Survey on HGAN, RP-DNN and Bi-GCN Rumor Detection Techniques on Social Media

Mehzabin Sadat Aothoi , Samin Ahsan , Fardeen Ahmed , Mohammed Julfikar Ali Mahbub , Motahar Mahtab and Annajiat Alim Rasel

Department of Computer Science and Engineering
Brac University
66 Mohakhali, Dhaka - 1212, Bangladesh
{mehzabin.sadat.aothoi, samin.ahsan, fardeen.ahmed, mohammed.julfikar.ali.mahbub, md.motahar.mahtab}@g.bracu.ac.bd
annajiat@bracu.ac.bd

***Abstract*—The task of detecting rumors has been one of the key areas of interest in the field of Machine Learning (ML), specifically in Natural Language Processing (NLP). Countless different approaches have been and regularly are being tried out to detect rumors successfully before they become contagious and potentially harmful. In this paper, we compare three different approaches to rumor detection, namely Heterogeneous Graph Attention Networks (HGAN), Rumor Propagation Based Deep Neural Network (RP-DNN), and Bi-Directional Graph Convolutional Networks (Bi-GCN), in order to better understand which method works best to hopefully help further research or application.**

***Index Terms*—Rumor Detection, Machine Learning, NLP, Heterogeneous Graph Networks, Rumor Propagation Based Deep Neural Network, Bi-Directional Graph Convolutional Networks**

I. Introduction

The notoriety rumor detection has been increasing with each passing day due to the fact that people are able to disperse rumors across social platforms much more easily. As such, the portion of false news or rumor spreaders is increasing significantly compared to preceding years. Detecting rumors has, therefore, become one of the biggest areas of interest for ML and NLP as manually sorting out the rumors would require a potentially infinite amount of human labor and time. Countless rumor detection techniques are developed and tested daily, setting the direction for future research. A typical rumor detection process contains four subtasks which are rumor detection, tracking, stance classification, and verification (Zubiaga et al., 2018)[1]. The papers our survey is based on follow a similar strategy but in varying ways.

Several methods have been used for rumor detection such as feature engineering-based machine learning methods and, more recently, deep neural network-based methods. However, these methods rely on the local semantic relations present in the rumor-spreading texts without taking into account the global semantic relations of the texts spreading different rumors. Hence, for our survey, we choose to study a method based on Heterogeneous Graph Attention Network (HGAN)[2] in which, the global semantic relations of different texts are captured and the information involved in the source tweets used for rumor detection is fused together.According to the authors, this has been the first work that constructs the text content and the source propagation of rumors as a heterogeneous tweet-word-user graph, where contains tweet, word, and user nodes. They explored a novel meta-path based heterogeneous graph attention network framework to capture the global semantic relations of text contents and integrate them with the information involved in source tweet propagations for rumor detection. The experiments on real-world Twitter datasets demonstrate that the proposed method outperforms the state-ofthe-art baselines and has a comparable ability in detecting rumors at an early stage.

Another approach we studied for our survey is RP-DNN for early rumor detection (Gao et al., 2020)[3] where the authors proposed a hybrid and context-aware deep neural network framework for tweet-level ERD, which is capable of not only learning textual contents of rumors but more importantly social-temporal contexts of their diffusion. In their model, the authors utilized social context content (CC) to provide insights about how public opinion evolves in early stages and social context meta-data (CM) to provide auxiliary information on how rumors spread how people react to rumors. A summarization of their contribution is: The authors proposed a hybrid deep learning architecture for rumor detection at the individual tweet level, in contradiction to recent work focused on event-level classification, to advance SoA performance on tweet-level ERD. Then they exploit a context-aware model that learns a unified and noise-resilient rumor representation from multiple correlated context inputs including SC, CC, and CM beyond the word-level modeling via a rumor task-specific neural language model and multi-layered temporal attention mechanisms. To then train the proposed model, they employed a large, augmented rumor data set (Han et al., 2019a)[4]. To examine its effectiveness and generalizability extensive experiments based on ablation study and LOO-CV are conducted. Gao and his group found that their model outperforms SoA models in tweet-level rumor detection and achieved comparable performance with SoA event-level rumor detection models.

