Journal of Artificial Intelligence Research 82 (2025) 1175-1204 Submitted 08/2024; published 03/2025

Rumor Detection with Adaptive Data Augmentation and Adversarial Training

Ying Wang WANGYING2010@JLU.EDU.CN
Key Laboratory of Symbol Computation and Knowledge Engineering
College of Computer Science and Technology, Jilin University
Changchun, Jilin, China

Fuyuan Ma MAFY23@MAILS.JLU.EDU.CN
College of Artificial Intelligence, Jilin University
Changchun, Jilin, China

Zhaoqi Yang ZQYANG21@MAILS.JLU.EDU.CN
Yaodi Zhu ZHUYD20@MAILS.JLU.EDU.CN
College of Computer Science and Technology, Jilin University
Changchun, Jilin, China

Bo Yang YBO@JLU.EDU.CN
Key Laboratory of Symbol Computation and Knowledge Engineering
College of Computer Science and Technology, Jilin University
Changchun, Jilin, China

Pengfei Shen SHEN_PF@QQ.COM
(Corresponding author)
Lei Yun YL@CEPREI.COM
(Corresponding author)
Key Laboratory Ministry of Industry and Information Technology
China Electronic Product Reliability and Environment Testing Research Institute
Guangzhou, Guangdong, China

Abstract

Rumors are widely spread on social media, which has a negative impact on social stability. To address this problem, many rumor detection methods have been proposed. However, most existing methods overlook the potential impact of noise and adversarial attacks on their detection performance, which could compromise their effectiveness when applied in an unknown environment. To overcome these challenges and improve the framework robustness to noise and adversarial attacks, we propose a novel rumor detection framework with Adaptive Data Augmentation and Adversarial Training, named ADAAT. Our framework utilizes the adaptive data augmentation module to calculate the importance of edges and features and adaptively modify the less important among them with a greater probability. In addition, it contains a hard sample generation module which generates adversarial representations through adversarial training. These adversarial representations are treated as hard samples, which are utilized in contrastive learning to learn essential features, thereby improving the robustness of the framework. Our framework proves superiority in rumor detection tasks, increasing the accuracy by an average of 3.6%, 4.5% and 2.5% over the state-of-the-art methods on Twitter15, Twitter16 and PHEME, respectively. When the ADAAT framework is applied to attacked test data, the detection accuracy decreases by only 1.3%, 1.4%, and 1.2%.

1. Introduction

With the rapid development of the Internet, social media, such as Twitter, Weibo, and Facebook (Aljabri et al., 2023), have become the main platforms for people to obtain information and communicate with others. However, due to their popularity and the convenience with which information can be spread, rumors are widely disseminated across the Internet. A large number of widely disseminated rumors mislead public cognition, seriously disrupt the harmony of the Internet and can even undermine social stability. Therefore, it is imperative to develop an effective framework for detecting rumors in social media and mitigating the damage caused by them.

Traditional methods for detecting rumors rely on feature engineering, which involves manually extracting features from user profiles, textual content, sentiment and other factors. These approaches include Decision Tree (Pathak et al., 2020), Support Vector Machine (SVM) (Pisner & Schnyer, 2020) and Latent Dirichlet Allocation (LDA) (Alsaif & Aldossari, 2023). Feature engineering can be time-consuming, resource-intensive and difficult to apply to high-dimensional data. In recent years, researchers have increasingly focused on deep learning methods for rumor detection, enabling the extraction of deep semantic information from events, such as user features, textual content, and propagation structures. These methods, variants of Long Short-Term Memory (LSTM) (Lindemann et al., 2021), Gated Recurrent Unit (GRU) (Joseph et al., 2022) and Recurrent Neural Network (RNN) (Ma et al., 2018) capture temporal features in the propagation paths of events. Additionally, Convolutional Neural Networks (CNNs) (Yu et al., 2017) can learn local spatial representations of event propagation paths, but they struggle to capture global information. To better capture structures in event propagation, some researchers have explored Graph Neural Networks (GNNs) (Zheng et al., 2022), which simultaneously process features and structural information for rumor detection.

Rumor detection methods based on GNNs (Bian et al., 2020; Wei et al., 2021; Liu et al., 2022; Zeng et al., 2023) are able to aggregate information based on the interrelationships between events. However, most of these methods have the following problems: (1) **Noise interference** in the data is rarely considered by existing rumor detection methods. Only a few methods (Hamilton et al., 2017; Yang et al., 2022; He et al., 2021; Adjeisah et al., 2023) that use data augmentation consider augmenting edges without taking into account node features. Moreover, these methods commonly use uniform sampling operations when augmenting graph structures, which might remove edges that are more important in the propagation process. This results in suboptimal performance of rumor detection. (2) These rumor detection methods do not consider the robustness of neural networks against **adversarial attacks** (Ghaffari Laleh et al., 2022; Wu et al., 2022; Lee & Han, 2023; Xu et al., 2023). Rumors on real-world social media platforms may be subject to adversarial attacks, deliberately forwarded or responded to by malicious users or bots. This can alter the spread and commentary of rumors, facilitating the dissemination of misinformation. Additionally, malicious users may engage in adversarial interference by misspelling words or using high-frequency vocabulary on purpose. These artificially designed adversarial perturbations can fool neural networks, reducing rumor detection performance and hindering the practical application of these methods. Therefore, it is urgent to propose a rumor detection framework that is **robust to noise and adversarial attacks**.

To overcome the above challenges, we propose a rumor detection framework named Rumor Detection with Adaptive Data Augmentation and Adversarial Training (ADAAT). Specifically, we introduce the adaptive data augmentation module, which utilizes node centrality to calculate the importance of edges and features respectively. The edges and features in the events are adaptively modified according to their importance, the higher the importance of the edges and features, the lower the probability of being modified. Thus, the diversity of data is enriched without affecting the labels of the events themselves, which helps the framework learn noise-insensitive representations during the training process. In addition, to counter adversarial perturbations designed by humans in reality, we introduce a hard sample generation module to enhance the robustness of the framework. It generates adversarial representations through adversarial training and utilizes these representations as hard samples. In contrastive learning, samples with the same label as the anchor point are treated as positive hard samples, while those with different labels are considered negative hard samples. They are combined with contrastive learning to compel the framework to learn invariant essential features from the data, thereby improving its robustness against adversarial attacks.

Our contributions can be summarized as follows:

- We develop an adaptive data augmentation module to enrich the data and enhance the robustness of our framework. This module endeavors to preserve important edges and features as much as possible, ensuring the integrity of significant information.
- We propose a hard sample generation module that utilizes Fast Gradient Method (FGM) and Projected Gradient Descent (PGD) for adversarial training to generate adversarial representations. By integrating this module with contrastive learning, our neural network can learn more invariant features from the data.
- We conduct experiments on three real-world social media rumor detection datasets. Experiments demonstrate that our model has superior rumor detection capabilities compared to baseline methods and shows good robustness against perturbations.

The remainder of this paper is structured as follows: Section 2 introduces the related work. Section 3 provides the definitions of symbols. Section 4 elaborates the framework of the proposed the ADAAT. Section 5 presents the results of experimental analyses. In section 6, we summarize the research content of our paper.

2. Related Work

2.1 Rumor Detection Methods

Traditional methods for detecting rumors have primarily focused on manually extracting content information from rumors and summarizing various statistical features. However, machine learning methods struggle to effectively learn high-dimensional and complex features. In recent years, deep learning methods have achieved great performance in rumor detection tasks. For instance, Ma et al. (Ma et al., 2016) utilize RNN to sequentially process each time step in the event propagation sequence for capturing potential temporal semantic information. Chen et al. (Chen et al., 2018) improve this method by combining

the attention mechanism with RNN, giving differential attention to text features of different importance and capturing more long-term dependencies. Selvaraj et al. (Selvaraj & L D, 2020) propose a dual-CNN classification method with CFNet as activation function for the early detection of rumor events. Wu et al.(Wu et al., 2021) propose that the MCAN model can extract spatial domain features and frequency domain features from images, and text features from text. Then, MCAN utilizes co-attention layers to achieve multimodal feature fusion, and utilizes the obtained representation for rumor detection.

However, most of these deep learning methods only focus on learning time and text features, ignoring the information contained in the event propagation structure. To utilize the structural information of event propagation, Bian et al. (Bian et al., 2020) further improve upon this method by using Graph Convolutional Networks (GCNs). Additionally, Wei et al. (Wei et al., 2021) utilize an Edge-enhanced Bayesian Graph Convolutional Network to explore the uncertainty of message event propagation for learning higher quality event representations. Liu et al. (Liu et al., 2019) propose a semi-supervised learning framework for co-detecting crowdturfing spammers and microblogs by modeling user behavior, content, and networks to address label sparsity on real-world datasets. Hao et al. (Hao et al., 2024) propose a framework named MsDD for dynamic disinformation detection using graph entropy to identify variable-length propagation stages, enabling accurate and timely detection. Liu et al. (Liu et al., 2022) propose a novel self-attention-based retweeting neural network to learn individual features from both content and users. By fusing node-level features with our global fabric embedding, the framework can generate a more comprehensive representation for rumor detection. Mosallanezhad et al. (Mosallanezhad et al., 2022) integrate ancillary information, such as user comments and user-news interactions, and leverage adaptive reinforcement learning models to address the limitations of existing automated fake news detection models. Zhou et al (Zhou et al., 2022) apply a Multi-task Sharing Layer, a task-specific Transformer Encoder and a Selection Layer to improve the diversity and stability of textual representation for rumor detection. Sun et al. (Sun et al., 2023d) propose a rumor detection framework integrating content, propagation structures, and temporal relations for online temporal networks. Zhang et al. (Zhang et al., 2022b) introduce a memory-efficient, self-supervised transformer-based model named GMAEs for graph representation learning, addressing training and memory challenges. Sun et al.(Sun et al., 2023c) proposes a hypergraph-based framework that integrates data mining techniques with sociological behavioral criteria. Xu et al.(Xu et al., 2022) propose a hierarchical aggregation graph neural network model that focuses on capturing different granularities of high-level representations of text content and fusing rumor propagation structures. The model applies a graph convolutional network (GCN) with a rumor propagation graph to learn a textual granular representation of event propagation, which helps form a final representation of events to detect rumors.

