RESEARCH ARTICLE

Dynamic graph convolutional networks with attention mechanism for rumor detection on social media

Jiho Choi, Taewook Ko, Younhyuk Choi, Hyungho Byun, Chong-kwon Kim*

Department of Computer Science and Engineering, Seoul National University, Seoul, Republic of Korea

* ckim@snu.ac.kr

Abstract

Social media has become an ideal platform for the propagation of rumors, fake news, and misinformation. Rumors on social media not only mislead online users but also affect the real world immensely. Thus, detecting the rumors and preventing their spread became an essential task. Some of the recent deep learning-based rumor detection methods, such as Bi-Directional Graph Convolutional Networks (Bi-GCN), represent rumor using the completed stage of the rumor diffusion and try to learn the structural information from it. However, these methods are limited to represent rumor propagation as a static graph, which isn't optimal for capturing the dynamic information of the rumors. In this study, we propose novel graph convolutional networks with attention mechanisms, named *Dynamic GCN*, for rumor detection. We first represent rumor posts with their responsive posts as dynamic graphs. The temporal information is used to generate a sequence of graph snapshots. The representation learning on graph snapshots with attention mechanism captures both structural and temporal information of rumor spreads. The conducted experiments on three real-world datasets demonstrate the superiority of Dynamic GCN over the state-of-the-art methods in the rumor detection task.

OPEN ACCESS

Citation: Choi J, Ko T, Choi Y, Byun H, Kim C-k (2021) Dynamic graph convolutional networks with attention mechanism for rumor detection on social media. PLoS ONE 16(8): e0256039. https://doi.org/10.1371/journal.pone.0256039

Editor: Chi-Hua Chen, Fuzhou University, CHINA

Received: March 3, 2021

Accepted: July 28, 2021

Published: August 18, 2021

Data Availability Statement: All relevant datasets are publicly available on [1] Figshare, and our model is available on our public GitHub repository [2]. [1] https://figshare.com/s/d8984fd39557a3d295e8 [2] https://github.com/jihochoi/dynamic-gcn.

Funding: This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIP) (No. (NRF-2016R1A5A1012966) and (No. NRF-2017R1A2A1A01007400). This work was also supported by the BK21 plus program through the National Research Foundation (NRF) funded by the

1 Introduction

Social media has been a great disseminator for new information and thoughts. Due to its accessibility of sharing information, however, social media has also become an ideal platform for propagations of rumors, fake news, and misinformation [1]. Although the definition of rumor may vary by literature, we use the term rumor to indicate messages in which the veracity labels are unknown at the time of diffusion [2, 3]. Rumors on social media not only mislead the users of online but also affect the real world immensely [4]. Thus detecting the rumors and preventing their spread became an essential task.

Early studies in rumor detection focused on understanding the characteristics of rumors [5, 6] and extracting prominent features of rumor from the textual contents or the users' profiles [7–11]. Also, the temporal features or propagation patterns were elaborated significantly in [12–17], respectively. These elaborated features show profound results in rumor detection

Ministry of Education of Korea. The authors deeply appreciate the administrative support for this work from the Institute for Industrial Systems Innovation of Seoul National University.

Competing interests: The authors have declared that no competing interests exist.

tasks. The manually extracted content-based, user-based, or propagation-based handcrafted features were used to train classical machine learning classifiers such as a decision tree, random forest, or SVMs. However, the limitation of using manually extracted features is that it fails to capture the high-dimensional patterns of rumors.

To solve the problem of using handcrafted features and avoid the feature engineering efforts [18–21], had adopted neural networks such as recurrent neural networks (RNNs) or convolutional neural networks (CNNs). The proposed rumor detection models were able to capture the high-dimensional representation from the textural contents, user profiles, and propagation structures. The models of using the propagation structure [20, 21] try to represent the skeptical or conflict opinions of the responsive posts such as retweets, replies, or comments toward the original message.

Recent advent in Graph Neural Networks (GNNs) and its variants such as Graph Convolutional Networks (GCN), GraphSAGE, and Graph Attention Networks (GAT) [22–25] have gained a lot of attention. The GNNs have shown promising results in graph inference tasks such as node classification, graph classification, and link prediction. [26, 27] successfully adopted GCN and GAT in the rumor detection domain, respectively. However, both models aren't considering the temporal dynamics of the rumor propagation, which only considers the static graph structure of the final state of rumor propagation.

In this study, motivated by the dynamic nature of rumor propagation, we present a novel graph convolutional network-based model, named *Dynamic GCN*, to better understand the evolving pattern of rumor propagation. The model includes two distinct ways of representing rumor propagation with graph snapshots: sequential and temporal snapshots. Fig 1 depicts how the rumor propagation can be represented with the sequence of snapshots. In the example scenario, the initial trust (Fig 1a) of the root post begins to gain doubts (Fig 1b), and the posts that reveal doubts are supported by others (Fig 1c). With this whole process, the veracity value of the root post can be inferred. The details of the representation will be discussed in section 4. The extended GCNs capture the spatial representation of rumor posts within a snapshot. And finally, the series of graph snapshot representations are combined with an attention mechanism. We evaluate the proposed model with three real-world datasets and show our model outperforms other state-of-the-art methods.

We summarize the main contributions as follows:

- We propose two distinct ways of depicting a dynamic graph by generating two variants of graph snapshots: sequential and temporal snapshots.

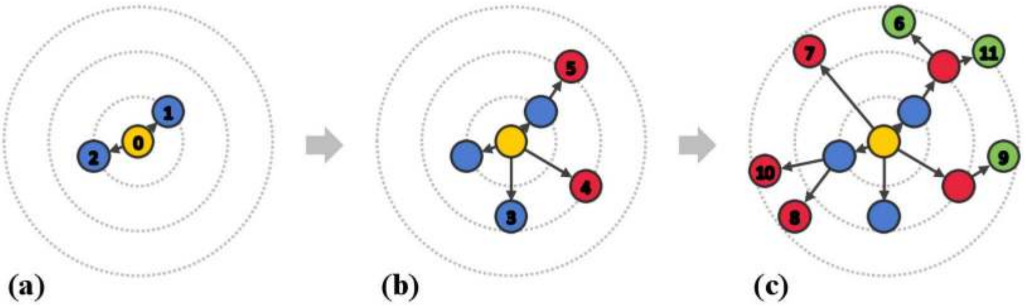

Fig 1. Example of an evolving rumor propagation network where each node represents a post. The link between nodes implies a direct reaction. The numbers in nodes are in the chronological ordering of the generation, and the distance from the root post represents the time interval. The colors depict users' latent stances on their parent's post (e.g., neutral (blue), suspect/doubt (red), trust/support (green)). Example scenario: there is an initial claim of Node 0. Node 1, 2, and 3 express neutral stances to the initial claim. Node 4, 5, 7, 8, and 10 express suspect/doubt and Node 6, 9, 11 express trust/support on their parent posts, respectively. Can we identify the veracity label of Node 0?

https://doi.org/10.1371/journal.pone.0256039.g001

- We propose a novel GCN-based rumor detection model that can capture the evolving pattern of rumor propagation by aggregating the structural representations of snapshot sequences.
- The conducted experiments on three real-world datasets demonstrate that our model accomplishes superior results on the rumor detection task compared to other state-of-the-art methods.

