

Article

Dynamic End-to-End Information Cascade Prediction Based on Neural Networks and Snapshot Capture

Delong Han ^{1,2,†} , Tao Meng ^{1,2,*,†}  and Min Li ^{1,2} 

¹ Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China; handl@sdas.org (D.H.); limin@qlu.edu.cn (M.L.)

² Shandong Provincial Key Laboratory of Computer Networks, Shandong Fundamental Research Center for Computer Science, Jinan 250353, China

* Correspondence: 10431210595@stu.qlu.edu.cn

† These authors contributed equally to this work.

Abstract: Knowing how to effectively predict the scale of future information cascades based on the historical trajectory of information dissemination has become an important topic. It is significant for public opinion guidance; advertising; and hotspot recommendation. Deep learning technology has become a research hotspot in popularity prediction, but for complex social platform data, existing methods are challenging to utilize cascade information effectively. This paper proposes a novel end-to-end deep learning network CAC-G with cascade attention convolution (CAC). This model can stress the global information when learning node information and reducing errors caused by information loss. Moreover, a novel Dynamic routing-AT aggregation method is investigated and applied to aggregate node information to generate a representation of cascade snapshots. Then, the gated recurrent unit (GRU) is employed to learn temporal information. This study's validity and generalization ability are verified in the experiments by applying CAC-G on two public datasets where CAC-G is better than the existing baseline methods.

Keywords: social platform; information cascade; popularity prediction; attention; global information; aggregate; snapshot



Citation: Han, D.; Meng, T.; Li, M. Dynamic End-to-End Information Cascade Prediction Based on Neural Networks and Snapshot Capture.

Electronics **2023**, *12*, 2875. <https://doi.org/10.3390/electronics12132875>

Academic Editors: Rashid Mehmood and Simeone Marino

Received: 22 May 2023

Revised: 20 June 2023

Accepted: 27 June 2023

Published: 29 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Nowadays, online public opinion has become an essential factor affecting society's sustainable and orderly development and maintaining social harmony and stability. Meanwhile, understanding the laws of information dissemination and of modeling and predicting the popularity of online content have become important research topics for social media. This is important for platforms or governments to control opinion trends [1]; improve security; and benefit numerous applications in public administration, business, and security-related fields. However, due to the openness of social networks, the platform has a vast number of users, and the intricate relationships between users and noisy social media data significantly affect prediction accuracy. In addition, the information dissemination process also involves time series information, and knowing how to combine time series information with structural information reasonably is an essential factor affecting the model prediction performance. Meanwhile, interpretability is also a vital issue in the study of information popularity prediction, and providing a theoretical basis for the working principle of the prediction model will also improve the credibility of the model prediction results. It follows that making fast and accurate predictions with regard to information is a complicated and challenging task.

Information dissemination on social platforms stimulates the emergence of an information cascade, and the prediction of information popularity can be regarded as a prediction of the scale of the information cascade [2]. The existence of inter-influence relationships

among users [3] allows us to explore the underlying laws of cascade diffusion better. In addition to the forwarding of information on social platforms, information cascades are widely used in scenarios such as paper citation, virus propagation [4], recommendation [5], and information security [6]. Early information prevalence prediction methods are mainly based on features [7,8]. The techniques usually extract features from text content, structural features, time features, and the unique attributes of the original user. However, these features are usually obtained based on people's experience and only show good results on specific platforms or data. In addition, this method is based on people's subjective judgment and has substantial uncertainty. Compared with the method proposed in this paper, the interpretability of such approaches needs to be more pronounced, and they need better generality and robustness.

The advent of generative methods further solves these problems [9,10]. The techniques usually use Poisson or Hawkes process [11,12] to model the information cascade to enhance interpretability and robustness. However, this method cannot fully utilize the hidden information, so the prediction effect is not ideal. Recently, the authors of [13–15] have sought to predict machine learning methods. Cheng et al. [16] proposed that an information cascade can be combined with a deep learning method, and the nodes in the cascade can be embedded and encoded into the deep learning model to realize end-to-end learning. Cascaded diffusion structures in deep learning are usually learned using graph neural networks (GNNs) [17,18]. However, the data from various social platforms are too noisy with low utilization rates, thus ensuring that the existing models cannot fully utilize their performance.

For the prediction problem of complex cascade size, this paper proposes a deep learning framework CAC-G. Firstly, the cascade is divided into multiple snapshots based on previous experience, and the cascade snapshots are learned through the CAC model, which is mainly proposed to perform feature extraction for complex cascades through the graph attention network (GAT) [19] and the convolutional neural network (CNN) [20]. The organic combination of node and edge features is fed into GAT for learning, while node features are also fed into CNN. Finally, the output of GAT and CNN is subjected to a splicing operation to obtain the output features of the cascade snapshot. The proposed CAC model effectively reduces information loss and is more suitable for dynamic cascade processing than previous methods. Next, send the node representation of cascaded snapshots output by CAC to Dynamic routing-AT for aggregation, and obtain the vector representation of snapshots. Dynamic routing-AT is proposed based on the Dynamic routing algorithm in the capsule network [21] combined with self-attention [22] to make aggregation more accurate and efficient. The temporal information is then hidden in the sequence of cascade snapshots output by Dynamic routing-AT, and GRU processes the hidden temporal information [23]. Then, a multilayer perceptron (MLP) [24] is applied to predict the final growth size of the cascade. To verify the effectiveness of CAC-G, we conducted experiments on six sub-datasets in two scenarios cited by Sina Weibo and the paper, respectively. The results show that CAC-G has better predictive power than the existing baseline methods. Meanwhile, sufficient ablation experiments are carried out to verify the effectiveness of each part of CAC-G. CAC-G provides a new method for learning complex cascade features and vector aggregation, a new idea for popularity prediction research, and a new baseline for future information cascade prediction.

The main contributions of this paper are listed as follows:

- (1) This paper provides a novel popularity prediction framework CAC-G, which takes the cascade network as an input and the final size of the cascade as the output. It is an end-to-end learning framework that fully considers the cascade's structural and time series information.
- (2) Our proposal is implemented for complex cascade processing. By organically combining GAT and CNN, the model can pay more attention to the overall situation information, thus reducing errors caused by information losses.

- (3) A novel vector aggregation method, Dynamic routing-AT, is presented in this paper. The research combines the self-attention weight calculation method with the capsule network's Dynamic routing algorithm, making the model prediction more stable and less time cost.

The rest of this paper is organized as follows. Section 2 reviews the relevant studies on prevalence prediction. Section 3 introduces the CAC-G model in detail. We explained our experiment specifically and introduced the data set and related parameter settings in Section 4. The experimental results are further analyzed and discussed in Section 5, and the validity of each part of the CAC-G model is verified. In Section 6, we summarize this paper and propose some future work.

2. Related Works

2.1. Feature-Based Methods

After obtaining datasets from social platforms, this method usually extracts various features, mainly including content features [25], structural features [25], time features [26,27], and the unique attributes of the original users, and then predicts them in the machine learning model. Szabo et al. [7] found an inseparable linear relationship between the future popularity of online content and its early popularity. Bakshy et al. [28] used a regression decision tree to deal with user influence and message content to prove that both features play an essential role in cascade predictions. Tsur et al. [29] learned many linguistic features, user features, message topics, and time series features and predicted future popularity with various machine learning models. Shulman et al. [8] found that the speed of the first few users forwarding the message strongly influences the final popularity. In conclusion, the feature-based approach provides a general method for the popularity prediction problem. However, the technique is limited by manual extraction, which makes it difficult to generalize from one domain to another and could be more conducive to generality and robustness.

