag{15}$$

where $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d'\times d_k}$ and $\mathbf{W}^O \in \mathbb{R}^{Hd_k\times d_Q}$; H is the number of heads of attention module. $\mathbf{Y} \in \mathbb{R}^{N\times d_Q}$ represents the vector list after semantic fusion.

4.4. Prediction module

The last component of HeDAN is a multi-layer perceptron (MLP) with one final output unit. Given the representation vector $\mathbf{pre}_i$, we calculate the popularity ΔS_i as:

$$\Delta S_i = MLP(\mathbf{pre}_i). \tag{16}$$

Our ultimate aim is to predict the final cascade size increment ΔS_i of message m_i, which can be done by minimizing the following loss function:

$$loss(\Delta S_i, \Delta\tilde{S}_i) = \frac{1}{M}\sum_{i=1}^{M}(\log \Delta S_i - \log \Delta\tilde{S}_i)^2, \tag{17}$$

where M is the number of messages, ΔS_i is the predicted popularity for message m_i, and $\Delta\tilde{S}$ is the ground truth.

4.5. Complexity analysis

To improve the completeness of the framework proposed in this paper, we further analyze its time complexity, which is expressed as $\mathcal{O}(|\mathcal{V}_u| \cdot d_u \cdot d' + |\mathcal{E}_{all}| \cdot d' + N^2 \cdot d')$. The specific analysis is as follows:

(1) For the node feature transformation, the complexity of mapping features by the projection matrix is $\mathcal{O}(|\mathcal{V}_u| \cdot d_u \cdot d' + |\mathcal{V}_m| \cdot d_m \cdot d')$, where the total number of users is $|\mathcal{V}_u|$, the original feature dimension of users is d_u, the total number of messages is $|\mathcal{V}_m|$, and the original dimension of message features is d_m. The projection matrix output dimension is d'. Since the number of users is much larger than the number of messages, i.e., $|\mathcal{V}_u| \cdot d_u \gg |\mathcal{V}_m| \cdot d_m$, this part of complexity is simplified to $\mathcal{O}(|\mathcal{V}_u| \cdot d_u \cdot d')$.

(2) For the parallel subgraph attention layers, the complexity mainly lies in the calculation of the dot product similarity in the process of calculating the attention coefficient. The complexity of dot product similarity for an edge is d', so the complexity of all edges is $|\mathcal{E}_{all}| \cdot d'$, where $|\mathcal{E}_{all}| = |\mathcal{E}_{F(u)}| + |\mathcal{E}_{I(u)}| + |\mathcal{E}_{IA}| + |\mathcal{E}_{IB}|$.

(3) For the multi-head self-attention mechanism, the computational complexity is $\mathcal{O}(N^2 \cdot d')$, where N is the total number of active user lists within the observation window. The computational complexity comes from the matrix product when calculating the similarity, which is expressed as: $(N \cdot d' \cdot N) = (N^2 \cdot d')$.

Furthermore, note that although the overall time complexity of the framework seems high, many of the matrix and vector operations could be accelerated by GPUs. Therefore, the actual running time depends not only on the total computational cost but also on the GPU resources used to train the model.

5. Evaluation

5.1. Dataset

We select three datasets containing both user social graphs and diffusion cascades for experiments. The detailed statistics are presented in Table 1. Since there are invalid cascades in the datasets, we preprocessed the datasets by removing the invalid cascade samples to eliminate their unpredictable impact. Table 1 lists the minimum length, average length and maximum length of the diffusion sequences in the preprocessed datasets.

Twitter [59]: The Twitter[1] dataset contains the tweets with URLs during October 2010. Each URL is interpreted as an information item spreading among users. The social relation of

[1] http://www.twitter.com.

Table 1
Statistics of the Twitter, Douban, and Weibo datasets.

Datasets	Twitter	Douban	Weibo
#Users	12,627	23,123	2,000,000
#Links	309,631	348,280	12,822,901
#Cascades	3442	10.662	22,767
#Train Casacdes	2768	8529	18,231
#Valid Cascades	345	1067	2265
#Test Cascades	344	1066	2271
Avg.Length	16.54	33.46	60.79
Min.Length	1.00	1.00	10.00
Max.Length	500.00	500.00	1400.00

users is the follower–followee relationship on Twitter. Since only cascade sequences are provided in the dataset, we add *interaction* connections between the current active user and all historical users. The dataset involves 12,627 users and 309,631 social relationships. We consider observation time windows of 1 h, 2 h and 3 h. For each observation window, only data whose number of forwarding behaviors is between 1 and 500 are considered. After preprocessing, we obtain 3442 cascades.

