DOI: 10.15849/IJASCA.231130.09

Int. J. Advance Soft Compu. Appl, Vol. 15, No. 3, November 2023
Print ISSN: 2710-1274, Online ISSN: 2074-8523

A Dynamic Deep-Learning Approach for Predicting Information Diffusion

Majed AlGarni [*1] and Mohamed Maher Ben Ismail[2]

King Saud University, College of Computer and Information Sciences,
Department of Computer Science.
mbenismail@ksu.edu.sa
* Correspondence: 435108679@student.ksu.edu.sa;

Abstract

Abstract: Social media exhibits an intrinsic capability to broadcast new information. One of its essential characteristics is its ability to transmit and exchange knowledge. With the growth in popularity of social networking sites and their use to spread information, it is more important than ever to spot rumors before they spread. Indeed, the large amount of information on social media makes recognizing the origin of a rumor or conspiracy theory even more challenging. Several deep-learning algorithms, such as recursive neural networks (RvNNs), have been employed to identify rumors based on how they spread. Moreover, machine learning algorithms employed in rumor detection consider patterns of deep information propagation instead of the broad dispersion structures in the data. Therefore, this research introduces a hybrid deep-learning model for information diffusion analysis and rumor detection called the Bidirectional Graph Convolutional Network integrated with Long Short-Term Memory (BiGCN-LSTM), which utilizes two different word embedding approaches: Word2Vec and BERT. Two standard datasets, a Sina Weibo microblog and Twitter16, are used to validate and assess the proposed approach where the main microblog and tweet and their responses are represented in a graph-tree-based structure for obtaining and retrieving important information. Especially the original main microblog and tweet information are integrated into each top-down and bottom-up GCN layer, increasing the effect of the messages delivered by the rumor spreaders. The experimental results showed that the proposed model achieved 92% and 88% accuracy using Word2Vec, and 90% and 95% accuracy using BERT for the Sina Weibo and Twitter16 datasets, respectively.

Keywords: *BiGCN-LSTM; Sina Weibo; Twitter16; deep learning; rumor detection*

1 Introduction

Social media services such as Twitter, Facebook, Sina Weibo, and Instagram are witnessing continuous growth in popularity [1–3]. They yield the creation, sharing, and spread of user-generated internet material, known as information. These platforms report tens of millions of data bits generated daily [4]. Typically, social network platform users can generate and share a wide range of content and discuss topics of mutual interest with one another, promoting the quick spread of information that speeds up the phenomenon of information cascades [5]. Cascade modeling has been the subject of extensive study in various fields, with

Received 6 July 2023; Accepted 7 August 2023

important implications for a wide range of applications, including viral marketing categorization [6], impact maximization [7,8], media ads [9], and false news identification [10,11]. Therefore, the cascade prediction issue is critical since it allows for regulating the flow of information in various circumstances [12].

Recurrent neural networks (RNNs) built of long short-term memory (LSTM) units have been typically used to address tweets' categorization problems [13]. They have proven capable of deriving the long-term trend of local high-dimensional characteristics where LSTM units are exploited to significantly alleviate the issue of erroneous prediction caused by the mixing of original data. The LSTM technique has been used in other study domains, for example, with wind power predictions using combined retrieved temporal information and physical features. For example, using real wind turbine operating data, researchers have demonstrated that the proposed LSTM model can combine multi-scale extended characteristics while delivering superior performance for short-term wind forecasting [14,15].

Existing techniques for predicting information diffusion suffer from shortcomings that fall into four primary groups: (a) scattering model-based approaches that rely significantly on presumed underlying diffusion models unsuitable for prediction analysis [16,17], (b) attribute-based approaches relying heavily on the extracted characteristics while requiring considerable domain knowledge that is difficult to transfer to new domains when generalized[18-20], (c) propagative approaches that consider each message individually, enabling knowledge about the popularity dynamics of information with less desired predictive ability gained by analyzing the intensity function of the arrival process [21,22], and (d) dynamic deep-learning approaches such as RNN-based systems that automatically learn temporal features yet fall short because they lack the fundamental structure knowledge of cascades required for cascade prediction [23].

Critical factors were considered when designing the proposed method to address the above shortcomings. Particularly, the lifecycle of true and fake news on social media [23], which plays a critical role in disseminating information [24], was used to support the proposed learning task. Indeed, the content producer starts the news journey on the social media platform when the news is generated, but once individuals are exposed to this news, they assume the role of information consumers.

According to the confirmation bias hypothesis of information processing, individuals are more likely to prefer, interpret, and share information that confirms or reinforces their existing ideas and ideologies [25]. Therefore, consumers who think a news article reinforces their prior worldview are more inclined to propagate it across their social networks, serving as content distributors. Since fake news is produced to trick readers into believing and disseminating false information, it is logical to infer that it is more readily distributed among its followers than actual news, which is impartial in its viewpoints and ideologies. For example, prior studies have revealed that false information travels substantially faster, further, deeper, and more broadly than correct information substantiated by evidence [26,27].

The model incorporates spreading and diffusing structural information from all source microblog content [28], reposts, and users involved in spreading online information on the Weibo platform. Although not friends or followers on Weibo, users 1 and 2 in this scenario both republished the same Weibo content (weibo1) simultaneously. Notably, although the three Weibo postings were completely unconnected in content, weibo2 and weibo3 shared neighbors, utilizing the same hashtags.

To address these shortcomings, we propose a novel approach to predicting information cascades on social media platforms, specifically on the Weibo and Twitter platforms. The proposed method incorporates spreading and diffusing structural information from all source microblog and Tweet contents, reposts, and users involved in spreading online information on the Weibo platform. The proposed global heterogeneous network includes local and global relationships between Weibo sources, reposts, and users in the dataset.. It comprises the following components: (a) word embeddings from other users' microblogs to recognize rumors, (b) a graph-aware network to learn about the concept of repost propagation, and (c) a graph to depict probable user interactions [29]. Figure 1 shows a simple prediction model designed as a tree.

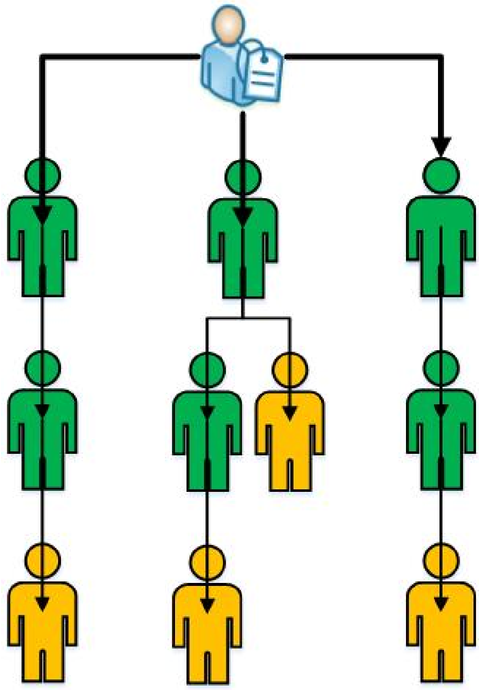


Figure 1. Prediction model tree.