We organize this paper as follows. In Section 2, we briefly review the rumor detection methods and the fundamental components of our model; GCNs and attention mechanisms. In Section 3, we formulate the rumor detection problem with the propagation structure of rumor. In Section 4, we introduce our model as follows: snapshot generation, graph convolution networks, readout layer, attention mechanisms, and prediction. In Section 5, the details of experiments and performance evaluation are described. And finally, we conclude this work in Section 6.

2 Related work

2.1 Rumor detection

Rumor is commonly defined as a message in which the veracity labels are unknown [2, 3]. Rumor detection on social media is a task of classifying messages or posts with their veracity labels. Traditional approaches in rumor detection and other misinformation detection are to extract handcrafted features with prior knowledge on rumors. The content-based method and user-based method were two main approaches [7–9, 11]. To elaborate different and additional features, the temporal or linguistic features were considered in [12–14]. Another characteristic feature of the rumor is its propagation structure. [15–17] utilize propagation patterns of rumor and show profound results on rumor detection. The manually extracted content-based, user-based, temporal, or propagation-based handcrafted features were used to train classical machine learning classifiers such as a decision tree, random forest, or SVMs. However, the limitation of models with handcrafted features is that they fail to capture the high-dimensional patterns of rumors. To solve the problem [18, 19], adopted deep learning models such as RNNs or CNNs variants to extract texture, image features, or user profile features from the rumor posts. Noticeably, models which utilize propagation structure as additional features that try to represent the skeptical or conflict opinions from the responsive posts. Recently, sophisticated models like GCN [26] or GAT [27] have successfully been adopted in the rumor detection domain.

2.2 Representation learning on graphs

Promising results on neural networks in various fields, encourage studies to bring deep learning to topological graph structures. Early studies of node embedding [28, 29] leverage sampling method like random walk for shallow node embedding. Recent advent in graph neural networks (GNNs) and its variants [22–25] made representation learning to be applied directly to a variety of graph structures such as social networks (friendship network, citation network, transaction network), knowledge graphs, computer networks, biochemical graph, and so on. One of the early and honored studies of GNNs is graph convolutional networks (GCNs) [23]. It approximates spectral filters with Chebyshev polynomial to extend convolutional operations on graphs. Another important study of the GNNs variant is GraphSAGE [24], which proposes different trainable aggregation functions from neighbor node embeddings with sampling methods. The proposed aggregation functions like mean, LSTM (random ordered), max-pooling are symmetric, where the ordering of neighbor nodes can be invariant. GAT [25] utilizes the attention mechanism for neighbor node embeddings. The GNNs have firmly established state-of-the-art performance in various graph inference tasks such as node classification, graph classification, link prediction, and community detection (clustering for the network structure). The fundamental component of GNNs is message passing architecture where the representation of the node is aggregated with its neighbors. The key differences in GNN variants are diverse neighborhood aggregation methods and different pooling approaches [30, 31].

2.3 Attention mechanism

The attention mechanism captures the importance of the input sequence by calculating the attention scores and weights. Compared to RNN-variants, such as Long Short-Term Memory (LSTM) [32], Gated Recurrent Units (GRU) [33], or Seq2Seq model [34], attention mechanisms have demonstrated outstanding results on both the efficiency and the performance in a variety of fields [35, 36]. Various attention mechanisms have been proposed depending on how they calculate the attention weights. [36] proposed additive attention, which adopts a feedforward neural network to calculate the importance of the input in the context of the input sequence. [35, 37] suggested dot-product attention and self-attention, which utilized dot-product similarity to capture the significance of certain input words from the set of words in the task of neural machine translation. Attention mechanism had also introduced and shown promising results in graph representation learning [25] where the node embeddings are calculated and attended over their neighbor nodes' features.

2.4 Representation learning on dynamic graph

Graph structure like social network contains the property of dynamics by its nature [38]. Different approaches have been proposed to capture the dynamics of graphs. Early studies [39, 40] have focused on the changes or graph properties such as clusters, centralities, and similarities in certain temporal points of graphs called graph snapshots. From the advancement of feature-based dynamic graph representations, architectures with triadic closure and RNNs [41, 42] were adopted to embed sequences of graph structures. [43] suggested Dyngem which utilizes the snapshot method with an autoencoder to embed the evolving graphs. As the GNN-based methods have shown promising results on graph embedding tasks [44, 45], proposed GCN architectures combined with LSTM, GRU for the dynamic graph embedding. [46] applied a self-attention mechanism for representing the dynamic graphs.

3 Problem definition

In this section, the rumor detection task on graph structure is described. Rumor detection aims to predict the veracity label of a message. We formulate the task as below.

Let $C = \{c_1, c_2, \cdots, c_m\}$ be the set of m claims, where each claim (or a conversational thread) c_i consists of n_i microblog posts $P_i = \{p_{i0}, p_{i1}, \cdots, p_{i(n_i-1)}\}$. The p_{i0} is the root post of c_i and $n_i - 1$ responsive posts are in chronological order by their post time. Each post p_{ij} is represented with F dimensional feature $x_{ij} \in \mathbb{R}^F$.

Propagated from the root post, responsive posts form a propagation tree $G_i = \langle V_i, E_i \rangle$, where each edge represents its direct responsiveness [15, 16]. The vertex set V_i is represented with the posts' features $\{x_{i0}, x_{i1}, \cdots, x_{i(n_i-1)}\}$ and the edge set E_i represents set of directed edges from source posts (root or responsive posts) to their direct responsive posts. A_i is an adjacency matrix for the directed graph G_i and $X_i = \{x_{i0}^T; x_{i1}^T; \cdots; x_{i(n_i-1)}^T\}^T \in \mathbb{R}^{n_i \times F}$ is the feature matrix for posts P_i. Upon representing the propagation tree as a static graph, to elaborate its evolving pattern, we define the diffusion graph with T step series of snapshots $S_i = \{S_i^{(1)}, S_i^{(2)}, \cdots, S_i^{(T)}\}$. The detail of the snapshot formulation will be discussed at section 4.1.

Each claim c_i is associated with its veracity label y_i, where y_i belong to one of four classes $\{T, F, U, N\}$ (True rumor, False rumor, Unverified rumor, or Non-rumor) or two classes $\{R, N\}$ (Rumor, Non-rumor) depending on the dataset [16, 18]. The definition of rumor labels that we borrowed is the messages in which the veracity labels are unknown at the stage of the propagation and later classified by human annotators as true, false, or unknown. (non-rumor messages are thoughts or simple admiration) [2, 3]. In this study, we define the task of rumor detection as a supervised graph classification problem, which the goal is to learn a mapping function f: $C \rightarrow Y$ to classify the veracity labels of c_i using S_i and X_i.

4 Dynamic GCN

In this section, we propose a dynamic graph representation learning model for rumor detection, named *Dynamic GCN* (DynGCN). The main components of the model are snapshot generation, graph convolutional networks, readout layer, and attention mechanisms. The components are respectively responsible for the following functionalities: rumor propagation representation, representation learning on a graph snapshot, node embedding aggregation for global graph representation, and sequential learning from the series of graph snapshots. Fig 2 is the overview of our dynamic rumor detection model with its layers and inputs' shapes.