The last approach that we considered for our survey is the Bi-Directional Graph Convolutional Networks (Bian et al. 2020)[5] where the authors proposed a novel bi-directional graph model with the view to explore both of the crucial characteristics of rumors, which are propagation and dispersion. In short, the entire Bi-directional GCN (Bi-GCN) method can be broken down into three parts. Firstly, in order to obtain the features of propagation Top-Down Graph Convolutional Networks (TD-GCN) are operated. In order to formulate rumor propagation, the TD-GCN forwards information from the parent node of a node in a rumor tree. Secondly, Bottom-Up graph convolutional Networks (BU-GCN) are utilized, aiming to acquire the features of dispersion. The BU-GCN aggregates information from the children nodes of a node in a rumor tree to represent rumor dispersion. Lastly, the outputs from the TD-GCN and BU-GCN are pooled, then concatenated via full connections, which produces the final results. While, to enhance the influences from the roots of rumors, the authors merged the features of the roots in rumor trees with the hidden features at each GCN layer. Furthermore, during the training phase, the authors employed DropEdge, which is a method used to reduce over-fitting for GCN based models (Rong et al. 2019)[6].

II. Related Work

A number of studies have been done to compare different methods for rumor detection. In this section, we go over some of these studies and review their findings.

Q. Li et al. (2019)[7] overviewed different studies on rumor detection. In their paper, they reviewed other studies based on the type of information exploited in the models. They review various studies including but not limited to approaches based on content information, rumor detection contests, joint learning for user stance, and rumor detection. They also point out some future directions of research for rumor detection such as utilizing cross-domain and cross-language rumor detection, early rumor detection, and knowledge base.

AR Pathak et al. (2020)[8] analyzed different methods for rumor detection in their paper. They highlight a number of approaches for rumor detection utilizing machine learning (both supervised and unsupervised) and deep learning-based approaches (CNN and RNN based), as well as hybrid approaches, i.e, approaches that make use of various methods. The authors conclude that more emphasis should be given on finding rumors from long texts and that more focus should be given on evidence to justify why some given piece of information is a rumor instead of focusing on the final decision that given information is a rumor. Lastly, they point out that rumors can spread through several mediums such as image, video, and audio rather than through text alone. Therefore these multiple mediums should be considered when performing rumor detection.

J Cao et al. (2018)[9] highlight three different paradigms for rumor detection: feature-based classification approach, credibility propagation approach, and neural network approaches. They also point out the problems regarding the datasets for rumor detection. Specifically, the number of rumors is less than the non-rumor samples. R Oshikawa et al. (2018)[10] suggest researchers investigate the combination of hand-crafted features with neural network models. Alzanin and Azmi (2018)[11] point out that some languages lack adequate research for rumor detection and urge the need to expand research to include many languages.

III. Models

This section gives a brief overview of HGAN, RP-DNN, and Bi-GCN models from the three papers our survey is covering.

A. Heterogeneous Graph Attention Network

The Heterogeneous tweet-word-user graph is the form used to build the rumor dataset. This graph contains the contents of the text and the information involved in the source tweets propagating the rumors. The authors decomposed the heterogeneous graph into subgraphs to capture the global semantic relation of the texts and the information in source tweet propagations. After the decomposition, 2 subgraphs are obtained:

- Tweet-word subgraph: In this subgraph, the nodes are the tweet and word nodes in the heterogeneous graph and the edges between nodes are the same as the edges between tweets and nodes in the heterogeneous graph
- Tweet-user subgraph: In this subgraph, the nodes are the tweet and user nodes in the heterogeneous graph and the edges between nodes are the same as the edges between a tweet and user nodes in the heterogeneous graph.

A framework subgraph attention network is used to solve rumor detection in a heterogeneous graph. The framework consists of a subgraph attention network and subgraph level attention.

Subgraph Attention Network: The subgraph attention network uses an attention mechanism similar to graph attention networks[12]. The network learns the importance of each node's neighbors and combines the representation of these neighbors with the importance of these neighbors to form each neighbor's representation.

Subgraph-level Attention: The subgraphs obtained after decomposition contain different information. The tweet-word subgraph contains the global semantic relation information of text contents, while the tweet-user subgraph contains the information involved in the propagation of source tweets. For accurate identification of rumors, the information contained in the two subgraphs needs to be combined. For this purpose, subgraph-level attention is used to learn subgraph weights for rumor detection.

Figure 1 shows the architecture of the heterogeneous graph attention network.