The blue node in Figure 1 represents the publisher of Weibo microblog content. The green node shows the users who forward the Weibo, while the yellow node represents users who do not want their Weibo to be sent. When we make the forecast, we use our best judgment [30]. The main contributions are then as follows:

- A BiGCN with an LSTM approach is associated to identify rumors.
- The proposed prediction model handles three users, specifically considering two social network users who promote Weibo information while a third does not.
- The latter characteristics of posts are shared with all users, given the designed graph convolutional layer of the model to exploit the core feature's information to improve the overall rumor detection performance.
- The proposed model is evaluated using two standard datasets and performance metrics. Namely, Twitter and Weibo application datasets are used to assess rumor detection performance in the context of information diffusion.

This remain sections of the manuscript are organized as follows. Section 2 covers related works, while Section 3 formalizes the problem and provides the suggested backdrop. Section 4 outlines the conducted experiments and the obtained results. Finally, the discussion and findings of this research, along with the main conclusions, are depicted in Section 5.

2. Related Works

Kipf et al. proposed a graph convolutional network (GCN) [31] as a semi-supervised learning approach using all graph nodes, including unlabeled ones, while employing a message-passing strategy. However, the primary flaw of the GCN was that it was not temporal and did not simulate changes in nodes and edges over time as they occurred. Therefore, to capture the temporal dynamics of the networks, Narayan et al. adopted LSTM networks in conjunction with long short-term memory networks. The convolutional neural network (CNN) architecture can be integrated with an RNN technique to enhance predictions, such as node classification. In particular, relevant features are extracted using a CNN model. Then, an RNN model is deployed and fed with the extracted features for sequence learning. As a result, there is a constraint that all nodes at all timestamps are accessible throughout training [32].

Microblogging systems enable users to classify their postings using hashtags they may design themselves. Over the last several years, the problem of selecting hashtags for use on microblogs has received increasing attention. However, most currently available approaches rely on human-crafted features. In another study, LSTM was used to learn the representation of a microblog post, motivated by its successful usage for various natural language processing tasks. Therefore, the observation that hashtags highlight the key subjects of microblog posts has led us to develop an attention-based LSTM model that includes topic modeling into the LSTM architecture through an attention mechanism [33].

The EvolveGCN model was presented in [34] to address the primary problem that Pareja et al. (2020) identified with the preceding techniques compared to other approaches: they underperformed when faced with additional nodes in the future. Although these approaches employed RNN to forecast the future in terms of time, they still needed the knowledge of the nodes throughout the whole-time span. Using GNNs exclusively as feature extractors with RNNs only as predictors and while stacking these models can prevent GNNs from learning to recognize temporal changes effectively. Therefore, GNN weights must be regularized using RNNs rather than being utilized as feature extractors, as was previously the case. Therefore, EvolveGCN executes this update of GNN weights at each time step, which the user controls, enabling a GCN to learn from changes in the network structure over time due to these modifications.

However, in the natural language processing domain, deep-learning models with word embedding aspire to represent nodes in a network with vectors for various downstream analyses. Using mathematical similarities between vectors, one may anticipate relationships between the nodes they contain and even propose individuals and items on social networks. For instance, Node2vec and Deepwalk are two examples of network embedding models commonly used in the literature [35,36]. Methods based on autoencoders have shown to be quite effective for discovering latent representations of networks, aiding in understanding the underlying structure of a network, which may be utilized for subsequent studies and investigations. Some autoencoder techniques, such as CAN and DANE, have recently been used on social networking sites, with methods based on graph convolution, which can be used for networks similar to how convolution filters are applied to pictures. Thus, convolution can be applied to networks by integrating information from neighbor nodes.

Typically, two types of convolution filters are applied to the graph: spatial filters applied to the adjacency matrix and spectral filters applied to the graph's spectrum [37]. The traditional method of extracting features for downstream analysis, such as social recommendation, is based on kernel functions [38], graph statistics [39], and feature engineering. However, deep-learning algorithms have surpassed traditional modeling techniques as deep-learning algorithms have improved over time.

According to a recent study by Tan et al. [39], traditional techniques are "extremely time-consuming and expensive, rendering them useless for many real-world applications" in contrast to deep learning's capacity to automatically discover essential characteristics in the network. The research provided ascribed impact maximization based on crowd emotion, seeking to apply user emotion and group attributes to multi-dimensional characteristics of information propagation. Therefore, the authors of [40] presented a possible impact user-finding approach based on the emotion aggregation mechanism to identify seed candidate sets via a two-factor information propagation model considering network complexity. Their suggested technique worked well on real-world datasets, so their findings surpassed heuristic approaches, essentially identical to greedy methods.

Researchers, businesses, and governments have worked diligently to identify disinformation. Traditional solutions have either evaluated intricate hand-crafted characteristics or have significantly depended on built-in believability networks to extract valuable signs. Nevertheless, these techniques have required subject expertise and significant feature engineering. Recently developed deep-learning approaches have enabled the discovery of diffusion patterns in text and visual information. However, despite their improvement, these systems have suffered from over-reliance on content features while failing to consider the effect of each individual participating in the rumor-spreading process. Indeed, the various user-aspect information plays diverse roles in rumor transmission [49,50].

In [29], the authors proposed bidirectional GCNs to investigate top-down and bottom-up rumor transmission where an opposing directed graph of rumor dissemination captured the structures of rumor dispersion. Moreover, each layer of GCN included information from source posts to amplify the rumors' foundations. Encouraging empirical data on numerous benchmarks have supported the suggested method's advantage over existing techniques. In [41], the author proposed novel approach called Dual-Dynamic Graph Convolutional Networks (DDGCN) for detecting rumors on social media using message propagation based on structural and temporal information. Using the Sin Webio dataset their model attained 0.948 accuracy rate.

3. Material and Methods

This section details the proposed rumor recognition approach that relies on rumor diffusion and propagation. Specifically, it couples a BiGCN and LSTM models with mine rumor dispersion and diffusion. In addition, the core GCN components consist of (a) multi-layer convolutional networks and (b) an advanced variant of RNNs, including LSTM units designed to simulate chronological sequences and long-range relationships. These two components are intended to increase the accuracy of demand forecasts while improving the overall decision-making quality in case false rumors spread throughout a population. Figure 2 depicts the proposed BiGCN-LSTM model designed to address rumor detection challenges.

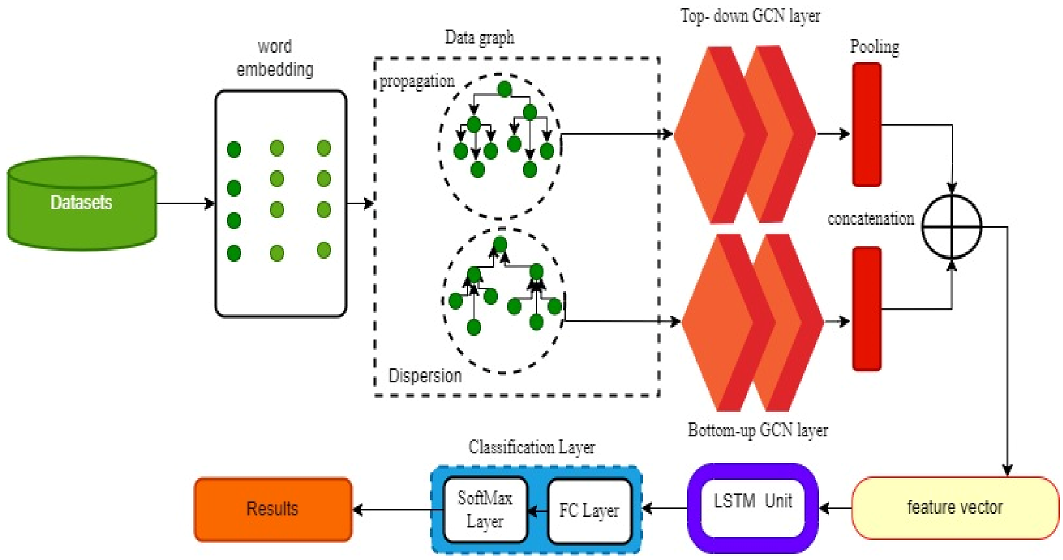


Figure 2. Structure of the proposed BiGCN-LSTM model.