4.1 Snapshot generation

To capture the evolving pattern of the rumor diffusion, we adopt the series of graph snapshots. We introduce two different ways of depicting the dynamic graphs as T step graph snapshots $S = \{S^{(1)}, S^{(2)}, \cdots, S^{(T)}\}$. One is with sequential snapshots, and the other is with temporal snapshots. In Fig 3, we illustrated the two different methods of snapshot generations. Here on the index i for the claim c_i will be omitted. $S^{(t)}$ is the graph snapshot at the time step t. Each graph snapshot in S will have separate adjacency matrices $A = \{A^{(1)}, A^{(2)}, \cdots, A^{(T)}\}$ with $S^{(t)} = \langle V^{(t)}, E^{(t)}\rangle$.

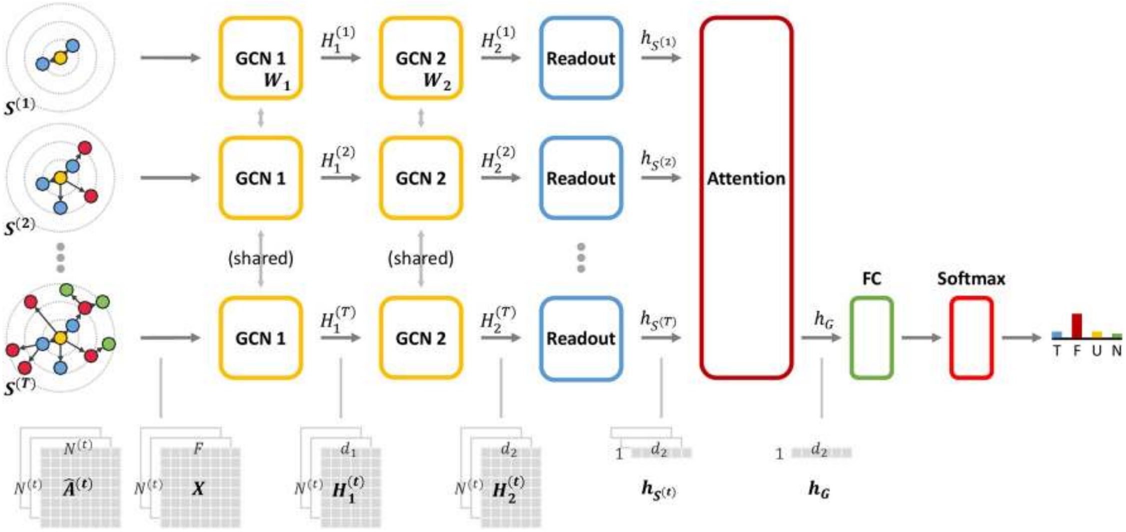

Fig 2. The overall architecture of Dynamic GCN rumor detection model with its layers and inputs' dimensions for each layer.

https://doi.org/10.1371/journal.pone.0256039.g002

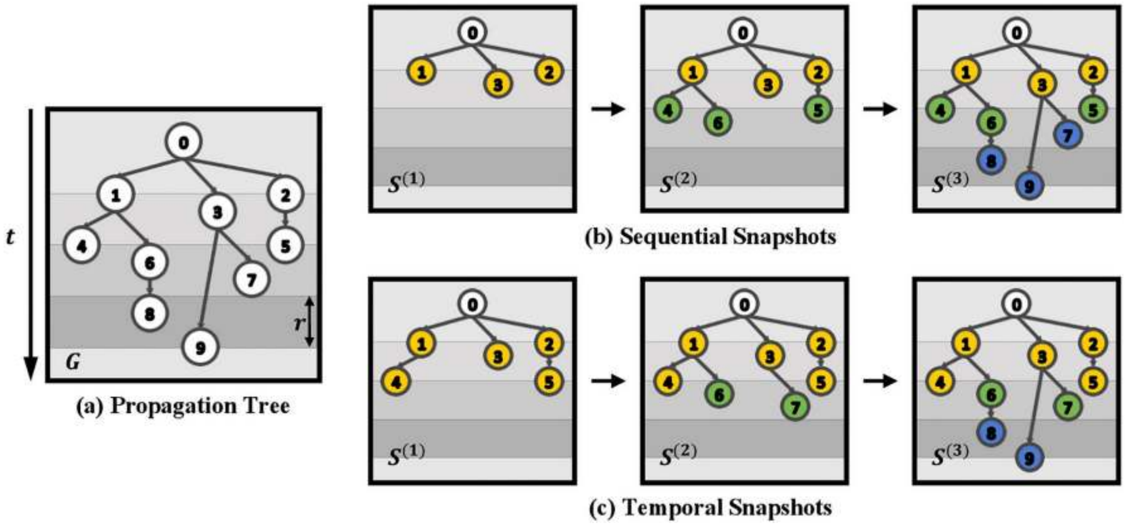

Fig 3. The disparity of snapshot generation results in sequential snapshots (b) and temporal snapshots (c) originated from the propagation tree (a). Sequential snapshots take account of chronological ordering and node counts, while temporal snapshots utilize timestamps.

https://doi.org/10.1371/journal.pone.0256039.g003

4.1.1 Sequential snapshots. Consider the ordering of the additional nodes and links of the propagation tree. Starting from $S^{(1)}$, the following graph snapshots will contain $\lceil (n-1)/T \rceil$ additional links (and nodes), where $n-1$ is the total number of responsive links. Eventually, each graph snapshot $S^{(t)}$ will contain $\lceil t \times (n-1)/T \rceil$ links. The edge set for the sequential snapshot is as:

$$E^{(t)} = \{e_1, e_2, \cdots, e_{\lceil t\times(n-1)/T \rceil}\} \quad (1)$$

4.1.2 Temporal snapshots. Consider temporal information of the propagation tree. Compared to the sequential snapshot which contains the equal counts of additional edges, temporal snapshots separate T step diffusion with the fixed time interval r. Time interval r is retrieved by dividing the time difference of the first and the last responsive posts with the time step T. The edge set for the temporal snapshot $S^{(t)}$ can be defined as:

$$\begin{aligned} E^{(t)} &= \{e | \tau_e - \tau_{e_1} \leq r \times (t), e \in E\}, \\ r &= \frac{\tau_{e_{(n-1)}} - \tau_{e_1}}{T} \end{aligned} \quad (2)$$

where τ_e is the timestamp of link e, and r is the time interval of the snapshots.

