



Multi-stage dynamic disinformation detection with graph entropy guidance

Xiaorong Hao¹ · Bo Liu^{1,2} · Xinyan Yang¹ · Xiangguo Sun³ · Qing Meng⁴ · Jiuxin Cao^{1,5}

Received: 12 August 2022 / Revised: 27 August 2023 / Accepted: 6 December 2023 /

Published online: 31 January 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

Online disinformation has become one of the most severe concerns in today's world. Recognizing disinformation timely and effectively is very hard, because the propagation process of disinformation is dynamic and complicated. The existing newest research leverage uniform time intervals to study the multi-stage propagation features of disinformation. However, uniform time intervals are unrealistic in the real world, cause the process of information propagation is not regular. In light of these facts, we propose a novel and effective framework *Multi-stage Dynamic Disinformation Detection with Graph Entropy Guidance*(MsDD) to better analyze multi-stage propagation patterns. Instead of traditional snapshots, we analyze the dynamic propagation network via graph entropy, which can work effectively in finding the dynamic and variable-length stages. In this way, we can explicitly learn the changing

This article belongs to the Topical Collection: *Special Issue on Privacy and Security in Machine Learning*
Guest Editors: Jin Li, Francesco Palmieri and Changyu Dong

✉ Bo Liu
bliu@seu.edu.cn

Xiaorong Hao
hxr@seu.edu.cn

Xinyan Yang
naynix@qq.com

Xiangguo Sun
xgsun@se.cuhk.edu.hk

Qing Meng
qmeng@hhu.edu.cn

Jiuxin Cao
jx.cao@seu.edu.cn

¹ Southeast University, Nanjing, China

² Purple Mountain Laboratories, Nanjing, China

³ The Chinese University of Hong Kong, Hong Kong SAR, China

⁴ Hohai University, Nanjing, China

⁵ Sanjiang University, Nanjing, China

pattern of propagation stages and support timely detection even at the early stages. Based on this effective multi-stage analysis framework, we further propose a novel dynamic analysis model to model both the structural and sequential evolving features. Extensive experiments on two real-world datasets prove the superiority of our model. We open the datasets and source code at <https://github.com/researchxr/MsDD>.

Keywords Disinformation detection · Multi-stage learning · Graph entropy · Dynamic analysis

1 Introduction

Recent years witness the whole world suffering from various disinformation with the help of online social networks. The rapid spread of various disinformation creates growing anxiety and panic among the public, making timely and effective disinformation detection urgently needed now than ever.

The disinformation propagation has some potential patterns, which attracts lots of researchers to consider this problem from the perspective of sequence, structure, and propagation stage. Earlier research explores sequential propagation patterns and temporal variation of contextual information [1-4]. However, the sequential patterns is not enough to model the information propagation, because the information in the real social network platform diffuses through large amount of network interactions. Then, some graph-structured models are proposed for disinformation representation learning and classification [5-8]. The researchers studied the spreading structure of disinformation based on graph neural network and integrated the semantic features for analyzing disinformation detection deeply. These works cannot perform well on timely detection and pay no attention to interaction evolution. Therefore, recent research [9,11] focus on studying propagation stage for dynamic disinformation spreading. This multi-stage approach is based on differences between the propagation patterns of real and false information. However, how to find information's propagation stages for better disinformation identification is still an open problem. Guo et al. [12] summarize some propagation stages of disinformation such as the incubation stage, the eruptive stage, the contagion stage, and the termination stage. Different propagation stages usually display different statuses during the process of information spreading. As shown in Figure 1, each propagation stage contains some source information and responsive information. The source information refers to the tweets posted by the user at a certain moment. Responsive informa-

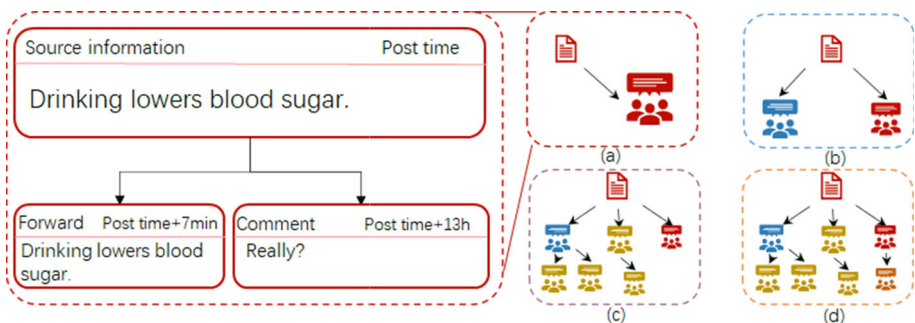


Figure 1 Visualization of disinformation propagation. The left half of this figure is a specific case of disinformation propagation. The right half of this figure is the structural evolution of disinformation propagation

tion is retweets and comments. Therefore, it is necessary to analyse diffusion dynamics by splitting these stages effectively. However, some works [15, 16] try to study the dynamics of disinformation by splitting time-varying propagation as several snapshots using uniform time intervals. These works are far from enough to properly learn propagation stages, because uniform time interval may cause duplicated stages and meaningless stages.

To achieve more timely and effective disinformation detection, we take information dynamics into consideration to recognize reasonable propagation stages. However, this is never easy because of the following difficulties: (1) How to find information propagation stages properly? The propagation stages are not regular and apparent, so it needs to design effective methods to measure the process of information propagation. (2) How to design an effective model to capture each stage's characteristics? After recognizing stages of information propagation, each stage should be modeled by capturing effective features. (3) How to learn the evolving pattern from the propagation stages? It is necessary to grasp the evolving patterns from a macro perspective. Because there exists differences of propagation patterns between different types of information.

To overcome these difficulties, we creatively design a graph entropy-based method to find proper propagation stages. Information diffusion is an irreversible process and dynamic, which has direct connections with the change of entropy over time [1]. Our motivation is to try to find the change rule of information diffusion structure, and graph entropy [2] can measure information gain of a graph. To this end, we apply graph entropy to quantify the structure change of the disinformation propagation. By analyzing changes in graph entropy, we can find some obvious stages in the propagation process of information. Some similar stage sub-graphs will be merged into one stage. Based on this, we propose a single-stage embedding module to capture both structural and semantic features. Then we further propose an effective model to learn the multi-stage evolution features. In summary, our main contributions are as follows:

- 1) To the best of our knowledge, this is the first job that dynamically divides propagation stages of disinformation with graph entropy. The propagation structure of disinformation diffusion has direct connections with the increase of graph entropy over time. Multi-stage propagation sub-graph divided by graph entropy is more aligned with the real-world disinformation propagation.
- 2) We build a novel **Multi-stage Dynamic Disinformation Detection (MsDD)** framework, which adapts well to multi-stage propagation sub-graph. Based on the multi-stage propagation sub-graph composed of proper propagation stages, this framework can effectively capture the evolution features of all sub-graphs.
- 3) Detailed experiments on two real-world social media data demonstrate the effectiveness of the proposed framework. Extensible studies also show the benefits of multi-stage propagation sub-graphs.

2 Preliminaries and problem definition

In this section, we introduce some concepts as a foundation for our method.

Information propagation graph We utilize $\mathcal{C} = \{C_0, C_1, \dots, C_{|C|}\}$ to denote disinformation datasets. A disinformation instance consists of both source (r_i) and responsive information ($x_{ij}, j \in [1, m]$), i.e., $C_i = \{r_i, x_{i1}, \dots, x_{im}\}$. Let $\mathcal{SG} = \{\mathcal{V}, \mathcal{E}, \mathcal{T}\}$ be an information propagation graph. Here, the node set \mathcal{V} shares the same content C_i . $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, usually represented by the interaction between source and responsive

information, such as forwarding, commenting et al. $\mathcal{T} = \{t_0, t_1, \dots, t_m\}$ is the time when information is posted.