Figure 2 depicts that the proposed BiGCN-LSTM model relies on three main components. First, a graph tree is deployed to represent the information diffusion and dispersion processes. Next, two convolutional layers are dedicated to extracting the post-text sequence features to produce feature vectors to the LSTM layer to determine the context. Finally, the relationship between feature vectors is conveyed to the SoftMax layer to perform the rumor classification task.

3.1. Word embedding

Word embedding is the process adopted in natural language processing to represent sets of words in given text sentences for data mining purposes. Thus, we deploy two word embedding approaches includes Word2Vec and BERT. Word2Vec [42] is defined as a text representation approach to convert the root posts and their responses in the considered datasets into numerical forms called real-valued vectors. BERT is a pre-trained language model developed by Google that uses deep learning techniques to generate word embeddings, which represent words as numerical vectors. BERT is trained on large amounts of text data and uses a transformer architecture that allows for bidirectional processing of text, enabling it to capture complex relationships between words and their contexts. The pre-trained BERT model can be fine-tuned on specific NLP tasks to improve performance, making it a widely used and effective approach for natural language processing [43]. Word embedding using Word2Vec encodes the meaning of words in the posts' content so that the words closer in vector space are expected to be similar in meaning and each word in the dataset is converted into a 64-dimensional vector. While using the Bert, the word embedding size was set to 768 vector dimensions. Bidirectionality: Word2Vec generates word embeddings based on the context of words that occur before or after a given word, while BERT uses a bidirectional architecture that allows it to consider the context of words both before and after the current position in the text. The output of word embedding is then conveyed to the next component of the model for representation in the propagation and dispersion trees.

3.2. Propagation and dispersion

The proposed rumor detection system is based on interactions between the main microblog and tweet posts, including their responses, relying on a typical propagation structure. The adjacency and feature matrices corresponding to the spreading tree of rumors appear in Figure 2, where the tree topology structure is solely comprised of edges linking the nodes at the top of the graph to those at the bottom. Notably, the conclusion of each training cycle included a fixed proportion of edges removed to prevent penitential overfitting.

The propagation and dispersion representations are aggregations of the top-down and bottom-up GCN node representations, respectively, generated from the node representations of the two networks. To aggregate this information, we concatenate the propagation and dispersion representations to integrate the information they contained. Finally, a label for event X is determined using several full connection layers and a SoftMax layer.

The training the proposed model parameters is accomplished by reducing the cross-entropy of the predictions and distributions' overall occurrences. Rumor detection, rumor representation, rumor propagation, and dispersion representation were all aspects of the rumor analysis; using microblog representations representing the mapping between word embedding and meaningful space. In particular, training the rumor detection module's classification function aims to predict the tags associated with the rumors.

3.3. Bidirectional GCN layer

A GCN is one category of neural network structure that uses a graph schema to accumulate node information from neighborhoods convolutionally. However, the considered BiGCN is constructed using two graph convolutional layers: top-down and bottom-up. Indeed, the BiGCN is intended to represent the content of the source post in the root node and its responses in child nodes, including the relationship between them using the graph's edges. Consequently, the suitable way to classify microblog content into different categories (i.e., true or false) is achieved through node classification. The fundamental idea of the BiGCN layers is to learn the exemplification of nodes through propagation and dispersion processes to classify the event of the main post in the Twitter dataset into either a true rumor, false rumor, non-rumor, or unverified rumor. In contrast, the two predefined classes are true and false rumors for the Weibo datasets.

In this module, two important subtasks such as propagation/diffusion and word embedding are conducted to create a tree structure for rumor representation, as shown in Figure 2 depicted above. The initial input x_i indicated a vector of words in a specified size of the vocabulary of the training and testing datasets in terms of Word2Vec features, where A refers to the adjacency matrix pointing to the connections among the microblog content. To improve the capability of model learning, we utilize two convolutional layers through the GCN structure, as depicted in Figure 2. The equations for the GCN layer are written as follows [44]:

$$\mathrm{H_1} = \mathrm{GCN(X, A)} = \widetilde{\mathrm{A}}\sigma\left(\widetilde{\mathrm{A}}\mathrm{X}\mathrm{W_0}\right)\mathrm{W_1} \qquad (1)$$

$$\mathrm{H_2} = \mathrm{GCN(\,H_1, A)} = \widetilde{\mathrm{A}}\sigma\left(\widetilde{\mathrm{A}}\,\mathrm{H_1}\mathrm{W_2}\right)\mathrm{W_3} \qquad (2)$$

where $\mathrm{H_1}$ and $\mathrm{H_2}$ symbolize the hidden features of two convolutional layers of GCN. $\widetilde{\mathrm{A}}$ indicated an adjacency matrix after regularization, while $\widetilde{\mathrm{A}} = \widetilde{\mathrm{D}}^{\frac{-1}{2}} \times \mathrm{A}\widetilde{\mathrm{D}}^{\frac{-1}{2}}$, D represented a degree matrix, and $\mathrm{X} \in \mathrm{R}^{\mathrm{n}\times\mathrm{d}}$ was a feature matrix. Moreover, $\mathrm{W_0}$, $\mathrm{W_1}$, $\mathrm{W_2}$ and $\mathrm{W_3}$ are trainable parameter matrices of the GCN, while σ is a non-linear activation function equivalent to the activation of the ReLU function. The retrieved adjacency matrix is formed through the formula expressed with the help of using the inner product and a sigmoid function to reconstruct the initial graph:

$$\widehat{\mathrm{A}} = \sigma\,(\mathrm{ZZ}^{\mathrm{T}}) \qquad (3)$$

where $\mathrm{Z} \in \mathrm{R}^{\mathrm{n}\times\mathrm{H}}$ indicates the matrix form of z, and σ denoted the inner-product operation. A good z should have made the recreated adjacency matrix $\widehat{\mathrm{A}}$ similar to the main adjacency matrix A. Equation 4 denotes the concatenation process of output of two GCN layers.

$$S = concat(H1, H2) \qquad (4)$$

3.4. LSTM unit

LSTM is adapted in the proposed methodology as a linked layer in the GCN model intended to detect rumors in social media networks, conveying 100 output dimensions to the next layer of the GCN-LSTM model. It was trained by taking the feature vector as input, enabling the network to understand long-term relations by applying to forget and remember gates that allowed the cell to determine the information to block or broadcast based on its strength and relevance. Graph convolution models demand long-term memory and are difficult to train, making them challenging to use. Nonetheless, it includes more extra special units that can keep information for a longer period with a memory cell, which is highly useful, in addition to a considerable number of parameters, such as learning rates and input and output biases. As a result, there is no need for precise modifications.

