

Research Article

Rumor Detection Based on Knowledge Enhancement and Graph Attention Network

Wanru Wang ^{1,2}, Yuwei Lv,³ Yonggang Wen,⁴ and Xuemei Sun^{2,3}

¹School of Software, Tiangong University, Tianjin 300387, China

²Tianjin Key Laboratory of Autonomous Intelligence Technology and Systems, Tianjin 300387, China

³School of Computer Science and Technology, Tiangong University, Tianjin 300387, China

⁴Tianjin University of Commerce Boustead College, Tianjin 300384, China

Correspondence should be addressed to Wanru Wang; tjpuwvr@126.com

Received 17 April 2022; Revised 11 July 2022; Accepted 17 September 2022; Published 6 October 2022

Academic Editor: Akbar Ali

Copyright © 2022 Wanru Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Presently, most of the existing rumor detection methods focus on learning and integrating various features for detection, but due to the complexity of the language, these models often rarely consider the relationship between the parts of speech. For the first time, this paper integrated a knowledge graphs and graph attention networks to solve this problem through attention mechanisms. A knowledge graphs can be the most effective and intuitive expression of relationships between entities, providing problem analysis from the perspective of “relationships”. This paper used knowledge graphs to enhance topics and learn the text features by using self-attention. Furthermore, this paper defined a common dependent tree structure, and then the ordinary dependency trees were reshaped to make it generate a motif-dependent tree. A graph attention network was adopted to collect feature representations derived from the corresponding syntax-dependent tree production. The attention mechanism was an allocation mechanism of weight parameters that could help the model capture important information. Rumors were then detected accordingly by using the attention mechanism to combine text representations learned from self-attention and graph representations learned from the graph attention network. Finally, numerous experiments were performed on the standard dataset Twitter, and the proposed model here had achieved a 7.7% improved accuracy rate compared with the benchmark model.

1. Introduction

In recent years, with the rapid development of social media platforms, various social media have been widely used, such as Sina Weibo and Twitter. The popularity of social media platforms has brought great convenience for people to collect information and news, but the development of social media platforms has also led to the spread of rumors. Rumors in social media have the characteristics of fast spread, wide influence range, large cost, and low efficiency of manual refuting. Therefore, rumor detection has become an extremely challenging research topic in the field of text classification, which has attracted close attention from academia and industry.

In recent studies, Ma et al. [1] used recurrent neural network and attention mechanisms to perform rumor detection, improving the most sophisticated detection effects at

the time. However, attention mechanisms may sometimes fail due to the complexity of linguistic morphology and syntax. To this end, Wang et al. [2] proposed syntax tree-based structure information for text. As knowledge is refined into a structured form, many domains of knowledge graph (KG) have been constructed. Comet [3] used triples in KG as a corpus to train generative pretraining for common sense learning. But this knowledge embedding fails to account for the impacts of the introduced knowledge on sentences. Recently, Zhang et al. [4] used graph neural networks to process representations of graphs learned from the dependency tree. Lv et al. [5] proposed rumor detection based on a time graph attention network. The method first aggregated the spread structure and text features of rumors through the graph attention neural network, then, recorded the historical state of the spread structure using the time attention

mechanism, and then, captured the features of the spread structure changing with time through the gated loop unit. This method innovatively proposes the concept of time for rumor detection but does not consider the deep connection between words as the characteristics of the text itself. Although these models have improved in some ways, their disadvantages cannot be ignored. The dependency trees constructed first ignore the connection between subjects and viewpoint words; second, introduced knowledge graphs also inevitably bring noise; and finally, only a small fraction of the dependent trees are associated with rumor detection, which does not require encoding the entire dependency trees.

To address these problems, for the first time, this paper proposed a method of rumor detection based on knowledge enhancement and graph attention network models.

First, motif of knowledge graph triples was injected for thematic enhancement, modifying the effects of mask mechanisms for noise reduction. Second, the syntax-dependent trees were reshaped so that the motif was dependent and pruning for the syntax-dependent trees, only retaining edges that had a direct dependence on the motif. This unified tree structure was able to focus not only on the connection between topics and potential opinion words, but also facilitated batch processing and parallel operation. Finally, our study fused the text representations learned by self-attention with the representation of the graphs learned by graph attention network. The Twitter dataset was extensively evaluated, and the experimental results demonstrated that our model outperformed the benchmark method.

Main contributions of the research:

- (1) For the first time, this paper proposed that knowledge graphs and syntactic-dependent trees could be used to reshape sentences to detect rumors, which has improved the effects of rumor detection.
- (2) In this paper, we improved the traditional graph neural network by using a graph attention network model based on text-classification and a dual attention mechanism, where node attention and edge attention could enhance each other.
- (3) The attention mechanism was adopted to integrate the text representation and the graph representation, and the fusion feature representation was used to enhance the text representation learned from the attention and the graph representation learned by the graph attention network, respectively.
- (4) Comparing the proposed model with the current benchmark model, we experimentally demonstrated that the proposed model could improve detection accuracy by 7.7%.

2. Related Work

The earliest study of rumor detection was based on traditional machine learning, which transformed the rumor detection into a second classification problem for processing. First, the rumor features were extracted, then, the machine learning model was applied for modeling, and finally, the rumor

characteristics were input into the model for training to realize the rumor detection. Zhao et al. [6] presented a model based on the decision tree classifier that had clustered the questioned and corrected phrases that were extracted from messages, and then a decision tree based on statistical features was built for rumor detection. Yang et al. [7] added content-based subject features, semantic bias, and comment information to the rumor detection model. Guo et al. [8] proposed that emotional features had an important role in rumor detection. With the development of information technology, deep learning has been successfully applied in information processing, mode recognition and artificial intelligence, and has achieved fruitful research results. Kwon et al. [9] used text structure information and linguistic features to capture the multimodal phenomena of rumor propagation by applying three classification models: support vector machine, random forest classifier, and decision tree. Ma et al. [1] first proposed using circular neural networks to realize rumor detection, considering the comment information and timing characteristics of the message in the transmission process. A circular neural network can automatically capture the changes of time signals in the process of rumor transmission, modeling the long-time context information on the time axis, thus, realizing the time series-based event representation and achieving good results in rumor prediction. Chen et al. [10] took advantage of the characteristics of the constantly changing importance of words during message transmission and the replication of rumors to introduce attention mechanisms in recurrent neural networks. Yu et al. [11] first applied the convolutional neural network to detect rumors, using the paragraph vectors to model the rumor information, taking the obtained paragraph vectors as the input to the convolutional neural network, and generating high-order abstract features through the underlying combination of the features to achieve the purpose of improving the rumor detection accuracy.