Multi-stage propagation sub-graph The information propagation graph usually has a continuous timestamp for different nodes. However, in the real world, the timestamps are usually so sparse that analyzing them for each timestamp is impossible. More importantly, in the scenario of disinformation propagation, we usually care more about the changes of information propagation graph during a period. To this end, we define the multi-stage propagation sub-graph of an information propagation graph as $SG = \{sg_0, sg_1, \dots, sg_{|SG|}\}$ where $|SG|$ is the total number of sub-graphs. Each stage sub-graph is denoted as $sg_k = \{V_k, E_k, T_k\}$, $sg_k \subseteq SG$.

Graph entropy Graph entropy is used to describe the amount of information encoded in a graph [2], which can be defined as: $H(sg) = -\sum f(i) \log f(i)$. Here, sg is a graph data denoted by $sg = \{V, E\}$, where $V = \{1, 2, \dots, n\}$ is the node set and $E = \{(i, j) \mid i, j \in V\}$ is the edge set. The function $f(v_i)$ is a mapping: $\mathbb{R}^{1 \times d} \rightarrow 0 \leq p \leq 1$, which takes each node's feature as input and outputs a scalar score p within $[0, 1]$. In this paper, we take each node's centrality features such as node degree as input and then output their importance measurement to denote the information contribution of the entire graph. It is worthwhile to note that the graph entropy of an empty graph without any node and edge is always zero. As more nodes and connections are added to the graph, the graph entropy should change with a different ratio. Therefore, it is of great value to analyze graph entropy trend for recognizing the dynamic propagation stage properly.

Disinformation detection The disinformation detection task is designed to reveal fabricated information and reduce the impact of misleading public opinion. We aim to identify disinformation by learning a classifier g , that is:

$$g : SG \rightarrow y, \quad (1)$$

where SG denotes the information propagation graph and y denotes the class label.

3 Disinformation detection

3.1 Overview

Our disinformation detection framework, MsDD, is shown in Figure 2, which includes three parts. The first part is an original sequence of information propagation. The second part is multi-stage awareness based on graph entropy. In this part, we divide the original sequence data into multi-stage propagation sub-graphs based on the theory of entropy. The third part is dynamic analysis, which captures the evolution features of each stage sub-graph. Each stage sub-graph includes structural and sequential attributes. We use graph attention network to learn structural features, and use sequence attention network to learn sequential features. The concatenation of structural and sequential features is fed into a multi-stage learning unit to extract the evolution features of all stage sub-graphs. Finally, the learned evolution representations are sent to MLP(multilayer perceptron) to recognize whether the information is false or not.

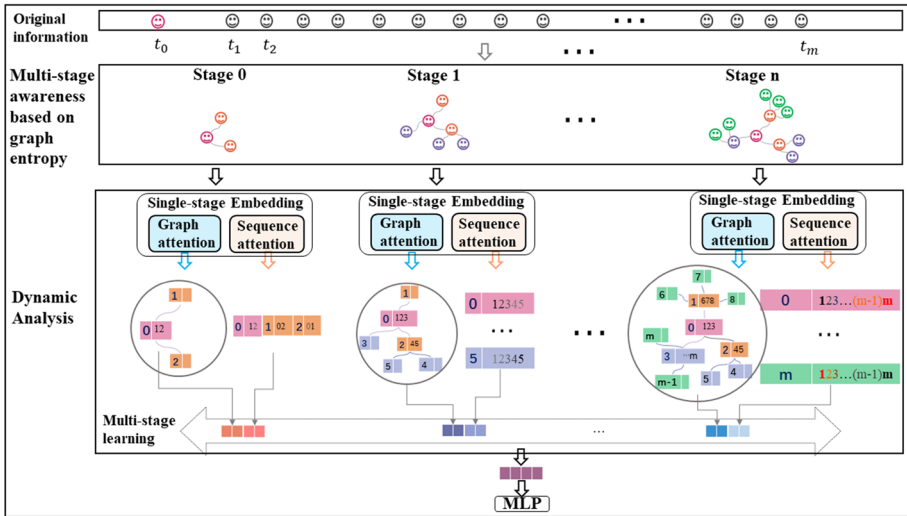


Figure 2 The flowchart of MsDD for disinformation detection

3.2 Multi-stage awareness based on graph entropy

Intuitively, disinformation is usually diffused via users’ tweets, retweets and comments. These events are not evolving steadily because they usually contain several “breaking times”. Modeling the non-steady evolution is essential by constructing different propagation stages. The graph entropy can be an indicator of information quantity and can help construct multi-stage propagation sub-graph. According to the increasing curve of graph entropy over time, we define the sub-graph entropy change ratio in a moment. Then, we divide stages by judging the tangent of the angle between two entropy change ratios of two adjacent stage sub-graphs. To be specific, the multi-stage awareness contains two steps: (1) Calculating graph entropy via node degree. (2) Constructing multi-stage propagation sub-graphs.

3.2.1 Calculating graph entropy via node degree

Information diffusion is naturally reflected by its propagation structures. Therefore the graph entropy $H(sg)$ we used should also be derived from its structures. Here we present a very simple yet effective graph entropy form via node degree, which can be calculated by:

$$H(sg) = - \sum_{i=1}^{|V_{cur}|} \frac{d(i)}{\sum_{j=1}^{|V_{cur}|} d(j)} \ln \frac{d(i)}{\sum_{j=1}^{|V_{cur}|} d(j)}, \tag{2}$$

where d_i is the degree of node i , and $|V_{cur}|$ is the number of nodes in the current graph. The graph entropy is zero on empty graph or the graph only contains a node. The graph entropy gradually increases as new nodes join the graph. Following these changes, we can easily obtain the graph entropy curve, which will be further analyzed to find the dynamic propagation stages properly.

Algorithm 1 Multi-stage awareness based on graph entropy.

Input: One information propagation graph $SG = \{\mathcal{V}, \mathcal{E}, \mathcal{T}\}$, parameters τ, μ .

1: Initialize empty sets SG_{temp} , the information propagation's total duration $d_t = \max\{\mathcal{T}\} - \min\{\mathcal{T}\}$

2: **loop**

3: $sg \leftarrow \emptyset, H(sg) \leftarrow 0, \nabla H(sg) \leftarrow 0$.

4: **for** $i \leftarrow 1$ to $\frac{d_t}{\tau}$ **do**

5: $sg' = \{\mathcal{V}_{\leq i*\tau}, \mathcal{E}_{\leq i*\tau}, \mathcal{T}_{\leq i*\tau}\}$.

6: compute $H(sg')$ by formula (2)

7: compute $\nabla H(sg')$ by formula (3)

8: compute $\tan \theta$ by formula (4)

9: **if** $\tan \theta > \mu$ **then**

10: add sg' into SG_{temp}

11: **end if**

12: $H(sg) = H(sg'), \nabla H(sg) = \nabla H(sg')$

13: **end for**

14: **end loop**

Output: $SG = SG_{temp}$

3.2.2 Constructing multi-stage graph propagation sub-graph

The most informative stages should bring about relatively larger entropy shifts. Here, we define sub-graph entropy change ratio as follows:

$$\nabla H(sg') := \frac{H(sg') - H(sg)}{\tau}. \quad (3)$$

The above equation explains the information change from sg to a new stage sub-graph sg' in a time unit τ . The time unit means a time interval that makes it possible to calculate the graph entropy every time unit during the process of information propagating. Next, we split the propagation process by making the following constraint satisfied.

$$\tan \theta := \left| \frac{\nabla H(sg') - \nabla H(sg)}{1 + \nabla H(sg') \cdot \nabla H(sg)} \right| > \mu. \quad (4)$$

Where θ describes the angle between two entropy change ratios of two adjacent stage sub-graphs. The computation of $\tan \theta$ is the same as the tangent of angle between any two lines [3]. Let sg' be a new stage sub-graph if $\tan \theta$ exceeds the threshold μ . Repeating this process to construct multi-stage propagation sub-graph. The workflow of multi-stage awareness based on graph entropy is outlined in Algorithm 1.