4.2 Graph convolutional networks

For the snapshot representation learning, we adopt graph convolutional architecture. Upon generating the graph snapshots $S = \{S^{(1)}, S^{(2)}, \cdots, S^{(T)}\}$ and their adjacency matrices $A = \{A^{(1)}, A^{(2)}, \cdots, A^{(T)}\}$, we conduct representation learning on the graph snapshots with the graph convolutional networks (GCNs) [23]. Introduced in [23], the approximated normalized graph Laplacian [47] is used for high-dimensional node representation learning. Together with an adjacency matrix $A^{(t)} \in \mathbb{R}^{N^{(t)} \times N^{(t)}}$, where $N^{(t)}$ is the number of nodes in the snapshot, and feature matrix $X \in \mathbb{R}^{N^{(t)} \times F}$, the learnable parameters $W_k \in \mathbb{R}^{d_{k-1} \times d_k}$ are trained, where k^{th} layer produce node embeddings $H_k \in \mathbb{R}^{N^{(t)} \times d_k}$. The GCN model that we adopted is as:

$$\begin{gathered} H_k = \sigma(\hat{A} H_{k-1} W_k), \\ \hat{A} = \hat{D}^{-1/2} \tilde{A} \hat{D}^{-1/2}, \tilde{A} = A + I_N, \tilde{D}_{ii} = \sum_j \tilde{A}_{jj} \end{gathered} \tag{3}$$

Trainable parameters W_* are shared between same level of GCNs with different snapshots steps. We use 2-layer GCNs with ReLU as activation function σ. We also adopt a skip-connection-like method [48] called feature enhancement [26] to enhance the information from a certain node, in this case, the root node. The root representations in a previous GCN layer bypass the layer as:

$$\tilde{H}_k = concat(H_k, (H_{k-1})_{root}) \tag{4}$$

And finally, inspired and introduced by [26, 49], instead of perceiving diffusion pattern as undirected graph, we adopt bi-directional GCNs which consider both direction of graph representation separately as:

$$H_k = concat(\vec{\tilde{H}}_k, \overleftarrow{\tilde{H}}_k) \tag{5}$$

The outputs $H_K^{(t)}$, produced by the last layer K of GCNs, are node embeddings of each graph snapshot $S^{(t)}$.

4.3 Readout layer

After the GCN layers embed node representation $H_K^{(t)} \in \mathbb{R}^{N^{(t)} \times d_K}$ of each graph snapshot $S^{(t)}$, the global graph pooling method is used to convert node representation to graph representation. The permutation invariant (symmetric) down-sampling method like max / mean / sum-pooling, or even sophisticated pooling method like [30, 31] can be used for the aggregation function in the readout layer. In this work, we empirically selected mean-pooling method for global graph pooling. The element-wise mean operation of node embeddings of the last layer K of GCN as:

$$h_{S^{(t)}} = MEAN(H_K^{(t)}) \tag{6}$$

for the global graph snapshot embedding at $t \in \{1, 2, \ldots, T\}$; $h_{S^{(t)}}$.

4.4 Attention mechanism

To apprehend the dynamic (temporal) information of graph snapshots, we use attention mechanisms. We adopt two well-known attention mechanisms: additive attention [35] and scaled dot-product attention [36]. From the graph snapshot embeddings $h_s = \{h_{s^{(1)}}, h_{s^{(2)}}, \cdots, h_{s^{(T)}}\}$, the goal is to learn the attention weights and use them to aggregate the weighted inputs.

Introduced in [20, 35], for the additive attention, we retrieve the context vector m_s by applying element-wise mean operation of embeddings of h_s. The context vector m_s is used as a query (Q) of the attention mechanism and h_s is used for the key (K) and value (V). For the additive attention, query and key are concatenated and fed to a feed-forward neural network to produce the attention scores z. Attention weights are calculated as:

$$\begin{aligned} Attention(Q, K, V) &= Softmax(MLP(Q; K))V \\ Softmax(z_i) &= \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, z = (z_1, z_2, \cdots, z_K) \in \mathbb{R}^K \end{aligned} \tag{7}$$

Scaled dot-product attention consider the dot-product similarity of the embeddings when calculating the attention scores. We adopt self-attention which the query (Q), key (K), value (V) is all $h_s = \{h_{s(1)}, h_{s(2)}, \cdots, h_{s(T)}\}$ as:

$$Attention(Q, K, V) = Softmax(\frac{(QK^T)}{\sqrt{d_k}})V \tag{8}$$

The softmax result of normalized similarity measures of snapshots is applied to calculate the attention weights for the h_s where d_k is the dimension of $h_{s(t)}$.

The outputs of the two different attention layers are both the weighted sequences of the snapshot embeddings. The element-wise average of the T snapshots where attention weights are applied is used to retrieve the global graph embedding h_G as:

$$h_G = \frac{1}{T}\sum_{i=1}^{T} h_{S(i)} \tag{9}$$

4.5 Training & prediction

For the graph classification task, the graph embedding h_G is fed to the multi-layer perception as:

$$\hat{y} = Softmax(MLP(h_G)) \tag{10}$$

The $\hat{y} \in \mathbb{R}^{|class|}$ is the probabilities of veracity labels where *class* = {*T*, *F*, *U*, *N*} or *class* = {*R*, *N*}.

Our supervised graph classification model is trained with the cross-entropy loss of the predictions and ground truth labels. The loss function of our model is defend as:

$$\mathcal{L} = \sum_{i=1}^{|class|} - y_i^T log\hat{y}_i \tag{11}$$

where y_i is the ground truth label for the claim c_i.

5 Experiments

In this section, we perform experiments on three real-world datasets and compare the performance of the proposed model, *Dynamic GCN*, with other rumor detection baselines. Furthermore, we conduct ablation studies and analyze the results on different snapshot counts and variants of the sequential learning methods.

5.1 Datasets

We evaluate the proposed model with three publicly available rumor detection datasets: *Twitter15* [13], *Twitter16* [16], and *Weibo* [18]. These datasets contain rumor propagation trees, where nodes are posts and links are responsive relations such as replies or retweets, with one of the four ground truth veracity labels (True rumor, False rumor, Unverified rumor, Non-

Table 1. The statistics of the rumor detection datasets.

	Twitter15	*Twitter16*	*Weibo*
# of root posts	1490	818	4664
# of users	276,663	173,487	2,746,818
# of posts	331,612	204,820	3,805,656
# of true rumors	372	205	0
# of false rumors	370	205	2313
# of unverified rumors	374	203	0
# of non-rumors	374	205	2351
avg. max. time (hours)	1,337	848	2,461
avg. # of posts / event	222.6	250.4	816.0
max. # of posts / event	1,768	2,765	59,318

https://doi.org/10.1371/journal.pone.0256039.t001

rumor) for *Twitter15* and *Twitter16* and two classes (Rumor, Non-rumor) for *Weibo* dataset. The detailed statistics of the datasets are provided in Table 1. We used the bag-of-words (BoW) features by selecting the top 5,000 vocabularies for the corpus by *TF-IDF*; thus, each post initially contains 5,000 features.

5.2 Baselines

We compare our Dynamic GCN model with the following rumor detection baseline models:

- DTC [7]: A decision tree-based classifier with handcrafted features to identify the credibility of microblog posts related to trending topics.
- RFC [11]: A random forest based-ranking method that elaborates the inquiry phrases of posts.
- SVM-TS [12]: An SVM model that captures the temporal characteristics of social context features of posts.
- SVM-TK [16]: An SVM model with a tree kernel that captures higher-order patterns of propagation structures of rumors.
- GRU [18]: An RNN-based model that learns contextual information from continuous representations of relevant posts over time.
- RvNN [21]: A recursive neural network-based model which captures the structural patterns of a top-down and bottom-up rumor propagation trees.
- Bi-GCN [26]: A graph convolutional network-based model, which captures propagation patterns with message passing architecture.
- DynGCN (Proposed): A graph convolutional network-based model with attention mechanisms to capture temporal dynamics of graph snapshots.

We haven't included the Propagation Path Classification (PPC) model [20] and Global-Local Attention Network (GLAN) model [27] as our baselines since both methods include crawled user profiles as additional input features (such as whether the user is suspended or verified), which could be too biased at the time of current work. A few years had passed since the initial collection of the datasets, the results could be distorted and might be too much depended on when the user profiles were crawled. Instead, we compare our model with the state-of-the-art model [26], which considers the posts relations without additional crawled user profiles.