The knowledge graph can automatically make use of massive unstructured text data and information to assist with manual analysis, research, understanding the big data, and provide an accurate, reliable, and efficient factual basis for rumor detection. Rospocher et al. [12] presented an event-centered knowledge graph that drew events from news reports, which included time, place, and participants established causal and coexisting relationships between events, and reconstructed the historical development and temporal evolution of events. Gottschalks et al. [13] presented a knowledge graph with event-centered multi-lingual temporal that drew 690,000 contemporary historical events and over 2.3 million temporal relationships from existing large knowledge graphs like DBpedia, YAGO, and Wikidata and integrated the extracted events, entities, and relationships. Hemes et al. [14] proposed a semantic representation of financial events that automatically processed and analyzed financial events to assist decision-making. Entities from the rumor dataset are extracted to connect these entities to the open knowledge graph DBpedia, based on which relationships and entities connected are obtained. The knowledge graph needed in this paper is formed by using these relationships and entities.

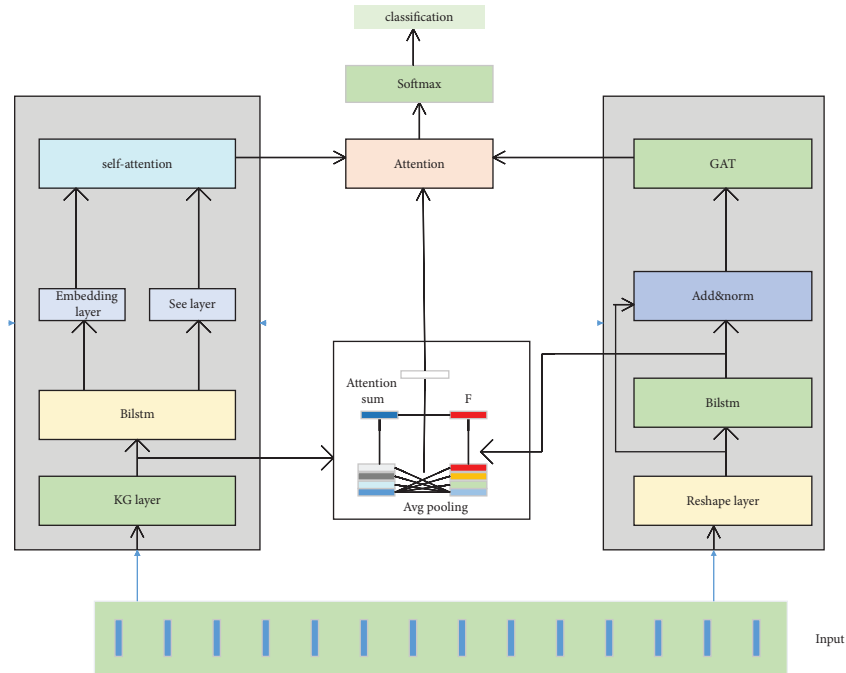


FIGURE 1: The overall framework of rumor detection based on knowledge enhancement and Graph Attention Network.

The attention mechanism was an allocation mechanism of weight parameters that could help the model capture important information. Some works provide the visualization of the learned attention, such as: Xu et al. [15] proposed a novel X-invariant Contrastive Augmentation and Representation learning framework to thoroughly obtain rotate-shear-scale invariant features. Shu et al. [16] proposed a novel Expansion-Squeeze-Excitation-Fusion Network to learn modal and channel-wise Expansion-Squeeze-Excitation attentions for attentively fusing the multimodal features in the modal and channel-wise ways. Shu et al. [17] present a Spatio-Temporal Context Coherence constraint and a Global Context Coherence constraint to capture the relevant motions and quantify their contributions to the group activity, an attention mechanism is employed to quantify the contribution of a certain motion by measuring the consistency between itself and the whole activity at each time step. Tang et al. [18] proposed a novel Skeleton-joint Co-Attention Recurrent Neural Network to capture the spatial coherence among joints and the temporal evolution among skeletons simultaneously on a skeleton-joint co-attention feature map in spatiotemporal space.

Since the first introduction of graph neural networks in Kipf et al. [19], Graph Neural Network (GNN) has recently shown its strong power in processing graphical structural representations in the context of natural language processing. Marcheggiani and Ivan [20] in 2017 proposed a semantic Graph Convolution Network (GCN) based role labeling model, at the word level; it effectively combined syntactic information and skillfully combined sequence model and graph representation, which alleviated the defects of the sequence model to a certain extent. Shu et al. [21] propose a novel graph LSTM-in-LSTM (GLIL) for group activity recognition by modeling the person-level actions

and the group-level activity simultaneously. Zhang et al. [22] used the GNN in the documentation and relational classification. The Graph Attention Network (GAN) was first introduced into the text classification task under document-word and word-word relations by Yao et al. For similar purposes, Huang and Carley [23] in 2019 explicitly established the dependence between words by using a graph attention network. However, these methods often ignore dependencies, which may determine the connection between words.

3. Model

In the model, we integrated the text representation learned by self-attention with the graph representation of syntactic dependency trees learned by the Graph Attention Network. Figure 1 shows the overall framework of the rumor detection model, which contains 3 main modules, the knowledge embedding module, dependency tree module based on the graph attention network, and the attention module.

In this section, the task of rumor detection is expressed as follows: In each document D contains sentences $\{s_1, s_2, \dots, s_i\}$, given a sentence s_i , $i \in \{1, 2, \dots, i\}$ and his subject a_k , $k \in \{1, 2, \dots, k\}$, the purpose of rumor detection is to predict whether an a_k event is a rumor or not by automatically learning the sentences related to a_k .

3.1. Dependency Tree Modules Based on Graph Attention Network. A Graph Neural Network (GNN) model for text classification was first introduced, and then a syntax dependency tree was used to model the text for rumor detection. Finally, a graph attention network model based on text classification with a novel dual attention mechanism was

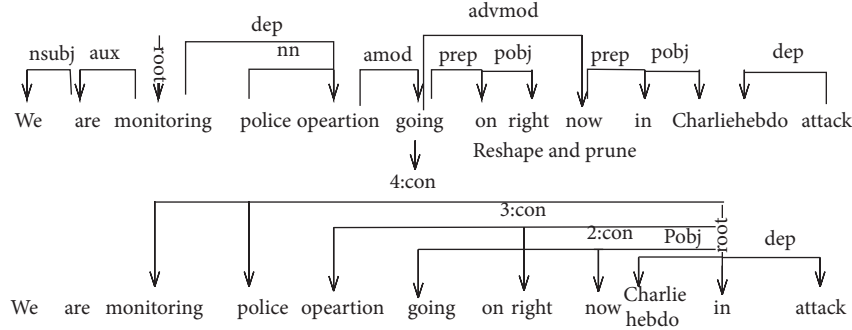


FIGURE 2: Schematic diagram of common dependency tree and modified and deleted topic dependency tree.

also applied in the research. Node attention was used to enhance edge attention and edge attention was used to enhance node attention.