3.3 Dynamic analysis

This subsection implements dynamic analysis for the multi-stage propagation sub-graph. Here, we provide analysis methods from two perspectives, i.e., single-stage embedding and multi-stage dynamic learning. The core of single-stage embedding module is to extract interaction features and the temporal features. The multi-stage dynamic learning module is intended to learn key shifts between two adjacent stage sub-graphs.

3.3.1 Single-stage embedding

For simplicity, we describe the embedding process on a stage sub-graph sg_k and similarly for other stage sub-graphs. We first initialize each information representation $\mathbf{v}_k^{1, \dots, |V_k|}$ using the word vector model and Bi-LSTM. We then update information embedding by designing a graph and sequence attention model.

Graph attention module The objective of this step is to extract the structural representation of each stage sub-graph. Let $\mathbf{h}_i^{(l=0)} = \mathbf{v}_k^i$ be the hidden representation output for the initial node/information i from the 0th layer. We first conduct a linear transformation for the lower layer node representation as $\mathbf{z}_i^l = \mathbf{W}^l \mathbf{h}_i^l$. Here, \mathbf{W} is the linear transformation weight matrix. Nodes' representation can be updated by attention aggregation as follows:

$$\mathbf{h}_i^{(l+1)} = \sigma \left(\sum_{x \in N(i)} \beta_{ix}^{(l)} \mathbf{z}_x^{(l)} + \mathbf{z}_i^{(l)} \right), \tag{5}$$

where the representation $\mathbf{h}_i^{(l+1)}$ of node i at the $l + 1$ is the weighted sum of node and its neighbours' representation at the l layer. σ is the softmax activation function. $\beta_{ix}^{(l)}$ is attention weight matrix between node i and its neighbours $N(i)$, which can be computed as (6).

$$\beta_{ix}^{(l)} = \frac{\exp(e_{ix}^{(l)})}{\sum_{y \in N(i)} \exp(e_{iy}^{(l)})} \tag{6}$$

$$e_{ix}^{(l)} = \text{LeakyReLU}(\vec{b}^{(l)\top} \cdot (|\mathbf{z}_i^{(l)} \parallel \mathbf{z}_x^{(l)}|)) \tag{7}$$

In the above formula: (7) computes an attention score between information i and its neighbour x . \vec{b} is a learnable weight vector, \parallel denotes concatenation, N_i denotes the neighbours of node i , and *LeakyReLU* is activation function. Equation (6) normalizes the obtained attention score to get $\beta_{ix}^{(l)}$.

Considering the superiority of the multi-head attention function, we randomly initialize different weight matrix \mathbf{W}^l , and then acquire node representations based on multi-head attention. We concatenate these node representations into a combined vector \mathbf{h}_i^L . Finally, we get the structural information representation \mathbf{I}_k^s in the k th stage sub-graph. This formula can be computed as (8).

$$\mathbf{I}_k^s = \sum_{i=1}^{|V_k|} \alpha_i \mathbf{h}_i^L \tag{8}$$

We achieve the attention score α_i by training (9), where \mathbf{p} is a trainable weight parameter vector.

$$\alpha_i = \frac{\exp(\mathbf{h}_i^L \cdot \mathbf{p})}{\sum_{j=1}^{|V_k|} \exp(\mathbf{h}_j^L \cdot \mathbf{p})} \tag{9}$$

Sequence attention module Information in each stage sub-graph is shared in chronological order, which is a very useful supplementary item for graph structural data. To this end, we build a sequence attention module based on Bi-LSTM to meet the above shortfalls. The input to this module is sequences of node representations $\mathbf{v}_k^{1, \dots, |V_k|}$ in each stage sub-graph. We can specify the current input \mathbf{v}_k^i and the previously generated hidden state $\overrightarrow{\mathbf{h}}_{i-1}$ as inputs to the single-direction LSTM. Here, \rightarrow is the forward direction of the LSTM unit. Equation (10) is

used for the determination of temporary cell state. \mathbf{W}_{Ce} and \mathbf{b}_{Ce} are trainable weight matrix and bias, respectively.

$$\vec{C}e_i = \tanh(\mathbf{W}_{Ce} \cdot [\vec{\mathbf{h}}_{i-1}, \mathbf{v}_k^i] + \mathbf{b}_{Ce}) \tag{10}$$

Single-direction LSTM units contain three gates. We denote the input gate \vec{in}_i as (11), the output gate \vec{ou}_i as (12), the forget gate \vec{fo}_i as (13).

$$\vec{in}_i = \sigma(\mathbf{W}_{in} \cdot [\vec{\mathbf{h}}_{i-1}, \mathbf{v}_k^i] + \mathbf{b}_{in}) \tag{11}$$

$$\vec{ou}_i = \sigma(\mathbf{W}_{ou}[\vec{\mathbf{h}}_{i-1}, \mathbf{v}_k^i] + \mathbf{b}_{ou}) \tag{12}$$

$$\vec{fo}_i = \sigma(\mathbf{W}_{fo} \cdot [\vec{\mathbf{h}}_{i-1}, \mathbf{v}_k^i] + \mathbf{b}_{fo}) \tag{13}$$

Where \mathbf{W}_{ou} , \mathbf{W}_{in} , and \mathbf{W}_{fo} are corresponding weight matrices. \mathbf{b}_{ou} , \mathbf{b}_{in} , and \mathbf{b}_{fo} are corresponding biases. Some valuable features of the current cell state are selected from previously generated cell state $\vec{C}e_{i-1}$ and the unit state update value $\vec{C}e_i$. Equation (14) gives a clear definition of this situation.

$$\vec{C}e_i = \vec{fo}_i * \vec{C}e_{i-1} + \vec{in}_i * \vec{C}e_i \tag{14}$$

The forward hidden state of the i_{th} node can be implemented with (15). The output gate and cell state control its values.

$$\vec{\mathbf{h}}_i = \vec{ou}_i \tanh(\vec{C}e_i) \tag{15}$$

We then use the reverse arrow to indicate the backward output $\overleftarrow{\mathbf{h}}_i$. The updated node embedding can be generated from (16), where the notation \oplus is the element-wise sum operation.

$$\hat{\mathbf{v}}_k^i = [\vec{\mathbf{h}}_i \oplus \overleftarrow{\mathbf{h}}_i] \tag{16}$$

The time-sequenced representation \mathbf{I}_k^t of the k th stage sub-graph can be aggregated through attention mechanism. The computation method of the attention score $\tilde{\alpha}_i$ is similar to (9). We formulate (17) to denote the time-sequenced information representation.

$$\mathbf{I}_k^t = \sum_{i=1}^{|V_k|} \tilde{\alpha}_i \hat{\mathbf{v}}_k^i \tag{17}$$

3.3.2 Multi-stage dynamic learning

We carry out dynamic learning on the multi-stage dynamic propagation sub-graph. Each stage sub-graph’s representation is learned by fusing structural information representation and time-sequenced information representation. The representation of stage sub-graph is denoted as $\tilde{s}g_k$.