2.2. Generative Method

The generative methods enhance information growth by independently modeling the intensity function of each message's arrival process according to the dissemination process of user-generated content [30,31]. Manuel et al. [32] applied survival theory to establish general additive and multiplicative risk models, considering that a node can increase or decrease the activation probability of another node. Zamam et al. [33] proposed that a Bayesian model was adopted for calculation based on the time series of information, forwarding time, and network structure to predict the information popularity. Shen et al. [9] put forward an enhanced Poisson process to simulate a single item to obtain the final popularity. Although generative methods are to some extent superior to feature-based methods, generative methods share common drawbacks; generative models simplify the arrival rate of events, limiting their learning ability on large-scale cascaded data [34]. Because the actual propagation process of messages is complex and diverse, it is difficult for the method to update the parameters according to the specific propagation process and fully use the message cascade's confidential information. Therefore, the prediction effect is not satisfactory.

2.3. Deep Learning-Based Methods

The cascade prediction method has developed rapidly in light of deep learning in recent years. Methods for cascaded graph processing have also received extensive attention [35,36]. Cheng et al. [16] proposed the first deep learning-based information cascade prediction framework, DeepCas. A set of node sequences is obtained through random walks, and it can automatically learn node representations through neural networks and predict the size of the cascade growth. However, it ignores the time information of the cascade. The DeepHawkes model [37] combines RNN and Hawkes models, interpretatively uses generative methods, and achieves end-to-end learning. However, since RNN is used

for encoding, the structural information of the cascade is not fully applied. Wang et al. [38] came up with a novel recurrent neural network TopoLSTM to model dynamic graphs, but it ignored the inherent structural information of cascades. It can be seen that most of the above methods use a structure similar to RNN for cascades, so such methods do not make the best of the complete topology of cascades in modeling, producing their unsatisfactory prediction in complex social networks. Following this idea, Chen et al. [39] proposed to use the CasCN model, which first applied random walks to obtain cascade subgraphs from the propagation graph, and then a graph convolutional network (GCN) can learn the structural features of the cascade graph. Finally, RNN is used to learn time series information. Combining the structural information and time series information, this method finally outputs cascade prediction. CoupledGNN [40] captured the network structure by two coupled graph neural networks to predict the cascade size. TempCas [41] applies BiGRU and CNN to combine cascade information and temporal information for forwarding prediction. CasFlow [42] uses a hierarchical variational information diffusion model to capture node-level and cascade-level uncertainty and to learn cascade distributions through variational inference and normalized flow. CCasGNN [43] used a combined framework of GAT and GCN, which simultaneously considered user information, structural features, and temporal features, improving the accuracy of cascade prediction. AECasN [44] applied an autoencoder to handle cascade graphs to improve learning time and prediction accuracy. These methods fully consider structural and temporal information to make predictions. Still, in existing social platforms, forwarding cascades are often dynamic, and these types of methods only apply to static cascade networks and are not ideal for dynamic cascade networks. CasSeqGCN [45] proposed transforming the cascade graph into a cascade snapshot input model and using graph convolutional network and LSTM to learn structural and temporal features for information cascade prediction. The method proposed is to be applied for dynamic cascade network prediction, but it needs to function better when facing complex social media data.

3. CAC-G Network

3.1. Problem Definition

Information cascade prediction modeling is a relatively complex process. To facilitate further elaboration, definitions and explanations of some concepts will be given in this section, followed by the presentation of the proposed model. The relevant symbols are shown in Table 1.

Table 1. Descriptions of key notations.

Notations	Descriptions
G	Static social network
V	Users/nodes
E	Edges
C	Cascade graphs
D_i^T	The state of the node
S_i^T	Cascade snapshot
ΔV_T^i	Growth scale of cascade network
p	Increment
e_{v_i, v_j}	The importance of nodes V_j to nodes V_i
N	Number of nodes
F	The feature dimension of the node
W	Weights
\vec{h}	Node eigenvalues
a	Attention mechanism
α	Importance factor
K	Number of attentions
\vec{h}	Output of the GAT layer
out	Output of CNN layer
$channel$	Number of channels
B	Output of the CAC layer
U	Affine-transformed

Table 1. Cont.

Notations	Descriptions
c_{ij}	Weight coefficient
v_j	Output of Dynamic routing-AT
r_t	Reset gate
z_t	Update gate
h_t	Hidden state
y^t	Output of the GRU layer
T_d	Observation time

3.1.1. Cascade Graph

Let $G = \{V, E\}$ be a static social network, where V denotes the set of users and E denotes the set of edges. Each message i propagated in the network forms an information cascade. Subgraphs $C_i = \{V_i, E_i\}$ are defined as cascade graphs, where V_i denotes the set of nodes through which the message passes, and E_i denotes the set of edges connections V_i . At time t , if a message passes through the node, the node's state is set to 1. Otherwise, it is 0. The node state is denoted by $D_i^T(t)$, and it captures the cascaded subgraph C_i^T of the node at time t , where T is the observation time window.

3.1.2. Cascade Snapshot

As shown in Figure 1, only one node in the uppermost picture in this figure is marked yellow, which proves that the message has just been sent out at this time, and the state vectors $D_i^T(t)$ of other nodes except for the marked yellow node are all 0. In the middle picture, except for the initial, there are three nodes marked yellow on the outside of the node, proving that three users have forwarded this message at this time. The cascaded subgraph C_i^T only includes these four nodes at this time, the state vector of these four nodes is 1, and the rest node is 0. The bottom graph indicates that all nodes participate in forwarding, the cascaded subgraph is the entire graph, and the state vectors of all nodes are 1. The captured node state diagram C_i^T and node state D_i^T are combined to build snapshot $S_i^T = \{V_i^T, E_i^T, D_i^T(t)\}$, which contains the structure of the cascade graph and the node state at time t .

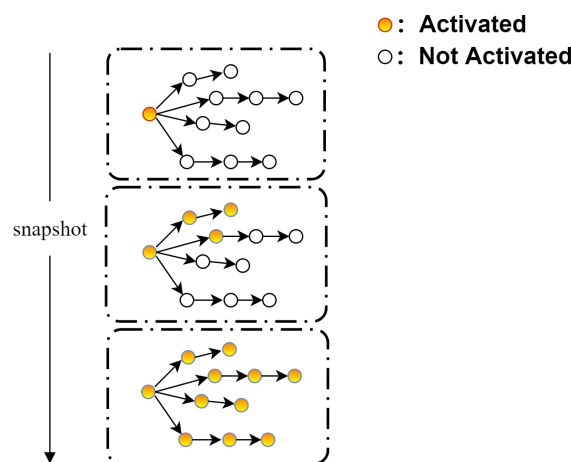


Figure 1. Snapshot.

3.1.3. Growth Scale of Cascade Network

The prediction of message popularity can be understood as the prediction of the growth scale of the information cascade. The more nodes the message reaches, the more times it is forwarded and the larger the cascade size (Figure 2). We predict the cascade growth scale ΔV_i^T of each cascade graph C_i in the next period by setting the observation time to determine the popularity of the news corresponding to the cascade.