3.1.1. Introduction of the GNN Model. With the proliferation of deep learning techniques, GNN has achieved great success in representation learning of graph structure data. In general, most of the existing GNN models follow a neighborhood aggregation strategy, and a GNN layer can be defined as shown in the following equation :

$$h_i^l = \text{AGGR}\left(h_i^{l-1}, \left\{h_j^{l-1} \mid \forall j \int N_i\right\}\right), \quad (1)$$

where h_i^l was the node representation of node i at the l -layer (usually using x_i as h_0^l), and N_i was the local domain set of node i . AGGR was the aggregation function of GNN, and there were many ways to implement it. GNN, which was with excellent performance in text classification, had the ability to capture long-distance interactions between entities. Most of the current methods are to build corpus-level document graphs and try to classify documents through semisupervised node classification. Despite of its success, most existing methods are computational flawed. Meanwhile, these methods are largely influenced by the use of simple graphs for modeling word interactions, which could limit text expressibility. Therefore, how to improve the computational power of the model expression is an important task to be solved.

3.1.2. Subject-Based Syntax-Dependent Trees. The subject-based dependency trees suggested that dependencies with direct connections to a certain subject could help the model focus more attention on relevant opinion words and, therefore, are more important than others. In addition, as was shown in Figure 2, the dependency trees that contained rich syntactic information were not usually rooted in the subject. However, the subject was the key target rather than the roots of the trees, while some relationships appeared somewhat redundant.

Based on the above observations, a novel subject-oriented dependent tree structure was adopted to reshape the original dependent trees to be subject-dependent. Algorithm 1 described the process, for the input statement, first, the dependent tree parser was applied to obtain its

TABLE 1: Dependence.

Label	Relationship type
Advmod	Adverbial modifier
Aux	Auxiliary
Conj	Conjunct
det	Determiner
Nsubj	Nominal subject
Prep	Prepositional modifier
Amod	Adjectival modifier
cc	Coordination
dep	Dependent
Nn	Noun compound modifier
Pobj	Object of a preposition

dependency trees; secondly, the subject was placed at the root, considered as an entity. Finally, the node with a direct connection to the subject was set to subnode, for which the original node dependency was retained.

There are many dependencies in sentences, such as: juxtaposition, dependencies, preposition modification, noun combination forms, correlations, and so on. The existence of these relationships may increase distance and computation. In order to shorten the distance between words and increase the calculation rate, we entered a character in the coordination(CC), which was unable to determine a more precise dependency between two words(DEP), determiner(DET), noun clause adjunct(ACL), nominal modifier(NMOD), etc. relationships remained stable after putting two characters in the dictionary. The distance was still equal after removing the edge in the dependency. Figure 2 shows the motif-oriented dependent trees constructed from the ordinary dependency trees. This subject-oriented structure has at least two advantages. First, each aspect has its own dependency trees and is less affected by joint-free points and relationships. Second, a motif contains a distance greater than 2, setting $n = \infty$ if the distance is greater than 4. This paper uses the natural language processing tool Stanford core NLP to construct the syntactic dependency tree of the rumor text. The common dependencies of this tool are shown in Table 1.

The dependency tree was defined as $G = (V, E)$, where $V = \{v_1, \dots, v_n\}$ represented the set of nodes in the graph, where each node represented a word in the sentence, $E = \{e_1, \dots, e_m\}$ represented the dependency, and the neighborhood information of the node e_i , v_i can be represented by N_i .

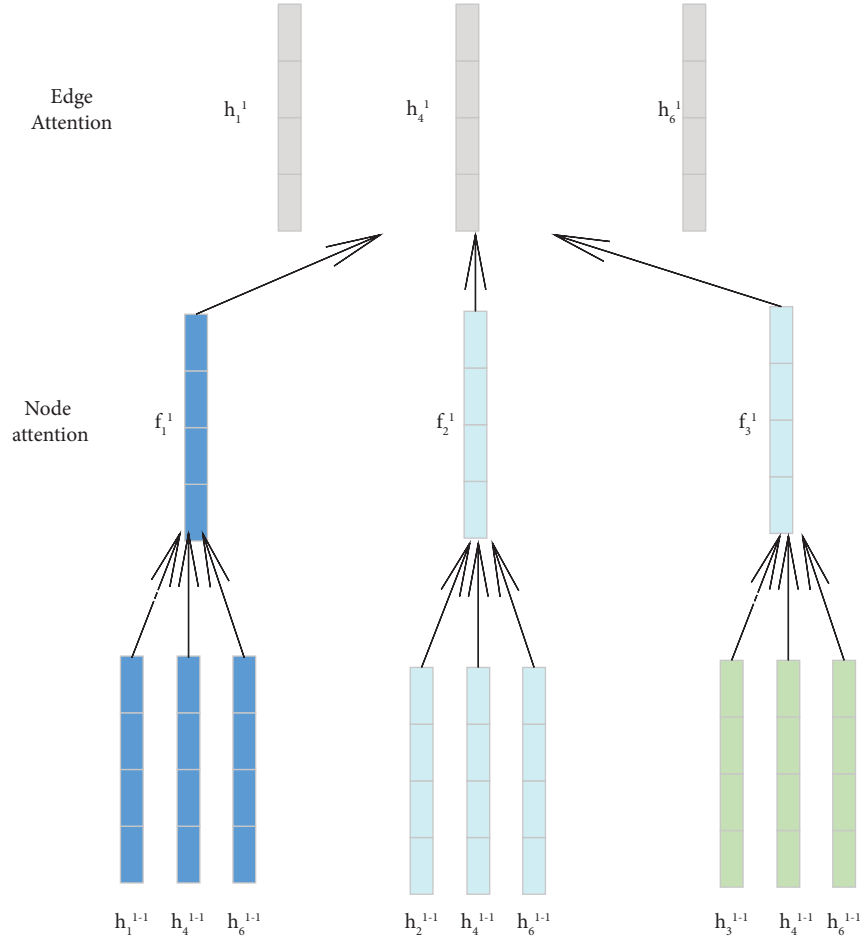


FIGURE 3: The overall architecture of Graph Attention Network information aggregation.

The topology of the syntax-dependent tree G was represented by the matrix $A \in \mathbb{R}^{n \times m}$, as defined in the following equation :

$$A_{ij} = \begin{cases} 1 & \text{if } v_i \in e_j \\ 0 & \text{if } v_i \notin e_j \end{cases} \quad (2)$$

Each node in the syntax-dependent trees had a d -dimensional attribute vector. Thus, all node properties could be expressed as $X = [x_1 x_2 \dots x_n]^T$, and we could further use $G = (A, X)$ to represent the entire syntax-dependent trees. For syntax-dependent trees, nodes represented words in a document, and the node attribute could be a thermal vector or pretrained word embeddings (e. g., word2vec, GloVe).