In this study, every LSTM unit performed pre-calculations for the input feature vector received from convolutional layers of the GCN, and its output is passed to the classification layer of the network. In every unit, four different computations are accomplished based on four gates: input ($\mathrm{i_t}$), forget ($\mathrm{f_t}$), candidate ($\mathrm{c_t}$), and output ($\mathrm{o_t}$). The LSTM structure appears in Figure 3.

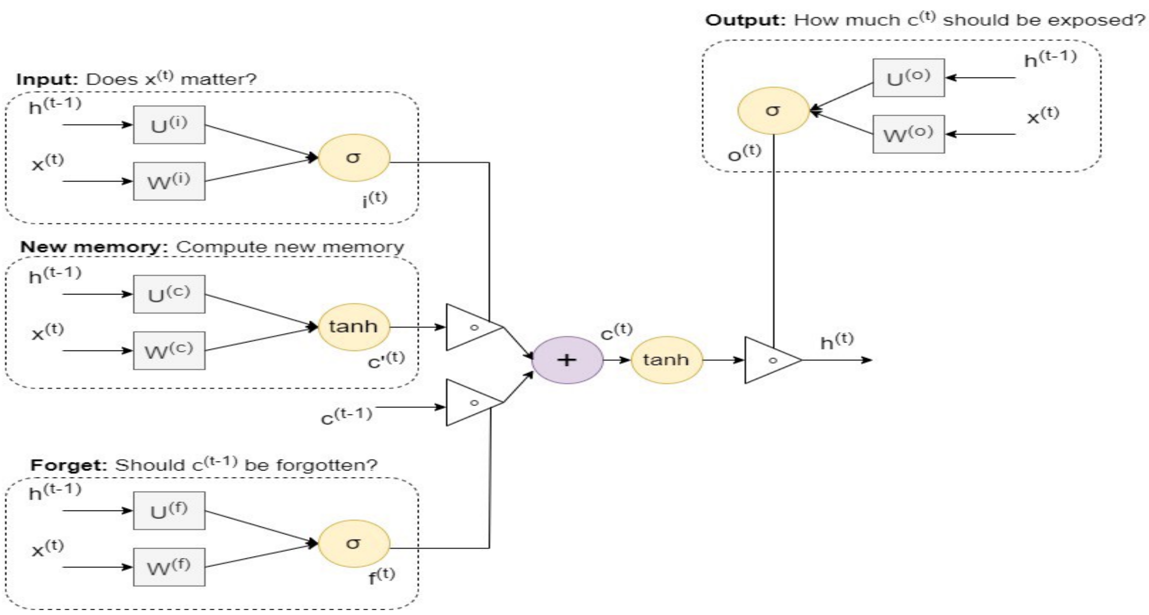


Figure 3. The considered LSTM structure.

The equations for these gates were defined as follows [45]:

$$f_t = \mathrm{sig}(Wf_{xt} + Uf_{ht} - 1 + b_f\,) \quad (5)$$

$$i_t = \mathrm{sig}(Wi_{xt} + Ui_{ht} \; - 1 + \; b_i) \quad (6)$$

$$O_t = \mathrm{sig}(Wo_{xt} + Uo_{ht} - 1 + b_o) \quad (7)$$

$$c{\sim}t = \tanh(wc_{xt} + Uc_{ht} - 1 + bc\,) \quad (8)$$

$$C_t = (f_{to}ct - 1 + i_{to}\; c{\sim}t\,) \quad (9)$$

$$h_t = O_{to} * \tanh(C_t) \quad (10)$$

$$\tanh(x) = \frac{1 - e^{2x}}{1 - e^{2x}} \quad (11)$$

where sig and tanh represent the sigmoid and tangent activation functions, respectively. Furthermore, X indicates the input data, while W and b represented the weight and bias factors of the LSTM, respectively. Similarly, C_t refers to the cell state, $c{\sim}t$ represents the candidate gate, and h_t denoted the output of the LSTM cell.

3.5. Classification layer

The classification layer is the last layer of the BiGCN-LSTM, known as the detector layer, which took the la tent representation of the sequences of post content as input and targets to categorize the event as a false or non-rumor for the Weibo dataset and a false rumor, non-rumor, true rumor, or unverified rumor in the Twitter dataset. In this layer, we utilize the mean-pooling operator to aggregate all nodes' information to event representation and the SoftMax activation function for the classification task, expressed as follows:

$$Z = (\mathrm{LSTM}(S)) \quad (12)$$

where $S \in R^{1\times H}$. Furthermore, we compute the event label using a fully connected layer using the Softmax function, calculated as follows:

$$\widehat{Y} = \mathrm{Softmax}(FL(Z)) \quad (13)$$

4. Experiments

4.1. Datasets

These experiments employed two standard datasets. Specifically, Sino Weibo and Twitter16 datasets collected by Ma et al. (2016, 2017) were used to validate and assess the proposed BiGCN-LSTM model. The Weibo and Twitter social media platforms were chosen as the most used platforms in China and the US.

In particular, we considered the users as graph nodes representing these datasets, while the graph edges denoted the relationships between response posts. Specifically, the Weibo dataset had two labels: false rumor (F) and true rumor (T). Specifically, the label of each event on Weibo was annotated according to the Sino community management center, which has received numerous allegations of disinformation [38–39]. The characteristics of the Webio data appear in Table 1.

Table 1. Characteristics of the Sina Weibo dataset.

Attributes	Count
Number of events	4,664
Non-rumors	2,351
False rumors	2,313
Users	2,746,818
Number of posts	3,805,656
False rumors	2,313
Users	2,746,818

For the Twitter16 dataset, four labels were considered: true rumor, false rumor, unverified rumor, and non-rumor. The corresponding class distribution appears in Table 2.

Table 2. Class distribution of the Twitter16 dataset.

Attributes	Count
Number of events	818
Non-rumors	205
False rumors	205
True rumors	205
Unverified rumor	203
Number of posts	204,820
Number of users	173,487

This research divided each dataset into 70% training, 10% validation, and 20% testing subsets. Table 3 reports the resulting collections.

Table 3. Dividing the datasets.

Dataset	Total number of events	Training set (70%)	Validation set (10%)	Testing set (20%)
Weibo	4,664	3,265	466	932
Twitter16	818	645	580	163

4.2. Training results

In the experimental setup, we implemented the BiGCN-LSTM to improve the detection rate of the rumored microblogs and tweets in the Sina Weibo and Twitter 16 datasets. For training, we evaluated the model performance using standard evaluation metrics: accuracy and F1 scores. Accuracy was adopted to calculate the corrected predictions divided by all predictions, while the F1 score was defined by the multiplication of precision and recall metrics divided by their addition. These metrics were used to measure the model performance for the classification of training and testing categories for the Weibo and Twitter datasets. The equations for these metrics were defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{FP} + \text{FN} + \text{TP} + \text{TN}} \tag{14}$$

$$\text{F1} - \text{score} = 2 * \frac{\text{precision} \times \text{sensitivity}}{\text{precision} + \text{sensitivity}} \tag{15}$$

where TP represented the total number of samples correctly classified into the positive class, TN was the total number of samples correctly predicted as the negative class, FP referred to false positives and indicated the total number of samples misclassified into the negative class, while FN denoted the misclassified samples of the negative class.