$$\tilde{s}g_k = \mathbf{I}_k^s \parallel \mathbf{I}_k^t \tag{18}$$

We still leverage the Bi-LSTM technique to extract dynamic features from the multi-stage propagation sub-graph. The specific calculation formula is just the same as before (10-16). The only difference is that we use sub-graph representation $\tilde{s}g_k$ instead of the i_{th} node’s representation \mathbf{v}_k^i . After the effective training of Bi-LSTM, the representation of each stage

sub-graph $s\hat{g}_k$ has fused other stage sub-graphs' information propagation features. Finally, by combining all stage sub-graphs' representation, which is given by

$$\hat{S} = \frac{1}{|SG|} \sum_{k=1}^{|SG|} s\hat{g}_k \quad (19)$$

where $\hat{S} \in \mathcal{R}^d$ is the final output representing the propagation features of the source information. A detection model is deployed on \hat{S} , and its detection score is defined as:

$$\hat{y} = MLP(\hat{S}) = softmax(\mathbf{W}_1\sigma(\mathbf{W}_2\hat{S} + \mathbf{b}_2) + \mathbf{b}_1) \quad (20)$$

where $MLP(\cdot)$ is a detection operation comprised of a multilayer perceptron. We use the cross-entropy loss function (21) for training our model,

$$\mathcal{L} = -\frac{1}{s} \sum_{is=1}^s (y_{is} \log(\hat{y}_{is}) + (1 - y_{is}) \log(1 - \hat{y}_{is})) \quad (21)$$

where s is the number of all samples, y_{is} is the actual label of the is_{th} sample, and \hat{y}_{is} is the detect score predicted by our model. The gradient descent algorithm is adopted to optimize the designed method.

3.4 Complexity analysis

According to our experiments, the main computational cost of the model lies in the multi-stage awareness and dynamic analysis module. For the multi-stage awareness partition, we need to deal with $|S|$ samples. Each sample is computed $w = \frac{d_i}{\tau}$ times according to the time unit τ , where d_i is the total propagation duration of each sample. Since w is a changeable value due to the variable duration of different samples, the complexity of this partition is less than $O(w_{\max} \times |S|)$. The dynamic analysis module includes graph and sequence attention neural network. Its cost is mainly due to the repeated computation of each stage sub-graph. In practical training, we use the same scale parameters for parallel computing on each stage sub-graph. To wit, the dimension of weight matrix \mathbf{W} of graph attention network is d^2 for each stage sub-graph. Here input and output feature share the same feature vector dimensions d . Thus, the complexity of graph attention module is $O(|V| \times d^2 + |E| \times d)$ for each stage sub-graph. $|V|$ and $|E|$ is the number of nodes and edges in each stage sub-graph, respectively. The dimension of weight matrices for the sequence attention is $2 \times 4(\frac{d}{2}(\frac{d}{2} + d) + \frac{d}{2})$ for each stage sub-graph. Here, 2 indicates LSTM neural network in both directions, and 4 indicates three gates and one cell state in each unit. $\frac{d}{2}$ denotes the number of hidden units, and d is the number of input features. Thus, the total complexity of the dynamic analysis module is approximated as $O(|V| \times d^2 + |E| \times d)$ for each stage sub-graph. Benefits of parallel computing for each stage sub-graph, this algorithm does not spend more time.

Table 1 Statistics of the datasets

Statistic	MisInfdect	PHEME
# of source information	11765	5444
# of disinformation	4085	1862
# of true information	7680	3582
# of responsive information	905691	98895
# of users	754273	48823
Avg.# of responsive information	77	18

4 Experiments and analysis

4.1 Experimental setting

4.1.1 Datasets

In our experiments, we select two public datasets to evaluate our method. One dataset is the PHEME [4], which contains 5444 source information from Twitter. The other dataset is the MisInfdect dataset ¹, which contains 11765 source information from Sina Weibo. These two datasets contain the following content: the content of source information and responsive information, the posted time of source information and responsive information. We show the statistics of datasets in Table 1.

4.1.2 Metric and training details

We use Accuracy(ACC), F1 score, and Area under the ROC curve(AUC) score to quantitatively evaluate the disinformation detection performance. Higher values indicate better performance. All the parameters are selected using a grid search strategy on different datasets. Several appropriate parameters are the following: the threshold μ is 0.007, the time unit τ is 15 minutes, the layer of Bi-LSTM is 2, four-head attention and one-layer Graph neural network give the optimizer results, an early stopping strategy halts training if the loss doesn't decrease for 25 epochs. The Adam algorithm is adopted to update parameters with a learning rate of 0.08. All experiments are executed in a server with Intel Xeon Silver 4216 (2.10 GHz) CPU, and GeForce RTX2080Ti-12GB GPU.

4.1.3 Baselines

To evaluate the effectiveness of our method, we compare it with some following state-of-the-art baselines.

- ACAMI [5]: The first work applies content and temporal co-attention to massive disinformation identification. It learns the event's distribution representation, and then an information representation in each event can be embedded according to sequence learning.
- Hierarchical GCN-RNN(HGCN-RNN) [6]: A hierarchical multi-task learning framework for disinformation detection and stance detection, which learns information features

¹ <https://weibo.com/weibopiyao>

Table 2 Results of disinformation detection

	PHEME(%)			MisInflect(%)		
	ACC	F1	AUC	ACC	F1	AUC
ACAMI	75.78	61.66	71.11	82.65	75.54	81.30
HGCN-RNN	85.28	81.18	<u>91.32</u>	89.11	83.10	94.77
Bi-GCN	84.97	80.32	91.09	88.03	83.03	93.47
StA-PLAN	83.36	79.78	90.58	87.67	83.04	94.07
RDLNP	84.31	80.78	91.18	88.01	83.13	93.12
DynGCN	86.78	<u>82.40</u>	90.41	88.70	84.37	94.28
MsDD_uni	<u>87.40</u>	82.28	90.83	<u>89.80</u>	<u>85.75</u>	<u>95.51</u>
MsDD	90.70	84.63	93.52	91.76	87.74	96.53

¹Bold values are the best results

²Underlined values denote the second best

from conversation structure and rumor stance evolution. Here, we only apply it to implement the disinformation detection task.

- Bi-GCN [7]: A graph neural network based model for disinformation detection, which adopts bi-directional graph convolutional networks to deal with both propagation and dispersion of information.
- StA-PLAN [8]: A Transformer with an attention mechanism for disinformation detection. It tries to improve the information's relation representation by modelling long-distance interactions between source information and its responsive information.
- RDLNP [9]: An embedding network with linear sequence learning and non-linear structure learning for disinformation identification. This work also considers the attention dependencies between source information and its responsive information.
- DynGCN [10]: Using uniform temporal snapshots capture the dynamics of information propagation for disinformation identification. For better results, we use the GAT network to replace the GCN network and train snapshots through Bi-LSTM.
- MsDD_uni: This is a variant of our model MsDD, which uses uniform temporal snapshots but not graph entropy guided multi-stage snapshots. The remaining part of this model is the same as MsDD.

These models consider both the content as well as the interactive features of information. We also fine-tune these baselines for optimal performance.

4.2 Overall performance on disinformation detection

Table 2 shows the disinformation detection performance of all the competitors on two different datasets. For ACAMI, its results are not as good as other methods as it neglects the propagation structure features. For StA-PLAN, it mines the tree structure information by using Transformer network and outperforms than ACAMI. For HGCN-RNN, Bi-GCN, RDLNP and DynGCN, they use Graph Neural Network to better embed the propagation structure and outperforms than StA-PLAN. For HGCN-RNN, it gives more temporal analysis on people's stance and has higher scores compared with Bi-GCN and RDLNP. For DynGCN, it analyzes temporal features by using uniform temporal snapshots and performs well on F1 than HGCN-RNN.

Our method significantly outperforms all baselines in each metric. We observe that our method improves over the strongest baseline by 3.30% (ACC), 2.23% (F1), 2.20% (AUC) on the PHEME dataset and 1.96% (ACC), 1.99% (F1), and 1.02% (AUC) on the MisInfect dataset. We attribute this superiority to the creation of the multi-stage propagation sub-graph. Compared with MsDD_uni, the dedicated design of the variable-length stage ensures that the subsequent neural network can distinguish the true information from disinformation. Each stage sub-graph contains the propagation structure and temporal evolution. And dynamic learning of all stage sub-graphs increases MsDD's discrimination capability for disinformation and true information. The results of MsDD on two datasets, therefore, have a great improvement over MsDD_uni.