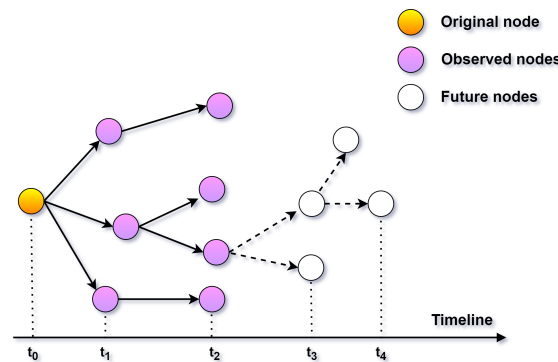


Figure 2. An example of cascade network (the yellow nodes in the figure are the source nodes; the purple nodes are the observation cascades of the input model; and the white nodes are the scale of the future diffusion of the cascades, which is also the cascade growth scale ΔV_i^T to be predicted by the model).

3.2. Network

The CAC-G network is presented in this sub-section, as shown in Figure 3. CAC layers combine GAT and CNN to stress multi-scale information extraction. Dynamic routing-AT is used for aggregation to generate snapshot vector representation, which is sent to GRU to learn hidden time information and finally sent to MLP to output cascade growth size.

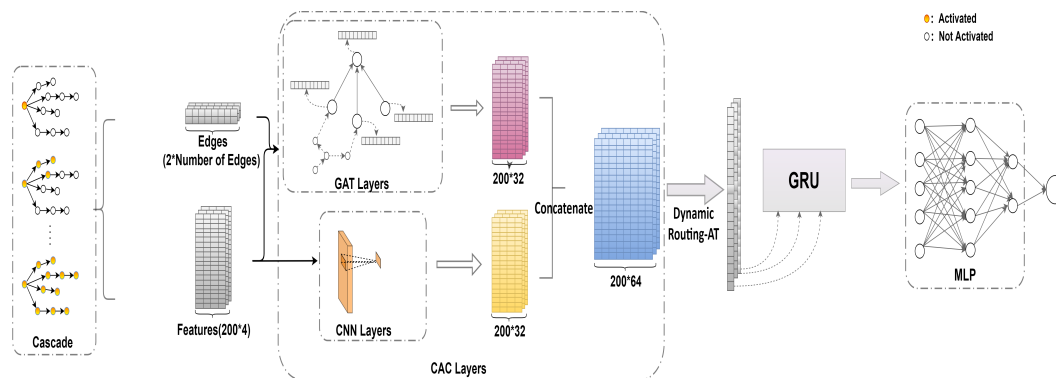


Figure 3. The framework of CAC-G.

3.2.1. Cascade Snapshot Sequence Generation

The model cannot directly use a cascade network as a complex structure. The input of our proposal is a cascade snapshot sequence. First, the cascade network is converted into the cascade graph. When the state of each node is set to 1, a message passes through this node to generate a new snapshot S_i^T . Combined snapshots are used to generate a cascade sequence of snapshots $\{S_i^T(t_0), S_i^T(t_1) \dots S_i^T(t_n)\}$. However, it will take a long time to process each snapshot. It is mentioned in CasSeqGCN [40] that a partial sampling strategy can be adopted. Therefore, the snapshot sequence is obtained by increment p . We tested from $p = 1$ to $p = 5$, respectively, and found that the prediction accuracy was reduced by about 20%, and the time-consuming was reduced by five times. Therefore, under careful consideration, we choose $p = 3$. When $p = 3$, the sequence is the most representative, and the model's prediction performance is best. After partial sampling, each cascade produces $1 + \lceil (|V_i^T| - 1) / p \rceil$ snapshots, where $|V_i^T|$ represents the modulus of V_i^T , which can also be understood as the size of V_i^T . After that, the node information and edge information in each snapshot are extracted, and a two-dimensional matrix is constructed, as shown in Figure 4.

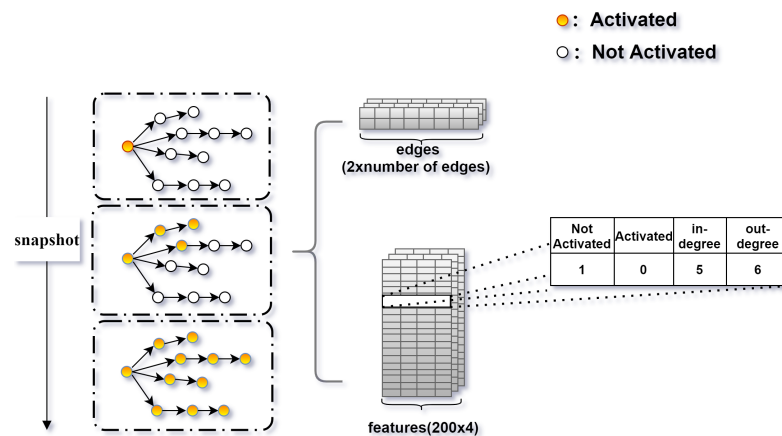


Figure 4. Characteristic matrix.

3.2.2. Cascade Attention Convolutional Networks (CAC)

After processing the accurate social platform data, it is found that the number of cascade graph edges is minimal or the repetition rate is high, which causes many cascades to be challenging to map. Therefore, this paper proposes CAC to process the cascade graph. CAC is an organic combination of GAT and CNN. The reason for this design is that due to the complexity of the social platform cascade network, the neighbor nodes of many nodes need to be clarified, and the attention mechanism of GAT is challenging to play a vital role. It will lead to the information loss of original nodes. In this context, CNN is introduced into the model to extract node information and splice it with the node information extracted by GAT. Additionally, it can recover the partial information lost caused by GAT.

First of all, we need to introduce the self-attention mechanism a . As we all know, Q , K , and V must be set separately for query, key, and value, respectively. The specific formula of the self-attention mechanism a is

$$a = \frac{QK^T}{\sqrt{d_K}} \tag{1}$$

where T represents the transpose and d_K represents the dimension of K .

The input of GAT is edge and node features. Each row of the node feature matrix $V \in R^{N \times F}$ represents a piece of node information, where N denotes the number of nodes, F represents the feature dimension of the node, and GAT first performs a shared linear transformation on each node, parameterized with a weight matrix W , and then applies a self-attention mechanism a to calculate the importance $e_{V_i V_j}$ of node V_j to node V_i . The formula is as follows:

$$e_{V_i V_j} = a(W \vec{h}_{V_i}, W \vec{h}_{V_j}), \tag{2}$$

where \vec{h} is the feature value of the node. Then, *LeakyReLU* is used for nonlinearization, and finally, we use softmax to normalize the neighbor nodes of the central node, where \top is the transpose operation worth the vector, as shown in Figure 5. The specific implementation formula is:

$$\alpha_{V_i V_j} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{a}^\top \left[W \vec{h}_{V_i} \parallel W \vec{h}_{V_j} \right]\right)\right)}{\sum_{k \in N_i} \exp\left(\text{LeakyReLU}\left(\vec{a}^\top \left[W \vec{h}_{V_i} \parallel W \vec{h}_{V_k} \right]\right)\right)} \tag{3}$$