The proposed model parameters were tuned using stochastic gradient descent, and the model was optimized using the Adam optimization method. The hidden feature vectors for each node had a size of 64 bytes. During the training process, 16 batch size, 50 iterations were performed, and early halting was implemented when the validation loss had not decreased by 5 iterations since the start of the procedure. A novel model was developed to analyze rumor spreading: a BiGCN-LSTM. The training results appear in Table 4.

Table 4. The training results obtained with the Sina Weibo dataset.

No. of Epoch	Train Loss	Train Accuracy	Train F1-Score	Validation Loss	Validation Accuracy
00	0.4978	0.8113	0.8115	0.4282	0.8906
01	0.4136	0.8995	0.9001	0.4425	0.8627
02	0.4052	0.9062	0.9065	0.4143	0.8927
03	0.3963	0.9180	0.9181	0.4080	0.9034
04	0.3798	0.9308	0.9311	0.4079	0.9056
05	0.3679	0.9456	0.9456	0.3851	0.9206
06	0.3559	0.9579	0.9579	0.3902	0.9227
07	0.3479	0.9660	0.9659	0.3841	0.9249
08	0.3442	0.9705	0.9704	0.3887	0.9206
09	0.3358	0.9796	0.9796	0.3874	0.9270
10	0.3341	0.9815	0.9815	0.3987	0.9163
00	0.4978	0.8113	0.8115	0.4282	0.8906
01	0.4136	0.8995	0.9001	0.4425	0.8627
02	0.4052	0.9062	0.9065	0.4143	0.8927
03	0.3963	0.9180	0.9181	0.4080	0.9034
04	0.3798	0.9308	0.9311	0.4079	0.9056
05	0.3679	0.9456	0.9456	0.3851	0.9206
06	0.3559	0.9579	0.9579	0.3902	0.9227
07	0.3479	0.9660	0.9659	0.3841	0.9249
08	0.3442	0.9705	0.9704	0.3887	0.9206
09	0.3358	0.9796	0.9796	0.3874	0.9270
10	0.3341	0.9815	0.9815	0.3987	0.9163
11	0.3308	0.9839	0.9839	0.3916	0.9163

12	0.3281	0.9866	0.9865	0.3835	0.9313
13	0.3251	0.9890	0.9889	0.3920	0.9163
14	0.3279	0.9863	0.9863	0.3860	0.9206
15	0.3268	0.9869	0.9869	0.3875	0.9163
16	0.3291	0.9853	0.9853	0.3896	0.9185
17	0.3257	0.9882	0.9879	0.3820	0.9335
18	0.3268	0.9871	0.9870	0.3976	0.9163
19	0.3227	0.9912	0.9912	0.3756	0.9335
20	0.3268	0.9877	0.9877	0.3738	0.9421
21	0.3224	0.9917	0.9916	0.3782	0.9356
22	0.3233	0.9912	0.9910	0.3792	0.9335
23	0.3235	0.9904	0.9902	0.4042	0.9034
24	0.3253	0.9877	0.9877	0.3823	0.9313
25	0.3221	0.9912	0.9910	0.3794	0.9270
26	0.3215	0.9925	0.9925	0.3905	0.9206
27	0.3206	0.9930	0.9929	0.3832	0.9227
28	0.3190	0.9946	0.9945	0.3801	0.9292
29	0.3198	0.9938	0.9937	0.3857	0.9206
30	0.3247	0.9890	0.9890	0.3965	0.9142
31	0.3237	0.9895	0.9930	0.3903	0.9185
32	0.3223	0.9912	0.9909	0.3927	0.9142
33	0.3209	0.9930	0.9929	0.3934	0.9163
34	0.3188	0.9946	0.9945	0.3889	0.9227
35	0.3185	0.9949	0.9948	0.3851	0.9292
36	0.3190	0.9944	0.9944	0.3909	0.9206
37	0.3184	0.9949	0.9948	0.3914	0.9163
38	0.3190	0.9946	0.9945	0.3853	0.9270
39	0.3262	0.9882	0.9881	0.3905	0.9185
40	0.3267	0.9874	0.9874	0.3789	0.9356
41	0.3201	0.9936	0.9934	0.3822	0.9270
42	0.3195	0.9944	0.9944	0.3782	0.9313
43	0.3190	0.9946	0.9945	0.3835	0.9313
44	0.3205	0.9936	0.9936	0.3768	0.9356
45	0.3207	0.9925	0.9925	0.3861	0.9249
46	0.3176	0.9960	0.9960	0.3793	0.9313
47	0.3178	0.9960	0.9960	0.3836	0.9249
48	0.3168	0.9965	0.9964	0.3844	0.9292
49	0.3188	0.9946	0.9945	0.3949	0.9120
40	0.3267	0.9874	0.9874	0.3789	0.9356
41	0.3201	0.9936	0.9934	0.3822	0.9270
42	0.3195	0.9944	0.9944	0.3782	0.9313
43	0.3190	0.9946	0.9945	0.3835	0.9313
44	0.3205	0.9936	0.9936	0.3768	0.9356
45	0.3207	0.9925	0.9925	0.3861	0.9249
46	0.3176	0.9960	0.9960	0.3793	0.9313
47	0.3178	0.9960	0.9960	0.3836	0.9249
48	0.3168	0.9965	0.9964	0.3844	0.9292
49	0.3188	0.9946	0.9945	0.3949	0.9120

As seen in Table 4, the model was trained for 50 iterations, and the training accuracy rose from 81% to 99%. Moreover, the model validation accuracy reached 91%. Similarly, the model training loss decreased from 49% to 31%, while the validation loss and the model error rate dropped from 42% to 39%.

4.3. Testing results

This subsection reports the testing results with comparing the performance of two word embedding approaches obtained from the rumor detection experiment using the proposed hybrid intelligent deep-learning model based on the BiGCN-LSTM. Table 5 summarizes the accuracy and F1 measures obtained using the proposed model and the testing set.

Table 5. The testing results of the BiGCN-LSTM model.

Word embedding	Testing Dataset	Test Loss	Test Accuracy	Test F1 Score
Word2Vec	Sina Weibo	0.3824	0.9272	0.9272
	Twitter16	0.4100	0.8845	0.87
BERT	Sina Weibo	0.1596	0.9583	0.9585
	Twitter16	0.3195	0.9034	0.9033

As shown in Table 5, the classification results revealed that the model using BERT embeddings achieved the highest performance when applied to the Sino Weibo and Twitter datasets, while providing satisfactory results when using Word2Vec for the same datasets. Figures 3 and 4 show the capacity of the trained model using Word2Vec word embedding approach. Depicting the training and validation accuracy and loss values throughout the 50 epochs, the results demonstrated that while training accuracy and loss were optimal after ten epochs, the validation accuracy remained practically the same in all epochs, lower than during the training and validating processes.

The validation loss plots imply an overfitting of the model, as the loss changes dramatically with an increasing trend. Adding an internal drop-out layer might increase the model's capacity to generalize, which will be further studied in upcoming work in the field.