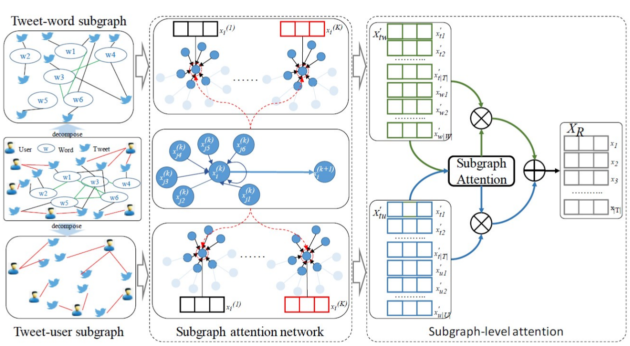


Fig. 1. The architecture of the heterogeneous graph attention networks for rumor detection.

B. Rumor Propagation Based Deep Neural Network

The overall architecture of the proposed tweet-level EDR using RP-DNN consists of four major parts which are: 1) data encoding layers, 2) stacked RNN layers, 3) stacked attention models and 4) classification layer.

The data encoding layers encode two types of raw context inputs. These important layers covert source tweets and conversation context into inputs for RNN layers for contextual modeling later in the process. The data encoding layers consist of a content embedding layer and a metadata encoding layer.

In the stacked RNN layers the authors proposed two simultaneous context embeddings to explore two correlated context inputs, and use two more layers of forward LSTMs to learn more about abstract features. The context output state H^i_{cc} at time t is abbreviated as:

$$\overrightarrow{h^i_{cc,t}} = \overrightarrow{LSTM_l}(\overrightarrow{h^i_{cc,t-1}}, v^i_{cc,t}), \forall t \in [0, j] \tag{1}$$

The context output state H^i_{cm} at time t is abbreviated as:

$$\overrightarrow{h^i_{cm,t}} = \overrightarrow{LSTM_l}(\overrightarrow{h^i_{cm,t-1}}, v^i_{cm,t}), \forall t \in [0, j] \tag{2}$$

For the Stacked Soft Attentions layer, the authors proposed the calculation of attention weight by providing information about all time steps for context embedding layers. They employed the idea of hierarchical attention networks (Yang et al., 2016b)[13] and adapted the context-aware model in their networks. The attention mechanism is applied to two layers in their architecture rather than computing them at once which are: 1) stacked RNN layers and 2) joint representation layer. The mechanism in this layer can be described with the following sequence of equations:

$$H^i_{cc\_new} = attention_1(H^t_{cc}) \tag{3}$$

$$H^i_{cm\_new} = attention_1(H^i_{cm}) \tag{4}$$

$$h^t_c = attention_2(h^t_{cc\_new} \oplus h^t_{cm\_new}) \tag{5}$$

$$v_c = \sum_t h^t_c \tag{6}$$

where h^t_c is the joint hidden states of context and v_c is the final context vector.

The Classification layer consists of two parts: Tweet content encoder and Conversational Context Metadata. For the tweet content encoder, the authors employed a SoA ELMo model finetuned for the task of rumor detection specifically (Han et al., 2019a)[4]. To produce the final short-text embeddings, averaging ELMo word vectors is employed using features from three layers of the ELMo model. For the second part, the authors let an unsupervised NLM automatically learn syntactic and semantic representations of input tweets.Figure 2 represents the structure of the proposed model.

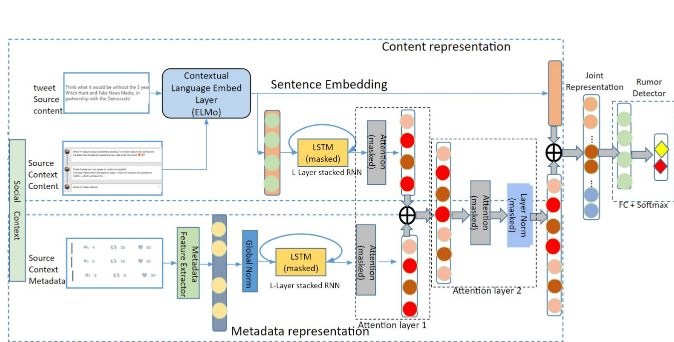


Fig. 2. Overview of the RP-DNN model architecture

C. Bi-directional Graph Convolutional Networks

The proposed Bi-directional Graph Convolutional Networks (Bi-GCN) is an effective GCN-based method. The main focus of the method is to adapt high-level representations from both of the characteristics of rumor which are propagation and rumor dispersion. The Bi-GCN models fundamental GCN components are two-layer 1stChebNet and the process of rumor detection by Bi-GCN is described briefly in four steps as follows:

Construct Propagation and Dispersion Graphs: Suppose, $C = \{c_1, c_2, c_3, \ldots, c_m\}$ is the dataset for rumor detection and c_i is the i-th event and m denotes the number of events. The authors constructed a propagate structure for a certain rumor event c_i based on the retweet and response relationships. Then, A is the corresponding adjacency matrix and X is the feature matrix of c_i. As mentioned previously, the Bi-GCN is made up of a Top-Down Graph Convolutional Network (TD-GCN) and a Bottom-Up Graph Convolutional Network (BU-GCN). The adjacency matrix for TD-GCN is $A^{TD} = A'$ and for BU-GCN the adjacency matrix is represented as $A^{BU} = A'^{T}$. Both TD-GCN and BU-GCN embrace the identical feature matrix, which is X.

Calculate the High-level Node Representations: As the DropEdge operation is completed, TD-GCN obtains the top-down propagation features. Where the BU-GCN obtains features of bottom-up propagation. Then, the hidden features of two layers (H_1 and H_2) are calculated for both TD-GCN (H_1^{TD} and H_2^{TD}) and BU-GCN (H_1^{BU} and H_2^{BU}). Also, to avoid any over-fitting issues "Dropout" (Srivastava et al. 2014)[14] is applied to the GCN layers.

Root Feature Enhancement: The authors implemented an operation with the view to improve the performance of the detection process by root feature enhancement. In this step, the hidden feature vectors of each node are concatenated with the hidden feature vector of the root node. As a result, a new feature matrix is constructed.

Representations of Propagation and Dispersion for Rumor Classification: In the final step, the author employed mean-pooling operators in order to aggregate information from the sets of TD-GCN (propagation) and BU-GCN (dispersion) node representations and get S^{TD} and S^{BU}, respectively. Then, both the representations are concatenated and the merged information goes through several layers of full connections and a layer of softmax. Figure 3 shows the overall architecture of the proposed Bi-GCN model.

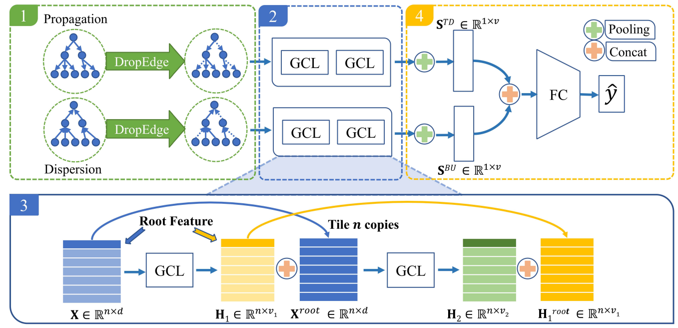


Fig. 3. The Bi-GCN model architecture

IV. Experiments

In this section we go over how the authors of each paper tested their model and the findings that surfaced consequently.

A. Tasks

For experimenting with the HGAN model, two publicly available Twitter datasets are used: Twitter15 and Twitter16. These 2 datasets were collected by Ma et al. [15]. Twitter15 contains 1490 tweets and Twitter16 contains 818 source tweets. Each of the source tweets in the datasets are labeled according to one of four labels: non-rumor, false rumor, true rumor, unverified rumor[16]. 10% of the datasets are selected as the validation set for model selection and the remaining is divided into training and test sets in the ratio of 3:1. This is done in order to ensure fairness of validation and is similar to the settings found in previous works [17], [18]. The statistics of the datasets are shown in Table-1.

TABLE I
Statistics of the HGAN datasets

Statistics	Twitter15	Twitter16
# of source tweets	1,490	818
# of users	276,663	173,487
# of tweets	331,612	204,820
# of non-rumors	374	205
# of false-rumors	370	205
# of true-rumors	372	207
# of unverified rumors	374	201

To train the RP-DNN model, the authors used a variety of different datasets. The first is, Aug-rnr (Han et al., 2019a)[4] which contains rumor and non-rumor source tweets and their contexts associated with six real-world breaking news events. Secondly, they used the Twitter 15/16 datasets which also contain rumor and non-rumor source tweets and their context in the form of propagation trees. Lastly, the PHEME (6392078; Kochinka et al., 2018)[19] dataset was used which contains manually labeled rumor and non-rumor source tweets and replies for 9 breaking news events. The four datasets are then combined to create a 7-fold LOO-CV dataset which is presented in

Table-2. The data from the datasets are then preprocessed to keep only those tweets that are informative and popular.