In this network, using multi-head GAT, the formula under multi-head attention is expressed as follows, where \parallel represents concatenation, N_i represents the neighborhood of node i in the cascade graph, and σ denotes the sigmoid function.

$$\vec{h}_{V_i} = \parallel_{k=1}^K \sigma \left(\sum_{V_j \in N_i} \alpha_{V_i V_j}^k W^k \vec{h}_{V_j} \right) \tag{4}$$

The input of CNN is the node feature, which is output after a layer of conv1d. The specific formula is expressed as:

$$out = bias + \sum_{i=1}^{channel} weight * input \tag{5}$$

Weight is a one-dimensional convolution kernel; the channel denotes the number of channels. After obtaining the GAT layer output and the CNN layer output, they are spliced to obtain the CAC output B. The specific formula is implemented as follows:

$$B = \left[\begin{matrix} \vec{h} \\ ||out \end{matrix} \right] \tag{6}$$

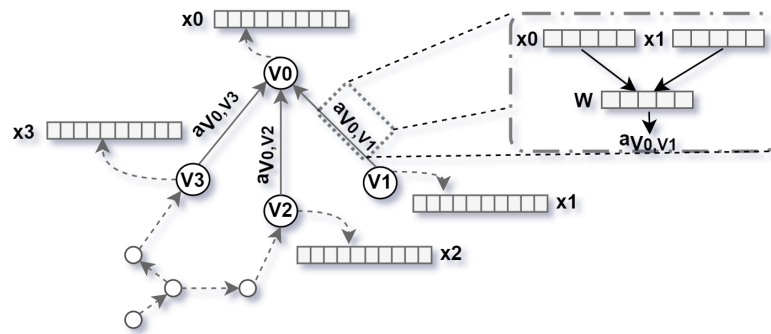


Figure 5. Node information extraction (each node in the cascade graph will enter the GAT model as a vector. The attention coefficient between the node V_i and the neighbor node V_j will be obtained through the attention mechanism, and the vector representation of the cascade graph will be output after processing. Among them, V_0 - V_3 represents the node, and X_0 - X_3 depicts the vector representation of the corresponding node).

3.2.3. Cascade Snapshot Expression

The output of CAC is the feature of each node. To better use node features and hidden temporal features, we aggregate the nodes of each snapshot as the feature representation of the snapshot. CasSeqGCN [45] has shown that the Dynamic routing algorithm in the capsule network can be used for aggregation. Still, the weight update by Dynamic routing requires multiple iterations to achieve better results followed by specific time loss. Inspired by the weight calculation method of self-attention, we combine attention with Dynamic routing to raise Dynamic routing-AT.

A dot product is carried out between the affine-transformed vector u_i of the user i and the Dynamic routing output v_j to obtain the weight coefficient of the user i . The input of Dynamic routing is the output from CAC, that is, the node feature matrix B . Firstly, a linear affine transformation is performed on the node representation vector, i.e.,

$$U = W_d B \tag{7}$$

where W_d denotes the mapping matrix. The weight coefficient c_{ij} is calculated as:

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}, b_{ij} = \text{soft max} \left(\frac{u_i \cdot v_{j-1}}{\sqrt{d} v_{j-1}} \right) \tag{8}$$

When the influence of the node representation in the snapshot representation is greater, the weight coefficient assigned to the i -th node is higher, and j represents the j -th iteration. The specific formula for the output of Dynamic routing-AT is as follows:

$$v_j = \sum_i c_{ij}u_i \tag{9}$$

3.2.4. Time Information Processing

The time effect in the prediction of message popularity cannot be ignored. For example, if a Weibo message is widely viewed and forwarded immediately after sending, the probability of the message becoming a hot topic will be very high. In this model, the time information is hidden in the sequence of cascade snapshots, and we apply GRU to process the temporal information. As shown in Figure 6, the cascaded snapshot sequence output by Dynamic routing-AT to GRU for processing is fed. The specific formula is:

$$r_t = \sigma(W_r V^t + U_r h_{t-1} + b_r) \tag{10}$$

and

$$z_t = \sigma(W_z V^t + U_z h_{t-1} + b_z) \tag{11}$$

where r_t denotes the reset gate, z_t denotes the update gate, σ denotes the sigmoid function, h_{t-1} denotes the hidden layer state before time t , and W and U are the weight matrices. The hidden state calculated by the reset gate \tilde{h} is:

$$\tilde{h}_t = \tan h(W_h V^t + U_h(r_t \otimes h_{t-1}) + b_h) \tag{12}$$

where \otimes denotes the multiplication of corresponding elements in the operation matrix. GRU selectively remembers the information containing the current node by selectively forgetting the original hidden state and then sums it up to obtain the output y^t and hidden state h_t , i.e.,

$$y^t = h_t = z_t \otimes h_{t-1} + (1 - z_t) \otimes \tilde{h}_t \tag{13}$$

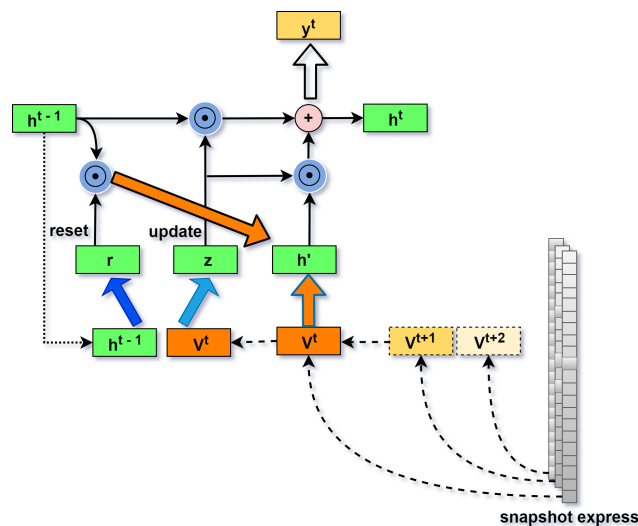


Figure 6. Cascaded snapshot sequences are sent to GRU for processing.

3.2.5. Growth Scale Prediction

The last module of CAC-G is the prediction module, which takes the output y^t of GRU as the input of MLP, and the output of MLP is the cascade growth scale, that is,

$$\Delta \hat{V}_i^{T_d} = MLP(y_i^t) \tag{14}$$

where $\Delta \hat{V}_i^{T_d}$ denotes the predicted cascade growth scale, and T_d denotes the observation time.

4. Experiments

4.1. Datasets

To evaluate the effectiveness of CAC-G, we conduct experiments on two public real-world datasets. The first dataset is the Sina Weibo dataset, one of the largest social platforms in the world. The second dataset is the DBLP paper citation dataset, which predicts the number of each article's citations. The data statistics are shown in Table 2.

The Sina Weibo dataset is provided by Zhang et al. [46]. The dataset is captured from Sina Weibo, which includes the following relationships and forwarding links between users. When user A delivers the message to user B, A is said to be a follower of B, and there will be a forwarding link from B to A. The data contain 300,000 famous information dissemination cascades. We adopt a data-preprocessing method similar to Wang et al. [45], eliminating the forwarding cascades fewer than ten times. This means that the message will not be propagated if it has not been propagated within 12 h. We construct three sub-datasets to predict the scale of message dissemination in the last 9 h, 12 h, and 24 h, respectively. Based on that, 70%, 10%, and 20% of each sub-dataset are randomly selected as the model's training, validation, and test data, respectively.