Syntax-dependent trees, that described grammatical relations among words, had been shown for its effectiveness in text representation learning. To exploit the syntax information for each word, a syntax dependency tree was first constructed for each text in the corpus. Inspired by the success of the hierarchical attention network, each syntactic information was viewed as an edge that could connect all the words in the sentence.

3.1.3. Graph Attention Network(GAT). Graph neural networks aggregate representations of neighborhood nodes along the dependent path. However, this process failed to account

for the dependencies, which may result in some important dependency information. Intuitively, there should be different effects for neighborhood nodes with different dependencies. This model used additional relational headers to extend the original GAT, which served as relational-level gates to control the flow of information from the neighborhood nodes. Figure 3 shows the overall architecture of the proposed method. So as to support the representation learning of the constructed dependent trees, this paper used the GAT model to aggregate the neighborhood node representation by using two different aggregation functions, iteratively updating each node representation, such as equations (3) and (5).

$$f_{\text{node } i}^{l+1} = \sum_{j \in N_i}^K a_{ij}^{lk} W_k^l h_j^l, \quad (3)$$

$\sum_{k=1}^K$ represented the concatenation of vectors from x_1 to x_k , and W_k^l was a trainable weight matrix. The a_{ij} represented the attention coefficient of the node_j , use the same method as literature [24] calculated as in the following formula:

$$a_{ij} = \text{attention}(\text{node}_i, \text{node}_j). \quad (4)$$

This model used edge attention to highlight the next layer representation of the learning node node_i . This procedure can be represented in the following form as:

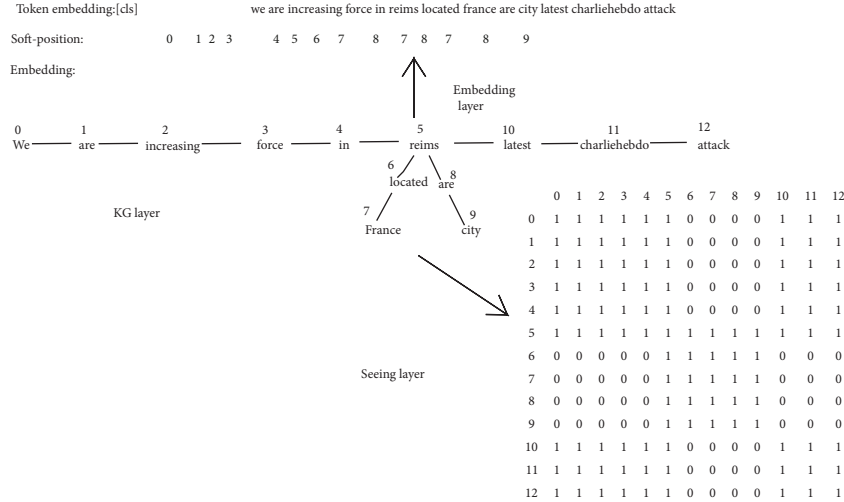


FIGURE 4: Schematic diagram of knowledge embedding layer, the upper left corner is embedding, and the lower right corner is the visible matrix.

$$h_{\text{edge } i}^{l+1} = \parallel_{m=1}^M \sum_{j \in N_i} B_{ij}^{lk} W_k^l f_j^l. \quad (5)$$

W_k^l was a weight matrix. B_{ij}^{lk} represented the attention coefficient of the edge e_j on the node v_i , which can be calculated by (6) and (7).

$$B_{ij}^{lm} = \frac{\exp(a^T v_i^{lm})}{\sum_j \int N_i(a^T v_j^{lm})}, \quad (6)$$

$$v_j = \text{Leak Relu}([w_2 f_j^{l+1} w_1 h_l j]). \quad (7)$$

Among these, a^T was a weight vector for computational dependencies, while \parallel was a connection operation. This model could not only capture higher-order word interactions, but also learn dependencies of dependent trees.

3.2. Knowledge Embedded Module. The knowledge embedding module consisted of four modules, namely, knowledge layer, embedding layer, seeing layer, and mask-self-attention. For the sentence input, the knowledge layer, first, injected the relevant triples into the sentence from the KG, and then, transformed the original sentence into knowledgeable sentence trees. The sentence tree was then input into both embedding layer and seeing layer and then converted to token-level embedding representations and visible matrices. A visible matrix was used to determine the visible region of each subject word, preventing changes in the original meaning of the sentence due to excessive knowledge edges injected. A simple example: “We are increasing our force in reims latest on charliehebdo attack”. For reims, triples “reims country France” were used and noise occurred when injecting triples into the sentence, the effect of the introduced sentences on the original sentence was reduced through the seeing layer.

Lastly, input the embedding layer and seeing layer to learn feature representation from self-attention, as shown in Figure 4. Among them, at the knowledge layer, for traditional knowledge graphs, such as SNOMED-CT, HowNet, were not suitable for rumor detection, so that triples of the grasped data from the DBpedia were injected into the sentence. Given the input sentence $S = \{w_0, w_1, w_2, \dots, w_n\}$, and KG, the output sentence tree $t = \{w_0, w_1, \dots, w_i \{r_i^0, w_i^0\}, \dots, (r_i^k, w_i^k)\}, \dots, w_n\}$. From input to output, it can be divided into two steps: knowledge query and knowledge injection. In the knowledge query, the motifs in the sentence were selected to query their corresponding triples from the KG. The knowledge injection could be expressed as $E = K\text{-Query}(S, K)$, where $E = \{(w_i^0, r_i^0, w_i^0), \dots, (w_i^k, r_i^k, w_i^k)\}$ is the set of the corresponding triples.

Next, K-Inject generated the sentence tree T by injecting the triples into sentence by adding the triples in E to its corresponding location. K-Inject could be represented as $T = K\text{-Inject}(S, E)$. Self-attention was adopted to prevent K-Inject changes from taking advantage of the sentence structure information in N . The formula was as follows: (8), (9), (10):

$$Q^{i+1}, K^{i+1}, V^{i+1} = h^i W_q, h^i W_k, h^i W_v, \quad (8)$$

$$G^{i+1} = \text{soft max} \left(\frac{Q^{i+1} K^{i+1} + M}{\sqrt{d_k}} \right) V, \quad (9)$$

$$G^{i+1} = G^{i+1} V^{i+1}. \quad (10)$$

Where W_q , W_k and W_v are trainable model parameters. h^i is the hidden state of the i -th self-attention blocks. d_k is the scaling factor. N is the visible matrix calculated by the seeing layer. if w_k is invisible to w_j , the M_{jk} will mask the attention score G^{i+1} , which means w_k make no contribution to the hidden state of w_j .