TABLE II
STATISTICS OF THE BALANCED LOO-CV DATASET FOR RP-DNN

LOO event	Training	Holdout	Test
charlie	4,674	496	680
ferguson	4,818	584	466
german	5,144	526	212
sydney	4,474	200	836
ottawa	4,676	536	578
twitter15	3,924	446	646
twitter16	4,600	514	382

To evaluate the Bi-GCN method, three datasets are used which are: Weibo (Ma et al. 2016)[20], Twitter15 (Ma, Gao, and Wong 2017)[15], and Twitter16 (Ma, Gao, and Wong 2017)[15]. In all of the datasets, the edges refer to retweet/response relationships, the nodes represent the users, and the extracted top-5000 words in terms of the TF-IDF values are the representation of the features (Bian et al. 2020)[5]. The Weibo dataset consists of two labels, False Rumor (F) and True Rumor (T). On the other hand, both Twitter15 and Twitter16 have four different labels, False Rumor (F), True Rumor (T), Non-rumor (N), and Unverified Rumor (U). Table 3 contains the statistics of the three mentioned datasets.

TABLE III
STATISTICS OF THE BI-GCN DATASET

Statistic	Weibo	Twitter15	Twitter16
# of posts	3,805,656	331,612	204,820
# of users	2,746,818	276,663	173,487
# of events	4664	1490	818
# of True rumors	2351	374	205
# of False rumors	2313	370	205
# of Unverified rumors	0	374	203
# of Non-rumors	0	372	205
Avg. time length/event	2,460.7 Hrs	1,337 Hrs	848 Hrs
Avg. # of posts/event	816	223	251
Max # of posts/event	59,318	1,768	2,765
Min # of posts/event	10	55	81

B. Experimental Setup

The proposed HGAN method is compared with different detection baselines based on machine learning methods: decision tree-based model(DTR)[21], decision tree based classifier (DTC)[22], random forest classifier (RFC)[23], linear SVM classifier utilizing features of Time Series (SVM-TS)[24], SVM classifier with a hybrid kernel (SVM-HK)[25], SVM classifier with a tree-based kernel (SVM-TK)[22], as well as deep learning-based methods: RNN with GRU units (RNN-GRU)[20], RNN based on traversal direction of propagation tree (BU-RvNN and TD-RvNN)[26], propagation-based classifier (PPC)[17] and a global-Local Attention Network(GLAN)[18]. To keep the comparison fair, the micro-average accuracy(Acc.) is used for all categories and the F1 measure of the precision and recall in each category is used to evaluate the performance of the models.

To evaluate the RP-DNN model through Accuracy (Acc.), precision (P), recall (R), and F1-measure (F1), the authors adopted two methods. Firstly, the LOO-CV (Ma et al., 2016[20]; Liu and Wu, 2018[17]; Chen et al., 2018[27]; Ma et al., 2018b[26]; Zhou et al., 2019[28]; Tarnpradab and Hua, 2019[29]) method where they adopted Leave one (event) out cross-validation as an approximate evaluation of their proposed models in realistic scenarios. Secondly, the K-fold CV method is adopted by performing 5-fold cross-validation to provide a comparative evaluation with more SoA methods. The baselines for evaluating the models are set with SoA models that are comparable and utilize conversational threads.

The authors compared their Bi-GCN method with some of the state-of-art methods, which are: DTC (Castillo, Mendoza, and Poblete 2011)[22], SVM-RBF (Yang et al. 2012)[30], SVM-TS (Ma et al. 2015)[24], (Ma, Gao, and Wong 2017)[15], RvNN (Ma, Gao, and Wong 2018)[26], PPC RNN+CNN (Liu and Wu 2018)[17]. The datasets were randomly split into five parts and 5 fold cross-validations were conducted to establish an unbiased comparison. The Weibo dataset was evaluated on the Accuracy (Acc.) basis over the two categories and Precision (Prec.), Recall (Rec.), F1 measure (F1) on each class. For the two Twiter datasets, the authors evaluated Acc. over the four categories and F1 on each class (Bian et al. 2020)[5]. Due to its exponential complexity, SVM-TK was not employed on the Weibo dataset.