In the paper citation scenario, the DBLP paper citation dataset [47] is applied by us. The process of paper citation is consistent with the underlying operating mechanism of the message-forwarding process. When paper B is cited by A, there will also be a transmission link from B to A while recording the number of days since the paper was published. The popularity prediction of the article can be converted into the size of the citation cascade. In a certain period, the greater the number of citations of the article, the stronger the popularity of the paper. We also adopt a data-preprocessing method similar to Wang et al. [45], eliminate the cascades of less than 10, and believe that a paper that has not been cited for three years will not generate a new cascade. Like the Sina Weibo dataset, three sub-datasets are constructed, predicting the citation scale of papers in the last 1, 2, and 3 years, respectively. On this basis, 70%, 10%, and 20% of each sub-dataset are randomly selected as the model's training, validation, and test data, respectively.

Unlike previous data partitioning, methods such as DeepHawkes [37] fix the observation time window and predict the cascade diffusion scale afterward. During the 9 h, 12 h, and 24 h for the Weibo dataset and the 1, 2, and 3 years for the DBLP paper citation dataset, we began to refer to the prediction period. They are different from the fixed observation time window setting. This treatment aims to enable dynamic changes in the observation time window closer to the actual usage scenario.

Table 2. Statistics of the datasets.

Dataset	Weibo			DBLP		
	9 h	12 h	24 h	1 year	2 years	3 years
Number of cascades	29,123	29,122	34,897	30,106	29,998	29,991
Train	20,386	20,385	24,428	21,074	20,999	20,994
Val	2912	2912	3490	3011	3000	2999
Test	5825	5825	6979	6021	5999	5998
Avg. observed nodes	39.005	38.018	26.977	32.008	31.665	31.226
Avg. observed edges	36.254	37.323	37.444	60.009	58.556	57.013
Avg. growth size	4.874	6.999	20.616	1.965	2.101	8.578

4.2. Baseline Methods

Three classes of cascade prediction methods in Section 2 are introduced by us, namely, feature-based methods, generative methods, and deep learning-based methods. We will

select several representative approaches from these methods as baseline methods and compare our method with them, which is further elaborated on below:

Feature-linear and feature-deep are feature-based methods. We manually extract the average in-degree and out-degree, number of nodes, leaf nodes, edges, and average activation time of nodes as cascade features. The extracted features are then fed into the model for prediction. Feature-linear applies linear regression to fit the scale of cascade growth, and feature-deep uses a two-layer fully-connected neural network to predict the scale of cascade growth.

DeepCas [16] is the first end-to-end model based on deep learning. The model obtains a set of node sequences through random walks. The model mainly adopts the cascade graph's structural and node information and passes the bidirectional GRU. At the same time, an attention mechanism is applied to predict the growth scale of the cascade.

DeepHawkes [37] proposes to convert the cascade graph into a forwarding path, use RNN for processing, and then combine the self-excitation process of the Hawkes model to perform cascade prediction. At the same time, the cascade graph's structural and time information are employed to improve the prediction efficiency and accuracy.

CasCN [39] proposes to divide the cascade graph into the form of cascade subgraphs; apply a graph convolutional neural network (GCN) to learn the subgraph representation, which is then fed into the LSTM to capture the cascade structure evolution; use a self-excitation mechanism and temporal decay mechanism, which can effectively capture both the subgraph structure and temporal features; and finally predict the scale of cascade growth.

AECasN [44] proposes to divide the information cascade network into different layers; obtain the initial representation of the cascade network; and then multiply the representation vector with the discrete vector of the time decay effect, input it into the auto-encoder for learning, and output the cascade growth scale. The model utilizes structural features and temporal features to improve prediction efficiency.

CasSeqGCN [45] seeks to divide the cascade graph into the form of cascade snapshots, use GCN to learn the node features, aggregate the cascade information and perform LSTM to know the time information, and finally output the cascade-prediction scale. The similarity between our model and CasSeqGCN is that the cascade graph is also divided into cascade snapshots, and the difference is that CasSeqGCN only applies GCN to consider the relationship between adjacent edges. It is challenging to extract deep node information and capture global details effectively. The CAC model proposed by us can solve this problem very well.

4.3. Experimental Setup

According to the existing research, we choose MSLE as the evaluation metric for the cascade prediction problem, which is convenient for us to compare with other baseline methods. The final state of the predicted cascade is reflected in the MSLE loss function, and each loss value output by MSLE is the result of comparing the predicted cascade size from the model output with the true size. The specific definition is as follows:

$$MSLE = \frac{1}{N} \sum_{i=1}^N \left(\log_2 \Delta \hat{V}_i^{T_d} - \log_2 \Delta V_i^{T_d} \right)^2 \quad (15)$$

where N denotes the total number of cascades, $\Delta V_i^{T_d}$ denotes the true growth scale of cascades, $\Delta \hat{V}_i^{T_d}$ denotes the predicted cascade growth scale, and T_d denotes the observation time.

Our code runs on a Windows Server 2019 server with 60 G memory, the CPU is Intel Xeon Processor (Icelake) 2.59 GHz manufactured by Intel Corporation, and the GPU uses NVIDIA A100-SXM4 manufactured by NVIDIA Corporation. Our code runs on a Windows Server 2019 server with 60G memory, the CPU is Intel Xeon Processor (Icelake)

Manufactured by Intel Corporation 2.59 GHz, and the GPU uses NVIDIA A100-SXM4 manufactured by NVIDIA Corporation.

Some hyperparameters and specific parameter settings will be given below in our method. Our model learning rate and batch size are 0.001 and 100, respectively. For CAC, there are two parts, GAT and CNN, which only have one layer, and then a four-head attention mechanism is applied in GAT. The output dimension of GAT is 32, the convolution kernel size of CNN is 7, the snapshot vector dimension is 64, and the input and output dimensions of Dynamic routing-AT and GRU are 64. GRU only sets one layer, and the dropout is 0.4. The input dimension of the MLP layer is 64, the output dimension is 1, and the dropout of the MLP layer is 0.5. The iteration number r of Dynamic routing-AT is 1.

The parameters of other baseline models are set as follows for MLP. On the one hand, the hidden layer is formed to 2, and the remote layer activation function is sigmoid. On the other hand, the dropout is selected as 0.5. For DeepCas and DeepHawkes, the node-embedding is set to 50, the learning rate is 0.0005, each GRU contains 32 hidden units, and the time interval is set to 3 h in the Weibo dataset and 1 year in the DBLP dataset. Other parameters are consistent with those used in related papers.

5. Results and Discussion

5.1. Performance Comparison

This paper designs and implements a dynamic end-to-end information cascade prediction model based on deep learning. To make it easier for readers to understand the working principle of CAC-G, specific scenarios are introduced for an explanation. For example, user A on Weibo publishes new content, and then user B forwards it. Then, a forwarding path is formed from A to B, and as more and more other users forward, a cascade is included in the network. User A can be understood as the originating node in the cascaded network, and the forwarding path can be understood as an edge in the cascaded network. This paper studies the transformation of the cascaded network into multiple snapshots, forming a snapshot every time a message passes through a node, which can also be called activating a node. Afterward, each snapshot is sent to CAC to obtain the feature representation of the snapshot. Since the feature representation output by CAC has a huge amount of information, the research uses Dynamic routing-AT to aggregate vector information to increase computing efficiency. At the same time, the time information is hidden in the snapshot sequence; the information output by Dynamic routing-AT is sent to the GRU for processing; and finally, the MLP is used to predict the growth scale of the cascade. The popularity of information depends on the size of the cascade scale. The larger the cascade scale, the more times the information is forwarded, and the higher the popularity of the information.