5.3 Experimental setup

We conducted 10 runs of 5-fold cross-validation and reported the average accuracies and F1 scores by each label. For the fair comparison, for the models with early stopping method [50] such as Bi-GCN and ours, we randomly splitted 4-fold of training set into 80% training set and 20% validation set, which eventually making 16:4:5 splits for train, validation, and test sets. The validation set was used for early stopping with patience of 10 epochs.

The model has 256 hidden dimensions for a single graph snapshot, including root feature enhancement and bi-directional representation. We set 2-layer GCNs and used rectified linear units for the non-linearity. We adopt the dropout [51] rate of 0.5 for GCN layers and DropEdge [52], graph data augmentation method, rate with 0.2. We train our model with Adam optimization algorithm [53] with the initial learning rate 5E-4 and a maximum of 200 epochs if not early stopped.

Our model is implemented in PyTorch [54] with PyTorch Geometric [55] for the message passing framework. For the baseline models, we conduct experiments with the authors' provided codes with the same hyperparameters that were reported, respectively. For the fair comparison, we directly cited (*) some of the metrics already reported in original papers [16, 18] with equivalent experimental settings due to some handcrafted features that are unavailable at the time of the reproduction.

5.4 Performance evaluations

Tables 2 and 3 summarize the overall performances of the rumor detection task of the proposed model with other baselines. The reported performances are accuracies and F1 scores of DynGCN with both additive attention (ADD) and dot-product attention (DOT) with the sequential (S) snapshots or temporal (T) snapshots size of 3. The accuracy results of DynGCN with additive attention shows (S) 0.818, (T) 0.827 in *Twitter15*, and (S) 0.828, (T) 0.836 in *Twitter16*. The accuracy results of DynGCN with dot-product attention shows (S) 0.819, (T) 0.821 in *Twitter15*, and (S) 0.829, (T) 0.824 in *Twitter16*. Although the two attention methods don't show significant performance differences, the model with additive attention and

Table 2. Overall performance of rumor detection task on *Twitter15* and *Twitter16*.

Model		*Twitter 15*					*Twitter 16*				
		Accuracy	TR	FR	UN	NR	Accuracy	TR	FR	UN	NR
			F1	F1	F1	F1		F1	F1	F1	F1
DTC [7]*		0.454	0.317	0.355	0.415	0.733	0.465	0.419	0.393	0.403	0.643
RFC [11]*		0.565	0.401	0.422	0.543	0.810	0.585	0.547	0.415	0.563	0.752
SVM-TS [12]*		0.544	0.404	0.472	0.483	0.796	0.574	0.571	0.420	0.526	0.755
SVM-TK [16]*		0.667	0.772	0.669	0.645	0.619	0.662	0.783	0.623	0.655	0.643
GRU [18]		0.641	0.688	0.634	0.571	0.684	0.633	0.577	0.715	0.527	0.617
RvNN [21]		0.723	0.821	0.758	0.654	0.682	0.737	0.835	0.743	0.708	0.662
Bi-GCN [26]		0.814	0.793	0.811	0.872	0.768	0.804	0.718	0.787	0.799	0.895
DynGCN w/additive attention	Sequential	0.818	0.860	0.793	0.761	0.779	0.828	0.765	0.736	0.826	0.637
	Temporal	**0.827**	0.837	0.769	0.820	0.746	**0.836**	0.880	0.804	0.853	0.741
DynGCN w/dot-product attention	Sequential	0.819	0.871	0.816	0.771	0.800	0.829	0.873	0.756	0.775	0.708
	Temporal	0.821	0.859	0.806	0.765	0.782	0.824	0.876	0.848	0.767	0.742

https://doi.org/10.1371/journal.pone.0256039.t002

Table 3. Overall performance of rumor detection task on *Weibo*.

Model	*Weibo*		
	Accuracy	Rumor	Non-rumor
		F1	F1
DTC [7] *	0.831	0.831	0.819
RFC [11] *	0.849	0.864	0.830
SVM-TS [12] *	0.857	0.861	0.857
GRU [18]	0.910	0.914	0.906
RvNN [21]	0.908	0.905	0.911
Bi-GCN [26]	0.928	0.928	0.928
DynGCN w/ ADD (Seq)	**0.936**	0.936	0.936
DynGCN w/ DOT (Seq)	0.932	0.932	0.932

https://doi.org/10.1371/journal.pone.0256039.t003

temporal snapshots outperformed others. Furthermore, both variants of attention models outperform other state-of-the-art models, such as Bi-GCN and RvNN, in both *Twitter15* and *Twitter16* with the aid of evolving patterns. The results indicate that taking account of the temporal information and evolving pattern of rumor propagation is beneficial. A similar result is shown for the binary classification task of the *Weibo* dataset. Although the timestamps aren't retrieval for *Weibo*, the experiment with the sequential snapshots with the snapshot size of 3 shows the improved performance.

It is demonstrated that the traditional machine learning-based methods with handcrafted features, (DTC, RFC, SVM-TS, SVM-TK), show lower performances compare to other deep learning-based methods (GRU, RvNN, BiGCN, DynGCN). However, SVM-TS and SVM-TK show superior results within the traditional handcrafted methods since these models are able to utilize temporal features. It is constructive to consider temporal information of rumor for rumor detection.

Finally, among the propagation-based baselines, a graph-based models, DynGCN and Bi-GCN, outperforms other baselines such as RvNN or GRU since graph convolutional network can better capture the structural representation of rumor diffusion.

5.5 Ablation study

In order to see the performance of our model in different settings, we report the following ablation studies. The performance results with different snapshot counts for sequential and temporal snapshots, with different learning algorithms for combining snapshot sequences, and attention weights of additive attention and dot-product attention.

5.5.1 Different snapshot counts. Fig 4 is the result of DynGCN with the snapshot counts of 1, 2, 3, 4, and 5 with dot-product attention. Although there aren't significant correlations in the aspect of accuracy with the counts, adopting multiple snapshots shows better performance compare to a single static snapshot in both sequential and temporal snapshots. However, we observed that simply applying larger snapshot counts won't produce a performance improvement and believe this can be a hyperparameter for the dataset.