Moreover, the learning curves of the proposed method with Word2Vec embedding on the datasets, the training, validation loss, and accuracy curves over the 50 epochs on the Sina Weibo dataset were plotted to better understand whether the models were over-fitted to the data. After considerable training, we performed testing on it. As a result, we found 92% testing accuracy, 92.78% F1 measure, and 38% loss rate using the Sina Weibo dataset.

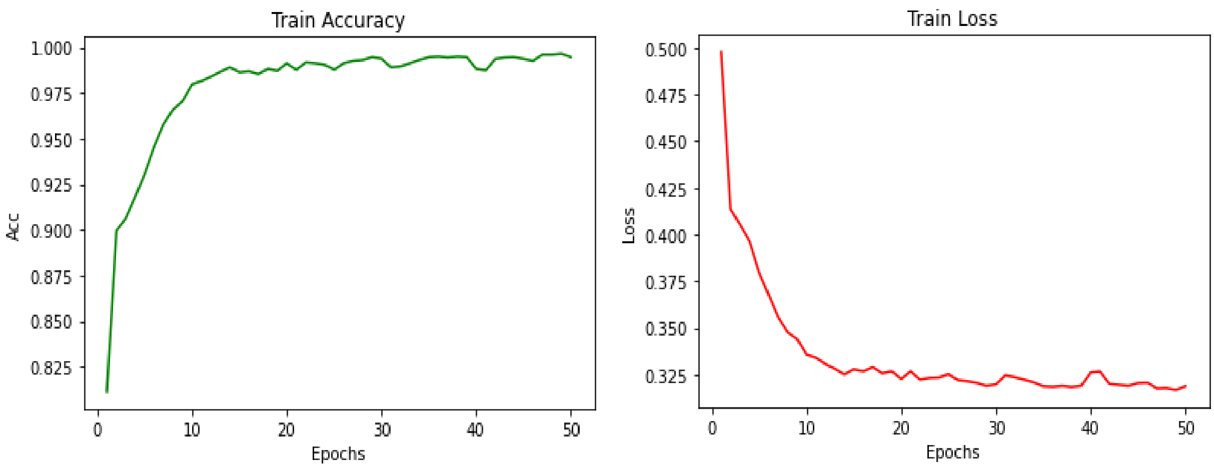


Figure 4. Training accuracy and loss obtained using the Sina Weibo dataset.

In Figure 4, the model training accuracy started at 0.82 and reached 0.99, while the model training loss decreased from 50% to 32%. These results demonstrate that the proposed model achieved the highly accurate detection of rumor propagation.

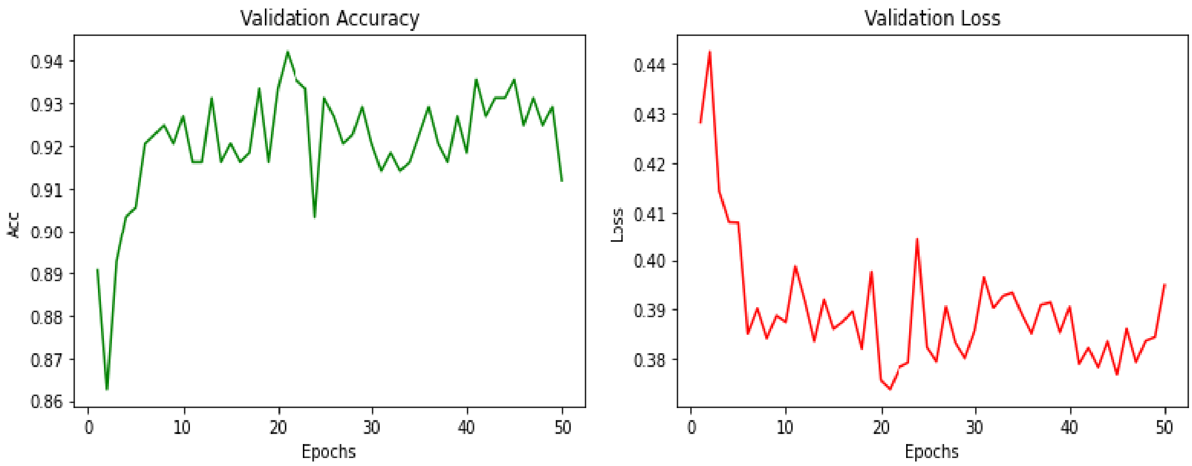
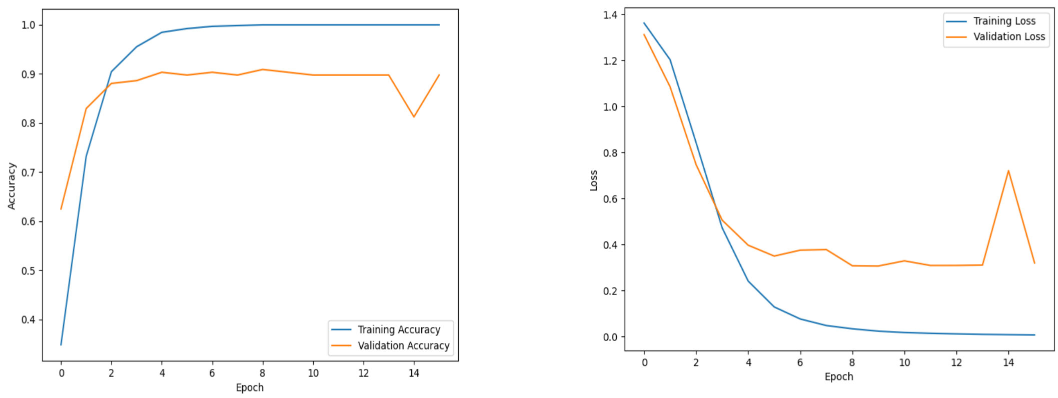


Figure 5. Validation accuracy and loss obtained using the Sina Weibo dataset.

In Figure 5, the validation accuracy increased from 0.86 to 0.92, whereas the validation loss decreased from 0.44 to 0.39.

Figures 6 and 7 show the performance plots of the model in case use BERT embedding approach for Twitter16 and Sina Weibo datasets and early stop is applied when the validation loss stops decreasing by 5 epochs.



(b)

(a)

Figure 6. Validation accuracy and loss obtained using the Twitter16 dataset.