C. Result & Analysis

In the case of the HGAN method, it performs better than all other baselines on two datasets as evident from Table-4 and Table-5. Their proposed framework achieves an accuracy of 91.1% and 92.4% respectively, increasing by 2.1% and 2.2% compared with the best baseline. This shows that the proposed framework can effectively capture the global semantic relations of the text contents in rumors, which is helpful for rumor detection. The baselines based on traditional machine learning methods do not perform as well as the deep learning-based methods. The authors observe that GLAN achieves the best performance among the baselines because it captures the local semantics and global structure information of the source tweet propagation of rumors while others capture only part of this information.

The authors' method performs very well on early rumor detection tasks as well. Their framework outperforms PPC in under 1 hour or less than 10 retweets. In this regard,

TABLE IV
HGAN RESULTS ON TWITTER15 DATASET

Method	Acc.	NR	FR	TR	UR
		F1	F1	F1	F1
DTR	0.409	0.501	0.311	0.364	0.473
DTC	0.454	0.733	0.355	0.317	0.415
RFC	0.565	0.810	0.422	0.401	0.543
SVM-TS	0.544	0.796	0.472	0.404	0.483
SVM-HK	0.493	0.650	0.439	0.342	0.336
SVM-TK	0.667	0.619	0.669	0.772	0.645
GRU-RNN	0.641	0.684	0.634	0.688	0.571
BU-RvNN	0.708	0.695	0.728	0.759	0.653
TD-RvNN	0.723	0.682	0.758	0.821	0.654
PPC	0.842	0.818	0.875	0.811	0.790
GLAN	0.890	0.936	0.908	0.897	0.817
HGAN	**0.911**	**0.953**	**0.929**	**0.905**	**0.854**

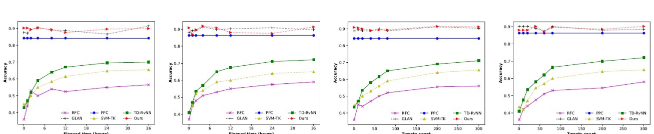
TABLE V
HGAN RESULTS ON TWITTER16 DATASET

Method	Acc.	NR	FR	TR	UR
		F1	F1	F1	F1
DTR	0.414	0.394	0.273	0.630	0.344
DTC	0.465	0.643	0.393	0.419	0.403
RFC	0.585	0.752	0.415	0.547	0.563
SVM-TS	0.574	0.755	0.420	0.571	0.526
SVM-HK	0.511	0.648	0.434	0.473	0.451
SVM-TK	0.662	0.643	0.623	0.783	0.655
GRU-RNN	0.633	0.617	0.715	0.577	0.527
BU-RvNN	0.718	0.723	0.712	0.779	0.659
TD-RvNN	0.737	0.662	0.743	0.835	0.708
PPC	0.863	0.843	0.898	0.820	0.837
GLAN	0.902	0.921	0.869	0.847	0.968
HGAN	**0.924**	**0.935**	**0.913**	**0.947**	0.899

HGAN method proposed by the author outperforms GLA. The result is visualized in Figure-4.

The RP-DNN model yielded SoA performance with larger test data comparable to all baseline models under two different evaluation techniques, as evident from Table-6 and Table-7. The full model achieved an F1 score of 0.817 in 5-fold CV and 0.727 in 7-fold LOO-CV which is an improvement of 7% over the best comparable SoA method. Several other variations of the full model, i.e., the source content only model (RPDNN-cxt), the context only model (RPDNN-SC), etc. achieved similar performance to the full model. The results show that the content of candidate source tweets can be considered as the most important and influential factor in ERD.

In the case of the Bi-GCN model, Table-7 and Table-8 holds the performance data of comparison between the Bi-GCN method and all the other mentioned methods



(a) Twitter15 (elapsed time) (b) Twitter16 (elapsed time) (c) Twitter15 (user-retweet count) (d) Twitter16 (user-retweet count)