2.2 Data Augmentation

Data augmentation (Zhao et al., 2021; Zhu et al., 2021; Ding et al., 2022; Liu et al., 2023) is a technique utilized to generate additional data from a limited sample set, with the goal of increasing the diversity and robustness of machine learning models. For example, Hong et al. (Hong et al., 2021) propose the StyleMix method to improve the accuracy of

image classification by generating new training samples through the combination of content and style features. While initially popularized in the field of Computer Vision (CV), more recent advances in GNNs have led to an increased interest in data augmentation for graph data. For example, GraphMAE, proposed by Hou et al. (Hou et al., 2022) leverages a masking strategy and scaling cosine error to improve feature reconstruction and model robustness. Similarly, Luo et al. (Luo et al., 2022) automate graph data augmentation through techniques such as node feature masking, node dropping, and edge perturbations to generate augmented graphs suitable for downstream tasks. Tan et al. (Tan et al., 2022) introduce random edge masking and reconstruction to improve self-supervised tasks and downstream task performance. In this work, we adopt an approach to adaptively augment graph data by modifying edges and masking features, specifically.

2.3 Contrastive Learning

Contrastive learning aims to learn a projection representation space, so that the same samples are close to each other, and different samples are pulled away from each other. It has been successfully applied to various tasks in multiple domains. For instance, Gao et al. (Gao et al., 2021) apply both unsupervised and supervised contrastive learning to natural language processing to enhance downstream task performance, as measured by the Spearman's correlation coefficient. Wang et al. (Wang & Qi, 2023) utilize contrastive learning to retrieve strongly augmented queries from a set of instances by comparing the distribution difference between weakly and strongly augmented images. Zhu et al. (Zhu et al., 2021) use correlated view pairs after data augmentation as inputs and train the model via contrastive loss to improve node representation quality. Zhang et al. (Zhang et al., 2022c) construct contrastive loss using the generated same-category augmented views as anchor points and positive samples, and different category augmentation views as negative samples, which is applied to graph tasks. Feng et al. (Feng et al., 2022) generate adversarial samples from the original clean graph and calculate the contrastive loss between the adversarial samples and the graph after data augmentation to enhance robustness of the model. Sun et al. (Sun et al., 2023b) propose a framework for designing robust scoring systems without ground truth, using a counter-empirical attacking mechanism and adversarial learning to iteratively improve scoring functions by ensuring compliance with empirical criteria. Moreover, supervised contrastive learning (Xue et al., 2023) is employed to further improve robustness of the model, where the final loss function comprises both contrastive loss (Wang et al., 2022; Huang et al., 2022) and supervised cross-entropy loss (Zhang et al., 2022a; Wang & Jang, 2023; Sun et al., 2023a).

3. Problem Definition

Rumor detection is commonly regarded as a classification task, where the goal is to develop a classifier. It can identify whether a given source post is a rumor or not, based on a set of events that includes the source posts and corresponding responses. Formally, we consider a rumor detection event set $C = \{c_1, c_2, \cdots, c_m\}$, where c_i is the i-th event and m is the number of events. Each event c_i in C consisting of a set of related tweets $c_i = r_i, r_{i0}, ..., r_{im}$. Here, r_i represents the source tweet, r_{i*} represents a comment (response) tweet related to the source tweet r_i, and m denotes the number of relevant comment tweets under the

Table 1: Summary of the main symbols

Description	Symbols	Detailed Descriptions
Rumor detection event set	$C = \{c_1, c_2, \cdots, c_m\}$	
Event	$c_i = \{G_i, y_i\}$	
The label	$y_i \in \{N, F, T, U\}$	
Propagation graph	$G_i = (V_i, E_i)$	
Node set	$V_i = \{v_{i,1}, v_{i,2}, v_{i,3}, \cdots, v_{i,n_i}\}$	n_i represents the number of posts
A set of directed edges	$E_i = \{e_{i.st} \mid s, t = 0, 1, 2, \cdots, n_i - 1\}$	$e_{i,st}$ denotes an edge from node s to node t
Adjacency matrix	A_i	
Feature matrix	X_i	
The importance of edge	$p^e_{i,st}$	
The importance of feature	$p^x_{i,b}$	
The augmented propagation graph	$G_i^{'}$	
The augmented adjacency matrix	$A_i^{'}$	
The augmented feature matrix	$X_i^{'}$	
A Top-Down adjacency matrix	$A_i^{'TD}$	
A Bottom-up adjacency matrix	$A_i^{'BU}$	
Averaged ponding operation	$MEAN()$	
Final representation	$z_{i,sum}$	Avector representation uses for classification

event c_i. We treat tweets within the event c_i as nodes, encodes the text within the tweets as vectors to serve as node features, and considers the comment (response) relationships between tweets as edges to construct the graph G_i. $G_i = (V_i, E_i)$ represents the propagation structure of the event, where $V_i = \{v_{i,1}, v_{i,2}, v_{i,3}, \cdots, v_{i,n_i}\}$ is the node set, $v_{i,j}$ refers to the j-th retweet and n_i represents the number of posts. $E_i = \{e_{i.st} \mid s, t = 0, 1, 2, \cdots, n_i - 1\}$ represents a set of directed edges, where $e_{i,st}$ denotes an edge from node s to node t. We denote $A_i \in \mathbb{R}^{n_i \times n_i}$ as an adjacency matrix that captures the initial relationship between nodes in the propagation graph.

$$a_{i,st} = \begin{cases} 1, & if\ e_{i,st} \in E_i \\ 0, & otherwise \end{cases}, \tag{1}$$

where $a_{i,st}$ denotes a element of A_i. The graph construction process for a single event instance is illustrated in Figure 1. Part (a) of the figure visualizes an event instance from the Twitter dataset. An event includes a source tweet and related comment tweets. Each tweet includes its corresponding text content, as well as information about the user who posts it. The comment relationships between tweets are represented by directed arrows. The parts (b), (c), and (d) of the figure illustrate the three steps of transforming the event from the raw tweet information into a graph representation. Part (b) corresponds to step one, where tweets are treated as nodes and assigned a unique node ID. Part (c) corresponds to step two, where the text content of a tweet is passed through a pre-trained BERT model to extract semantic features from the text, converting the textual information into numerical vectors to be used as node features. Part (d) corresponds to step three, where the comment (response) relationships between tweets are used as edges to connect the nodes and construct the graph.

Each event c_i is related to a ground-truth label $y_i \in \{T, F\}$ (i.e. False Rumor or True Rumor). In some datasets, the label is defined more specifically as $y_i \in \{N, F, T, U\}$, corresponding to Non-rumor, False Rumor, True Rumor, and Unverified Rumor, respectively.

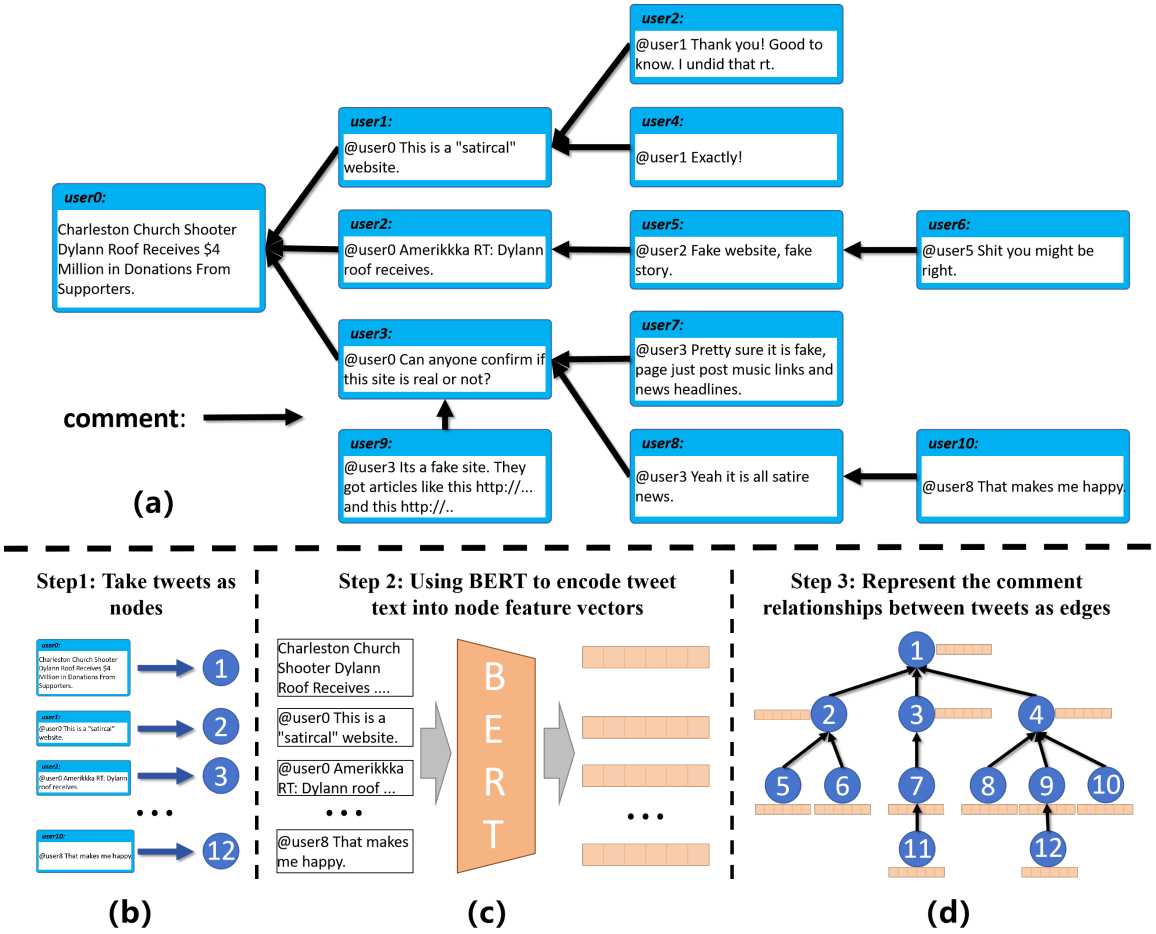

Figure 1: Visualization of the graph construction for an event instance.