Input: aspect $a = \{w_i^a, w_{i+1}^a, \dots, w_k^a\}$, sentence = $\{w_1^s, w_2^s, \dots, w_n^s\}$, dependency tree T , and dependency relations r .
Output: subject dependency tree T_1 . 1:

- (1) Construct the root R for T_1 ;
- (2) for x to k do
- (3) for $y=1$ to n do
- (4) if $b_y^s \rightarrow b_x^a$ then,
- (5) $b_y^s \rightarrow R$
- (6) else if $b_y^s \leftarrow b_x^a$ then,
- (7) $b_y^s \leftarrow R$
- (8) else
- (9) $b_y^s = \text{change word}((b_y^s))$
- (10) $n = \text{distance}(x,y)$
- (11) $b_y^s \rightarrow {}^n \text{con}R$
- (12) end if
- (13) end for
- (14) end for
- (15) return T_1

ALGORITHM 1: subject Dependency Tree.

3.3. Attention Module. The Attention Module is an interaction between the knowledge-embedding module and the subject-dependent tree module. The graph-based representations that guided the updates learned by self-attention could also guide the graph-based representations learned by the graph neural network.

Assuming the existence of three input matrices: $Q \in R_n \times D_k$, $K \in R_m \times D_k$, $V \in R_m \times D_v$, which, respectively, represented query matrix, key matrix, and value matrix. n and m are the length of two inputs; text representations learned by self-attention guide the updates of graph neural network. Text representations were converted to key and value, namely, K and V , and graph representation was converted to queries, namely, Q . The calculation procedure was shown in the following formula:

$$\begin{aligned} Q' &= \text{Attention}(Q, K, V) \\ &= \text{soft max} \left(\frac{QK^T}{\sqrt{d_k}} \right) V. \end{aligned} \quad (11)$$

The updated graph representation was then spliced with the original graph representation, and finally, converted to the original dimension via the full connection layer without the activation function. The calculation procedure of Q is shown in the following formula.

$$Q = \text{Linear}([Q', Q]). \quad (12)$$

The graph representation Q was averagely pooled to obtain the final graph representation Q' . The update of text representations learned by graph-based representation via graph neural networks was similar to the update flow of text representations. Graph representations were transformed to key and value, text representations were transformed to query, and text features were updated to H' via an attention mechanism.

Finally, the attention weights of the nodes after each update were calculated through a two-layer feedforward neural network, and both the updated graph representation

and the text representation were fused by the attention weights, with the calculation process shown in the following formula.

$$\begin{aligned} \text{attention} &= \text{soft max}(w_2 \tanh(w_1 [Q', H'] + b_1) + b_2), \\ F &= \text{Rule}(\text{Linear}(\text{attention}[Q', H'])). \end{aligned} \quad (13)$$

4. Simulation Experiment

4.1. Dataset. This experiment used the standard Twitter dataset publicly available proposed by Ma et al. The Twitter dataset, presented in 2016, was quickly recognized by academics and researchers, are now widely used for text classification tasks, is a classic dataset on text classification issues. The model used the Twitter ID of the Twitter dataset, texts, and entities, where the specific data information was shown in Table 1. The original dataset is divided into two mutually exclusive datasets, namely, training set and testing set by calling the function, the training set and the test set account for three and seven parts of the dataset, respectively, with 4061 events as the training set, 1741 events as testing set to test the model in this paper. As shown in Table 2.

Knowledge graphs in many fields have been constructed, such as SNOMED-CT in the medical field and HowNet in the Chinese concept, the knowledge graph used in this paper is based on Wikipedia. Since Wikidata has more than 24.7 million knowledge entities, the search is very time-consuming, so we manually search out the entities used in the dataset to establish the knowledge graph.

4.2. Parameters and Environment. A Biaffine Parser was used for the dependency solution, and the dimension of the dependency embedding was set to 300. The batch was set to 32, and the number of training rounds was 30. This model was trained on the GPU of python3.6, pytorch1.2.0; specific superparameters are shown in the following Table 3:

TABLE 2: Data information of Twitter.

Statistics	Twitter
User#	49,345
Posts#	103,212
Events#	5,802
Rumors#	1,972
Nonrumors	3,830
Avg.hours per event	33.4
Avg.# of posts per event	17
Max # of posts per event	346
Min # of post per event	1

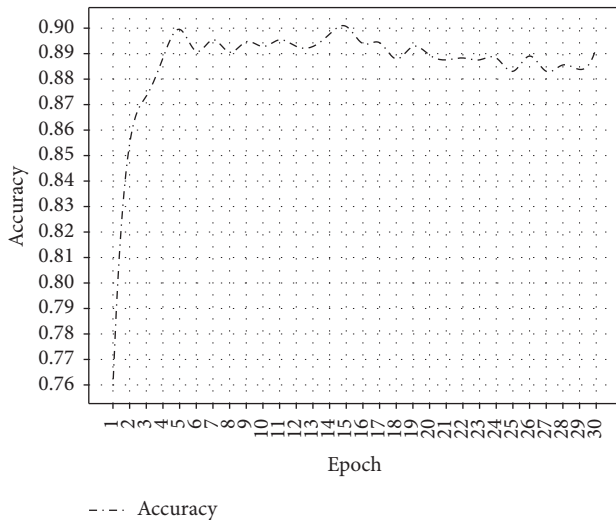


FIGURE 5: Schematic diagram of evaluation index accuracy changes.

4.3. *An Introduction of the Comparison Model.* Some mainstream models of rumor detection were used for comparison, including:

Circulating Neural Network (CNN): Considering the comment information re-transmitted and the timing characteristics of messages in the propagation process, the long-term context information was modeled on the timeline to realize the time-series-based representation of events.

Long and Short-Term Memory (LSTM): LSTM can learn long-distance dependencies through gate structure and memory units to capture text features from local continuous word sequences.

Gated Recurrent Unit (GRU): The GRU improved the LSTM to integrate forgetting date and the input gate as a single update gate. It also mixed cell and hidden states, thus increasing the speed of the model processing data.

Transformer [25]: The transformer used an attention mechanism to model the dependence of the input-output sequences without consideration of their distance in the sequence.

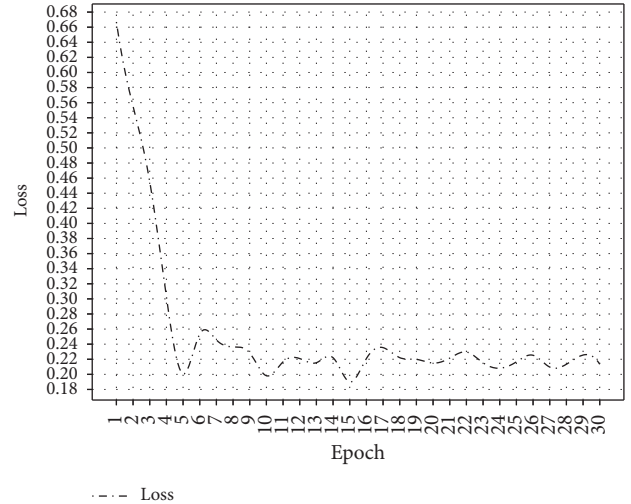


FIGURE 6: Schematic diagram of evaluation index loss changes.