Figure 6 shows that the proposed model achieved highly accurate detection of rumor propagation, as evidenced by the model's training accuracy improving from 0.25 to 1.00, and the training loss decreasing from 100.38% to 0%. Where the model validation accuracy started at 0.65 and reached to 0.90 and its loss decreased from 100.38 % to 30%

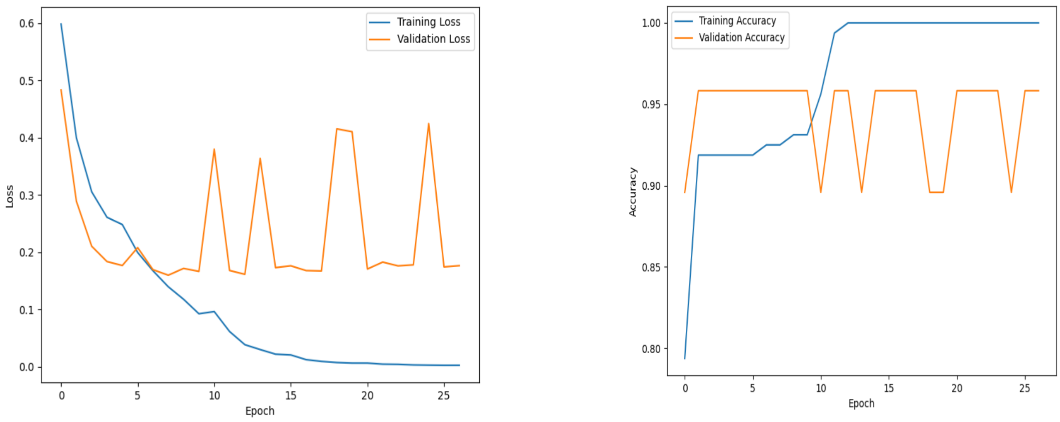


Figure 7. Validation accuracy and loss obtained using the Sina Weibo dataset.

In general, when comparing the aforementioned word embedding approaches, it is evident that the proposed model achieved the highest detection rate using the BERT approach which used to capture the textual semantics for rumor detection.

5. Comparative Analysis

This subsection compares the performance of our model with state-of-the-art methods for rumor detection using accuracy metric. DTC [46]: A method that uses a Decision Tree Classifier to identify rumors based on handcrafted features. SVM-RBF [48]: An SVM-based model with RBF kernel that employs handcrafted features based on overall statistics of the posts. SVM-TS [28]: A linear SVM classifier model that considers the time-series structure to model the variation of social context features from contents, users, and diffusion patterns. MVAE [47]: A multimodal variational autoencoder combined with a classifier for the task of rumor detection. RvNN [2]: An approach that uses GRU units to learn rumor representations via tree structure. Only-GCN [44]: A model that utilizes GCN to learn textual and propagation information without structure reconstruction, evaluated in a comparative experiment. AE-GCN [44]: A model that uses GCN as an encoder and GAE as a decoder. VAE-GCN[44]: A model that uses GCN as an encoder and Variational GAE as a decoder.

We have evaluated our model against these baselines to determine its effectiveness in detecting rumors. Table 6 shows the comparison results between the proposed model and existing ones.

Method	**Accuracy**	
	Weibo dataset	Twitter16 dataset
DDGCN	0.948	0.88
DTC [46]	0.831	0.473
SVM-RBF [48]	0.879	0.553
SVM-TS [28]	0.885	0.574
MVAE [47]	0.873	0.631
RvNN [2]	0.908	0.737
Only-GCN	0.935	0.852

AE-GCN	0.942	0.881
VAE-GCN	0.944	0.868
Proposed BiGCN-LSTM	**0.9583**	**0.9033**

Table 6. Comparison of the results of the proposed BiGCN-LSTM vs. existing methods.

5.Conclusion

Social media's intrinsic potential for quickly spreading new knowledge correlates directly with the phenomenon of information cascades. Because of the vast information accessible on social media, establishing rumor sources is becoming more challenging. Several deep-learning algorithms based recurrent neural networks (RNNs) and others, have previously been deployed to identify rumors based on how they are conveyed. Furthermore, deep-learning algorithms employed in rumor detection can only evaluate patterns of deep propagation without considering the structures of wide dispersion in the present data. They are differentiated by their potential to distribute and share knowledge, one of their most essential features.

This study presents new model to evaluate rumor propagation: BiGCN-LSTM. Analysis and classification of instances from Sina Weibo and Twitter16 datasets into predefined classes were carried out using this model, which gained knowledge of the patterns of rumor distribution by combining a GCN with a top-down and bottom-up directed graph of propagating rumors. Using BiGCN-LSTM with BERT embeddings for rumor detection can potentially result in even better performance than using BERT with LSTM alone. The BiGCN-LSTM model incorporates a Graph Convolutional Network (GCN) that can capture the structural information of the social network, which can be useful for detecting rumors that spread through a network. The LSTM component can learn the temporal dynamics of the rumor spread, while the BERT embeddings can capture the contextual information and meaning of words in the rumor content. By combining all three components, the model can potentially leverage the strengths of each to improve the accuracy of rumor detection. Notably, the proposed BiGCN-LSTM model outperformed the relevant models by considering the causal aspects of rumor propagation along connection chains via a top-to-down propagation pattern and structural features of rumor dispersion within communities via a bottom-up gathering pattern.

6.1 Limitations and future work

Despite the noticeable progress in the core field of information diffusion, research questions such as spreading information, handling false rumors, and manipulating information have still attracted researchers' interest. In particular, limitations like the convolution model and layering mishandling represent open problems for the research community. Therefore, in future work, we intend to investigate convolutional layering based on linear learning.

Acknowledgments: The authors would like to thank Deanship of scientific research in King Saud University for funding and supporting this research through the initiative of DSR Graduate Students Research Support (GSR)

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] Jenders, M.; Kasneci, G.; Naumann, F. Analyzing and predicting viral tweets. In Proceedings of the 22nd International Conference on World Wide Web 2013, pp. 657–664.

[2] Li, C.; Ma, J.; Guo, X.; Mei, Q. DeepCas: An end-to-end predictor of information cascades. In Proceedings of the 26th International Conference on World Wide Web 2017, pp. 577–586.

[3] Cao, Q.; Shen, H.; Cen, K.; Ouyang, W.; Cheng, X. DeepHawkes: Bridging the gap between prediction and understanding of information cascades. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management 2017, pp. 1149–1158.

[4] Shen, H.; Wang, D.; Song, C.; Barabási, A. L. Modeling and predicting popularity dynamics via reinforced Poisson processes. In Proceedings of the AAAI Conference on Artificial Intelligence 2014, 28, p. 1.

[5] Lazer, D.M.; Baum, M.A.; Benkler, Y.; Berinsky, A.J.; Greenhill, K.M.; Menczer, F.; Metzger, M.J.; Nyhan, B.; Pennycook, G.; Rothschild, D.; Schudson, M. The science of fake news. Science 2018, 359, pp. 1094–1096.

[6] Leskovec, J.; Adamic, L.A.; Huberman, B.A. The dynamics of viral marketing. ACM Transactions on the Web (TWEB) 2007, 1, p. 5–es.

[7] Tang, J.; Tang, X.; Xiao, X.; Yuan, J. Online processing algorithms for influence maximization. In Proceedings of the 2018 International Conference on Management of Data 2018, pp. 991–1005.

[8] Huang, K.; Wang, S.; Bevilacqua, G.; Xiao, X.; Lakshmanan, L.V. Revisiting the stop-and-stare algorithms for influence maximization. Proceedings of the VLDB Endowment 2017, 10, pp. 913–924.

[9] Li, H.; Ma, X.; Wang, F.; Liu, J.; Xu, K. On popularity prediction of videos shared in online social networks. In Proceedings of the 22nd ACM International Conference on Information and Knowledge Management 2013, pp. 167–178.

[10] Vosoughi, S.; Roy, D.; Aral, S. The spread of true and false news online. Science 2018, 359, pp. 1146–1151.