4.3 Ablation study

We further conduct ablation experiments for each sub-module. We consider three groups of ablation experiments: temporal-based method, structural-based method and a combination of them.

- *Temporal-based method*: These experiments only consider the sequence correlation of information propagation. MsDD_T detects disinformation by learning the sequential features when the input is a information propagation graph. MsDD_MT detects disinformation by learning the sequential features when the input is multi-stage propagation sub-graphs.
 - **MsDD_T**: This experiment first learns the text representation of information. Then a propagation graph is built with source information and its responsive information, whose representation is updated by a sequence attention learner. The source information is classified by using the multilayer perceptron .
 - **MsDD_MT**: MsDD_MT uses sequential learning to capture features from each stage's sub-graph. Then this method has a dynamic analysis for multiple stages.
- *Structure-based method*: These methods only consider the propagation structure of the information propagation graph without considering temporal features. MsDD_T detects disinformation by learning the structural features when the input is a information propagation graph. MsDD_MT detects disinformation by learning the structural features when the input is multi-stage propagation sub-graphs.

Table 3 Results of ablation experiments

Model	PHEME(%)			MisInfect(%)		
	ACC	F1	AUC	ACC	F1	AUC
MsDD_T	83.36	78.88	88.68	88.79	83.38	94.60
MsDD_MT	86.43	81.34	91.53	89.38	86.06	95.44
MsDD_S	84.45	80.62	89.87	89.89	85.65	95.23
MsDD_MS	86.78	82.65	92.49	90.31	86.35	95.93
MsDD_TS	84.60	80.92	90.31	90.74	86.29	95.46
MsDD	90.70	84.63	93.52	91.76	87.74	96.53

¹Bold numbers here are that our method outperforms ablation methods

- **MsDD_S**: The embedding of the propagation graph is learned via a GAT model. It is different from **MsDD_T** which uses a sequence attention learner.
- **MsDD_MS**: MsDD_MT uses GAT to capture features from each stage’s sub-graph. Then this method performs a weighted attention sum of representations from multi-stage propagation sub-graph.
- **Combination**: This type of methods combine the temporal and structural features of information propagation. MsDD_TS detects disinformation by learning temporal and structural features when the input is a information propagation graph. MsDD detects disinformation by learning the temporal and structural features when the input is multi-stage propagation sub-graphs.
 - **MsDD_TS**: This experiment is equivalent to our single-stage sub-graph learning process.
 - **MsDD**: Our framework, a multi-stage propagation sub-graph learning method for disinformation detection, takes into account the content, sequence, structure, and multi-stage evolution of information propagation.

Table 3 shows the results of ablation experiments. By comparing MsDD_T and MsDD_S, we observe that there is a slight preponderance of structure aggregation over sequence learning. The evaluation score of MsDD_S is 1.09% (ACC), 1.74% (F1), 1.19% (AUC) on the PHEME dataset, and 1.10% (ACC), 2.27% (F1), 0.63% (AUC) on the MisInfected dataset higher, respectively, than that of the MsDD_T. MsDD_TS behaves much better than MsDD_T and MsDD_S due to the fact that it combines the benefits of both. The creation of the multi-stage propagation sub-graph has provided new opportunities for disinformation detection. It makes the performance of disinformation detection achieve an advanced level, e.g. MsDD_MS reaches an AUC value of 92.49% on the PHEME dataset and 95.93% on the MisInfected dataset. Most of the time, nearly all of the results of MsDD_MS are higher than the baselines. On the PHEME dataset, MsDD_MS improves the ACC, F1, and AUC by up to 1.5%, 1.47%, 1.17% (MsDD_MS vs. HGCN-RNN). On the MisInfected dataset, MsDD_MS improves the ACC, F1, and AUC by up to 0.2%, 3.25%, 1.16% (MsDD_MS vs. HGCN-RNN). Undoubtedly, our model is the combination of multi-stage sequence attention, multi-stage sub-graph structure aggregation, and multi-stage dynamic learning, which results in a further improvement on the basis of MsDD_MS. Figure 3(a), (b) have a better visualization of ablation experiments.

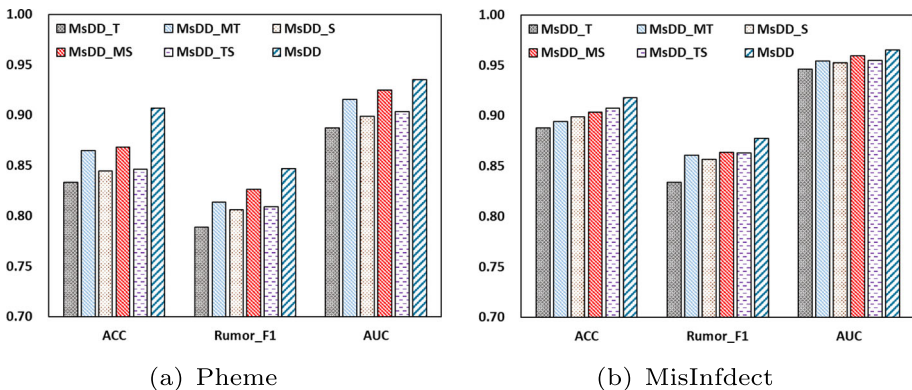


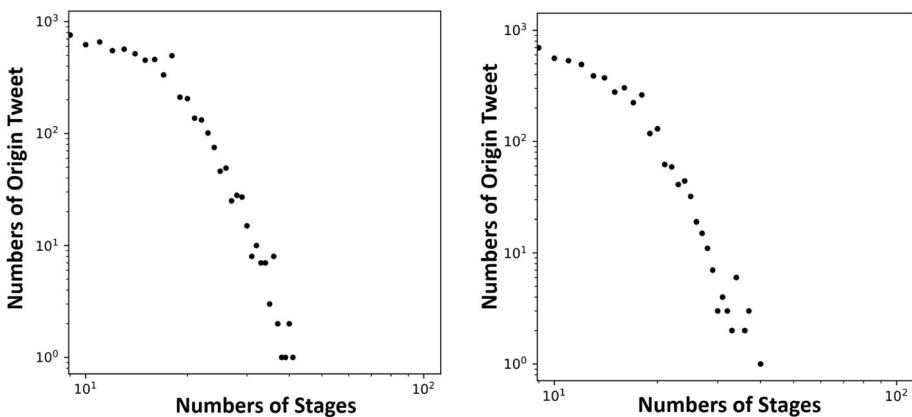
Figure 3 Ablation results of disinformation detection over two datasets

4.4 Effectiveness of graph entropy based multi-stage awareness

In this subsection, we present some stage division results to illustrate the rationality of dividing stages and to further investigate the capability of multi-stage awareness.

The division result of the multi-stage propagation sub-graph is related to two important parameters: time unit τ , and threshold μ . The time unit τ is a time interval during the process of information propagating and it is the basis of computing the change ratio of graph entropy. The size of parameter τ should reflect the change of the graph entropy. There are two principles to choose the parameter τ . Firstly, we'd better choose a time unit that can capture finer changes of the graph entropy. Secondly, we should avoid choosing overly small τ when the change of graph entropy is relatively stable, because the overly small τ will induce unnecessary entropy computations. The parameter μ is a threshold of entropy change ratio, which decides if the current time unit τ is a cut-off point for a new stage. The entropy change corresponds to the dynamic change of information propagation graph structure. So μ measures the change ratio of graph structure. Selecting a large threshold may lead to fewer propagation stages, while an overly small threshold can result in excessive segmentations that might not carry meaningful information. Thus, a rational threshold can help us to find the key shift of the graph structure.