We evaluate our proposed model on the Weibo dataset and DBLP dataset. In addition, the model's predictive performance is affected by many potential factors. When we tried to reduce the data volume of the Weibo dataset to 1000 cascades; the model had already entered the overfitting state when it was trained to the 10th epoch. Therefore, we applied a dataset with over 29,000 cascades to prevent overfitting. In addition, when we use two-layer GAT network, the model cannot be fully fitted, and the prediction performance is greatly reduced. The model can fully play its performance when we only use a single-layer GAT network. In addition, the settings of hyperparameters, such as learning rate and batch-size, will also greatly impact the model's predictive ability. The most appropriate hyperparameter settings are chosen after multiple experiments. The specific comparison between CAC-G and other baseline methods is shown in Table 3. Experiments show that CAC-G is better than the baseline mentioned in this paper. For the Weibo forwarding dataset, CAC-G is about 1% higher than the existing best baseline methods. For the DBLP paper forwarding dataset, the prediction accuracy has increased at least by about 2%. To thoroughly verify the model's effectiveness proposed in this paper, we also compared CAC-G with the baseline method on the precision indicator. As shown in Table 4, since the Weibo forwarding data set is the most typical social network cascading data, we only use

the Weibo data-focused comparisons. The results demonstrate that CAC-G outperforms existing baseline methods.

Considering the problem of forgetting mentioned in *cicp* [48], we use a variable observation time window. Through our comparative experiments, it is found that the unfixed observation time window makes the prediction task more accessible in the baseline methods. At the same time, the prediction effect of feature-linear and feature-deep methods is better than some methods based on deep learning but not as good as the method we proposed. Therefore, manually extracted features also have an important influence on prediction. However, these features are only for specific scenarios, and it is not easy to consider the cascade's structural information. For DeepCas [16], random wandering is applied to obtain node representations, which fails to consider the integrity of temporal and spatial structure information, thus producing limited prediction. Regarding DeepHawkes [37], although it feels user-friendly and combines the Hawkes process with deep learning, it needs to be revised to extract spatial structure information for actual propagation cascades. It is challenging to cope with complex information cascades. That is why the method is sometimes worse than feature-based methods. CasCN [39] proposes using cascade subgraphs as a structural and temporal information combination. However, the method needs to pay more attention to integrating the two, limiting the use of temporal information. AECasN [44] applied an autoencoder to the cascade graph, which learns the temporal information but does not sufficiently consider the network structure. In doing so, it makes the prediction effect limited. The experimental results show that the above methods give comparable prediction results for dynamic cascade prediction. Although the above methods have promising results in static cascade networks, they do not perform well for dynamic cascade networks and do not apply to dynamic cascade networks. CasSeqGCN [45] proposes to divide the cascade graph into cascade snapshots for dealing with dynamic cascades, taking into account structural and temporal information. However, for real complex cascades, only GCN is aimed at considering the relationship between neighboring edges, which makes it challenging to extract deep node information and capture global information effectively. After fully considering the shortcomings of the above methods, we propose CAC-G. Our method has significantly improved the speed and accuracy of cascade prediction compared to previous methods. At the same time, CAC-G has shown excellent performance in handling dynamic cascades.

Table 3. Comparative experimental results in two datasets under different scenarios (MSLE).

Dataset	Weibo			DBLP		
	9 h	12 h	24 h	1 year	2 years	3 years
Dataset selection	Test set					
Feature Linear	1.047	1.196	1.726	0.367	0.814	0.886
Feature Deep	0.982	1.187	1.635	0.310	0.666	0.865
DeepCas	0.979	1.185	1.534	0.355	0.721	0.874
DeepHawkes	0.983	1.190	1.550	0.520	0.787	0.929
CasCN	0.980	1.181	1.522	0.323	0.597	0.734
AECasN	0.980	1.190	1.545	0.362	0.718	0.853
CasSeqGCN	0.475	0.611	0.964	0.158	0.335	0.357
CAC-G	0.469	0.607	0.958	0.138	0.301	0.348

Table 4. Comparative experimental results in Weibo dataset (precision).

Dataset	Weibo		
	9 h	12 h	24 h
Dataset selection/Evaluation metric	Test set/Precision (%)		
Feature Deep	55.10	54.56	52.12
DeepCas	68.95	68.02	65.86
DeepHawkes	68.53	67.91	65.56
CasCN	69.25	68.24	65.73
AECasN	69.35	68.92	66.02
CasSeqGCN	69.92	69.53	66.55
CAC-G	70.31	69.90	66.85

5.2. Ablation Experiments

In our framework, the structural information of the cascade graph is learned by our proposed CAC model, which is an organic combination of GAT and CNN. To prove the contributions of these two parts, we delete CNN and only keep the GAT design of the variant experiment CAC-G-noCNN. The experimental results are shown in Table 5. It can be seen that the prediction effect of CAC-G-noCNN is worse than CAC-G on both the Weibo dataset and the paper forwarding dataset. Therefore, the components of CAC layers are more suitable for complex cascade learning. Attention to multi-scale information can reduce information loss, and prediction accuracy can be further improved. At the same time, we also compared CAC-G-noCNN with the previous baseline methods and found that when the model only has GAT, the prediction effect has little effect on the Weibo dataset. However, in the DBLP paper citation dataset, the prediction performance is better than that of the baseline method, which also proves the effectiveness of GAT in this framework.

Table 5. Ablation experimental results in two datasets under different scenarios (MSLE).

Dataset	Weibo			DBLP		
	9 h	12 h	24 h	1 Year	2 Years	3 Years
Dataset selection	Test set					
CasSeqGCN	0.475	0.611	0.964	0.158	0.335	0.357
CAC-G-noCNN	0.474	0.614	0.965	0.145	0.312	0.354
CAC-G	0.469	0.607	0.958	0.138	0.301	0.348

Dynamic routing-AT is used in the node vector aggregation part. Before that, Dynamic routing was applied for vector aggregation and achieved good results, but this method requires multiple iterations featuring unstable prediction. An aggregation method Dynamic routing-AT is presented that integrates self-attention. Dynamic routing-AT replaces the previous method, and the variant experiment CAC-G-Dynamic routing is performed. The results are shown in Table 6. The prediction effect of CAC-G is more stable and better than that of CAC-G-Dynamic routing. Meanwhile, we also compare the average time spent training an epoch on each dataset by these two methods. According to Figures 7 and 8, it can be seen that the time consumption of CAC-G is much lower than that of CAC-G-Dynamic routing.

Table 6. Ablation experimental results in two datasets under different scenarios (MSLE).