5.5.2 Different learning methods for the sequence. The attention layer of our model can be replaced with other Seq2Seq [34] models since the inputs to the attention layer are a sequence of snapshot representations. Fig 5 is the result of different sequence learning methods (Bi-LSTM, Bi-GRU, additive attention, and dot-product attention (self-attention)) with the snapshots count of 3. Attention mechanisms that are used for a weighted sum of sequential

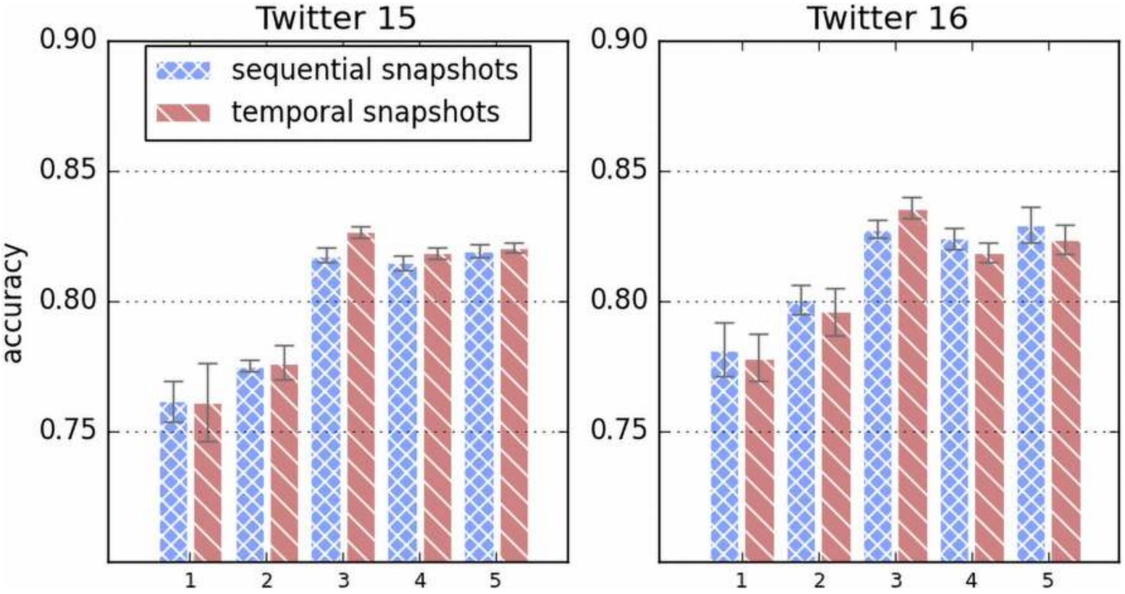

Fig 4. Difference in accuracy for different snapshot counts.

https://doi.org/10.1371/journal.pone.0256039.g004

and temporal snapshot representations outperform the other RNN-based models. Bidirectional LSTM/GRU show low performance in aggregating the temporal representation of graph snapshots. We suspect the results of the relatively low performance of Bi-LSTM and Bi-GRU are due to the short sequence of global graph snapshots. [34]

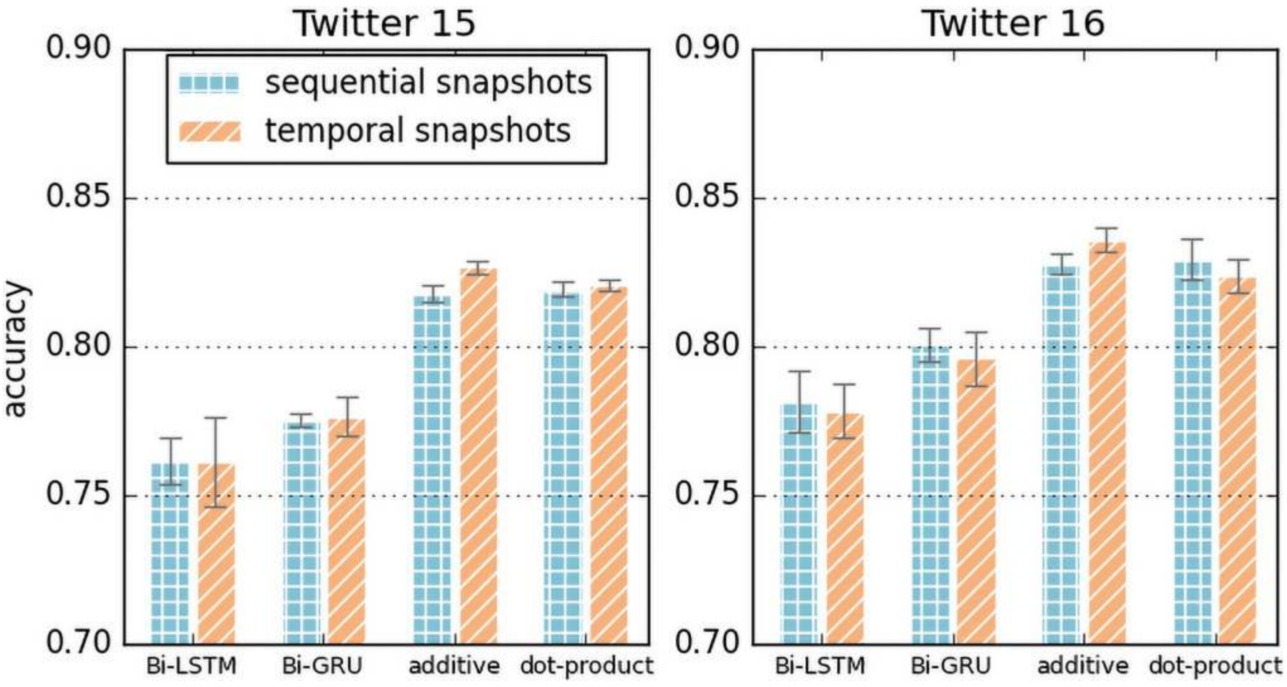

Fig 5. Differences in accuracy of various sequence learning methods.

https://doi.org/10.1371/journal.pone.0256039.g005

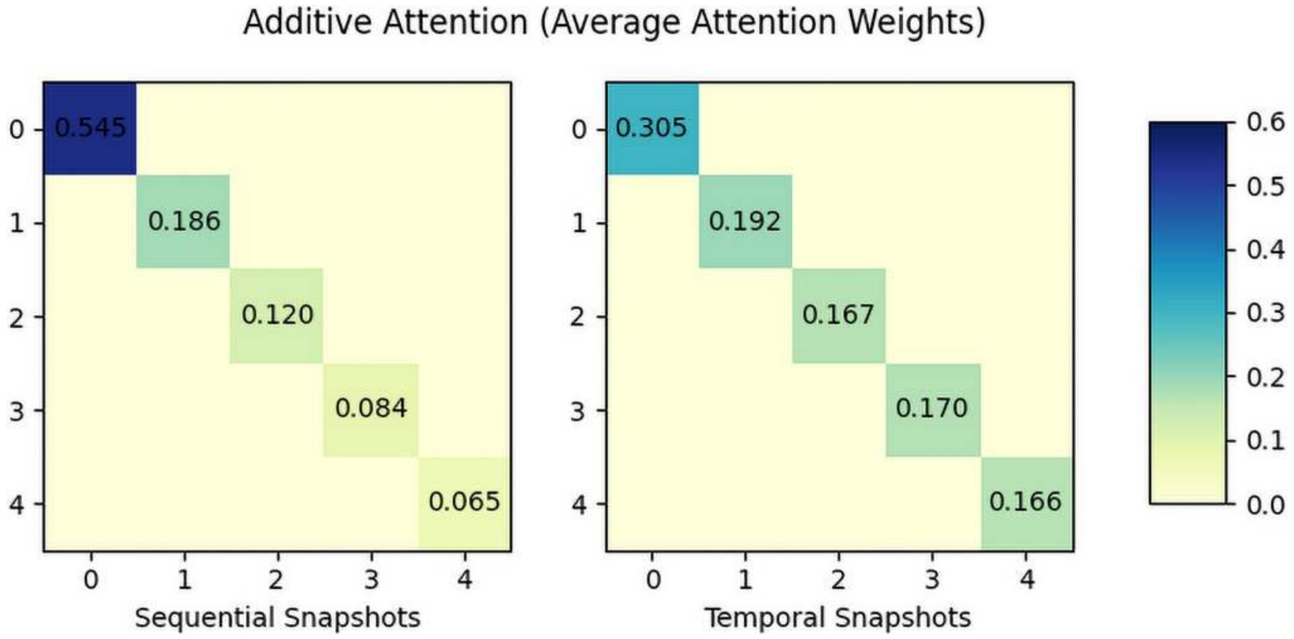

Fig 6. Average attention weights of additive attention on *Twitter16*.