Douban [59]: The Douban[2] dataset is collected from a social website where users can update their book reading and movie watching status and follow the status of other users. Each book or movie is considered an information item, and a user is infected if he or she reads or watches it. The social relation of users is the co-occurrence relation, i.e., if two users take part in the same discussion more than 20 times, they are considered a friend. The dataset involves 23,123 users and 348,280 social relationships. We set the observation time windows to 1 year, 2 years and 3 years. For each observation window, only data whose number of forwarding behaviors is between 1 and 500 are considered. After preprocessing, we obtained 10,662 cascades. Since only cascade sequences are provided in the dataset, we add *interaction* connections between the current active user and all previously active users.

Weibo: The Weibo[3] dataset contains approximately 30,000 source messages and their corresponding diffusion dynamics (i.e., cascade paths) from the Sina Weibo platform. These messages were forwarded more than 17.84 million times in total, involving approximately 8 million users, and the subsequent relationships between users exceeded 700 million. Due to the large scale of the dataset, we trimmed the dataset by keeping only 2 million users and the corresponding social relations. We consider observation time windows of 2 h, 4 h and 6 h. For each observation window, only data whose number of forwarding behaviors is between 10 and 1400 are considered. After preprocessing, we obtained 22,767 cascades.

5.2. Comparing methods

To evaluate the effectiveness of HeDAN, we select four methods from the existing deep learning-based macro-level information diffusion prediction methods for comparison. For the sequential representation-based methods, we select DeepCas [41] and DeepHawkes [34]. For the graph representation-based methods, we select DeepCon+Str [27] and CoupledGNN [30]. The baselines and their implementation details are as follows:

- **DeepCas** [41] represented a cascade graph as a set of random walk paths and utilized GRU and an attention mechanism for modeling and predicting casacde size. It is the first deep learning-based method for information popularity prediction, which implemented popularity prediction in an end-to-end manner. It is also a classic sequential representation-based method for information popularity prediction, specifically using GRU to learn the changes between cascade paths.
- **DeepHawkes** [34] combined the predictive power of sequence representation learning and the interpretability of the Hawkes process to predict information popularity. It considered three key factors of the Hawkes process, i.e., influence of users, self-exciting mechanism, and time decay effect. This method belongs to both generative approaches and deep learning-based methods.
- **DeepCon+Str** [27] represented intercascade associations by building a higher-order graph between cascades based on content and structural proximity and obtained the embedding of the entire cascade graph through random walks and semi-supervised language models. It directly considered the associations between cascades and generated a representation at the cascade level instead of generating a representation for each node in the cascade.
- **CoupledGNN** [30] captured the cascading effect explicitly on the social network to model the activation state of a target user given the activation state and influence of his or her neighbors and specifically utilized two coupled graph neural networks to capture the interplay between node activation states and the spread of influence. It is currently one of the best-performing graph representation-based methods considering both social information and cascade information for information popularity prediction.

5.3. Evaluation metrics

For macro-level information diffusion prediction methods, the commonly used evaluation metrics include the mean square error (MSE), mean square logarithm transformation error (MSLE), and median square logarithm transformation error (mSLE). Following the existing works [25,26,34,41], we choose MSLE and mSLE as the evaluation metrics of the experiments, and the calculation formula is shown as:

(1) Mean Square Logarithm Transformation Error (MSLE):

$$MSLE = \frac{1}{M}\sum_{i=1}^{M} SLE^i. \tag{18}$$

(2) Median Square Logarithm Transformation Error (mSLE):

$$mSLE = median(SLE^1, \ldots, SLE^M). \tag{19}$$

5.4. Parameter settings

To tune the hyperparameters, we randomly select 10% of the training cascades as the validation set. Note that all training cascades, including the validation set are used to train the final model for testing. Therefore, 10% of each dataset is used for validation, 10% for testing, and the rest for training.

For a fair comparison, we set the embedding dimension of all the algorithms to 64. For baselines, we use their best-performing parameters as reported in their original papers. Specifically, for DeepCas, DeepHawkes and DeepCon+Str, we sample K = 200 paths each with length $T = 10$ from the cascade graph, and set the hidden layer of each GRU to 32 units and the hidden dimensions of the two fully connected layers to 32 and 16. For CoupledGNN, the number of GNN layers is 3, and the number of hidden units in the state gating mechanism is 20. The other

[2] http://www.douban.com.
[3] http://www.weibo.com.

Table 2
Experimental results on Twitter dataset.