Fig. 4. Results of HGAN early rumor detection

TABLE VI
RP-DNN PERFORMANCE COMPARISON WITH OVERALL LOO-CV RESULT

Methods	P	R	F1	Acc.
RPDNN	**0.648**	0.834	**0.727**	0.684
RPDNN-cxt	0.626	0.838	0.715	0.667
RPDNN-SC	0.621	0.796	0.694	0.648
RPDNN-CC	0.631	0.800	0.705	0.654
RPDNN-CM	0.625	**0.862**	0.723	0.669
RPDNN-Att	0.643	0.814	0.717	0.679
RPDNN-SC-CM	0.59	0.862	0.697	0.625
RPDNN-SC-CC	0.568	0.519	0.514	0.544
(Han et al., 2019a)	**0.716**	**0.614**	**0.656**	**0.685**
(Zubiaga et al., 2017)	0.692	0.559	0.601	–

on the Twitter datasets. So, the three most important takeaways from the experiment are that firstly, deep learning methods outperform those using hand-crafted features among the baseline algorithms. Secondly, in terms of all the performance measures, the Bi-GCN method performs significantly better than the PPC RNN+CNN method. Lastly, Bi-GCN is notably superior to the RvNN method.

TABLE VII
BI-GCN RESULTS ON TWITTER15 DATASET

Method	Acc.	N	F	T	U
		F1	F1	F1	F1
DTC	0.454	0.415	0.355	0.733	0.317
SVM-RBF	0.318	0.225	0.082	0.455	0.218
SVM-TS	0.544	0.796	0.472	0.404	0.483
SVM-TK	0.750	0.804	0.698	0.765	0.733
RvNN	0.723	0.682	0.758	0.821	0.654
PPC RNN+CNN	0.477	0.359	0.507	0.300	0.640
Bi-GCN	**0.886**	**0.891**	**0.860**	**0.930**	**0.864**

TABLE VIII
BI-GCN RESULTS ON TWITTER16 DATASET

Method	Acc.	N	F	T	U
		F1	F1	F1	F1
DTC	0.473	0.254	0.080	0.190	0.482
SVM-RBF	0.553	0.670	0.085	0.117	0.361
SVM-TS	0.574	0.755	0.420	0.571	0.526
SVM-TK	0.732	0.740	0.709	0.836	0.686
RvNN	0.737	0.662	0.743	0.835	0.708
PPC RNN+CNN	0.564	0.591	0.543	0.394	0.674
Bi-GCN	**0.880**	**0.847**	**0.869**	**0.937**	**0.865**

Therefore, analyzing the results from the different papers we can say that, all the models are applicable in different use cases and are the best performer in those areas. The HGAN model would be the optimal choice for understanding semantic relations in social media posts while the RP-DNN model would yield better results in analyzing the context of a particular statement. The Bi-GCN model, on the other hand, would perform the best in tracking rumor propagation and dispersion.

V. Conclusion

In this survey paper, three different methods were presented for rumor detection. From those 3 methods, the HGAN method was observed to have performed very well. HGAN can take into account the global semantic of different texts which gives it an advantage over other methods that only take into account the local semantic relation within a given text. Indeed, it was seen that HGAN outperforms several machine learning and deep learning-based methods. Furthermore, early rumor detection was also possible with this method and even in this regard, it outperformed all other baseline models. Lastly, the framework proposed opens up directions for future research. The task of integrating the social relationship of users into the heterogeneous tweet-word-user graph to improve the performance of early rumor detection can be considered an emerging research problem.

Through the analysis of the RP-DNN model, we highlighted the fact that the SoA performance achieving model addressed the task of message-level ERD in early development stages of social media rumors where limited information is available. This model also provides further research directions such as the incorporation of social network structure, examination of recent neural language models with larger context size, and generating larger training data to allow a deeper NN architecture.

As for the Bi-GCN model, its inherent GCN model gives the authors' proposed method the ability of processing graph/tree structures and learn higher-level representations more conducive to rumor detection. The experimental results on three real-world datasets demonstrate that the GCN-based approaches outperform state-of-the-art baselines by far in terms of both accuracy and efficiency. Particularly, the Bi-GCN model achieves the best performance by considering both the causal features of rumor propagation along relationship chains from top to down propagation pattern and the structural features from rumor dispersion within communities through the bottom-up gathering.

References

[1] Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., and Procter, R. (2018). Detection and resolution of rumours in social media: A survey. ACM Comput. Surv.,51(2):32:1–32:36, February.

[2] Q. Huang, J. Yu, J. Wu, B. Wang. 2020 Heterogeneous Graph Attention Networks for Early Detection of Rumors on Twitter. arXiv preprint arXiv:2006.05866v1, .