Dataset	Weibo			DBLP		
	9 h	12 h	24 h	1 Year	2 Years	3 Years
Dataset selection	Test set					
CAC-G-Dynamic routing	0.470	0.611	0.964	0.145	0.312	0.355
CAC-G	0.469	0.607	0.958	0.138	0.301	0.348

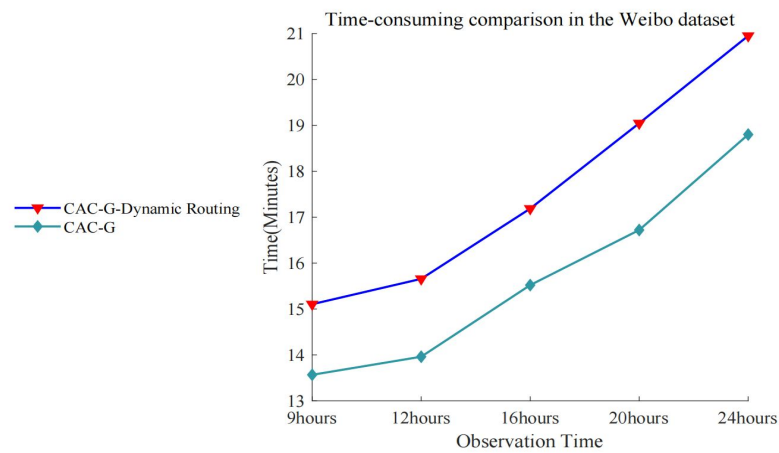


Figure 7. Average time consumption (Weibo).

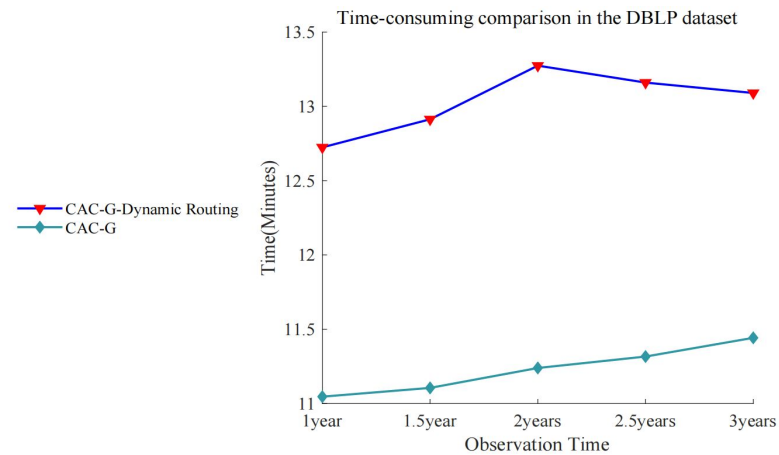


Figure 8. Average time consumption (DBLP).

6. Conclusions

An end-to-end deep learning framework CAC-G is proposed in this paper. In this framework, a model CAC for complex cascade processing is creatively raised considering the global information of the cascade. Moreover, Dynamic routing-AT, a vector aggregation method, is also innovated, improving prediction stability and time efficiency. In terms of predictive performance, on the one hand, low data volume may lead to overfitting problems. At the same time, an overly complex model will also reduce predictive performance. On the other hand, the model’s predictive ability will also be affected by hardware limitations, system environment, and software implementation. We thoroughly considered and avoided these potential factors affecting the model’s prediction performance. CAC-G achieved better results than other baselines on both the Weibo and DBLP paper citation datasets. Therefore, this study can inspire researchers to design information cascade prediction models and provide new solutions to the popularity prediction problem.

The proposal of CAC-G also meets the demand for dynamic capture and fast and accurate prediction of public opinion hotspot information, which can help the government

or platform accurately control the direction of public opinion and avoid adverse effects. In addition, the cascade forecasting method can also be extended to the prediction of epidemic spread. According to the scale and speed of the initial outbreak applied to predict the growth scale in a certain period, this will be a beneficial study for society. Furthermore, it can help the government respond in advance and protect the safety of people's lives and property.

However, CAC-G can only macroscopically predict the diffusion scale of the cascade, making it challenging to achieve microscopic-level accurate prediction to the following forwarding individuals. In addition, the model also relies on large-scale data for training. Future research should focus on better combining macroscopic and microscopic prediction and explore whether meta-learning methods can be combined with cascade prediction to enable small-scale data training.

Author Contributions: Conceptualization, T.M.; methodology, T.M. and D.H.; software, M.L.; validation, T.M.; formal analysis, M.L.; investigation, D.H.; resources, D.H.; data curation, D.H. and T.M.; writing—original draft preparation, T.M.; writing—review and editing, M.L.; visualization, D.H.; supervision, M.L.; project administration, D.H. and T.M.; and funding acquisition, D.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Key R&D Program of Shandong Province, China (2022CXGC020106) and National Key Research and Development Program of China (2022YFB4004401).

Data Availability Statement: Datasets can be accessed upon request to the corresponding author.

Acknowledgments: The authors would like to thank all of the anonymous reviewers for their insightful comments and constructive suggestions.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Ta, N.; Li, K.; Yang, Y.; Jiao, F.; Tang, Z.; Li, G. Evaluating public anxiety for topic-based communities in social networks. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 1191–1205. [[CrossRef](#)]
2. Zhou, F.; Xu, X.; Trajcevski, G.; Zhang, K. A survey of information cascade analysis: Models, predictions, and recent advances. *ACM Comput. Surv.* **2021**, *54*, 27. [[CrossRef](#)]
3. Liu, B.; Yang, D.; Shi, Y.; Wang, Y. Improving Information Cascade Modeling by Social Topology and Dual Role User Dependency. In *Database Systems for Advanced Applications, Proceedings of the 27th International Conference (DASFAA 2022), Virtual Event, 11–14 April 2022*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 425–440.
4. Robles, J.F.; Chica, M.; Cordon, O. Evolutionary multiobjective optimization to target social network influentials in viral marketing. *Expert Syst. Appl.* **2020**, *147*, 113183. [[CrossRef](#)]
5. Wu, Q.; Gao, Y.; Gao, X.; Weng, P.; Chen, G. Dual sequential prediction models linking sequential recommendation and information dissemination. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019*; pp. 447–457.
6. Zhao, L.; Chen, J.; Chen, F.; Jin, F.; Wang, W.; Lu, C.T.; Ramakrishnan, N. Online flu epidemiological deep modeling on disease contact network. *Geoinformatica* **2020**, *24*, 443–475. [[CrossRef](#)]
7. Szabo, G.; Huberman, B.A. Predicting the popularity of online content. *Commun. ACM* **2010**, *53*, 80–88. [[CrossRef](#)]
8. Shulman, B.; Sharma, A.; Cosley, D. Predictability of popularity: Gaps between prediction and understanding. In *Proceedings of the International AAAI Conference on Web and Social Media, Cologne, Germany, 17–20 May 2016*; Volume 10, pp. 348–357.
9. Shen, H.; Wang, D.; Song, C.; Barabási, A.L. Modeling and predicting popularity dynamics via reinforced poisson processes. In *Proceedings of the AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014*; Volume 28.
10. Zhang, X.; Aravamudan, A.; Anagnostopoulos, G.C. Anytime Information Cascade Popularity Prediction via Self-Exciting Processes. In *Proceedings of the International Conference on Machine Learning (PMLR), Baltimore, MD, USA, 17–23 July 2022*; pp. 26028–26047.
11. Tan, W.H.; Chen, F. Predicting the popularity of tweets using internal and external knowledge: An empirical Bayes type approach. *Adv. Stat. Anal.* **2021**, *105*, 335–352. [[CrossRef](#)]
12. Ling, C.; Tong, G.; Chen, M. Nestpp: Modeling thread dynamics in online discussion forums. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media, Online, 13–15 July 2020*; pp. 251–260.
13. Ebiaredoh-Mienye, S.A.; Esenogho, E.; Swart, T.G. Artificial neural network technique for improving prediction of credit card default: A stacked sparse autoencoder approach. *Int. J. Electr. Comput. Eng.* **2021**, *11*, 4392. [[CrossRef](#)]