The values of these two parameters depend on the statistical number of stages on the whole dataset. The number of stages generated by different time unit is different. An appropriate time unit is set as 15 minutes. The statistical result of the number of stages is shown in Figure 4(a) based on this time unit. When the time unit is smaller than 15 minutes, the number of stages will not have a big increase. For example, the number of stages with a time unit of 5 minutes is similar to that of 15 minutes. However, when the time unit is larger than 15 minutes like in Figure 4(b), the number of stages will reduce a lot, which makes the multi-stage aware algorithm unable to detect important changes in information propagation. The threshold μ also influences the division of stages. When μ is larger than 0.35, there will be no stages divided on all samples. When μ is 0, it will produce a new stage sub-graph in each time unit. After a lot of training and verification, we have selected an appropriate



(a) The time unit contains 15 min

(b) The time unit contains 30 min

Figure 4 The statistic w.r.t the number of stages on logarithmic coordinates system

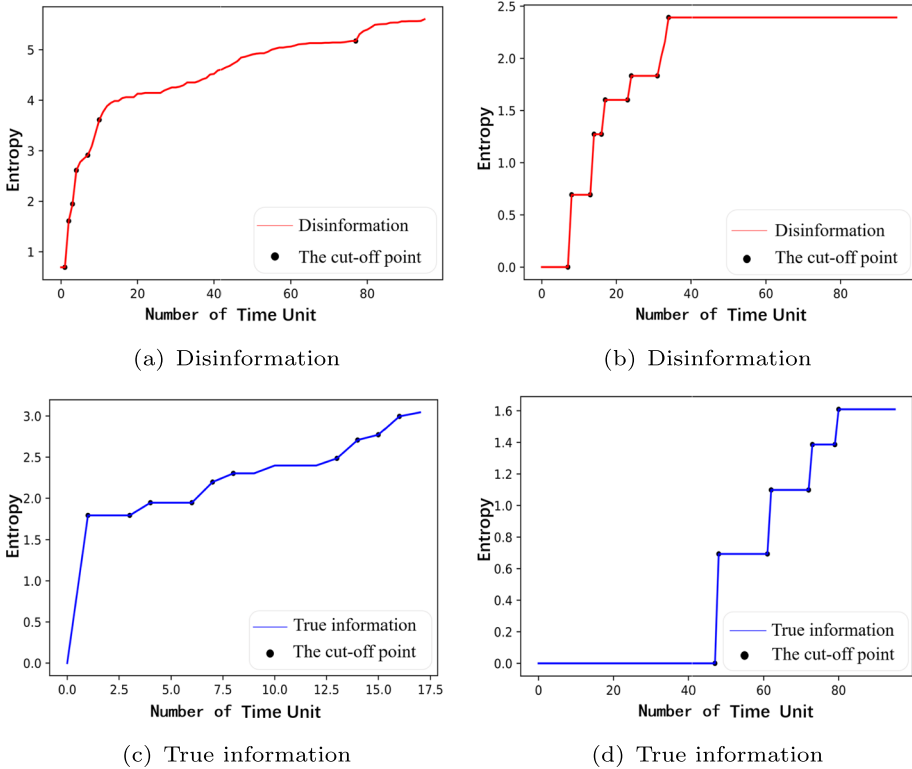


Figure 5 Stage division examples based on graph entropy. (a) and (b) shows the relationship between the entropy of the disinformation propagation graph and the number of time unit. (c) and (d) shows the relationship between the entropy of the true information propagation graph and the number of time unit. The horizontal axis represents the number of time unit. Each time unit contains 15 minutes. On the vertical axis, the entropy of the information propagation graph increase with the inclusion of the responsive information. Some cut-off points are marked with black dots when graph entropy changes a lot

threshold of 0.007. As shown in Figure 4, we found that the number of stage sub-graphs on most samples is quite small, and the maximum number of stages does not exceed 45.

Next, we conduct a further analysis of the division results of stage sub-graphs based on the above appropriate parameters. We select several specific propagation structures and show the stage division results. Figure 5 contains four samples of information stage division, with various distributions in graph entropy. That is to say, some graph entropy changed gently with time. Some changed dramatically and the rest of the samples are medium. From these figures, we can see that our multi-stage awareness algorithm can accurately find the divide where graph entropy has changed greatly around a moment. Additionally, the figures reveal that no cut-off points when the graph entropy has no change or has minor changes. For instance, Figure 5(a) shows no cut-off points between 20 and 80 time unit. Figure 5(b) exhibits no cut-off points after 40 time unit. These results are beneficial for the neural network to explore the propagation pattern of the source information. The reason is that we perform effective stage selection for information propagation. These stages are then fed into a neural network to learn the change patterns of information propagation. A few stages help the neural unit memorizing critical features.

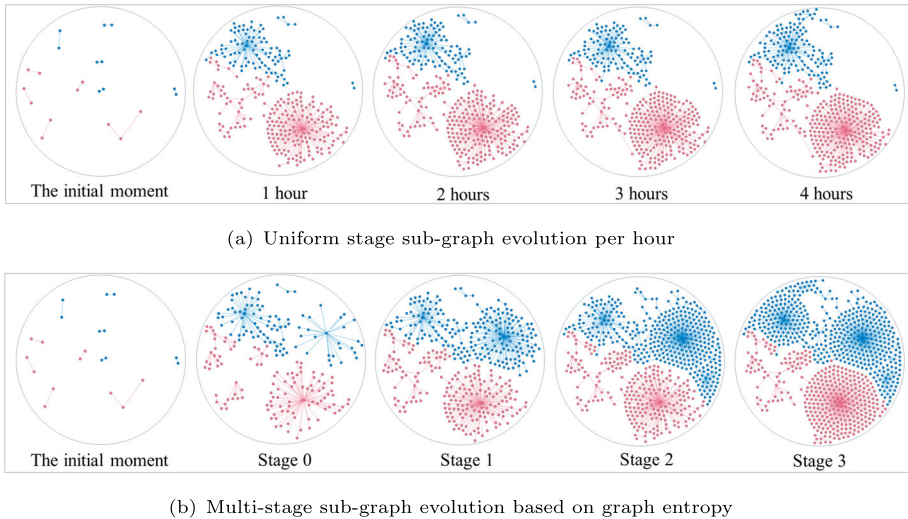


Figure 6 Visualization of sub-graph evolution from uniform stage and multi-stage. Red nodes represent the propagation of disinformation, blue nodes represent the propagation of true information. This figure depicts two different types of stage evolution states with same samples

To clearly demonstrate the capability of multi-stage awareness, we select false and true information with the same proportion from our datasets. Figure 6(a) is a visualization of information evolution based on uniform intervals. Starting from the initial moment, we present the propagation sub-graph at intervals of an hour. There are a few changes between some adjacent stages. So the uniform stages result in that important changes are difficult to be captured, because these important changes are covered in redundant stages. However, Figure 6(b) displays the information evolution stage based on graph entropy. We can observe that the process of information propagation shows an obvious change in adjacent stages. Our designed multi-stage propagation sub-graph based on graph entropy can retain informative stages and greatly reduces the number of redundant stages. Compared with uniform stage method, a multi-stage dynamic learning algorithm prefer to important changes rather than minor changes.

4.5 Effectiveness on timely detection

It's crucial and challenging to accurately judge disinformation at the early stage. Here, we evaluate our model's performance at the early stage of information propagation. Specifically, we provide the results of early detection at the moment of 3rd, 6th, 12th, 24th and 48th hours respectively. It has to be noted that the detection time is not our stage divide time. For example, the experiment results of timely detection at the 12th hour are acquired by training models on the first 12 hours' data of the whole datasets.