Dataset	Twitter					
Observation time	1 h		2 h		3 h	
Evaluation metric	MSLE	mSLE	MSLE	mSLE	MSLE	mSLE
DeepCas	1.3770	0.2788	1.3227	0.3092	1.3180	0.2992
DeepHawkes	0.9322	0.1710	0.8953	0.1615	0.8222	0.1552
DeepCon+Str	0.8847	0.1366	0.8521	0.1288	0.7297	0.1243
CoupledGNN	0.7867	0.1301	0.7660	0.1247	0.7045	0.1208
HeDAN	**0.7766**	**0.1263**	**0.7349**	**0.1211**	**0.6606**	**0.1127**

Table 3
Experimental results on Douban dataset.

Dataset	Douban					
Observation time	1 year		2 years		3 years	
Evaluation metric	MSLE	mSLE	MSLE	mSLE	MSLE	mSLE
DeepCas	1.1293	0.2564	1.0997	0.2369	0.9452	0.2408
DeepHawkes	0.8135	0.1820	0.7335	0.1788	0.7020	0.1665
DeepCon+Str	0.7026	0.1738	0.6854	0.1692	0.6663	0.1673
CoupledGNN	0.6330	0.1696	0.6262	0.1633	0.6035	0.1631
HeDAN	**0.6260**	**0.1676**	**0.6158**	**0.1618**	**0.5927**	**0.1618**

Table 4
Experimental results on Weibo dataset.

Dataset	Weibo					
Observation time	2 h		4 h		6 h	
Evaluation metric	MSLE	mSLE	MSLE	mSLE	MSLE	mSLE
DeepCas	2.2237	1.2260	2.2343	1.2394	2.2053	1.2102
DeepHawkes	1.5527	0.9886	1.5332	0.9727	2.5096	0.9662
DeepCon+Str	1.3724	0.9224	1.3522	0.9103	1.3042	0.8962
CoupledGNN	1.2228	0.7228	1.1055	0.6888	1.0134	0.6835
HeDAN	**1.1139**	**0.6905**	**1.0264**	**0.5707**	**0.9734**	**0.5664**

optimization parameters of the models are set to their default values.

For HeDAN, we perform a simple greedy-based manual tuning strategy to select the hyperparameters, i.e., each time the value of one parameter is tuned by fixing the other parameters. Specifically, the number of GAT layers is 2, the number of heads H_1 in the multi-head attention is 8, and the dimension of each head is 8. For the multi-head self-attention mechanism, we set the number of heads H_2 to 4, and the dimension of each head is 64. For the optimization algorithm, we employ the Adam optimizer with the learning rate set to 0.005 and the weight decay (L2 penalty) set to 0.001. Furthermore, we adopt the dropout mechanism with a dropout ratio of 0.6 to prevent the model from overfitting. Our model is implemented via the Deep Graph Library (DGL) [63] package under PyTorch frameworks. For the reproducibility of the results, we have made the source code publicly available.[4]

6. Results

6.1. Overall performance

Tables 2, 3 and 4 show the performance of all methods on the three datasets, where the best results are shown in bold.

Tables 2, 3 and 4, we can see that HeDAN outperforms the state-of-the-art methods by a significant margin. For example, on the Weibo dataset, HeDAN outperforms the second-best method (CoupledGNN) by an average of nearly 10% under MSLE and mSLE. Specifically, we have the following observations:

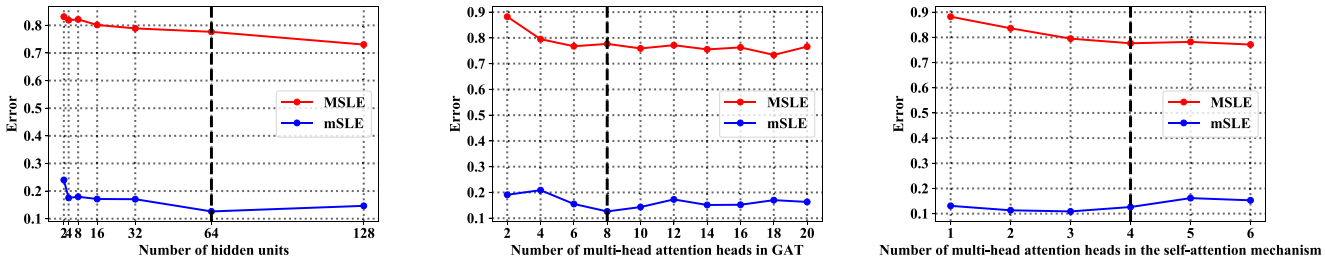
- As the first deep learning-based method, DeepCas significantly lags behind other methods. This is mainly because DeepCas only utilized RNN to learn user representations but ignored the influence of other factors such as social relationships, temporal information, real forwarding paths, or diffusion structures. In contrast, the DeepHawkes method considered three key factors of the Hawkes process, so it had better performance (over 20% improvement over DeepCas under MSLE and mSLE on all three datasets).
- Compared with DeepHawkes, other graph representation-based methods show superior performance (over 10% improvement in MSLE and more than 5% improvement in mSLE on three datasets). This indicates that graph representation-based methods can learn more graph structural information than sequential representation-based methods in information diffusion modeling.
- Compared with DeepCon+Str, HeDAN achieves approximately 10% improvement under MSLE and over 5% improvement under mSLE on the Twitter and Douban datasets and approximately 20% improvement under MSLE and mSLE on the Weibo dataset. The DeepCon+Str method manually extracted the features to establish high-level similarity between cascades and employed the random walk and semi-supervised language model to train features. However, since this method ignores fine-grained user-message interactions, it might lose detailed information for users and cascades. In contrast, HeDAN directly constructs the "Interaction" and "Interest" relationships, which significantly reduces the loss of information compared with DeepCon+Str. Moreover, HeDAN uses a graph representation model such as GAT to learn the node representation, which better captures the internal structure of the cascades compared with the semi-supervised language model. Thus, HeDAN exhibits better prediction performance than DeepCon+Str. The experimental results indicate that it is important to preserve both the cascade graph structure and social relationships to capture the user-message interactions for modeling information diffusion.
- Compared with CoupledGNN, HeDAN achieves over 5% improvement under MSLE and mSLE on three datasets. Compared with CoupledGNN, which directly places a separate diffusion sequence on the social network to capture the cascade effect, HeDAN considers coprocessing of the cascades to capture the interactions between users and the relationship between cascades, which has a boosting effect in predicting the popularity of cascades. The experimental results indicate the importance of collaborative processing of all cascades and the effect of mining users' preferences for cascade popularity prediction.

6.2. Parameter sensitivity

As mentioned above, we employed a simple greedy-based manual tuning strategy to select the hyperparameters, including the number of hidden units d', the number of multi-head attention heads H_1 in GAT, and the number of multi-head attention heads H_2 in the self-attention mechanism. To clearly show the influence of these hyperparameters, we report the MSLE and mSLE results of HeDAN under different parameter settings on the Twitter and Douban datasets, as shown in Figs. 3 and 4, respectively.

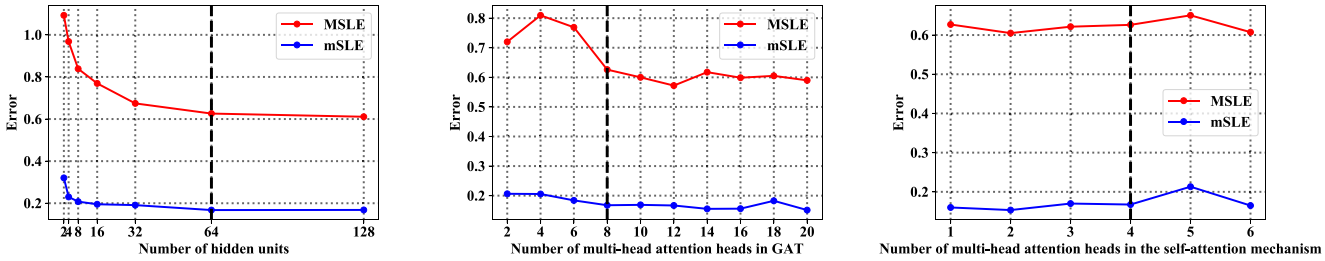
Impact of hidden units d' on model performance. The number of hidden units d' determines the dimensionality of the feature space after feature projection through Eqs. (1) and (2). Figs. 3(a) and 4(a) show how it affects the prediction performance of HeDAN on the Twitter and Douban datasets. Intuitively, the

[4] https://github.com/Shuey-Q/HeDAN-master.



(a) Impact of d' on model performance. (b) Impact of H_1 on model performance. (c) Impact of H_2 on model performance.

Fig. 3. Parameter sensitivity analysis for HeDAN on the Twitter dataset under one-hour observation time.



(a) Impact of d' on model performance. (b) Impact of H_1 on model performance. (c) Impact of H_2 on model performance.

Fig. 4. Parameter sensitivity analysis for HeDAN on the Douban dataset under one-year observation time.

higher the dimensionality of the feature space is, the richer the semantic information contained in the embedding vector, and better model performance will be achieved. From Figs. 3(a) and 4(a), we can observe the following: (1) The model performance under both datasets improves positively with the increase of dimensions of the embedding space. (2) The performance improvement becomes insignificant when d' exceeds 64. Therefore, d' is finally set to 64.