The classification task for rumor detection aims to learn a classifier that assigns events represented as graphs to their corresponding categories. In this paper, we use a classifier implemented using a graph neural network extracts both semantic features and propagation patterns from the rumor information to perform classification:

$$f : G_i \rightarrow y_i, \tag{2}$$

The task of rumor detection involves using a trained classification model to identify rumor events from events of unknown authenticity. We summarize and explain the symbols used in the paper, as detailed in Table 2.

4. The Proposed Framework

This paper proposes a framework for rumor detection based on GCNs, which combines adaptive data augmentation and adversarial training, named ADAAT. The overall architecture of our framework is illustrated in Figure 2. Specifically, our framework first employs an adaptive data augmentation module to generate more diverse data without changing the labels

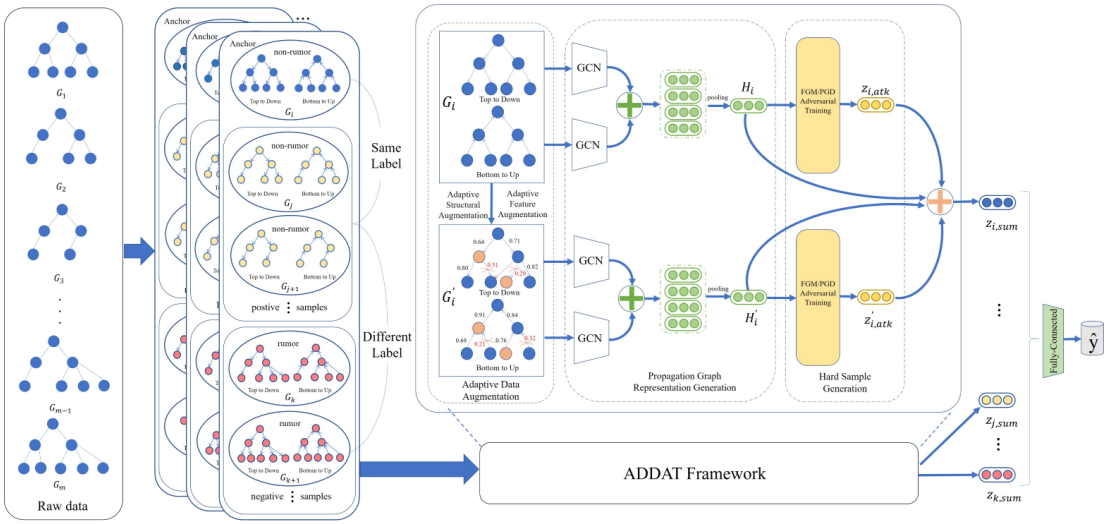

Figure 2: Overview of ADAAT.

of the events themselves, which helps the framework learn noise-insensitive representations during the training process. Second, we utilize the propagation graph representation generation module to capture both the Top-Down and Bottom-Up propagation graphs, thereby obtaining the overall representation of events. Third, a hard sample generation module is introduced to generate adversarial representations through adversarial training. These representations are treated as hard samples in contrastive learning to help the framework learn invariant essential features from data and improve the robustness of the framework.

4.1 Adaptive Data Augmentation

Applying data augmentation in the rumor propagation graph can modify the edges and features to increase the diversity of the training data, and improve the generalization and robustness of the rumor detection model. Most of the existing data augmentation methods randomly delete, misarrange edges or mask node features of a certain dimension with a fixed probability. Utilizing them on the rumor propagation graph will damage the propagation structures and affect the performance of rumor detection. For example, randomly deleting edges may cause important edges to be deleted, resulting in the destruction of normal propagation structure and the decrease of rumor detection performance. To solve this problem, we propose an adaptive data augmentation module that simultaneously augments the graph structure and features. It modifies unimportant edges and features with a greater probability, while retaining important edges and features as much as possible to ensure the integrity of the propagation graph. Specifically, structural data augmentation adaptively deletes and misarranges edges, and feature data augmentation adaptively masks features in some dimensions.

For structural data augmentation, we utilize probability $P_{i,st}$ to represent the likelihood that the propagation structure is preserved. It is defined as follows:

$$P_{i,st} = 1 - p^e_{i,st}, \tag{3}$$

where $p^e_{i,st}$ reflects the importance of edge $e_{i,st}$, the more important the edge, the smaller the value of $p^e_{i,st}$.

In the rumor propagation graph G_i, edge $e_{i,st}$ represents the relationship between node s and node t. This importance is closely related to the centrality of the nodes. To measure the importance $p^e_{i,st}$ of the edge $e_{i,st}$, we utilize its edge centrality $\omega^e_{i,st}$, which is calculated based on the centrality of its neighboring nodes $\Theta_i(\cdot)$. In this paper, degree centrality is utilized to measure node centrality $\Theta_i(\cdot)$. Specifically, edge centrality $\omega^e_{i,st}$ is defined as follows:

$$\omega^e_{i.st} = \frac{\Theta_i(s) + \Theta_i(t)}{2}. \tag{4}$$

In order to reduce the influence of nodes with high degree on edge importance, adaptive data augmentation module sets $k_{i,st} = log(\omega^e_{i.st})$. Then, the edge centrality is normalized to obtain the importance $p^e_{i,st}$ of each edge, which is defined as:

$$p^e_{i,st} = min\left(\frac{k_{i,max} - k_{i,st}}{k_{i,max} - k_{i,avg}} \times \rho^e, p^e_t\right), \tag{5}$$

where $k_{i,max}$ and $k_{i,avg}$ represent the maximum and average values of $k_{i,st}$, respectively. ρ^e is a hyperparameter that controls the probability of deleting and misarrange edges, and p^e_t is the truncation probability used to prevent excessive deletion and misarrangement of edges from affecting the structure of the rumor propagation graph.

For the adaptive misarrangement of edges, we select two edges $e_{i,st}$ and $e_{i,s't'}$ with large deletion probabilities $p^e_{i,st}$ and $p^e_{i,s't'}$ in the propagation graph, and misarrange these edges as $e_{i,st'}$ and $e_{i,s't}$. The probabilities are calculated in the same way as the adaptive deletion of edges.

For feature data augmentation, we first sample to obtain a masking vector $Q_i \in \{0,1\}^F$, where F denotes the dimension of the feature X. The values of each dimension are drawn independently from the same Bernoulli distribution, $Q_{i,b} \sim Bren(p^x_{i,b}), \forall b \in \{1, 2, \cdots, F\}$, where $p^x_{i,b}$ represents the importance of the b-th dimension feature. The new feature matrix $X_{i,ad}$ can be generated by this masking vector Q_i, which can be defined as:

$$X_{i,ad} = [x_1 \circ Q_i; x_2 \circ Q_i; \cdots; x_{n_i} \circ Q_i]^T, \tag{6}$$

where $[\cdot;\cdot]$ represents the concatenation operator, $\circ$ denotes the element-wise multiplication, and x_j denotes the feature vector of node j, $j \in \{1, 2, \cdots, n_i\}$.

To compute $p^x_{i,b}$, we first need to compute the importance weight $\omega^x_{i,b}$ of the b-th dimension feature. Features in the datasets are continuous. It can be assumed that feature dimensions which occur frequently in nodes of high importance and have large absolute values are relatively important. Hence, the importance weight $\omega^x_{i,b}$ of the b-th dimension feature can be formally defined as follows:

$$\omega^x_{i,b} = \sum_{j \in n_i} |x_{i,jb}| \cdot \Theta_i(j), \tag{7}$$

where $|x_{i,jb}|$ represents the absolute value of the b-th dimensional feature of node j. $\Theta_i(\cdot)$ represents the centrality of nodes j.

Similar to structural data augmentation, we set $s^x_{i,b} = log(\omega^x_{i,b})$ and normalize it to obtain $p^x_{i,b}$, which represents the importance of the b-th dimension feature. It can be formally defined as follows:

$$p^x_{i,b} = min\left(\frac{s_{i,max} - s^x_{i,b}}{s_{i,max} - s^x_{i,avg}} \times \rho^x, p^x_t\right), \tag{8}$$

where $s_{i,max}$ and $s_{i,avg}$ represent the maximum and average values of $s^x_{i,b}$, respectively. ρ^x is the hyperparameter that controls the amplitude of node feature augmentation, and p^x_t is the truncation probability to prevent excessive destruction of node attributes.

After adaptive data augmentation, we get the augmented propagation graph $G^{'}_i$. The adjacency matrix A_i from $G^{'}_i$ is converted into $A^{'}_i$ and the feature maxtrix X_i is converted into $X^{'}_i$, correspondingly.

4.2 Propagation Graph Representation Generation

Rumors on social media not only contain textual content, but also have structural information. The spread and diffusion of tweets are crucial aspects of an event, and the propagation pattern is vital for determining the nature of the event. The information dissemination process yields distinct information when moving from the original tweet outward and from retweet endpoints back to the original tweet, necessitating the modeling of these distinct directions separately. We utilize the approach similar to previous work (Bian et al., 2020) to obtain the graph representation. Both the original propagation graph G_i and the augmented propagation graph $G^{'}_i$ are built respectively as a Top-Down propagation graph and a Bottom-Up propagation graph, and are applied into GCNs for obtaining the propagation graph representation.

Taking the original propagation graph $G^{'}_i$ as an example, we divide it into two parts: a Top-Down rumor propagation graph (TDGraph) $G^{'TD}_i$ and a Bottom-Up rumor propagation graph (BUGraph) $G^{'BU}_i$. The adjacency matrices of two propagation graphs are different. For TDGraph, the corresponding adjacency matrix is $A^{'TD}_i = A^{'}_i$. For BUGraph, the corresponding adjacency matrix is $A^{'BU}_i = A^{'T}_i$. The feature matrix $X^{'}_i$ represents the encoding of the text content corresponding to the nodes and is pre-trained using Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019).