TABLE 3: Hyperparametric table.

Hyperparametric	Numerical value
num_heads	6
Dropout	0.2
learning_rate	$5e-5$
Epsilon for adam optimizer	$1e-8$
num_train_epochs	30
hidden_size	200
batch_size	32
embedding_dim	300

BERT [26]: A language model that trained text bidirectional by using the encoder part of self-attention could capture longer-distance dependencies more efficiently.

Bi-GCN [27]: The GCN was based on the rumor detection model to model the text by using the two-way propagation structure.

5. Experimental Analysis and the Results Analysis

5.1. *Training Loss and Accuracy.* During the training of the model, we set up 30 rounds of training, and the indicators obtained by the model became stabilized as the number of training rounds increased. A schematic diagram of the evaluation index accuracy changes was shown in Figure 5. Schematic diagram of evaluation index loss change was shown in Figure 6. In the first four rounds, the accuracy value obtained by the model was relatively low; the rumor detection effect of this model was poor, and in the fifth round, the accuracy value of the model increased rapidly. As the number of training rounds increased, the results of the model on the three metrics, finally, floated around the optimal value.

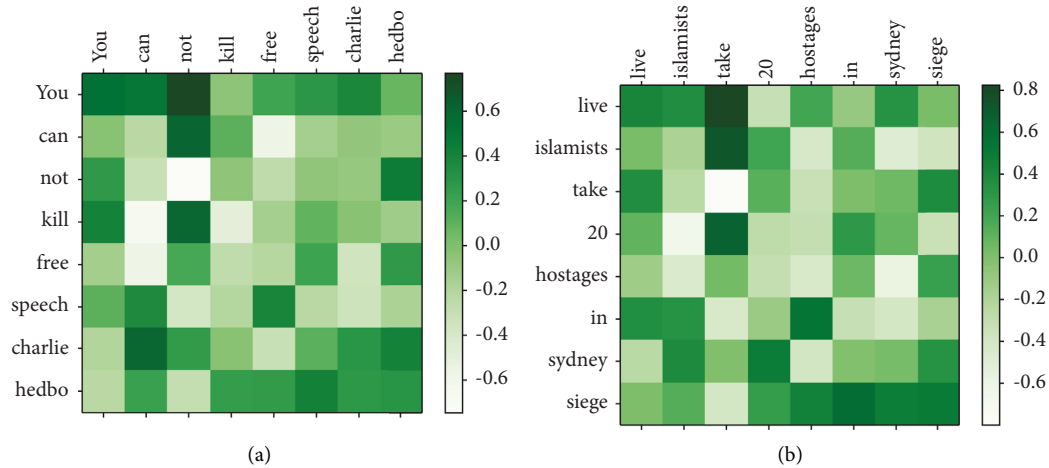


FIGURE 7: Case Study: a testing example demonstrates that the information of dependency tree contributes to the classification performance; our model generated a proper attention distribution with the assistance of dependency tree. Darker cell color indicates higher attention value. (a) The attention matrix of nonrumor by our method. (b) The attention matrix of rumor by our method.

In conclusion, the number of rounds trained had an important impact on the experimental results. As the number of training rounds increased, the accuracy of this model in rumor detection continuously improved and stabilized after a certain number of rounds. The model presented in this paper showed good detection effects after its stability.

5.2. Comparison Models. The overall performance of all the models was listed in Table 4, from the observation of Table 4, it can be concluded that the accuracy and F1 value of the traditional machine learning baseline model are relatively low, because the traditional machine learning manual feature extraction is cumbersome, so in the direction of rumor detection, deep learning is more effective than traditional machine learning. The deep learning models CNN, RNN, LSTM, and GRU in the table all belong to the simple recurrent neural network model, with accuracy and F1 above 60%. In terms of constructing and mining rumor features, the simple recurrent neural network model does not consider the characteristics of important spatial levels among objects. The performance of the method of modeling rumors based on graph and tree structure is better than that of the method of baseline time series. The effect of graph neural network model in identifying rumors should reach more than 80%, but the correlation between Bi-GCN and computing vertices is weak. The GAT model can fully consider the correlation between vertices and pay attention to the relationship between words. From which some observations could be noted. First, the GAT model outperformed most benchmark models. Second, in a subject-based dependent tree structure, GAT performance was significantly improved when combined with text word embedding representations. It also outperformed most of the baseline models, demonstrating that our GAT encodes the grammatical information better. After self-attention + GAT, this powerful model had further improved and achieved better

detection results. These results demonstrated the effectiveness of our self-attention + GAT in terms of the syntactic structure of rumor detection.

5.3. Case Study and Attention Distribution Exploration.

In order to observe the effect of attention on the model, a nonrumor and rumor were selected for the study. As shown in Figure 7(a), our method predicted that the post you can not kill free speech Charlie Hebdo was positive. In order to find out the reason, we studied the attention matrix and found that the model paid more attention to negative words and verbs, the common dependency tree was reconstructed to make it dependent on Charlie Hebdo. According to the dependency tree, not and kill form a dependency relationship, kill and free speech form a dependency relationship, and finally, free kill and Charlie Hebdo form a dependency relationship. Finally, it was identified as nonrumor. As shown in Figure 7(b), our method predicted that the post “live Islam take 20 hostages in Sydney College” was a rumor. In order to find out the reason, we studied its attention matrix. This model paid more attention to verbs and the words around them. Islam and in form a prep relationship, and in and Sydney College form a pobj, which was finally determined as a rumor.

5.4. Effects of the Different Parsers. Dependent analysis played a crucial role in this model. To assess the impact of the different parsers, we conducted a study based on GAT models by using two well-known dependency parsers.

After using the comparison, the better the Biaffine parser was, the higher the classification accuracy would be, as shown in Table 5. Furthermore, it further implied that our approach had the potential to be further improved with analytic techniques while existing parsers could correctly capture syntactic structure information.

TABLE 4: Comparison of detection accuracy on Twitter.