The impact of multi-head attention mechanisms H_1 and H_2. Figs. 3(b)(c) and 4(b)(c) show the impact of multi-head attention heads in GAT (H_1) and the self-attention mechanism (H_2) on the model performance. From Figs. 3(b) and 4(b), where the value of H_1 is chosen from {2, 4, 6, 8, 10, 12, 14, 16, 18, 20}, we see that a small number of attention heads (e.g., $H_1 \leq 4$) may result in inaccurate model performance. However, when H_1 is higher than 10, further increasing its value will no longer benefit the model performance. Moreover, a large size of attention heads will incur more computational cost. Therefore, considering the balance between model accuracy and training efficiency, we finally set H_1 to 8. A similar phenomenon is observed for H_2, whose value is chosen from {1, 2, 3, 4, 5, 6}, and the results are shown in Figs. 3(c) and 4(c). Based on the results and considering the accuracy and efficiency balance, the value of H_2 is finally set to 4.

6.3. Ablation experiments

To show the relative importance of each module in HeDAN, we perform a series of ablation studies over the key modules of the model. We conduct ablation experiments on the Twitter dataset with an observation time window of 1 h. The experimental results are presented in Table 5. The ablation studies are conducted as follows:

(1) Heterogeneous diffusion graph module.

(1.1) w/o "Friendship": Removing the friendship subgraph attention layer and randomly initializing the friendship-based user representations.

(1.2) w/o "Interaction": Removing the interaction subgraph attention layer and randomly initializing the interaction-based user representations.

(1.3) w/o "Interest": Removing the interest subgraph attention layer and randomly initializing the interest-based message representations.

(2) Node-level attention module.

(2.1) GCN: Replacing graph attention layers with graph convolutional layers.

(3) Semantic-level attention module.

(3.1) Mean-Pooling: Replacing the multi-head self-attention mechanism with a mean-pooling mechanism.

Table 5
Ablation study on Twitter dataset.

Twitter under 1 h observation time					
Evaluation metric	MSLE	mSLE	Evaluation metric	MSLE	mSLE
HeDAN	**0.7766**	**0.1263**	**HeDAN**	**0.7766**	**0.1263**
(1.1) w/o "Friendship"	0.7943	0.2202	(2.1) GCN	0.7954	0.1583
(1.2) w/o "Interaction"	0.8460	0.1667	(3.1) Mean-pooling	0.7760	0.1308
(1.3) w/o "Interest"	1.0723	0.4314	–	–	–

Table 5 gives the overall performance on several variant methods of HeDAN. Referring to the experimental results in the table, we can observe the following:

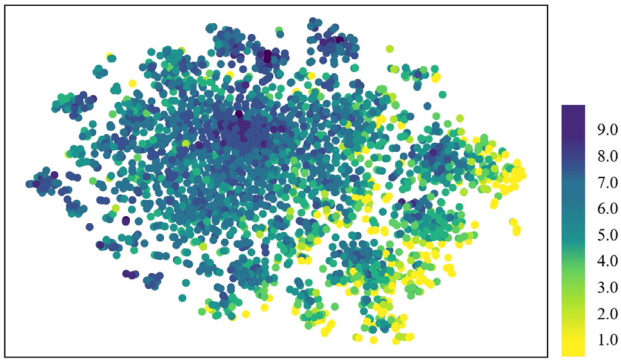
- Comparing the results of variant models (1.1), (1.2) and (1.3) with HeDAN, it can be seen that the error rate (MSLE or MSLE) of the prediction model increases to varying degrees after removing a certain type of relationship in the heterogeneous diffusion network, indicating that all three relationships are valuable for modeling information diffusion. The comprehensive consideration of the three relations improves the prediction performance of the model.
- Comparing the results of the variant model (2.1) and HeDAN, it is found that when the GAT module of the subgraph attention network module is replaced by the GCN module, the error rate (MSLE or MSLE) of the prediction model increases because each user has different preferences and

plays a different role. Compared with the equal-weighted neighborhood aggregation in the GCN module, the attention mechanism in the GAT model can learn the importance within users as well as between users and information, indicating that a node-level attention layer can facilitate the prediction effect.