GCNs apply convolution operation to graph structure and utilize it to aggregate neighbor node information to obtain high-quality representation, which have excellent ability to aggregate graph data information. We utilize them to obtain the representation of the graph, and the formula is defined:

$$H_{k+1} = \sigma(\hat{A}H_kW_k), \tag{9}$$

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}, \tag{10}$$

$$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}, \tag{11}$$

$$\tilde{A} = A + I, \tag{12}$$

where I represents the identity matrix, $\tilde{A}$ represents the adjacency matrix with self-connection, $\tilde{D}_{ii}$ represents the degree matrix of $\tilde{A}$ and $\hat{A}$ represents the normalized adjacency matrix. W_k represents the learnable weight of the GCNs at layer k. H_k denotes the representation of the k-th hidden layer, where $H_0 = X$, and X is the initial feature matrix. σ denotes the RELU activation function.

In this paper, a two-layer graph convolutional network is used to generate the Top-Down and Bottom-Up representation of the propagation graph, which is formulated as follows:

$$H_{i,2}^{'TD} = \sigma\left(\tilde{A}_i^{'TD}\sigma\left(\tilde{A}_i^{'TD}X_i^{'}W_0^{TD}\right)W_1^{TD}\right), \tag{13}$$

$$H_{i,2}^{'BU} = \sigma\left(\tilde{A}_i^{'BU}\sigma\left(\tilde{A}_i^{'BU}X_i^{'}W_0^{BU}\right)W_1^{BU}\right). \tag{14}$$

Finally, the mean pooling operation is utilized to aggregate the representation of all nodes in the propagation graph $G_i^{'}$. The graph representation $H_i^{'}$ of the propagation graph $G_i^{'}$ is generated, which is defined as follows:

$$H_i^{'TD} = MEAN(H_{i,2}^{'TD}), \tag{15}$$

$$H_i^{'BU} = MEAN(H_{i,2}^{'BU}), \tag{16}$$

$$H_i^{'} = H_i^{'TD} \oplus H_i^{'BU}, \tag{17}$$

where $\oplus$ denotes additive operation.

4.3 Hard Samples Generation

Rumors in the open space of social media contain not only noise perturbations, but also adversarial perturbations crafted by malicious users. Adversarial perturbations trick the rumor detection model so that a rumor is incorrectly detected as non-rumor, causing the rumor to continue to spread. For example, rumor publishers can deliberately misspell some words to inject adversarial perturbations, causing wrong detection results. These publishers can also remove responses against rumors, making rumors close to non-rumors, thereby fooling deep neural networks in detection models. Malicious users also use bots to post induced comments containing specific high-frequency words, changing the stance of replying content. These examples are shown in Figure 3. The red bold text in the figure represents the change in content after the attack. A red cross indicates that the edge is maliciously deleted. The red dotted line represents the increased induced comments. Although the adaptive data augmentation module is utilized, it cannot solve the problem of degraded detection performance when the framework is applied on data with adversarial perturbations. To solve this problem, we propose a hard sample generation module which utilizes adversarial training to simulate real-world adversarial attacks and generate adversarial representations. These adversarial representations are used as hard samples in contrastive learning to force the framework to learn the invariant essential features in the data and improve the robustness of the framework against adversarial attacks.

In this paper, Fast Gradient Method (FGM) (Miyato et al., 2017) and Projection Gradient Descent (PGD) (Madry et al., 2018) are utilized for adversarial training on the graph representations H_i and $H_i^{'}$ obtained in the propagation graph representation generation

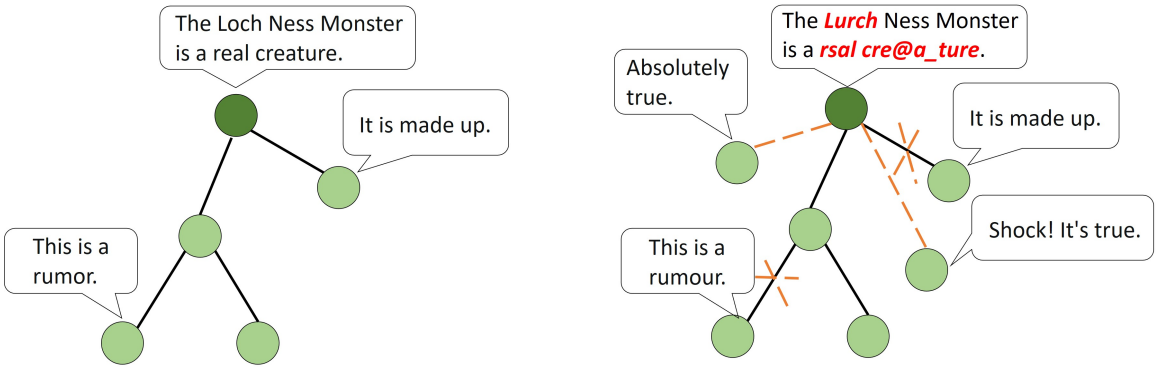

Figure 3: Comparison of normal and attacked event propagation processes.

module. H_i and $H_i^{'}$ generate adversarial representations through a fully connected layer with adversarial training and introduce them as hard samples into contrastive learning.

Without loss of generality, take the graph representation $H_i^{'}$ of the propagation graph $G_i^{'}$ as an example. The FGM method takes the $H_i^{'}$ as input. Its loss ℓ for forward propagation and its gradient g for back propagation are computed through the attacked neural network. The formula is defined as follows:

$$g = \nabla_{H_i^{'}} \ell(\phi, H_i^{'}, y_i), \tag{18}$$

where ℓ represents the loss function and ϕ_i represents the learnable weights. Based on the computed gradient g, adversarial perturbation r_{atk} is generated and added to the representation $H_i^{'}$, which is formulated as follows:

$$r_{atk} = \gamma \frac{g}{\|g\|_2}, \tag{19}$$

$$H_{i,atk}^{'} = H_i^{'} + r_{atk}, \tag{20}$$

where γ is a hyperparameter that controls the perturbed size. $\|\cdot\|_2$ denotes the l_2 norm.

Then, according to the obtained $H_{i,atk}^{'}$, the loss function of forward propagation and the adversarial gradient g_{atk} of back propagation. The g_{atk} is summed up with the gradient g to get g_{sum}, whose formula is as follows:

$$g_{atk} = \nabla_{H_{i,atk}^{'}} \ell(\phi, H_{i,atk}^{'}, y_i), \tag{21}$$

$$g_{sum} = g_{atk} + g. \tag{22}$$

Finally, the learnable weight ϕ is updated according to g_{sum} and the initial $H_i^{'}$.

The FGM method generates the adversarial perturbation in one step, while the PGD method splits the final adversarial perturbation generation process into multiple small steps, and sets the total number of small steps to T. PGD first generates adversarial samples r_{atk}^{t+1}

based on the gradient g^{t+1} calculated in small step t. Then, r_{atk}^{t+1} is added to the graph representation $H_i^{'t}$ to generate the representation $H_{i,atk}^{'t+1}$, which is defined as:

$$H_{i,atk}^{'t+1} = H_i^{'t} + r_{atk}^{t+1}. \tag{23}$$

Finally, if t is not the last step, the gradient obtained in the previous step is not considered in the calculation and the new gradient $g_{atk}^{t+1}$is directly calculated. If t is the last step, the initial calculated gradient is added to obtain the final perturbed gradient g_{atk}^{T}, whose formula is as follows:

$$\begin{cases} g_{atk}^{t+1} = \nabla_{H_{i,atk}^{'t+1}} \ell(\phi_i, H_{i,atk}^{'t+1}, y_i), & if t < T-1 \\ g_{atk}^{T} = \nabla_{H_{i,atk}^{'t+1}} \ell(\phi_i, H_{i,atk}^{'t+1}, y_i) + g^0, & if t = T-1 \end{cases}. \tag{24}$$

According to the calculated gradient g_{atk}^{T} and the initial representation $H_i^{'0}$, the learnable weight ϕ of the rumor detection model is updated, an adversarial attack process is completed.

We utilize H_i and $H_i^{'}$ as inputs to obtain the adversarial representation of the graph through the fully connected layer and the normalized layer. The learnable weights of the fully connected layer are obtained after FGM or PGD attacked, and the formula is expressed as follows:

$$z_{i,atk} = Norm((H_i W_{atk,0} + b_0) W_{atk,1} + b_1), \tag{25}$$

$$z_{i,atk}^{'} = Norm((H_i^{'} W_{atk,0} + b_0) W_{atk,1} + b_1), \tag{26}$$

where W_{atk} represents the learnable weights obtained by the fully connected layer after adversarial training, and b represents the bias. The obtained adversarial representations $z_{i,atk}$ and $z_{i,atk}^{'}$ are used as the hard samples of contrastive learning. Those with the same label as the anchor are hard positive samples, and those that are different are hard negative samples. The hard samples contain more learnable information, which can make the contrastive learning model learn more knowledge and better distinguish between positive and negative samples.

Concatenate $z_{i,atk}$ and H_i as well as $z_{i,atk}^{'}$ and $H_i^{'}$ to obtain z_i and $z_i^{'}$. Then, we can obtain the representation $z_{i,sum}$ of the final input loss function of the event by concatenating z_i and $z_i^{'}$, whose formula is as follows:

$$z_i = concat(z_{i,atk}, H_i), \tag{27}$$

$$z_i^{'} = concat(z_{i,atk}^{'}, H_i^{'}), \tag{28}$$

$$z_{i,sum} = concat(z_i, z_i^{'}), \tag{29}$$

where $concat(\cdot)$ represents the concatenating function. $z_{i,sum}$ is the vector representation when calculating the loss function.