In order to better evaluate our work, we need to analyze the distribution of datasets at an earlier time, as shown in Figure 7. Here we select 150 nodes to better present the changing of information propagation at an earlier time. The max propagation numbers of the PHEME dataset and the MisInfected dataset are 345 and 1984 respectively. At an earlier time, most disinformation spreads faster than true information on the MisInfected dataset. The PHEME dataset has a minor gap. Anomaly values outside the box plot show that some information

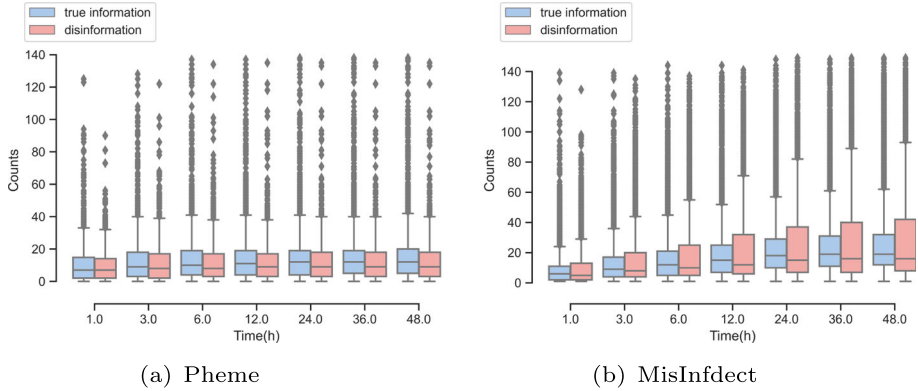


Figure 7 Datasets distribution statistic of timely detection. The box plot provides the visualization of the increase in the counts of information propagation over time

has a faster propagation speed. Compared to the PHEME datasets, the propagation change of MisInfect is more obvious, which means retweet nodes increased a lot from 3 hours to 6 hours. The dissimilar distribution patterns in these two datasets provide support for validating the robustness of timely disinformation detection.

Figure 8 denotes the early detection results over two datasets. Each model shows different trends, with evaluation results varying at different time points. An apparent example from the PHEME dataset is that the ACC of MsDD at the time of 24th hours is lower than at the time of 12th hours, while the Bi-GCN has the opposite result. From Figure 8(a), we can see that the ACC score peaks at about 12 hours and then begins to decrease. We attribute these phenomena to the differing impact of propagation structures at different stages on model detection. Notably, there is a differentiation in the performance of each model in terms of early detection capability. We see that the performance of our proposed model is superior to other models whether in early or later-stage detection. Figure 8(e) also demonstrates the increasing efficiency of detection performance over time.

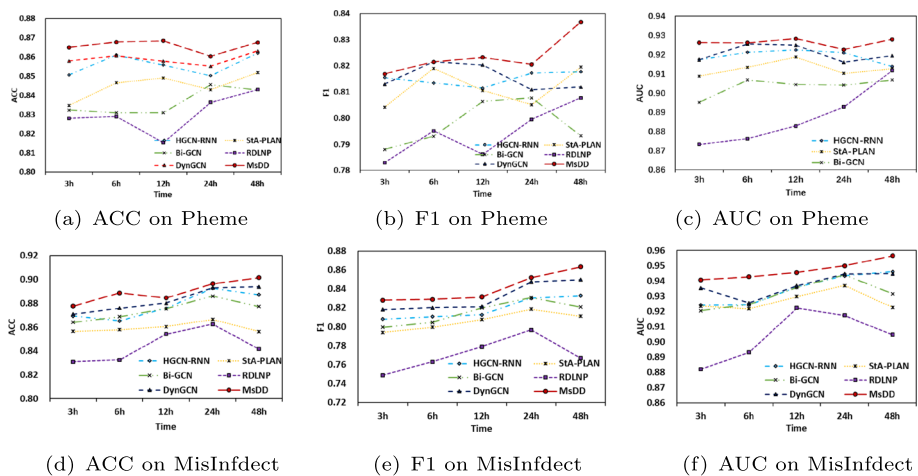


Figure 8 Timely detection of disinformation over two datasets. The horizontal axis represents the detection time. It means we evaluate the performance of every model in using datasets before the current time. For our model, there may be one or a few stage sub-graphs. The vertical axis is the corresponding evaluation index

4.6 Computational cost of multi-stage awareness

In this subsection, we measure the computation cost related to multi-stage awareness module. Considering the multi-stages awareness includes graph entropy calculation and stage division, we estimate the proportion of execution time and memory usage for these two parts over two datasets. For the PHEME dataset, the computation of graph entropy takes an average time of 4.53s for every 3 hours of information propagation. This execution time accounts for approximately 0.8% of the MsDD. The process of stage division takes an average time of 0.35s for every 3 hours of information propagation. This execution time accounts for approximately 0.06% of the MsDD. For the MisInfect dataset, the calculation of graph entropy takes an average time of 17.42s for every 3 hours of information propagation. This execution time accounts for approximately 0.2% of the MsDD. The process of stage division takes an average time of 0.54s for every 3 hours of information propagation. This execution time accounts for approximately 0.007% of the MsDD. The memory usage of multi-stage awareness bellow 3.36GB over two datasets. Overall, the computation cost of multi-stage awareness maintains a lower ratio in MsDD.

5 Related work

In this section, we review related work from two aspects: context-concerned disinformation detection and structure-aware disinformation detection.

5.1 Context-concerned disinformation detection

Context-concerned methods mainly rely on manual features to conduct disinformation detection. Ideal features for this task are mainly user-profile-based features (content features, user features), network-based features (communication structure features), and other different aspects. Castillo et al. [11] systematically proposes the following types of features in the study of information credibility: content features, user features, topic features, communication features, etc. And then these annotation features are used to explore how users spread, support, or deny information. The work of extracting features by Castillo et al. [11] make a pioneering contribution to the research domain of disinformation detection. Most follow-up research explores new features based on this work. These features include URL and hashtag features related to social platforms, the distribution of hashtag labels between disinformation or normal message [12], a client type and the location for users to post information [13], the average number of followers daily, the average number of tweets, the proportion of questioning comments, and the comment attributes of correcting the total number of comments [14], etc. Such approaches usually depend on the prior knowledge of researchers and are difficult to ensure the completeness of picked features. This may lead to some limitations in the model generalization.

5.2 Structure-aware disinformation detection

Structure-aware methods have particular benefits for disinformation detection, owing to their ability to mine deep features from the content and graph structure. Considering the spread of information can be simplified into a sequential problem based on the order of retweeted information, academics have developed new types of deep learning models based on recur-

rent neural networks (RNNs, LSTM). Ma et al. [15] employ a shared RNN layer to encode information. The adoption of the RNN model further allows learning of sequential features between the source and its forwarded information. The final neural layers of the RNN are designed to distinguish between disinformation and true information. Recognizing the challenge of dealing with long sequences of retweeted information, Guo et al. [16] view the entire sequence propagation as an event. This event is further divided into several subevents at fixed time intervals. The authors define a hierarchical structure that captures the relationship between the event and its subevents. They build a hierarchical structure network with an attention mechanism to embed posts for misinformation detection. Numerous recent studies focus on disinformation detection by incorporating extra knowledge resources, such as stance distributions [17], users' biases, and their social neighbors' opinions [18], among others. There are also efforts to support disinformation detection involving images [19] and videos [20]. However, these approaches may struggle to simulate the real propagation interaction structure of disinformation.