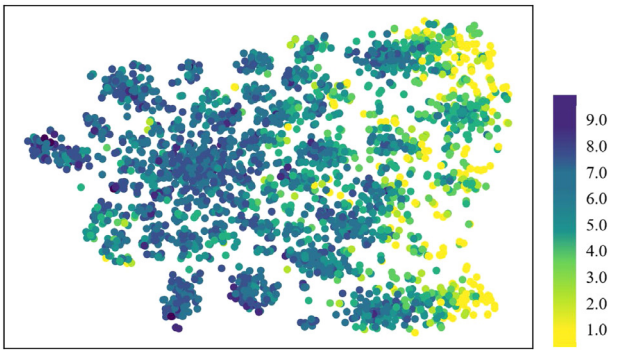
- Comparing the results of the variant model (3.1) and HeDAN, it is found that the error rate (MSLE or MSLE) of the prediction model increases after the multi-head self-attention mechanism is replaced by a mean-pooling mechanism. This indicates that the semantic-level attention module weights the message representation and user representation and characterizes the implicit semantics of different relations for information diffusion, which promotes the prediction effect of the model.

6.4. Interpretability analysis

In this section, we analyze whether HeDAN can provide some interpretability. Specifically, we utilize the t-SNE [64] algorithm for dimensionality reduction and visualization of the final prediction representations and message representations, as shown in Fig. 5(a) and (b), respectively. It should be noted that the values (as shown in the color bar) range from approximately 1.0 to 9.0 because we have performed logarithmic transformation on the original popularity values. From the results we found that (1) after reducing the dimensionality to 2 dimensions based on the distances between datapoints and their probability distributions, the final prediction representations and message representations have relatively similar spatial distributions, which shows that they both have the same tendency in predicting the popularity of information. (2) Most of the points with larger labels in Fig. 5(a) (darker color) are concentrated in the upper left part, and most of the points with smaller labels (lighter color) are concentrated in the lower right part, which indicates that the difference in true popularity between samples is proportionally similar to the difference between the final predicted representations. (3) We find a clear change in the popularity distribution in Fig. 5 (darker from right to left), and datapoints are aggregated rather than scattered, which indicates that the latent representations learned by HeDAN are expressive in the popularity prediction problem.



(a) Dimensionality reduction visualization of final prediction representations for samples from Twitter by t-SNE.



(b) Dimensionality reduction visualization of message representations for samples from Twitter by t-SNE.

Fig. 5. Dimensionality reduction visualization of latent representations for samples from Twitter by t-SNE. Each datapoint represents a cascade. The t-SNE algorithm reduces the dimensionality to two dimensions for visual representation based on the distances between data points and their probability distributions. The darker the color of the datapoint is, the higher its popularity value is. Conversely, the lighter the color of the datapoint, the lower the corresponding predicted value.

7. Conclusion

In this paper, we studied the information popularity prediction problem on social media platforms. To comprehensively consider various factors that affect information diffusion, we proposed a novel heterogeneous diffusion attention network model to characterize both the user and message representations through hierarchical attention. Specifically, we learned various subgraph structures through node-level attention and creatively integrated the roles of friendship, user interaction and user preference through semantic-level attention. We conducted experiments on three real-world datasets. The experimental results indicate that our model has achieved significant improvements over state-of-the-art models.

For future work, on the one hand, we will consider extending the idea of this model to the micro problem (e.g., user behavior prediction) to address the two problems simultaneously. On the other hand, we will consider extending this framework from the aspect of multi-modality, which can integrate multi-media data such as text, image, and video information into the framework, thus improving its scalability.

CRediT authorship contribution statement

Xueqi Jia: Conceptualization, Writing – original draft, Methodology, Software. **Jiaxing Shang:** Supervision, Writing – review & editing, Funding acquisition. **Dajiang Liu:** Data Curation, Writing – review & editing. **Haidong Zhang:** Software, Validation. **Wancheng Ni:** Supervision, Writing – review & editing.

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 was supported in part by the following: the National Natural Science Foundation of China (Nos. 61966008, U2033213).

References

[1] S. Bhattacharya, K. Gaurav, S. Ghosh, Viral marketing on social networks: An epidemiological perspective, Physica A 525 (2019) 478–490.
[2] J. Shang, S. Zhou, X. Li, L. Liu, H. Wu, CoFIM: A community-based framework for influence maximization on large-scale networks, Knowl.-Based Syst. 117 (2017) 88–100.