4.4 The Joint Loss Function

We propose a joint loss function that combines cross-entropy loss and contrastive loss to optimize the rumor detection task. The cross-entropy loss is a widely used classification

loss function that leverages label information to enhance classification performance. The contrastive loss in this paper is a supervised contrastive loss, which can effectively learn the features of hard samples like unsupervised contrastive loss, and further improve the detection performance and robustness of the framework. In summary, the joint loss function consists of cross-entropy loss $loss_{entropy}$ and contrastive loss $loss_{conv}$, which is defined as follows:

$$loss = (1-\beta) * loss_{entroy} + \beta * loss_{conv}, \tag{30}$$

where β is a hyperparameter and is the trade-off coefficient that controls the proportion of two loss functions. $loss_{conv}$ shortens the distance between similar samples and widens the distance between different samples. We set the event c_i as the anchor, samples with the same label as the anchor are positive samples, and samples with different labels from the anchor are negative samples. The $loss_{entroy}$ and $loss_{conv}$ are defined as follows:

$$loss_{entroy} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{ca=1}^{|CA|} y_{i,ca} log\hat{y}_{i,ca}, \tag{31}$$

$$\hat{y}_i = softmax(W_i z_{i,sum} + b_i) \tag{32}$$

$$\begin{aligned} loss_{conv} = -\sum_{i\in S_v} log\Bigg\{ \frac{1}{S_p(i)} \sum_{j\in S_p(i)} \\ \frac{exp(\cos(z_{i,sum}, z_{j,sum})/\tau)}{\sum_{k\in S_n(i)} \exp(\cos(z_{i,sum}, z_{k,sum})/\tau)} \Bigg\}, \end{aligned} \tag{33}$$

In the cross-entropy loss function $loss_{entropy}$, CA represents the class of labels, y_i represents the true label of event c_i, and $\hat{y}_{i,ca}$ represents the predicted label of c_i. In the contrastive loss function $loss_{conv}$, S_v is the index set of events. The event c_i corresponding to index i serves as the anchor, whose final representation is $z_{i,sum}$. The index j corresponds to the event c_j, with its final representation denoted as $z_{j,sum}$. Similarly, the event c_j corresponding to index j has the same label as the anchor c_i and is a positive sample, with its final representation denoted as $z_{j,sum}$. The index k corresponds to the event c_k, which is eventually represented as $z_{k,sum}$ and different from the anchor c_i label, which is a negative sample. $S_p(i)$ represents the positive sample set when the event c_i is the anchor, and $S_n(i)$ represents the negative sample set when c_i is the anchor. $\cos(\cdot)$ indicates cosine similarity. τ is the temperature coefficient, which controls the shape of the distribution.

Based on the joint loss, our optimization objective is defined as learning all the trainable parameters of our model by minimizing the loss function:

$$\min_{\theta}(1-\beta) * loss_{entroy} + \beta * loss_{conv} \tag{34}$$

where the cross-entropy loss guides our model to learn representations capable of detecting rumors, and the contrastive loss guides the model to learn representations that can differentiate between samples. The complete training process of the ADAAT model is shown in Algorithm 1.

Algorithm 1 The ADAAT training algorithm

Input: A set of propagation graphs G^{TD} and G^{BU}
Output: The parameter of the classifier model θ_f
Initialize θ_f with random weight values
for epoch $\leftarrow$ 1,2,... **do**
    Generate copies G' using data augmentation
    Generate propagation graph representation H' with Equation (13)-(17)
    Adversarial train the adversarial attack model with Equation (18)-(24)
    Generate adversarial representations z'_{atk} with Equation (25)(26)
    Calculate the final graph representation z using Equation (27)-(29)
    Compute joint loss $loss$ using Equation (30)-(33)
    Backward and optimize the classifier model's parameter: $\theta_f \leftarrow \theta_f - lr\nabla(\theta_f)$
end for
return The parameter of the classifier model θ_f

5. Experiment

In this section, to verify the effectiveness of the ADAAT, we perform comprehensive experiments on three real rumor detection datasets, trying to answer the following five important questions:

Q1. How does the ADAAT compare in performance to other current rumor detection methods across different datasets?

Q2. How well does ADDAT perform in the early rumor detection task compared to baselines?

Q3. How does each proposed module of framework contribute to the performance of ADAAT?

Q4. How does the ADAAT perform when additional perturbations are introduced into the data?

Q5. Is the framework sensitive to some hyperparameters?

5.1 Datasets

We select three public datasets to verify the effectiveness of ADAAT. Table 2 shows the statistics of these datasets, which are detailed as follows:

Table 2: Statistics of the datasets

Statistic	Twitter15	Twitter16	PHEME
(# source tweets)	1,490	818	6,425
(# non-rumors)	374	205	4,023
(# false rumors)	370	205	2,402
(# unverified rumors)	374	203	-
(# true rumors)	372	205	-
(# users)	276,663	173,487	48,843
(# posts)	331,612	204,820	197,852

Twitter15: The Twitter15 dataset is constructed by Ma et al. (Ma et al., 2017). It contains 1490 popular source tweets along with the content and propagation structures of their reposts/responses. This dataset contains four kinds of labels: Non-rumor (N), False rumor (F), True rumor (T), and Unverified rumor (U), with a relatively balanced proportion of each kind of label.

Twitter16: The Twitter16 dataset is constructed by Ma et al. (Ma et al., 2017) and consists of 818 popular source tweets with many reposts/responses. The structure of dataset is similar to Twitter15, but the propagation structures are smaller. It also contains four kinds of labels.

PHEME: The PHEME dataset is constructed by Zubiaga et al. (Zubiaga et al., 2018), which contains discussion information with clear discussion nature and communication structures screened from nine topics closely related to politics and people livelihood. The dataset divides 6,425 events into Rumors (R) and Non-rumors (N) based on rumor detection, stance judgment, and other criteria.

5.2 Baselines

In order to verify the effectiveness of the proposed framework ADAAT on the rumor detection task, it is compared with nine baseline methods. We have used the same experimental settings for all baseline methods as used in our method, and we have set the same number of network layers and hidden layer neurons to ensure that all methods have similar complexity. The details of these methods are as follow:

- DTC (Castillo et al., 2011): Decision Tree uses decision trees as classifiers for rumor detection tasks, utilizing manual features to obtain information credibility.
- SVM-TS (Ma et al., 2017): It models the temporal information sequence of events using manual features, and employs a linear SVM model to capture changes in event context features for rumor detection.
- CNN (Yu et al., 2017): It uses CNN to extract local spatial features of events to distinguish between rumors and non-rumors.
- BERT (Devlin et al., 2019): It is a popular pre-trained model that can extract linguistic features of rumor text from event textual information for rumor detection.
- RvNN (Ma et al., 2018): It is a classifier based on a tree structure that uses RNN to capture the bidirectional propagation structure of event information, learning event representations, and subsequently determining whether an event is a rumor or not.
- GCAN (Lu & Li, 2020): It utilizes a bidirectional attention mechanism to capture features from both the original text and the propagation structure of events, as well as the correlation between the original text and user features, and is capable of generating explanatory information during rumor detection.
- UDGCN (Bian et al., 2020): It applies GCN to the propagation graph corresponding to an event to capture structural patterns during information propagation, and also performs feature enhancement on the root node to improve rumor detection effectiveness.

- BiGCN (Bian et al., 2020): It uses two GCNs to extract structural patterns from the propagation and diffusion processes of an event, and like UDGCN, it also enhances the features of the root node.
- EBGCN (Wei et al., 2021): It employs edge-enhanced Bayesian graph neural networks to capture the propagation structure features of events, adaptively considering the potential relationships between graph nodes and exploring the uncertainty of event propagation for rumor detection.
- ADAAT: ADAAT is the rumor detection method proposed in this paper, which enhances the robustness of the model through an adaptive data augmentation module and an adversarial training module. The paper provides corresponding versions of the model under two adversarial training methods: ADAAT-FGM and ADAAT-PGD.

5.3 Experimental Settings

We adhere to the same experimental settings as other baseline methods. The experimental settings of other baselines are set according to the description of their papers. In this experiment, the datasets are randomly divided into five parts for cross-validation. The training set comprises 80% of the data, while the test set contains the remaining 20%. The learning rate is set to 0.001 for all datasets.

For Twitter15 and Twitter16 datasets, we utilize Accuracy (Acc) and F1-measure to evaluate the effectiveness of all methods. For PHEME, the Accuracy (Acc.), Precision (Prec.), Recall (Rec.) and F1-measure (F1) are adopted as evaluation metrics. We employ an early stopping mechanism to select the converged model, with the convergence metric set to no better metric results for eight consecutive epochs. For the hyperparameters involved in the model, based on practical experience, we set gamma to 0.3 in FGM and beta to 0.01 in final loss.

To maintain consistency, all experiments were conducted on a single hardware platform equipped with 1TB of RAM and an NVIDIA GeForce GTX 1080 Ti GPU, which has 11264MiB of storage.

5.4 Overall Performance

To answer question Q1, we list the rumor detection performance of different methods in Tables 3, 4 and 5, with the best result in bold. The two methods ADAAT-FGM and ADAAT-PGD, which are bolded in the Methods column of the table, are specific models obtained by combining the framework proposed in this paper with different adversarial training methods. The other methods are existing works used for comparison, and their information is introduced in Section 5.2. The effects of these methods are based on the best results reported in their related papers. We can observe that:

(1)*Methods with hand-crafted features typically exhibit the poorest performance on benchmarks. Since manually obtaining high-quality feature representations from the data is challenging. Deep learning-based methods can address this challenge and enhance rumor detection performance. For instance, SVM-TS has an average detection accuracy that is more than 5 % lower than CNN, which illustrates the advantage of deep learning for rumor detection.*

Table 3: Rumor detection performance in Twitter15 dataset

Method	ACC	N	F	T	U
		F_1	F_1	F_1	F_1
DTC	0.454	0.415	0.355	0.733	0.317
SVM-TS	0.642	0.811	0.434	0.639	0.600
CNN	0.718	0.807	0.601	0.635	0.730
RvNN	0.723	0.682	0.758	0.821	0.654
BERT	0.735	0.731	0.722	0.730	0.705
GCAN	0.842	0.844	0.846	0.889	0.800
UDGCN	0.834	0.827	0.866	0.885	0.756
BiGCN	0.886	**0.891**	0.860	0.930	0.864
EBGCN	0.892	0.869	0.897	**0.934**	0.867
ADAAT-FGM	0.899	0.875	0.950	0.896	**0.868**
ADAAT-PGD	**0.900**	0.871	**0.954**	0.903	0.859

Table 4: Rumor detection performance in Twitter16 dataset

Method	ACC	N	F	T	U
		F_1	F_1	F_1	F_1
DTC	0.473	0.254	0.080	0.190	0.482
SVM-TS	0.691	0.763	0.483	0.722	0.690
CNN	0.700	0.688	0.666	0.810	0.615
RvNN	0.737	0.662	0.743	0.835	0.708
BERT	0.804	0.777	0.525	0.824	0.787
GCAN	0.871	0.857	0.688	0.929	0.901
UDGCN	0.867	0.789	0.846	0.903	0.878
BiGCN	0.880	0.847	0.869	0.937	0.811
EBGCN	0.915	0.879	0.906	0.947	**0.910**
ADAAT-FGM	**0.928**	**0.927**	**0.931**	**0.961**	0.877
ADAAT-PGD	0.922	0.926	0.927	**0.961**	0.865

(2) *Methods based on GNNs generally outperform other deep learning methods.* CNN captures local spatial information of event propagation, while RvNN captures temporal features within the propagation. Both CNN and RvNN overlook the global structure of event propagation, which restricts the learning of structural features. In contrast, GNN-based methods can aggregate the neighborhood information of nodes in propagation graphs continuously to learn global structural features.