14. Obaido, G.; Ogbuokiri, B.; Swart, T.G.; Ayawei, N.; Kasongo, S.M.; Aruleba, K.; Mienye, I.D.; Aruleba, I.; Chukwu, W.; Osaye, F.; et al. An interpretable machine learning approach for hepatitis b diagnosis. *Appl. Sci.* **2022**, *12*, 11127. [[CrossRef](#)]
15. Ebiaredoh-Mienye, S.A.; Swart, T.G.; Esenogho, E.; Mienye, I.D. A machine learning method with filter-based feature selection for improved prediction of chronic kidney disease. *Bioengineering* **2022**, *9*, 350. [[CrossRef](#)]
16. Li, C.; Ma, J.; Guo, X.; Mei, Q. Deepcas: An end-to-end predictor of information cascades. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 577–586.
17. Liang, F.; Qian, C.; Yu, W.; Griffith, D.; Golmie, N. Survey of graph neural networks and applications. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9261537. [[CrossRef](#)]
18. Xu, M.; Liu, H. Road Travel Time Prediction Based on Improved Graph Convolutional Network. *Mob. Inf. Syst.* **2021**, *2021*, 7161293. [[CrossRef](#)]
19. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
20. Kim, P. *Matlab Deep Learning with Machine Learning, Neural Networks and Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2017.
21. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic routing between capsules. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3859–3869.
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.
23. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
24. Tolstikhin, I.O.; Housley, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. Mlp-mixer: An all-mlp architecture for vision. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 24261–24272.
25. Feng, X.; Zhao, Q.; Liu, Z. Prediction of information cascades via content and structure proximity preserved graph level embedding. *Inf. Sci.* **2021**, *560*, 424–440. [[CrossRef](#)]
26. Ducci, F.; Kraus, M.; Feuerriegel, S. Cascade-LSTM: A tree-structured neural classifier for detecting misinformation cascades. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020; pp. 2666–2676.
27. Horawalavithana, S.; Skvoretz, J.; Iamnitchi, A. Cascade-LSTM: Predicting information cascades using deep neural networks. *arXiv* **2020**, arXiv:2004.12373
28. Bakshy, E.; Hofman, J.M.; Mason, W.A.; Watts, D.J. Everyone’s an influencer: Quantifying influence on twitter. In Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, Hong Kong, China, 9–12 February 2011; pp. 65–74.
29. Tsur, O.; Rappoport, A. 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, Seattle, WA, USA, 8–12 February 2012; pp. 643–652.
30. Saha, A.; Ganguly, N. A gan-based framework for modeling hashtag popularity dynamics using assistive information. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, Ireland, 19–23 October 2020; pp. 1335–1344.
31. Cui, L.; Hawkes, A.; Yi, H. An elementary derivation of moments of Hawkes processes. *Adv. Appl. Probab.* **2020**, *52*, 102–137. [[CrossRef](#)]
32. Gomez-Rodriguez, M.; Leskovec, J.; Schölkopf, B. Modeling information propagation with survival theory. In Proceedings of the International Conference on Machine Learning (PMLR), Atlanta, GA, USA, 17–19 June 2013; pp. 666–674.
33. Zaman, T.; Fox, E.B.; Bradlow, E.T. A bayesian approach for predicting the popularity of tweets. *Ann. Appl. Stat.* **2014**, *8*, 1583–1611. [[CrossRef](#)]
34. Gao, X.; Cao, Z.; Li, S.; Yao, B.; Chen, G.; Tang, S. Taxonomy and evaluation for microblog popularity prediction. *ACM Trans. Knowl. Discov. Data* **2019**, *13*, 15. [[CrossRef](#)]
35. Sun, X.; Zhou, J.; Liu, L.; Wei, W. Explicit time embedding based cascade attention network for information popularity prediction. *Inf. Process. Manag.* **2023**, *60*, 103278. [[CrossRef](#)]
36. Zeng, Y.; Xiang, K. Persistence Augmented Graph Convolution Network for Information Popularity Prediction. *IEEE Trans. Netw. Sci. Eng.* **2023**. [[CrossRef](#)]
37. Cao, Q.; Shen, H.; Cen, K.; Ouyang, W.; Cheng, X. Deephawkes: Bridging the gap between prediction and understanding of information cascades. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1149–1158.
38. Wang, J.; Zheng, V.W.; Liu, Z.; Chang, K.C.C. Topological recurrent neural network for diffusion prediction. In Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM), Orleans, LA, USA, 18–21 November 2017; pp. 475–484.
39. Chen, X.; Zhou, F.; Zhang, K.; Trajcevski, G.; Zhong, T.; Zhang, F. Information diffusion prediction via recurrent cascades convolution. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, China, 8–11 April 2019; pp. 770–781.
40. Cao, Q.; Shen, H.; Gao, J.; Wei, B.; Cheng, X. Popularity prediction on social platforms with coupled graph neural networks. In Proceedings of the 13th International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 February 2020; pp. 70–78.

41. Tang, X.; Liao, D.; Huang, W.; Xu, J.; Zhu, L.; Shen, M. Fully exploiting cascade graphs for real-time forwarding prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 582–590.
42. Xu, X.; Zhou, F.; Zhang, K.; Liu, S.; Trajcevski, G. Casflow: Exploring hierarchical structures and propagation uncertainty for cascade prediction. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 3484–3499. [[CrossRef](#)]
43. Wang, Y.; Wang, X.; Jia, T. Ccasgnn: Collaborative cascade prediction based on graph neural networks. In Proceedings of the 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Hangzhou, China, 4–6 May 2022; pp. 810–815.
44. Feng, X.; Zhao, Q.; Li, Y. AECasN: An information cascade predictor by learning the structural representation of the whole cascade network with autoencoder. *Expert Syst. Appl.* **2022**, *191*, 116260. [[CrossRef](#)]
45. Wang, Y.; Wang, X.; Ran, Y.; Michalski, R.; Jia, T. CasSeqGCN: Combining network structure and temporal sequence to predict information cascades. *Expert Syst. Appl.* **2022**, *206*, 117693. [[CrossRef](#)]
46. Zhang, J.; Liu, B.; Tang, J.; Chen, T.; Li, J. Social influence locality for modeling retweeting behaviors. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013.
47. Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; Su, Z. Arnetminer: Extraction and mining of academic social networks. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 990–998.
48. Zhou, F.; Jing, X.; Xu, X.; Zhong, T.; Trajcevski, G.; Wu, J. Continual information cascade learning. In Proceedings of the 2020 IEEE Global Communications Conference (GLOBECOM 2020), Taipei, Taiwan, 7–11 December 2020; pp. 1–6.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.