[3] C. Huang, H. Xu, Y. Xu, P. Dai, L. Xia, M. Lu, L. Bo, H. Xing, X. Lai, Y. Ye, Knowledge-aware coupled graph neural network for social recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 4115–4122.
[4] S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, Science 359 (6380) (2018) 1146–1151.
[5] C. Song, W. Hsu, M.L. Lee, Temporal influence blocking: Minimizing the effect of misinformation in social networks, in: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), IEEE, 2017, pp. 847–858.
[6] Y. Liu, Y.-F. Wu, Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018.
[7] Y. Liu, Y.-F.B. Wu, Fned: a deep network for fake news early detection on social media, ACM Trans. Inform. Syst. (TOIS) 38 (3) (2020) 1–33.
[8] J. Zhu, S. Ghosh, W. Wu, Robust rumor blocking problem with uncertain rumor sources in social networks, World Wide Web 24 (1) (2021) 229–247.
[9] M.A. Manouchehri, M.S. Helfroush, H. Danyali, Temporal rumor blocking in online social networks: A sampling-based approach, IEEE Trans. Syst. Man Cybern. Syst. (2021).
[10] O. Tsur, A. 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, 2012, pp. 643–652.
[11] D.M. Romero, C. Tan, J. Ugander, On the interplay between social and topical structure, in: Seventh International AAAI Conference on Weblogs and Social Media, 2013.
[12] H. Pinto, J.M. Almeida, M.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, 2013, pp. 365–374.
[13] A. Anderson, D. Huttenlocher, J. Kleinberg, J. Leskovec, M. Tiwari, Global diffusion via cascading invitations: Structure, growth, and homophily, in: Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 66–76.
[14] 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.
[15] H. Shen, D. Wang, C. Song, A.-L. Barabási, Modeling and predicting popularity dynamics via reinforced poisson processes, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 28, 2014.
[16] J. Gao, H. Shen, S. Liu, X. Cheng, Modeling and predicting retweeting dynamics via a mixture process, in: Proceedings of the 25th International Conference Companion on World Wide Web, 2016, pp. 33–34.
[17] S. Xiao, J. Yan, X. Yang, H. Zha, S. Chu, Modeling the intensity function of point process via recurrent neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, 2017.
[18] J. Wang, V.W. Zheng, Z. Liu, K.C.-C. Chang, Topological recurrent neural network for diffusion prediction, in: 2017 IEEE International Conference on Data Mining (ICDM), IEEE, 2017, pp. 475–484.
[19] Z. Wang, C. Chen, W. Li, A sequential neural information diffusion model with structure attention, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 1795–1798.
[20] M.R. Islam, S. Muthiah, B. Adhikari, B.A. Prakash, N. Ramakrishnan, Deepdiffuse: Predicting the'who'and'when'in cascades, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 1055–1060.
[21] C. Yang, M. Sun, H. Liu, S. Han, Z. Liu, H. Luan, Neural diffusion model for microscopic cascade study, IEEE Trans. Knowl. Data Eng. (2019).
[22] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, J. Tang, Deepinf: Social influence prediction with deep learning, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2110–2119.
[23] S. Molaei, H. Zare, H. Veisi, Deep learning approach on information diffusion in heterogeneous networks, Knowl.-Based Syst. 189 (2020) 105153.
[24] X. Tang, Y. Liu, N. Shah, X. Shi, P. Mitra, S. Wang, Knowing your fate: Friendship, action and temporal explanations for user engagement prediction on social apps, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 2269–2279.
[25] F. Zhou, X. Xu, K. Zhang, G. Trajcevski, T. Zhong, Variational information diffusion for probabilistic cascades prediction, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 1618–1627.
[26] 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), IEEE, 2019, pp. 770–781.
[27] X. Feng, Q. Zhao, Z. Liu, Prediction of information cascades via content and structure integrated whole graph embedding, in: IJCAI, 2019.
[28] B. Shulman, A. Sharma, D. Cosley, Predictability of popularity: Gaps between prediction and understanding, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 10, 2016, pp. 348–357.
[29] Z. Wang, C. Chen, W. Li, Joint learning of user representation with diffusion sequence and network structure, IEEE Trans. Knowl. Data Eng. (2020).
[30] 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, 2020, pp. 70–78.
[31] Y. Zhao, N. Yang, T. Lin, S.Y. Philip, Deep collaborative embedding for information cascade prediction, Knowl.-Based Syst. 193 (2020) 105502.
[32] L. Weng, F. Menczer, Y.-Y. Ahn, Predicting successful memes using network and community structure, in: Eighth International AAAI Conference on Weblogs and Social Media, 2014.
[33] L. Hong, O. Dan, B.D. Davison, Predicting popular messages in twitter, in: Proceedings of the 20th International Conference Companion on World Wide Web, 2011, pp. 57–58.
[34] 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, 2017, pp. 1149–1158.
[35] P. Bao, H.-W. Shen, X. Jin, X.-Q. Cheng, Modeling and predicting popularity dynamics of microblogs using self-excited hawkes processes, in: Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 9–10.
[36] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[37] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), 2014.
[38] Y. Wang, H. Shen, S. Liu, J. Gao, X. Cheng, Cascade dynamics modeling with attention-based recurrent neural network, in: IJCAI, 2017, pp. 2985–2991.
[39] Z. Wang, C. Chen, W. Li, Attention network for information diffusion prediction, in: Companion Proceedings of the the Web Conference 2018, 2018, pp. 65–66.
[40] H. Wang, C. Yang, Information diffusion prediction with latent factor disentanglement, 2020, arXiv preprint arXiv:2012.08828.
[41] 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, 2017, pp. 577–586.
[42] C. Yang, J. Tang, M. Sun, G. Cui, Z. Liu, Multi-scale information diffusion prediction with reinforced recurrent networks, in: IJCAI, 2019, pp. 4033–4039.
[43] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: A review of methods and applications, AI Open 1 (2020) 57–81.
[44] F. Zhang, J. Tang, X. Liu, Z. Hou, Y. Dong, J. Zhang, X. Liu, R. Xie, K. Zhuang, X. Zhang, et al., Understanding WeChat user preferences and "wow" diffusion, IEEE Trans. Knowl. Data Eng. (2021).
[45] C. Yuan, J. Li, W. Zhou, Y. Lu, X. Zhang, S. Hu, DyHGCN: A dynamic heterogeneous graph convolutional network to learn users' dynamic preferences for information diffusion prediction, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2020, pp. 347–363.
[46] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710.
[47] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.
[48] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, 2017, arXiv preprint arXiv:1710.10903.
[49] X. Chen, K. Zhang, F. Zhou, G. Trajcevski, T. Zhong, F. Zhang, Information cascades modeling via deep multi-task learning, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 885–888.
[50] M. Schlichtkrull, T.N. Kipf, P. Bloem, R.v.d. Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: European Semantic Web Conference, Springer, 2018, pp. 593–607.
[51] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, P.S. Yu, Heterogeneous graph attention network, in: The World Wide Web Conference, 2019, pp. 2022–2032.
[52] S. Zhu, C. Zhou, S. Pan, X. Zhu, B. Wang, Relation structure-aware heterogeneous graph neural network, in: 2019 IEEE International Conference on Data Mining (ICDM), IEEE, 2019, pp. 1534–1539.
[53] Z. Liu, C. Huang, Y. Yu, B. Fan, J. Dong, Fast attributed multiplex heterogeneous network embedding, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 995–1004.
[54] Y. Yang, J.-Q. Yang, R. Bao, D.-C. Zhan, H. Zhu, X.-R. Gao, H. Xiong, J. Yang, Corporate relative valuation using heterogeneous multi-modal graph neural network, IEEE Trans. Knowl. Data Eng. (2021).