Method accuracy	Twitter			
	Rumors		Nonrumors	
	Precision	F1	Precision	F1
Baseline	0.548	0.581	0.558	0.577
CNN	0.653	0.655	0.643	0.663
LSTM	0.796	0.706	0.723	0.724
RNN	0.785	0.712	0.692	0.701
GRU	0.736	0.729	0.734	0.745
BERT	0.752	0.792	0.819	0.782
Bi-GCN	0.880	0.924	0.930	0.853
Ours	0.885	0.937	0.942	0.856

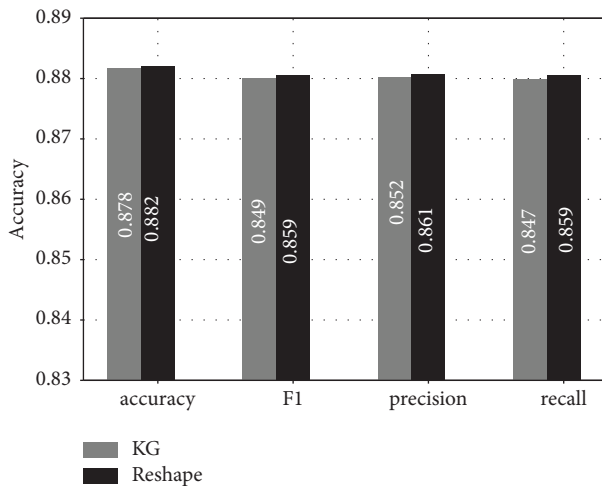


FIGURE 8: Schematic diagram of comparative analysis of knowledge graph and reshaped dependency tree.

TABLE 5: GAT results based on two different parsers, unlabeled attachment score, and labeled attachment score are indicators for evaluating parsers. The higher the score, the better the performance.

Parser	Performance		Dataset twitter
	UAS	LAS	
Stanford	94.10	91.49	0.885
Biaffine	95.74	94.08	0.853

5.5. Knowledge Graph & Comparative Analysis of the Remodeling-Dependent Trees. In this paper, two methods of a graph neural network-based rumor detection model were testified through subject enhancement. KG represented the application of the common dependency trees. Reshape represented the application of text unprocessed. The experimental results are shown in Figure 6. Rumor detection results were testified by using text and ordinary dependent tree methods, respectively, by comparison of four evaluation measures, accuracy, F1, precision, and recall. The effect of rumor detection by using the remodeling dependent trees was 0.004, 0.04, 0.01, 0.009, and 0.012 higher than those under the common dependency trees, respectively. It proved that the remodeling structure of the dependent trees had improved the accuracy of the model, as shown in Figure 8.

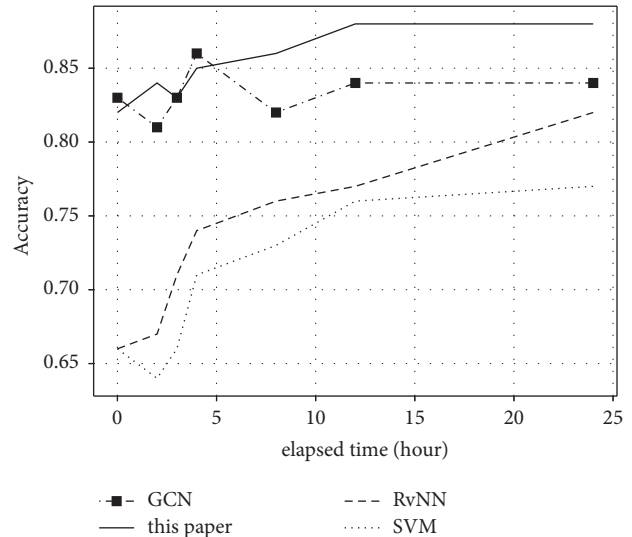


FIGURE 9: Early detection accuracy of rumors, accuracy obtained from Twitter datasets sent by posts and test posts scanned incrementally.

TABLE 6: accuracy and F1 obtained after adding five typical ablation conditions.

Ablation	Twitter	
	Accuracy (%)	F1 (%)
Transformer + GAT	88.5	85.7
(i) Mask	88.3	85.4
(ii) Reshape	87.8	84.9
(iii) BiAffine	87.1	84.3
(iv) KG	88.2	84.1
(v) Attention	88.1	85.2

5.6. Ablation Experiments. As was shown in Table 6, we investigated and reported five typical ablation conditions. “-Mask” indicated that we had removed the mask mechanism of self-attention, “Reshape” indicated that we had only used a common syntactic dependent tree by removing the sentence reconstruction trees. “-BiAffine” represented that we had removed the BiAffine process and used the output of the BiLSTM structure. So we could conclude that the BiAffine process was critical for our model. “-KG” means the removal of knowledge embedding without processing the text. “-attention” represented the representation of the removed middle graph and the interaction module of text representation.

5.7. Early Rumor Detection. In order to take precautions to prevent the spread in time, it is important to expose them in the early stages of spread. In the early detection task, we compared different detection methods to assess performance with the accuracy obtained for Twitter datasets being sent from posts and testing posts scanned incrementally.

Figure 9 shows the comparison of the method of this model with the GCN, RNN, and SVM models. It could be observed that a phenomenon that the early performance of

all methods fluctuated more or less. This was because as the posts spread, there was more semantic and structural information. Meanwhile, the amount of noise information increased correspondingly. The method of this model could achieve better results in the Twitter dataset after its publication 4 hours later, indicating the superior performance of this model in the early detection of rumors.

6. Conclusions

This paper presented a graph-based neural network model under topic enhancement. First, the topic of news text was injected into the knowledge graph for topic enhancement, which could modify the effect of the mask mechanism to achieve noise reduction. Second, reshaping of the dependent trees enabled it to be subject-dependent. We only retained edges that had a direct dependence on the motif by pruning the trees. Finally, this paper integrated the text representations learned by self-attention with the graph-based representations learned by the graph attention network.

Dependency graphs can guide and facilitate text representation learning. The final text representations derived by self-attention could be used for perceptual classification together with the motif-based dependency plots. The Twitter dataset was extensively evaluated, the accuracy of this model had outperformed many previous mature models, and on various other metrics, the present model also outperformed the benchmark model, and the experimental results showed that our model outperformed the baseline method. It is a promising direction to apply graph neural networks to rumor detection in the following research [28–37].

Data Availability

The data used to support the findings of this study are openly available at https://figshare.com/articles/PHEME_dataset_of_rumours_and_non-rumours/4010619. The code implementation of this paper is placed on Gitee (<https://gitee.com/wertr/rumor-detection-based-on-knowledge-enhancement-and-graph-attention-network/tree/master/>).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank the anonymous reviews for their helpful suggestions. And the work was supported by the Technical Innovation Guidance Special Fundation of Tianjin (Grant No. 21YDTPJC00130).