[11] Song, C.; Hsu, W.; Lee, M.L. Temporal influence blocking: Minimizing the effect of misinformation in social networks. In 2017 IEEE 33rd International Conference on Data Engineering (ICDE) 2017, pp. 847–858.

[12] Liu, Y.; Wu, Y.F. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In Proceedings of the AAAI Conference on Artificial Intelligence 2018, 32, pp. 13-122.

[13] Ye, Q.; Li, Y.; Chen, Y.; Zhao, H.V. Modeling content interaction in information diffusion with pre-trained sentence embedding. In 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) 2019, pp. 499–507.

[14] Brownlee, J. Long short-term memory networks with Python: Develop sequence prediction models with deep learning. Machine Learning Mastery 2017.

[15] Xiong, B.; Lou, L.; Meng, X.; Wang, X.; Ma, H.; Wang, Z. Short-term wind power forecasting based on attention mechanism and deep learning. Electric Power Systems Research 2022, 206, p. 107776.

[16] Gomez-Rodriguez, M.; Leskovec, J.; Schölkopf, B. Modeling information propagation with survival theory. In International Conference on Machine Learning 2013, pp. 666–674.

[17] Kempe, D.; Kleinberg, J.; Tardos, É. Maximizing the spread of influence through a social network. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2003, pp. 137–146.

[18] Romero, D.; Tan, C.; Ugander, J. On the interplay between social and topical structure. In Proceedings of the International AAAI Conference on Web and Social Media 2013, 7, pp. 516–525.

[19] Bao, P.; Shen, H.W.; Huang, J.; Cheng, X.Q. Popularity prediction in microblogging network: A case study on Sina Weibo. In Proceedings of the 22nd International Conference on World Wide Web 2013, pp. 177–178.

[20] Petrovic, S.; Osborne, M.; Lavrenko, V. Rt to win! Predicting message propagation on Twitter. In Proceedings of the International AAAI Conference on Web and Social Media 2011, 5, pp. 586–589.
[21] Pinto, H.; Almeida, J.M.; Gonçalves, M.A. Using early view patterns to predict the popularity of YouTube videos. In Proceedings of the Sixth ACM International Conference on Web Search and Data Mining 2013, pp. 365–374.
[22] Mishra, S.; Rizoiu, M.A.; Xie, L. Feature-driven and point-process approaches for popularity prediction. In Proceedings of the 25th ACM International Conference on Information and Knowledge Management 2016, pp. 1069–1078.
[23] Wang, Y.; Shen, H.; Liu, S.; Gao, J.; Cheng, X. Cascade dynamics modeling with attention-based recurrent neural network. In IJCAI 2017, pp. 2985–2991.
[24] Shu, K.; Mahudeswaran, D.; Wang, S.; Lee, D.; Liu, H. FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. Big Data 2020, 8, pp. 171–188.
[25] Shu, K.; and Liu, H. Detecting fake news on social media. Synthesis Lectures on Data Mining and Knowledge Discovery 2019, 11, pp. 1–129.
[26] Zhang, X.; Ghorbani, A.A. An overview of online fake news: Characterization, detection, and discussion. Information Processing & Management 2020, 57, p. 102025.
[27] Allcott, H.; Gentzkow, M. Social media and fake news in the 2016 election. Journal of Economic Perspectives 2017, 31, pp. 211–236.
[28] Ma, J.; Gao, W.; Wei, Z.; Lu, Y.; Wong, K.F. Detect rumors using time series of social context information on microblogging websites. In Proceedings of the 24th ACM International Conference on Information and Knowledge Management 2015, pp. 1751–1754.
[29] Bian, T.; Xiao, X.; Xu, T.; Zhao, P.; Huang, W.; Rong, Y.; Huang, J. Rumor detection on social media with bi-directional graph convolutional networks. In Proceedings of the AAAI Conference on Artificial Intelligence 2020, 34, pp. 549–556.
[30] Liu, H.; Li, Y. Weibo information propagation dissemination based on user behavior using ELM. Mathematical Problems in Engineering 2015.
[31] Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 2016.
[32] Narayan, A.; Roe, P.H.N. Learning graph dynamics using deep neural networks. IFAC-PapersOnLine 2018, 51, pp. 433–438.
[33] Li, Y.; Liu, T.; Jiang, J.; Zhang, L. Hashtag recommendation with topical attention-based LSTM. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers 2016, pp. 3019–3029.
[34] Pareja, A.; Domeniconi, G.; Chen, J.; Ma, T.; Suzumura, T.; Kanezashi, H.; Kaler, T.; Schardl, T.; Leiserson, C. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In Proceedings of the AAAI Conference on Artificial Intelligence 2020, 34, pp. 5363–5370.
[35] Liu, L.; Cheung, W.K.; Li, X.; Liao, L. Aligning Users Across Social Networks Using Network Embedding. In Ijcai 2016, 16, pp. 1774–1780.
[36] 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, pp. 701–710.
[37] Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. Advances in Neural Information Processing Systems 2016, 29.
[38] Ma, J.; Gao, W.; Mitra, P.; Kwon, S.; Jansen, B. J.; Wong, K.F.; Cha, M. Detecting rumors from microblogs with recurrent neural networks. In Ijcai 2016, pp. 3818–3824.

[39] Ma, J.; Gao, W.; Wong, K.F. Detect rumors in microblog posts using propagation structure via kernel learning. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics 2017.
[40] Li, W.; Li, Y.; Liu, W.; Wang, C. An influence maximization method based on crowd emotion under an emotion-based attribute social network. Information Processing & Management 2022, 59, p. 102818.
[41] Sun, M.; Zhang, X.; Zheng, J Ma.; G. Ddgcn: Dual dynamic graph convolutional networks for rumor detection on social media. In Proceedings of the AAAI Conference on Artificial Intelligence 2022, 36, 4, p. 4611-4619.
[42] Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 2013.
[43] Devlin, J.; Chang, M. W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 2018.
[44] Lin, H.; Zhang, X.; Fu, X. A graph convolutional encoder and decoder model for rumor detection. In 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA) 2020, pp. 300–306.
[45] Alzahrani, M. E.; Aldhyani, T. H.; Alsubari, S. N.; Althobaiti, M. M.; Fahad, A. Developing an intelligent system with deep learning algorithms for sentiment analysis of e-commerce product reviews. Computational Intelligence and Neuroscience, 2022.
[46] Castillo, C.; Mendoza, M.; Poblete, B. Information credibility on Twitter. In Proceedings of the 20th International Conference on World Wide Web 2011, pp. 675–68.
[47] Khattar, D.; Goud, J.S.; Gupta, M.; Varma, V. MVAE: Multimodal variational autoencoder for fake news detection. In the World Wide Web Conference 2019, pp. 2915–2921.
[48] 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, pp. 1–7.
[49] Raad Z. Homod et al., "Deep clustering of Lagrangian trajectory for multi-task learning to energy saving in intelligent buildings using cooperative multi-agent," in Applied Energy, Volume 351, 1 December 2023, 121843, doi.org/10.1016/j.apenergy.2023.121843.
[50] S. Goswami, A. K. Sagar, P. Nand and O. I. Khalaf, "Time Series Analysis Using Stacked LSTM Model for Indian Stock Market," 2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET), Arad, Romania, 2022, pp. 399-405, doi: 10.1109/GlobConET53749.2022.9872386.