In real scenarios, the interaction structure of disinformation is not purely sequential. To better simulate the actual diffusion process, researchers attempt to model the propagation graph structure using deep learning methods. Some scholars develop CNN-based models capable of capturing interactive structural features [5, 21, 22]. Ma et al. [23] conceptualize this task as a recursive feature learning process within a tree structure. The propagation tree's root node represents the source information, and its descendants correspond to responsive information. Khoo et al. [8] explore a hierarchical token and post-level attention model that considers user interactions. Recent studies show prospect in modeling interaction graph structures using Graph Neural Networks (GNNs), such as Graph Convolutional Neural Networks (GCNs) [24] and Graph Attention Networks (GAT) [25]. Wei et al. [6] utilize GCNs to encode structural contexts of propagation, enhancing information representations. Bian et al. [7] propose the bi-GCN model, which addresses both propagation and dispersion of information. Structural features are extracted from misinformation dispersion through the aggregation of neighbors. Hu et al. [26] also consider factors from sibling nodes. These studies modeled interaction networks at a local level, without considering the evolution of the whole network. Thus, some researchers [10] model interaction networks as multiple snapshots in uniform time units, which can capture the evolution features related to the propagation structure of disinformation across different stages. However, the uniform division of information propagation structure is limited to mine the non-regular propagation features. Thus, our proposed Multi-Stage Disinformation Detection (MsDD) method explores dependencies from a multi-stage viewpoint. It learns structural and sequential feature representations from each stage's propagation sub-graph and extracts evolving features from all stage sub-graphs.

6 Conclusion

This study proposes a multi-stage dynamic analysis model with graph entropy guidance that can divide stages appropriately for disinformation detection. We first propose a multi-stage awareness algorithm to divide stages of information propagation with graph entropy based on node degree. Then we propose a single-stage embedding module to capture the evolution structure and sequence features in each stage through graph and sequence attention neural network. The multi-stage dynamic learning module can support evolving pattern learning on all dynamic propagation stages. Extensive experiments on two different datasets demonstrate that our model achieves better performance. Ablation studies validate the effectiveness of

the designed multi-stage propagation sub-graph. In addition, a further visualization analysis of multi-stage guided by graph entropy also demonstrates the superiority of our proposed method. In the future, we plan to improve the graph entropy definition by utilizing the content feature of information. This may better reflect the real information propagation scenarios.

Acknowledgements This work is supported by National Key R&D Program of China under Grants No. 2022YFB3104300, National Natural Science Foundation of China under Grants No. 61972087, Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No. BM2003201, and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No. 93K-9.

Author Contributions **Xiaorong Hao:** Methodology, Formal analysis, Visualization, Writing - Original draft, Writing - Review & editing. **Bo Liu:** Supervision, Conceptualization, Writing - Review & editing. **Xinyan Yang:** Methodology, Validation, Visualization, Data curation. **Xiangguo Sun:** Writing - Review & editing. **Qing Meng:** Writing - Review & editing. **Jiuxin Cao:** Writing - Review & editing.

Funding This work is supported by National Key R&D Program of China under Grants No. 2022YFB3104300, National Natural Science Foundation of China under Grants No. 61972087, Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No. BM2003201, and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No. 93K-9.

Availability of Supporting Data All data that support the findings of this study are openly available. The Pheme datasets are included in the published article [4]. The MisInfected datasets are available at <https://weibo.com/weibopiyao>.

Declaration

Competing interests All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

Ethical Approval and Consent to participate Not applicable.

Human and Animal Ethics Not applicable

Consent for publication Not applicable

References

1. Tsallis, C.: Entropy. *Encyclopedia* **2**(1), 264–300 (2022)
2. Anand, K., Krioukov, D., Bianconi, G.: Entropy distribution and condensation in random networks with a given degree distribution. *Phys. Rev. E* **89**(6), 062807 (2014)
3. Schmid, J., Jr.: The relationship between the coefficient of correlation and the angle included between regression lines. *J. Educ. Res.* **41**(4), 311–313 (1947)
4. Zubiaga, A., Liakata, M., Procter, R., Bontcheva, K., Tolmie, P.: Crowdsourcing the annotation of rumourous conversations in social media. In: *Proceedings of the 24th International Conference on World Wide Web (The Web Conference)*, pp. 347–353 (2015)
5. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: Attention-based convolutional approach for misinformation identification from massive and noisy microblog posts. *Comput. Secur.* **83**, 106–121 (2019)
6. Wei, P., Xu, N., Mao, W.: Modeling conversation structure and temporal dynamics for jointly predicting rumor stance and veracity. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4789–4800 (2019)
7. Bian, T., Xiao, X., Xu, T., Zhao, P., Huang, W., Rong, Y., Huang, J.: Rumor detection on social media with bi-directional graph convolutional networks. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 549–556 (2020)

8. Khoo, L.M.S., Chieu, H.L., Qian, Z., Jiang, J.: Interpretable rumor detection in microblogs by attending to user interactions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8783–8790 (2020)
9. Lao, A., Shi, C., Yang, Y.: Rumor detection with field of linear and non-linear propagation. In: Proceedings of the Web Conference 2021, pp. 3178–3187 (2021)
10. Choi, J., Ko, T., Choi, Y., Byun, H., Kim, C.-k.: Dynamic graph convolutional networks with attention mechanism for rumor detection on social media. *Plos one* **16**(8), 0256039 (2021)
11. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: Proceedings of the 20th International Conference on World Wide Web (The Web Conference), pp. 675–684 (2011)
12. Qazvinian, V., Rosengren, E., Radev, D., Mei, Q.: Rumor has it: identifying misinformation in microblogs. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1589–1599 (2011)
13. Yang, F., Liu, Y., Yu, X., Yang, M.: Automatic detection of rumor on sina weibo. In: Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, pp. 1–7 (2012)
14. Liang, G., He, W., Xu, C., Chen, L., Zeng, J.: Rumor identification in microblogging systems based on users' behavior. *IEEE Trans. Comput. Soc. Syst.* **2**(3), 99–108 (2015)
15. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B.J., Wong, K.-F., Cha, M.: Detecting rumors from microblogs with recurrent neural networks. In: 25th International Joint Conference on Artificial Intelligence (IJCAI), pp. 3818–3824 (2016)
16. Guo, H., Cao, J., Zhang, Y., Guo, J., Li, J.: Rumor detection with hierarchical social attention network. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM), pp. 943–951 (2018)
17. Yu, J., Jiang, J., Khoo, L.M.S., Chieu, H.L., Xia, R.: Coupled hierarchical transformer for stance-aware rumor verification in social media conversations. Association for Computational Linguistics (2020)
18. Saxena, A., Hsu, W., Lee, M.L., Leong Chieu, H., Ng, L., Teow, L.N.: Mitigating misinformation in online social network with top-k debunkers and evolving user opinions. In: Companion Proceedings of the Web Conference 2020, pp. 363–370 (2020)
19. Ghai, A., Kumar, P., Gupta, S.: A deep-learning-based image forgery detection framework for controlling the spread of misinformation. *Inf. Technol, People* (2021)
20. Khattar, D., Goud, J.S., Gupta, M., Varma, V.: Mvae: multimodal variational autoencoder for fake news detection. In: The World Wide Web Conference, pp. 2915–2921 (2019)
21. Liu, Y., Wu, Y.-F.: 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, vol. 32 (2018)
22. Liu, B., Sun, X., Meng, Q., Yang, X., Lee, Y., Cao, J., Luo, J., Lee, R.K.-W.: Nowhere to hide: online rumor detection based on retweeting graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.*(TNNLS), (early access), 1–12 (2022)
23. Ma, J., Gao, W., Wong, K.-F.: Rumor detection on twitter with tree-structured recursive neural networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL) (2018)
24. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. [arXiv:1312.6203](https://arxiv.org/abs/1312.6203) (2013)
25. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. *stat* **1050**, 20 (2017)
26. Hu, D., Wei, L., Zhou, W., Huai, X., Han, J., Hu, S.: A rumor detection approach based on multi-relational propagation tree. *J. Comput. Res. Dev.* **58**(7), 1395 (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.