References

- [1] A. Ma, W. Gao, P. Mitra et al., “Detection rumors from microblogs with recurrent neural networks,” *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, vol. 16, pp. 3818–3824, 2016.
- [2] S. Wang, S. Mazumder, and B. Liu, “Target-sensitive memory networks for aspect sentiment classification,” *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 957–967, 2018.
- [3] A. Bosselut, H. Rashkin, and M. Sap, “COMET: commonsense transformers for automatic knowledge graph construction,” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, vol. 57, pp. 4762–4779, 2019.
- [4] C. zhang, Q. Li, and D. A. Song, “Aspect-based sentiment classification with aspectspecific graph convolutional networks,” *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, vol. 9, pp. 4560–4570, 2019.
- [5] Y. Lv, M. Sun, Y. Wen, and W. Wang, “Rumor detection based on time graph attention network,” in *Proceedings of the 2022 4th International Conference on Advances in Computer Technology*, Suzhou, China, April 2022.
- [6] Z. Zhao, P. Rensnick, and Q. Mei, “Enquiring minds: Early detection of rumors in social media from enquiry posts,” *Proceedings of the 24th international conference on world wide web*, vol. 24, pp. 1395–1405, 2015.
- [7] F. Yang, Y. Liu, X. Yu, and M. Yang, “Automatic detection of rumor on sina weibo,” *Proceedings of the ACM SIGKDD workshop on Mining Data Semantics*, vol. 12, pp. 1–7, 2012.
- [8] C. Guo, J. Cao, X. Zhang, K. Shu, and M. Yu, “Exploiting Emotions for Fake news detection on social media,” 2019, <https://arxiv.org/abs/1903.01728>.
- [9] S. Kwon, M. Cha, K. Jung, W. Chen, and W. Wang, “Prominent feature of rumor propagation in online social media,” *IEEE 12th international conference on Data Mining*, vol. 12, pp. 1103–1108, 2013.
- [10] T. Chen, L. Wu, X. Li, H. Yin, and J. Zhang, “Call attention to rumors: deep attention based recurrent neural networks for early rumor detection,” Article ID 059735, 2017, <https://arxiv.org/abs/1704.05973>.
- [11] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, “A convolutional approach for misinformation identification,” *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, vol. 17, pp. 3901–3907, 2017.
- [12] M. Rospocher, M. Vanerp, P. Vossen et al., “Building event-centric knowledge graphs from news,” *Journal of Web Semantics*, vol. 37–38, pp. 132–151, 2016.
- [13] S. Gottschalks and E. Demidovae, “Event KG: A multilingual event-centric temporal knowledge graph,” in *Proceedings of the European Semantic Web Conference*, vol. 15, pp. 272–287, Hersonissos, Greece, June, 2018.
- [14] M. Hernes and A. Bytniewski, “Knowledge representation of cognitive agents processing the economy events,” *Asian Conference on Intelligent Information and Database Systems*, vol. 1, pp. 392–401, 2018.
- [15] B. Xu, X. Shu, and Y. Song, “X-Invariant contrastive augmentation and representation learning for semi-supervised skeleton-based action recognition,” *IEEE Transactions on Image Processing*, vol. 31, no. 5, pp. 3852–3867, 2022.
- [16] X. Shu, J. Yang, R. Yan, and Y. Song, “Expansion-squeeze-excitation fusion network for elderly activity recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, 2022.
- [17] X. Shu, L. Zhang, G. J. Qi, W. Liu, and J. Tang, “Spatio-temporal co-attention recurrent neural networks for human-skeleton motion prediction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3300–3315, 2022.
- [18] J. Tang, X. Shu, R. Yan, and L. Zhang, “Coherence constrained graph LSTM for group Activity recognition,” *IEEE*

- Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 636–647, 2022.
- [19] T. N. Kipf and M. W. Thomas, “Semi-supervised classification with graph convolutional networks,” *International Conference on Learning Representations*, vol. 2016, Article ID 02907, 2016 <https://arxiv.org/abs/1609.02907>.
- [20] D. Marcheggiani and D. Ivan, “Encoding sentences with graph convolutional networks for semantic role labeling,” *Conference on Empirical Methods in Natural Language Processing*, vol. 17, pp. 1506–1515, 2017.
- [21] X. Shu, L. Zhang, Y. Sun, and J. Tang, “Host–parasite: graph LSTM-in-LSTM for group activity recognition,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 663–674, 2021.
- [22] Z. Zhang, F. Zhuang, H. Zhu, Z. Shi, H. Xiong, and Q. He, “Relational graph neural network with hierarchical attention for knowledge graph completion,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 5, pp. 9612–9619, 2020.
- [23] B. Huang and K. Carley, “Parameterized convolutional neural networks for aspect level sentiment classification,” *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, vol. 18, pp. 1091–1096, 2018.
- [24] X. Yang, H. Ma, and M. Wang, “Rumor detection with bi-directional graph attention networks,” *Security and Communication Networks*, vol. 2022, Article ID 4840997, 2022.
- [25] A. Vaswani, N. Shazeer, and N. Parmar, “Attention is all you need,” *Proceedings of the 31st International Conference on Neural Information Processing Systems*, vol. 31, pp. 6000–6010, 2017.
- [26] J. Devlin, M. Chang, and K. Lee, “BERT: pre-training of deep bidirectional transformers for language understanding,” 2018, <https://arxiv.org/abs/1810.04805>.
- [27] T. Bian, X. Xiao, T. Xu et al., “Rumor detection on social media with Bi-directional graph convolutional networks,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 549–556, 2020.
- [28] S. Ai, S. Hong, X. Zheng, Y. Wang, and X. Liu, “CSRT rumor spreading model based on complex network,” *International Journal of Intelligent Systems*, vol. 36, no. 5, pp. 1903–1913, 2021.
- [29] X. Chen, F. Zhou, F. Zhang, and M. Bonsangue, “Modeling microscopic and macroscopic information diffusion for rumor detection,” *International Journal of Intelligent Systems*, vol. 36, no. 10, pp. 5449–5471, 2021.
- [30] R. Zhang and D. Li, “Identifying influential rumor spreader in social network,” *Discrete Dynamics in Nature and Society*, vol. 2019, Article ID 8938195, 2019.
- [31] J. Borge-Holthoefer and Y. Moreno, “Absence of influential spreaders in rumor dynamics,” *Physical Review*, vol. 85, no. 2, Article ID 026116, 2012.
- [32] S. Srinivasan and B. Dhinesh, “A social immunity based approach to suppress rumors in online social networks,” *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 5, pp. 1281–1296, 2021.
- [33] F. Liu and M. Li, “A game theory-based network rumor spreading model: based on game experiments,” *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 6, pp. 1449–1457, 2019.
- [34] Q. Huang, C. Zhou, J. Wu, L. Liu, and B. Wang, “Deep spatial–temporal structure learning for rumor detection on twitter,” *Neural Computing & Applications*, vol. 32, 2020.
- [35] M. T. Rashid and D. Wang, “CovidSens: a vision on reliable social sensing for COVID-19,” *Artificial Intelligence Review*, vol. 54, pp. 1–25, 2021.
- [36] S. Raza and C. Ding, “News recommender system: a review of recent progress, challenges, and opportunities,” *Artificial Intelligence Review*, vol. 55, no. 1, pp. 749–800, 2022.
- [37] Q. Li, C. Zeng, W. Xu, and Y. Xiao, “A social rumor and anti-rumor game diffusion model based on sparse representation and tensor completion,” *Journal of Network and Computer Applications*, vol. 201, Article ID 103343, 2022.