https://doi.org/10.1371/journal.pone.0256039.g006

5.5.3 Effects of the attention mechanisms. Finally, Figs 6 and 7 are the visualization of the average attention weights of additive attention and dot-product attention. Notice that additive attention takes a context vector as a query for the attention; thus, the attention matrix is diagonal, while dot-product attention is a self-attention. The result shows the additive attention considers the early stage of the rumor propagation while the dot-product attention

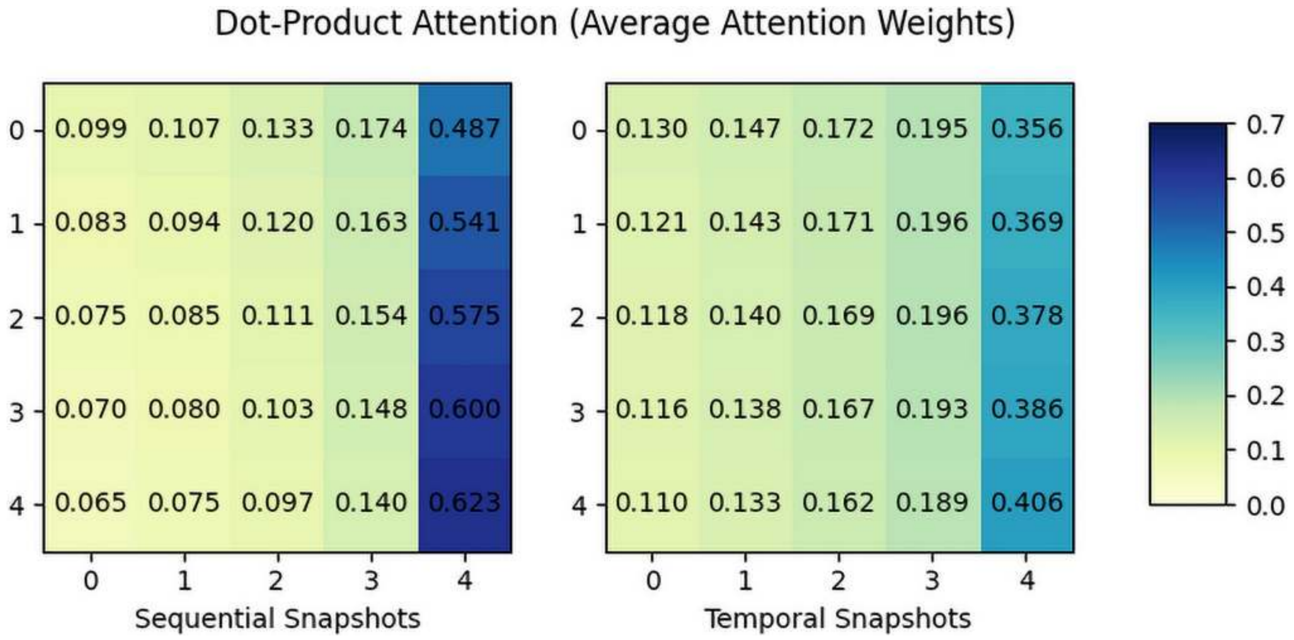

Fig 7. Average attention weights of dot-product attention on *Twitter16*.

https://doi.org/10.1371/journal.pone.0256039.g007

significantly considers the snapshots in the end-stage. This can be interpreted as that the additive attention reply on the context query to understand the global or overall propagation while dot-product attention relies on the input sequence to jointly understand the overall pattern. Although the weight itself depends on the dataset, we could see that each attention mechanism represents the propagation structure in its own way.

6 Conclusion

In this research, we propose Dynamic GCN, an end-to-end GCN-based model with attention mechanisms, for rumor detection. The model is able to capture the dynamics of rumor propagations using sequential snapshots and temporal snapshots. We empirically evaluate our model with three real-world datasets and compare the performance of the rumor detection (veracity classification) task with other rumor detection baselines. The results show that our model outperforms other state-of-the-art methods. The ablation studies report performance differences with snapshots counts, learning sequence variants, and the weights for the different attention mechanisms. We believe there is still room for improvement in the context of GCNs variants, global graph pooling, and additional features from different contexts.

Author Contributions

Conceptualization: Jiho Choi, Taewook Ko, Chong-kwon Kim.

Data curation: Jiho Choi, Younhyuk Choi.

Formal analysis: Jiho Choi, Younhyuk Choi, Hyungho Byun.

Funding acquisition: Chong-kwon Kim.

Methodology: Jiho Choi, Taewook Ko.

Resources: Jiho Choi.

Software: Jiho Choi, Hyungho Byun.

Supervision: Chong-kwon Kim.

Validation: Younhyuk Choi, Hyungho Byun.

Visualization: Jiho Choi, Younhyuk Choi.

Writing – original draft: Jiho Choi, Taewook Ko.

Writing – review & editing: Jiho Choi, Younhyuk Choi, Hyungho Byun, Chong-kwon Kim.

References

1. Shu K, Sliva A, Wang S, Tang J, Liu H. Fake news detection on social media: A data mining perspective. ACM SIGKDD Explorations Newsletter. 2017; 19(1):22–36. https://doi.org/10.1145/3137597.3137600
2. Li Q, Zhang Q, Si L, Liu Y. Rumor Detection on Social Media: Datasets, Methods and Opportunities. In: Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda. Hong Kong, China: Association for Computational Linguistics; 2019. p. 66–75.
3. Qazvinian V, Rosengren E, Radev DR, Mei Q. Rumor has it: Identifying misinformation in microblogs. In: Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics; 2011. p. 1589–1599.
4. Zubiaga A, Aker A, Bontcheva K, Liakata M, Procter R. Detection and resolution of rumours in social media: A survey. ACM Computing Surveys (CSUR). 2018; 51(2):1–36. https://doi.org/10.1145/3161603

5. Lazer DM, Baum MA, Benkler Y, Berinsky AJ, Greenhill KM, Menczer F, et al. The science of fake news. Science. 2018; 359(6380):1094–1096. https://doi.org/10.1126/science.aao2998 PMID: 29590025

6. Vosoughi S, Roy D, Aral S. The spread of true and false news online. Science. 2018; 359(6380):1146–1151. https://doi.org/10.1126/science.aap9559 PMID: 29590045

7. Castillo C, Mendoza M, Poblete B. Information credibility on twitter. In: Proceedings of the 20th international conference on World wide web; 2011. p. 675–684.

8. Yang F, Liu Y, Yu X, Yang M. Automatic detection of rumor on Sina Weibo. In: Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics; 2012. p. 1–7.

9. Kwon S, Cha M, Jung K, Chen W, Wang Y. Prominent features of rumor propagation in online social media. In: 2013 IEEE 13th International Conference on Data Mining. IEEE; 2013. p. 1103–1108.