(3) *ADAAT generally achieves the best performance among all GNN-based methods.* This is primarily due: a) Events are often disrupted by noise during the propagation process. GCAN, UDGCN, and BiGCN solely focus on leveraging propagation structure information, neglecting the impact of noise on both the structure and content. However, ADAAT employs adaptive data augmentation on both structures and features, thereby preserving crucial structures and features while also enhancing the diversity and randomness of the data. Consequently, the framework's robustness to noise is enhanced. b) Rumors

Table 5: Rumor detection performance in PHEME dataset

Method	Acc	Class	Pre	Rec	F_1
SVM-TS	0.685	R	0.553	0.539	0.539
		N	0.758	0.762	0.757
CNN	0.747	R	0.683	0.512	0.584
		N	0.768	0.872	0.816
RvNN	0.763	R	0.689	0.587	0.631
		N	0.796	0.858	0.825
BERT	0.807	R	0.736	0.695	0.713
		N	0.842	0.866	0.853
GCAN	0.834	R	0.769	0.758	0.761
		N	0.871	0.874	0.872
UDGCN	0.805	R	0.752	0.673	0.708
		N	0.831	**0.875**	0.852
BiGCN	0.824	R	0.753	0.734	0.741
		N	0.861	0.872	0.865
ADAAT-FGM	**0.846**	R	**0.767**	0.798	**0.779**
		N	0.889	0.874	**0.880**
ADAAT-PGD	0.842	R	0.758	**0.805**	0.776
		N	**0.896**	0.859	0.876

circulating in the open space of social media may be targeted by malicious users with adversarial perturbations, altering their propagation structures and comment content. These artificial adversarial perturbations are often more subtle and damaging than noise. Although EBGCN acknowledges the uncertainty of propagation, it overlooks the noise in the content and the influence of adversarial attacks on rumor detection. ADAAT employs FGM/PGD adversarial training to simulate realistic adversarial attacks in high-dimensional space, enabling the framework to learn their features during training. Moreover, the generated adversarial representations are employed as hard samples. Integrated with contrastive learning, the framework is compelled to learn invariant essential representations from these adversarial samples, thereby enhancing the framework's robustness against adversarial attacks and improving the performance of rumor detection.

5.5 Early Rumor Detection

Early rumor detection involves identifying whether an event is a rumor during its initial stages of propagation. The earlier a rumor is detected, the sooner social media can implement measures to curtail its spread. To address Q2, we conduct an early rumor detection experiment. To establish the early rumor detection task, we adhere to the methodology outlined by Ma et al. (Ma et al., 2018), using a detection deadline and defining five cut-off moments spanning 0 to 100 minutes. Only the event information up to each cut-off moment is considered. The experimental outcomes are presented in Figure 4. The following observations are made:

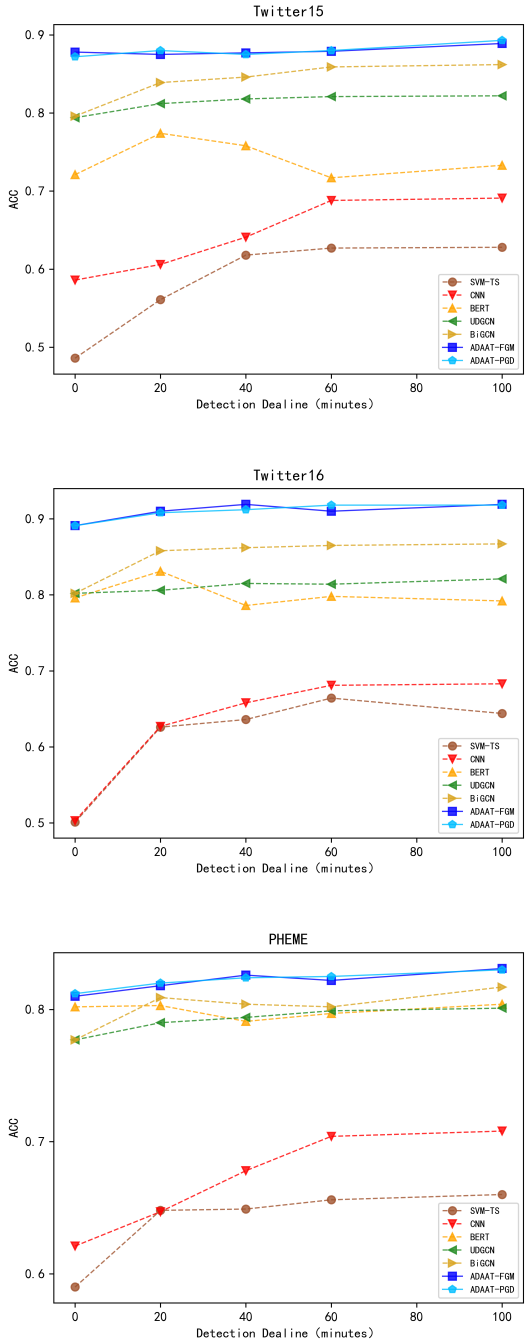

Figure 4: Results of rumor early detection on three datasets

(1) *All methods exhibit the lowest detection accuracy at time 0.* This is attributed to the insufficient availability of features and structural information at time 0, which leads to a lower quality of event representation and subsequently affects detection accuracy. As time

progresses, more textual and structural information becomes available, leading to a gradual improvement in the detection accuracy of most methods.

(2) *GNN-based methods outperform other baselines in early rumor detection, with ADAAT achieving the highest accuracy on the benchmark.* The rich structural information enhances rumor detection performance. GNNs are adept at leveraging the structural information within rumor propagation, thereby yielding a high early rumor detection performance. Furthermore, in the early stages, there is a reduced amount of structural and comment content, making it more susceptible to perturbations. ADAAT employs adaptive data augmentation and adversarial training to mitigate the impact of perturbations on detection performance, resulting in the highest accuracy.

5.6 Ablation Study

To answer question Q3, verifying the contribution of different modules to ADAAT, ablation study is performed. Without loss of generality, the experiment compares ADAAT-FGM with the following four variants:

Table 6: Results of ablation study on three datasets

Model	Twitter15	Twitter16	PHENE
	Acc	Acc	Acc
ADAAT	0.899	0.928	0.846
ADAAT-NoAA	0.896	0.926	0.840
ADAAT-NoBi	0.846	0.857	0.782
ADAAT-NoAtk	0.885	0.908	0.835
ADAAT-NoCL	0.862	0.911	0.827

ADAAT-NoAA does not utilize adaptive data augmentation module and directly learns the representations only using raw propagation graphs.

ADAAT-NoBi only utilizes the GCNs for the Top-Down propagation graphs and does not consider the wide dispersion of events.

ADAAT-NoAtk removes the hard sample generation module, and the framework is unable to generate adversarial representations as hard samples for contrastive learning.

ADAAT-NoCL only utilizes cross-entropy loss without contrastive loss, which cannot capture the commonality between same label samples and characteristics between different label samples.

Table 6 shows the results of the ablation experiments. We make the following observations:

(1) *ADAAT exhibits the highest performance among all variants.* This indicates that the presence of noise and adversarial perturbations in the data influences the framework's detection performance, and that both the adaptive data augmentation module and the hard sample generation module are effective in mitigating these effects. Furthermore, the inclusion of contrastive loss aids in distinguishing rumors from non-rumors.

(2) *The hard sample generation module contributes more significantly to the framework's performance than the adaptive data augmentation module.* In comparison to ADAAT-NoAA, ADAAT-NoAtk experiences a decrease in detection accuracy of 1.1 %, 1.8 %, and

0.5 % on the three datasets, respectively. This is due to the more pronounced impact of adversarial perturbations in the data on the framework's detection accuracy. Furthermore, hard samples play a crucial role in contrastive learning, facilitating the extraction of essential features from the data. Without this module, it is challenging to fully exploit the benefits of contrastive loss.

(3) *ADAAT-NoBU experiences the most significant drop in accuracy compared to all other variants.* In comparison to ADAAT, ADAAT-NoBU experiences a decrease of 5.3 %, 6.9 %, and 6.3 % on the three datasets, respectively, a greater decline than that observed in other variants. This is a result of utilizing GCNs exclusively for directed Top-Down propagation graphs, neglecting the diffuse nature of event propagation. This leads to the failure to capture certain structural features, resulting in a severe drop in detection accuracy.

5.7 Stability Study

To address Q4 and assess the performance of ADAAT under additional perturbations, we conduct stability experiments, which are divided into two parts: experiments against adversarial perturbations and experiments against random perturbations.