[3] Gao, J., Han, S., Song, X., Ciravegna, F. 2020. RP-DNN: A Tweet level propagation context based deep neural networks for early rumor detection in Social Media. arXiv preprint arxiv:2002.12683v2

[4] Han, S., Gao, J., and Ciravegna, F. (2019a). Neural language model based training data augmentation for weakly supervised early rumor detection. In Proceedings of 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. IEEE.

[5] Bian, T., Xiao, X., Xu, T., Zhao, P., Huang W., Rong, Y., Huang, J. (2020) Rumor detection on Social Media with Bi-Directional Graph Convolutional Networks in arXiv preprint arXiv:2001.06362v1,

[6] Rong, Y.; Huang, W.; Xu, T.; and Junzhou, H. 2019. The truly deep graph convolutional networks for node classification. arXiv preprint arXiv:1907.10903.

[7] Q. Li, Q. Zhang, L. Si, Y. Liu, 2019 Rumor Detection on Social Media: Datasets, Methods and Opportunities. arXiv preprint. arXiv:1911.07199v1

[8] A.R. Pathak, A. Mahajan, K. Singh, A. Patil, A. Nair, Analysis of Techniques for Rumor Detection in Social Media, Procedia Computer Science, Volume 167, 2020, Pages 2286-2296, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.03.281.

[9] J. Cao, J. Guo, X. Li, Z. Jin, H. Guo, J. Li. 2018. Automatic Rumor Detection on Microblogs: A Survey, in arXiv preprint arXiv:1807.03505

[10] R. Oshikawa , J. Qian, W.Y. Wang, 2018 A Survey on Natural Language Processing for Fake News Detection" in arXiv preprint arXiv:1811.00770v2

[11] Samah M. Alzanin, Aqil M. Azmi, Detecting rumors in social media: A survey, Procedia Computer Science, Volume 142, 2018, Pages 294-300, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.10.495

[12] P. Velickovi ˇ c, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Ben- ´ gio, "Graph attention networks," arXiv preprint arXiv:1710.10903, 2017

[13] Hierarchical attention networks for document classification. In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pages 1480–1489.

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

[15] J. Ma, W. Gao, and K.-F. Wong, "Detect rumors in microblog posts using propagation structure via kernel learning," in ACL, 2017, pp. 708–717.

[16] A. Zubiaga, M. Liakata, R. Procter, G. W. S. Hoi, and P. Tolmie, "Analysing how people orient to and spread rumours in social media by looking at conversational threads," PloS one, vol. 11, no. 3, 2016.

[17] Y. Liu and Y.-F. B. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in AAAI, 2018

[18] C. Yuan, Q. Ma, W. Zhou, J. Han, and S. Hu, "Jointly embedding the local and global relations of heterogeneous graph for rumor detection," arXiv preprint arXiv:1909.04465, 2019

[19] All-in-one: Multi-task learning for rumour verification. In Proceedings of the 27th International Conference on Computational Linguistics, pages 3402–3413. Association for Computational Linguistics, August.

[20] Ma, J.; Gao, W.; Mitra, P.; Kwon, S.; Jansen, B. J.; Wong, K.-F.; and Cha, M. 2016. Detecting rumors from microblogs with recurrent neural networks. In Ijcai, 3818–3824

[21] Z. Zhao, P. Resnick, and Q. Mei, "Enquiring minds: Early detection of rumors in social media from enquiry posts," in WWW, 2015, pp. 1395– 1405

[22] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in WWW, 2011, pp. 675–684

[23] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in ICDM, 2013, pp. 1103– 1108.

[24] J. Ma, W. Gao, Z. Wei, Y. Lu, and K.-F. Wong, "Detect rumors using time series of social context information on microblogging websites," in CIKM, 2015, pp. 1751–1754

[25] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on sina weibo by propagation structures," in ICDE, 2015, pp. 651–662.

[26] J. Ma, W. Gao, and K.-F. Wong, "Rumor detection on twitter with tree-structured recursive neural networks," in ACL (Volume 1: Long Papers), 2018, pp. 1980–1989

[27] Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In Pacific-Asia Confer-

ence on Knowledge Discovery and Data Mining, pages 40–52. Springer.

[28] Early rumour detection. 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), pages 1614–1623, Minneapolis, Minnesota, June. Association for Computational Linguistics.

[29] Attention based neural architecture for rumor detection with author context awareness. CoRR, abs/1910.01458

[30] Yang, F.; Liu, Y.; Yu, X.; and Yang, M. 2012. Automatic detection of rumor on sina weibo. In Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, 13. ACM.