10. Gupta A, Lamba H, Kumaraguru P, Joshi A. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In: Proceedings of the 22nd international conference on World Wide Web; 2013. p. 729–736.

11. Zhao Z, Resnick P, Mei Q. Enquiring minds: Early detection of rumors in social media from enquiry posts. In: Proceedings of the 24th international conference on world wide web; 2015. p. 1395–1405.

12. Ma J, Gao W, Wei Z, Lu Y, Wong KF. Detect rumors using time series of social context information on microblogging websites. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management; 2015. p. 1751–1754.

13. Liu X, Nourbakhsh A, Li Q, Fang R, Shah S. Real-time rumor debunking on twitter. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management; 2015. p. 1867–1870.

14. Kwon S, Cha M, Jung K. Rumor detection over varying time windows. PloS one. 2017; 12(1). https://doi.org/10.1371/journal.pone.0168344 PMID: 28081135

15. Wu K, Yang S, Zhu KQ. False rumors detection on sina weibo by propagation structures. In: 2015 IEEE 31st international conference on data engineering. IEEE; 2015. p. 651–662.

16. Ma J, Gao W, Wong KF. Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics; 2017. p. 708–717. Available from: https://www.aclweb.org/anthology/P17-1066.

17. Wu L, Liu H. Tracing fake-news footprints: Characterizing social media messages by how they propagate. In: Proceedings of the eleventh ACM international conference on Web Search and Data Mining; 2018. p. 637–645.

18. Ma J, Gao W, Mitra P, Kwon S, Jansen BJ, Wong KF, et al. Detecting Rumors from Microblogs with Recurrent Neural Networks. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. IJCAI'16. AAAI Press; 2016. p. 3818–3824.

19. Yu F, Liu Q, Wu S, Wang L, Tan T. A Convolutional Approach for Misinformation Identification. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. IJCAI'17. AAAI Press; 2017. p. 3901–3907.

20. Liu Y, Wu YFB. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In: Thirty-Second AAAI Conference on Artificial Intelligence; 2018.

21. Ma J, Gao W, Wong KF. Rumor Detection on Twitter with Tree-structured Recursive Neural Networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics; 2018. p. 1980–1989. Available from: https://www.aclweb.org/anthology/P18-1184.

22. Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems; 2016. p. 3844–3852.

23. Kipf TN, Welling M. Semi-Supervised Classification with Graph Convolutional Networks. In: Proceedings of the 5th International Conference on Learning Representations. ICLR'17; 2017.

24. Hamilton W, Ying Z, Leskovec J. Inductive Representation Learning on Large Graphs. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors. Advances in Neural Information Processing Systems 30. Curran Associates, Inc.; 2017. p. 1024–1034. Available from: http://papers.nips.cc/paper/6703-inductive-representation-learning-on-large-graphs.pdf.

25. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph Attention Networks. International Conference on Learning Representations. 2018;.

26. Bian T, Xiao X, Xu T, Zhao P, Huang W, Rong Y, et al. Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34; 2020. p. 549–556.

27. Yuan C, Ma Q, Zhou W, Han J, Hu S. Jointly Embedding the Local and Global Relations of Heterogeneous Graph for Rumor Detection. In: 2019 IEEE International Conference on Data Mining (ICDM); 2019. p. 796–805.

28. Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining; 2014. p. 701–710.

29. Grover A, Leskovec J. node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining; 2016. p. 855–864.

30. Ying Z, You J, Morris C, Ren X, Hamilton W, Leskovec J. Hierarchical graph representation learning with differentiable pooling. In: Advances in neural information processing systems; 2018. p. 4800–4810.

31. Lee K, Lee I, Kang J. Self-attention graph pooling. In: 36th International Conference on Machine Learning, ICML 2019. 36th International Conference on Machine Learning, ICML 2019. International Machine Learning Society (IMLS); 2019. p. 6661–6670.

32. Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997; 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 PMID: 9377276

33. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. In: NIPS 2014 Workshop on Deep Learning, December 2014; 2014.

34. Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. In: Advances in neural information processing systems; 2014. p. 3104–3112.

35. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015; Conference date: 07-05-2015 Through 09-05-2015; 2015.

36. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems; 2017. p. 5998–6008.

37. Devlin J, Chang MW, Lee K, Toutanova K. 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, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics; 2019. p. 4171–4186. Available from: https://www.aclweb.org/anthology/N19-1423.

38. Leskovec J, Kleinberg J, Faloutsos C. Graph evolution: Densification and shrinking diameters. ACM transactions on Knowledge Discovery from Data (TKDD). 2007; 1(1):2–es. https://doi.org/10.1145/1217299.1217301

39. Berger-Wolf TY, Saia J. A Framework for Analysis of Dynamic Social Networks. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD'06. New York, NY, USA: Association for Computing Machinery; 2006. p. 523–528. Available from: https://doi.org/10.1145/1150402.1150462.

40. Tang L, Liu H, Zhang J, Nazeri Z. Community Evolution in Dynamic Multi-Mode Networks. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD'08. New York, NY, USA: Association for Computing Machinery; 2008. p. 677–685. Available from: https://doi.org/10.1145/1401890.1401972.

41. Zhou L, Yang Y, Ren X, Wu F, Zhuang Y. Dynamic network embedding by modeling triadic closure process. In: Thirty-Second AAAI Conference on Artificial Intelligence; 2018.

42. You J, Ying R, Ren X, Hamilton W, Leskovec J. GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models. In: Proceedings of the 35th International Conference on Machine Learning. vol. 80 of Proceedings of Machine Learning Research. PMLR; 2018. p. 5708–5717.

43. Goyal P, Kamra N, He X, Liu Y. Dyngem: Deep embedding method for dynamic graphs. arXiv preprint arXiv:180511273. 2018;.

44. Pareja A, Domeniconi G, Chen J, Ma T, Suzumura T, Kanezashi H, et al. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. In: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence; 2020.

45. Manessi F, Rozza A, Manzo M. Dynamic graph convolutional networks. Pattern Recognition. 2020; 97:107000. https://doi.org/10.1016/j.patcog.2019.107000

46. Sankar A, Wu Y, Gou L, Zhang W, Yang H. DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks. In: Proceedings of the 13th International Conference on Web Search and Data Mining; 2020. p. 519–527.

47. Chung FR, Graham FC. Spectral graph theory. 92. American Mathematical Soc.; 1997.

48. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–778.

49. Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE transactions on Signal Processing. 1997; 45(11):2673–2681. https://doi.org/10.1109/78.650093

50. Yao Y, Rosasco L, Caponnetto A. On early stopping in gradient descent learning. Constructive Approximation. 2007; 26(2):289–315. https://doi.org/10.1007/s00365-006-0663-2

51. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research. 2014; 15(1):1929–1958.

52. Rong Y, Huang W, Xu T, Huang J. Dropedge: Towards deep graph convolutional networks on node classification. In: International Conference on Learning Representations. https://openreview.net/forum; 2020.

53. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. In: Bengio Y, LeCun Y, editors. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings; 2015.

54. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems; 2019. p. 8024–8035.

55. Fey M, Lenssen JE. Fast Graph Representation Learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds; 2019.