In the experiments targeting adversarial perturbations, ADAAT and its variants are tested on both unperturbed and perturbed test data. The perturbed test data is generated by adding vectors in the direction of gradient ascent to each dimension of the initial feature. The experimental outcomes are presented in Tables 7 and 8, with "(atk)" indicating that the test data has been perturbed. ADAAT-NoAA*Atk represents the variant lacking the adaptive data augmentation module and the hard sample generation module. Since both ADAAT-NoAA*Atk and ADAAT-NoAtk variants lack the hard sample generation module and are equivalent to ADAAT-FGM and ADAAT-PGD, the same experimental results are used for the corresponding rows in the table. The following observations are made:

Table 7: Stability experiment with adversarial perturbations results for ADAAT-FGM

Model	Twitter15	Twitter16	PHENE
	Acc	Acc	Acc
ADAAT-FGM	0.899	0.928	0.846
ADAAT-FGM(atk)	0.886	0.914	0.834
ADAAT-NoAtk(atk)	0.868	0.884	0.817
ADAAT-NoAA(atk)	0.890	0.921	0.837
ADAAT-NoAA*Atk(atk)	0.878	0.901	0.822

(1) *Our framework is effective to defend the against adversarial perturbations.* Compared with ADAAT-FGM and ADAAT-PGD, the detection accuracy of ADAAT-FGM(atk) and ADAAT-PGD(atk) only decreases by 1.3%, 1.4%, 1.2% and 0.6%, 1.6%, 1.3% respectively on the three datasets, the decline being very low. Furthermore, from Tables 5 and 6, it can be observed that compared with ADAAT-NoATk(atk), the detection accuracy of ADAAT-NoATk is reduced by 1.7%, 2.4% and 1.8%, respectively. Its decrease is more than ADAAT-FGM, which indicates the effectiveness of hard sample generation module to defend against adversarial perturbations.

Table 8: Stability experiment with adversarial perturbations results for ADAAT-PGD

Model	Twitter15	Twitter16	PHENE
	Acc	Acc	Acc
ADAAT-PGD	0.900	0.922	0.842
ADAAT-PGD(atk)	0.884	0.906	0.829
ADAAT-NoAtk(atk)	0.868	0.884	0.817
ADAAT-NoAA(atk)	0.886	0.920	0.835
ADAAT-NoAA*Atk(atk)	0.878	0.901	0.822

(2) *In the presence of adversarial perturbations, the adaptive data augmentation module proves counterproductive.* When compared to ADAAT, ADAAT-NoAA demonstrates higher detection accuracy when applied to data perturbed by adversarial perturbations. Moreover, the detection accuracy of ADAAT-NoAtk(atk) surpasses that of ADAAT-NoAA*Atk(atk), indicating that the adaptive data augmentation module may have a detrimental effect on defending against adversarial perturbations. This finding underscores the limitations of previous rumor detection methods that solely rely on data augmentation. Despite the use of data augmentation, it remains challenging to effectively counter adversarial perturbations.

The stability experiments involving random perturbations are analogous to those with adversarial perturbations. The key difference is that random perturbations involve setting 30 % of the feature vectors in the test data to zero. The experimental outcomes are presented in Tables 9 and 10, with "(per)" indicating that random perturbations have been added to the test data. The following observations are made:

Table 9: Stability experiment with random perturbations results for ADAAT-FGM

Model	Twitter15	Twitter16	PHENE
	Acc	Acc	Acc
ADAAT-FGM	0.899	0.928	0.846
ADAAT-FGM(per)	0.895	0.923	0.840
ADAAT-NoAtk(per)	0.889	0.913	0.839
ADAAT-NoAA(per)	0.892	0.920	0.838
ADAAT-NoAA*Atk(per)	0.885	0.909	0.833

Table 10: Stability experiment with random perturbations results for ADAAT-PGD

Model	Twitter15	Twitter16	PHENE
	Acc	Acc	Acc
ADAAT-PGD	0.900	0.922	0.842
ADAAT-PGD(per)	0.897	0.918	0.837
ADAAT-NoAtk(per)	0.889	0.913	0.839
ADAAT-NoAA(per)	0.891	0.911	0.836
ADAAT-NoAA*Atk(per)	0.885	0.909	0.833

(1) *ADAAT is effective to defend the against random perturbations.* Compared with ADAAT, the detection accuracy of ADAAT(atk) only drops by 0.4%, 0.5%, 0.6% and 0.3%, 0.4%, 0.5% on the three datasets, the decrease being very low. From Tables 5 and 8, it can be seen that on the three datasets, ADAAT has less accuracy decrease than its variants in the face of test data with noise. This phenomenon indicates that ADAAT has higher resistance to random perturbations its variants.

(2) *The adaptive data augmentation module has better resistance to random perturbations than the hard sample generation module.* From Tables 5 and 8, the accuracy of ADAAT-NoAA on the perturbed test data is 0.4%, 0.6% and 0.7% lower than that of the unperturbed test data, respectively. In contrast, the accuracy of the ADAAT-NoAtk is increased by 0.9%, 0.5%, and 1.1% on perturbed data and unperturbed data. The performance of the variant containing only the adaptive data augmentation module improves, while the performance of the variant containing only the hard sample generation module decreases. This demonstrates that the adaptive data augmentation module is more resistant to random perturbations than the hard sample generation module.

(3) *Adversarial perturbations have more negative impacts on the rumor detection framework than random perturbations.* Comparing the detection accuracy of the corresponding variants in Tables 6 and 8, as well as Tables 7 and 9, it can be found that the accuracy of the framework under the influence of noise is higher than that against adversarial the attack. This is normal, since adversarial perturbations are artificially designed specifically to fool the neural networks and it has more negative impacts than random perturbations.

5.8 Hyperparameter Sensitivity Analysis

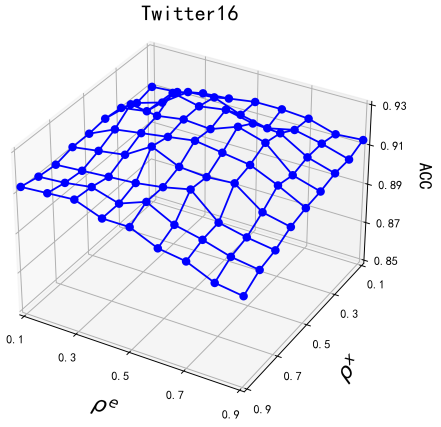

Figure 5: Hyperparameter sensitivity analysis

To answer question Q5, we perform a sensitivity analysis on the hyperparameters ρ_e and ρ_x. They affect the computation of edge and feature importance in the adaptive data augmentation module, respectively. We take the Twitter16 dataset as an example and set ρ_e and ρ_x to [0.1,0.9] with a modification interval of 0.1. The truncation probabilities p_t^e

and p_t^x are set to 0.9 and the other parameters are constant. The experimental results are shown in Figure 5. It can be observed that:

ADAAT is little sensitive to hyperparameters ρ_e and ρ_x. On the one hand, when ρ_e and ρ_x are not set too large, the rumor detection accuracy is high and stable. On the other hand, when at least one of the two is large, especially when $\rho_e = \rho_x = 0.9$, all edges and features are almost modified, and the detection accuracy decreases more. However, since not only augmented propagation graphs but also raw propagation graphs are utilized, the accuracy drop is only within 4% even if the augmented propagation graphs are severely corrupted. The above two points prove that ADAAT is little sensitive to hyperparameters ρ_e and ρ_x.

Based on all the experiments above, we draw the following conclusions about the proposed ADAAT model: our model demonstrates outstanding performance in rumor detection tasks, exhibiting remarkable efficiency and stability across rumor detection scenarios at different time periods. Moreover, the model shows strong robustness, maintaining stable rumor detection performance even when faced with adversarially attacked data. The bidirectional propagation module and contrastive learning loss introduced in our model prove to be highly effective, significantly enhancing the performance of rumor detection.

6. Conclusion

In this paper, we introduce a rumor detection framework named ADAAT. The framework employs an adaptive data augmentation module to assess the importance of edges and features within raw propagation graphs, and then modifies them to generate augmented propagation graphs. Subsequently, GCNs are utilized to concurrently capture the textual and propagation structure features of events, thereby generating graph representations. Lastly, a hard sample generation module is integrated to conduct adversarial training on the graph representations. The adversarial representations thus obtained are employed as hard samples, which are integrated with contrastive learning to compel the framework to discern essential features within the data that are resilient to adversarial perturbations. Experimental outcomes indicate that ADAAT surpasses existing methods on three public rumor detection datasets, showcasing strong detection performance and robustness.

To enhance the model's stability against attacks and noise, we employ various techniques such as data augmentation, bidirectional propagation, contrastive learning, and adversarial training. While these techniques improve the model's performance, they also increase its complexity, limiting its scalability in large-scale data or real-time detection scenarios. In our future research, we will focus on improving the model's performance while balancing its scalability and complexity. Besides, there exists a vast volume of data within social media, posing a challenge in the timely identification of rumors. Furthermore, the majority of existing rumor detection methods concentrate solely on textual information and propagation structures, disregarding other forms of information such as images and videos. Hence, our future endeavors will prioritize enhancing the scalability of rumor detection methods and also delve into multimodal approaches capable of leveraging text, images, and other forms of content simultaneously.

Acknowledgments

We express gratitude to anonymous reviewers for their hard work and kind comments. This work is supported by the Foundation of the Major Project of Science and Technology Innovation2030-New Generation of Artifcial Intelligence(No.2021ZD0112500), the National Natural Science Foundation of China(No.62272191), the Science and Technology Development Program of Jilin Province(No.20220201153GX), Jilin Provincial Science and Technology Department Project(No.20240402067GH) and the Open Project of Key Laboratory Ministry of Industry and Information Technology(HK202303528).

References

Adjeisah, M., Zhu, X., Xu, H., & Ayall, T. A. (2023). Towards data augmentation in graph neural network: An overview and evaluation. *Computer Science Review*.

Aljabri, M., Zagrouba, R., Shaahid, A., Alnasser, F., Saleh, A., & Alomari, D. M. (2023). Machine learning-based social media bot detection: a comprehensive literature review. *Social Network Analysis and Mining*.

Alsaif, H. F., & Aldossari, H. D. (2023). Review of stance detection for rumor verification in social media. *Engineering Applications of Artificial Intelligence*.