[55] P. Goyal, S.R. Chhetri, A. Canedo, Dyngraph2vec: Capturing network dynamics using dynamic graph representation learning, Knowl.-Based Syst. 187 (2020) 104816.
[56] Z. Liu, C. Huang, Y. Yu, J. Dong, Motif-preserving dynamic attributed network embedding, in: Proceedings of the Web Conference 2021, 2021, pp. 1629–1638.
[57] D. Wang, Z. Zhang, Y. Ma, T. Zhao, T. Jiang, N. Chawla, M. Jiang, Modeling co-evolution of attributed and structural information in graph sequence, IEEE Trans. Knowl. Data Eng. (2021).
[58] S. Bourigault, S. Lamprier, P. Gallinari, Representation learning for information diffusion through social networks: an embedded cascade model, in: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, 2016, pp. 573–582.
[59] F. Zhou, X. Xu, G. Trajcevski, K. Zhang, A survey of information cascade analysis: Models, predictions, and recent advances, ACM Comput. Surv. 54 (2) (2021) 1–36.
[60] L. Yu, P. Cui, F. Wang, C. Song, S. Yang, From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics, in: 2015 IEEE International Conference on Data Mining, IEEE, 2015, pp. 559–568.
[61] C. Zhang, D. Song, C. Huang, A. Swami, N.V. Chawla, Heterogeneous graph neural network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 793–803.
[62] Z. Hu, Y. Dong, K. Wang, Y. Sun, Heterogeneous graph transformer, in: Proceedings of the Web Conference 2020, 2020, pp. 2704–2710.
[63] M.Y. Wang, Deep graph library: Towards efficient and scalable deep learning on graphs, in: ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.
[64] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).