Bian, T., Xiao, X., Xu, T., Zhao, P., Huang, W., Rong, Y., & Huang, J. (2020). Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 549–556.

Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pp. 675–684.

Chen, T., Li, X., Yin, H., & Zhang, J. (2018). Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 40–52.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186.

Ding, K., Xu, Z., Tong, H., & Liu, H. (2022). Data augmentation for deep graph learning: A survey. In *ACM SIGKDD Explorations Newsletter*, pp. 61–77.

Feng, S., Jing, B., Zhu, Y., & Tong, H. (2022). Adversarial graph contrastive learning with information regularization. In *Proceedings of the ACM Web Conference 2022*, pp. 1362–1371.

Gao, T., Yao, X., & Chen, D. (2021). Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910.

Ghaffari Laleh, N., Truhn, D., Veldhuizen, G. P., Han, T., van Treeck, M., Buelow, R. D., Langer, R., Dislich, B., Boor, P., Schulz, V., et al. (2022). Adversarial attacks and

adversarial robustness in computational pathology. *Nature communications*, *13*(1), 5711.

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pp. 1024–1034.

Hao, X., Liu, B., Yang, X., Sun, X., Meng, Q., & Cao, J. (2024). Multi-stage dynamic disinformation detection with graph entropy guidance. *World Wide Web*, *27*(2).

He, Z., Li, C., Zhou, F., & Yang, Y. (2021). Rumor detection on social media with event augmentations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2020–2024.

Hong, M., Choi, J., & Kim, G. (2021). Stylemix: Separating content and style for enhanced data augmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14862–14870.

Hou, Z., Liu, X., Dong, Y., Wang, C., Tang, J., et al. (2022). Graphmae: Self-supervised masked graph autoencoders..

Huang, B., Alhudhaif, A., Alenezi, F., Althubiti, S. A., & Xu, C. (2022). Balance label correction using contrastive loss. In *Information Sciences*.

Joseph, J., Vineetha, S., & Sobhana, N. (2022). A survey on deep learning based sentiment analysis. In *Materials Today: Proceedings*, pp. 456–460.

Lee, Y., & Han, S. W. (2023). Cagcn: Causal attention graph convolutional network against adversarial attacks. *Neurocomputing*.

Lindemann, B., Müller, T., Vietz, H., Jazdi, N., & Weyrich, M. (2021). A survey on long short-term memory networks for time series prediction. In *Procedia CIRP*.

Liu, B., Sun, X., Meng, Q., Yang, X., Lee, Y., Cao, J., Luo, J., & Lee, R. K.-W. (2022). Nowhere to hide: Online rumor detection based on retweeting graph neural networks. In *IEEE Transactions on Neural Networks and Learning Systems*.

Liu, B., Sun, X., Ni, Z., Cao, J., Luo, J., Liu, B., & Fu, X. (2019). Co-detection of crowdturfing microblogs and spammers in online social networks. *World Wide Web*, *23*, 573 – 607.

Liu, G., Zhao, T., Xu, J., Luo, T., & Jiang, M. (2022). Graph rationalization with environment-based augmentations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1069–1078.

Liu, R., Liu, W., Zheng, Z., Wang, L., Mao, L., Qiu, Q., & Ling, G. (2023). Anomaly-gan: A data augmentation method for train surface anomaly detection. *Expert Systems with Applications*, *228*, 120284.

Lu, Y.-J., & Li, C.-T. (2020). Gcan: Graph-aware co-attention networks for explainable fake news detection on social media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 505–514.

Luo, Y., McThrow, M., Au, W. Y., Komikado, T., Uchino, K., Maruhash, K., & Ji, S. (2022). Automated data augmentations for graph classification. *arXiv preprint arXiv:2202.13248*.

Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K.-F., & Cha, M. (2016). Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 3818–3824.

Ma, J., Gao, W., & Wong, K.-F. (2017). Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 708–717.

Ma, J., Gao, W., & Wong, K.-F. (2018). Rumor detection on twitter with tree-structured recursive neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 1980–1989.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations*. OpenReview.net.

Miyato, T., Dai, A. M., & Goodfellow, I. (2017). Adversarial training methods for semi-supervised text classification. In *5th International Conference on Learning Representations*. OpenReview.net.

Mosallanezhad, A., Karami, M., Shu, K., Mancenido, M. V., & Liu, H. (2022). Domain adaptive fake news detection via reinforcement learning. In *Proceedings of the ACM Web Conference 2022*.

Pathak, A. R., Mahajan, A., Singh, K., Patil, A., & Nair, A. (2020). Analysis of techniques for rumor detection in social media. In *Procedia Computer Science*.

Pisner, D. A., & Schnyer, D. M. (2020). Chapter 6 - support vector machine. In *Machine Learning*, pp. 101–121.

Selvaraj, S., & L D, D. B. (2020). Earlier detection of rumors in online social networks using certainty-factor-based convolutional neural networks. *Social Network Analysis and Mining*.

Sun, C., Li, H., Song, M., & Hong, S. (2023a). A ranking-based cross-entropy loss for early classification of time series. In *IEEE Transactions on Neural Networks and Learning Systems*.

Sun, X., Cheng, H., Dong, H., Qiao, B., Qin, S., & Lin, Q. (2023b). Counter-empirical attacking based on adversarial reinforcement learning for time-relevant scoring system. *IEEE Transactions on Knowledge and Data Engineering*.

Sun, X., Cheng, H., Liu, B., Li, J., Chen, H., Xu, G., & Yin, H. (2023c). Self-supervised hypergraph representation learning for sociological analysis. *IEEE Transactions on Knowledge and Data Engineering*, *35*(11), 11860–11871.

Sun, X., Yin, H., Liu, B., Meng, Q., Cao, J., Zhou, A., & Chen, H. (2023d). Structure learning via meta-hyperedge for dynamic rumor detection. *IEEE Trans. on Knowl. and Data Eng.*, *35*(9), 9128–9139.

Tan, Q., Liu, N., Huang, X., Chen, R., Choi, S.-H., & Hu, X. (2022). Mgae: Masked autoencoders for self-supervised learning on graphs. *arXiv preprint arXiv:2201.02534*.

Wang, J.-Y., & Jang, J.-S. R. (2023). Training a singing transcription model using connectionist temporal classification loss and cross-entropy loss. In *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Wang, X., & Qi, G.-J. (2023). Contrastive learning with stronger augmentations. In *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 5549–5560. IEEE.

Wang, Y., Wang, J., Cao, Z., & Farimani, A. B. (2022). Molecular contrastive learning of representations via graph neural networks. In *Nat. Mach. Intell.*, pp. 279–287.

Wei, L., Hu, D., Zhou, W., Yue, Z., & Hu, S. (2021). Towards propagation uncertainty: Edge-enhanced bayesian graph convolutional networks for rumor detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 3845–3854.

Wu, T., Yang, N., Chen, L., Xiao, X., Xian, X., Liu, J., Qiao, S., & Cui, C. (2022). Ergcn: Data enhancement-based robust graph convolutional network against adversarial attacks. In *Information Sciences*.

Wu, Y., Zhan, P., Zhang, Y., Wang, L., & Xu, Z. (2021). Multimodal fusion with co-attention networks for fake news detection. In *Findings of the association for computational linguistics: ACL-IJCNLP 2021*, pp. 2560–2569.

Xu, C., Zhang, C., Yang, Y., Yang, H., Bo, Y., Li, D., & Zhang, R. (2023). Accelerate adversarial training with loss guided propagation for robust image classification. *Information Processing Management*.

Xu, S., Liu, X., Ma, K., Dong, F., Riskhan, B., Xiang, S., & Bing, C. (2022). Rumor detection on social media using hierarchically aggregated feature via graph neural networks. In *Applied Intelligence*.

Xue, T., Zhang, F., Zhang, C., Chen, Y., Song, Y., Golby, A. J., Makris, N., Rathi, Y., Cai, W., & O'Donnell, L. J. (2023). Superficial white matter analysis: An efficient point-cloud-based deep learning framework with supervised contrastive learning for consistent tractography parcellation across populations and dmri acquisitions. *Medical Image Analysis*.

Yang, Y., Miao, R., Wang, Y., & Wang, X. (2022). Contrastive graph convolutional networks with adaptive augmentation for text classification. *Information Processing Management*.

Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T., et al. (2017). A convolutional approach for misinformation identification.. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 3901–3907.

Zeng, Z., Sun, S., & Li, Q. (2023). Multimodal negative sentiment recognition of online public opinion on public health emergencies based on graph convolutional networks and ensemble learning. *Information Processing Management*.

Zhang, J., Zhao, L., Zeng, J., Qin, P., Wang, Y., & Yu, X. (2022a). Deep mri glioma segmentation via multiple guidances and hybrid enhanced-gradient cross-entropy loss. *Expert Systems with Applications*, *196*, 116608.

Zhang, S., Chen, H., Yang, H., Sun, X., Yu, P. S., & Xu, G. (2022b). Graph masked autoencoders with transformers. *arXiv preprint arXiv:2202.08391*.

Zhang, Y., Zhu, H., Song, Z., Koniusz, P., & King, I. (2022c). Costa: Covariance-preserving feature augmentation for graph contrastive learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2524–2534.

Zhao, T., Liu, Y., Neves, L., Woodford, O., Jiang, M., & Shah, N. (2021). Data augmentation for graph neural networks. In *Proceedings of the aaai conference on artificial intelligence*, pp. 11015–11023.

Zheng, X., Liu, Y., Pan, S., Zhang, M., Jin, D., & Yu, P. S. (2022). Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082*.

Zhou, H., Ma, T., Rong, H., Qian, Y., Tian, Y., & Al-Nabhan, N. (2022). Mdmn: Multi-task and domain adaptation based multi-modal network for early rumor detection. *Expert Systems with Applications*, *195*, 116517.

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., & Wang, L. (2021). Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, pp. 2069–2080.

Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., & Procter, R. (2018). Detection and resolution of rumours in social media: A survey. In *ACM Computing Surveys (CSUR)*, pp. 1–36.