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

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.

Keywords: Rumor Detection · Machine Learning · NLP · Heterogeneous Graph Networks · Rumor Propagation Based Deep Neural Network · Bi-Directional Graph Convolutional Networks

1 Introduction

The notoriety of 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. In a typical rumor detection process, there are usually four subtasks: 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 applied for detecting rumors 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 source tweet associated information used for rumor detection is fused together. According to the authors, this is the first approach where the text content and the source propagation of rumors are constructed as a heterogeneous tweet-word-user graph, with the graph containing tweet, word, and user nodes. They explored a heterogeneous graph attention network framework based on a novel meta-path to capture the global semantic relations of text contents and integrated them with the information involved in source tweet propagation for rumor detection. It can be seen from experiments conducted on real world Twitter datasets that the proposed method outperforms the state-of-the-art baseline methods 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 context-aware and hybrid deep neural network framework for tweet-level Early Rumor Detection (ERD), which is unique in that it not only learns the text contents of rumors but also the social and temporal contexts of their diffusion. In their model, the authors utilized social context content (CC) and social context meta-data (CM). The purpose of CC is to provide insights about how public opinion evolves in the early stages whereas CM provides additional information on how rumors spread and 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 contrast to recent work focused on event-level classification, to improve SoA performance on tweet-level ERD. Then they make use of a context-aware model that learns a noise-resilient and unified rumor representation from multiple correlated context inputs including SC, CC, and CM beyond the word-level modeling through a rumor task-specific neural language model and multi-layered temporal attention mechanisms. For the purpose of training the proposed model, they employed a large, augmented version of a rumor data set (Han et al., 2019a)[4]. Extensive experiments based on ablation study and LOO-CV are conducted to test the effectiveness and generalizability of the model . 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. The TD-GCN passes on the information from a node's parent node in a rumor tree so that rumor propagation can be formulated. Secondly, Bottom-Up graph convolutional Networks (BU-GCN) are utilized, aiming to acquire the features of dispersion. Information from the child nodes of a node in a rumor tree are aggregated by the BU-GCN in order to represent rumor dispersion. Lastly, pooling from the outputs of the TD-GCN and BU-GCN is done, then concatenated via full connections, which produces the final results. The authors merged the features of the roots in rumor trees with the hidden features at each GCN layer to enhance the influences from the roots of rumors. Furthermore, during the training phase, the authors employed DropEdge, a method used to reduce over-fitting for GCN based models (Rong et al. 2019)[6].

2 Related Work

Quite a few number of studies have been carried out to compare different methods of 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 few approaches for detecting rumors utilizing machine learning (both supervised and unsupervised) and approaches based on deep learning (CNN and RNN based), as well as hybrid approaches, i.e, approaches that make use of various methods. The authors conclude that finding rumors from long texts should be emphasized more 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 end 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.

3 Models

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

3.1 Heterogeneous Graph Attention Network

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

- Tweet-word subgraph: In this subgraph, the nodes are comprised of the tweet as well as the word nodes of the heterogeneous graph and the edges are the same as the edges between tweet and word nodes in the same graph.
- Tweet-user subgraph: In this subgraph, the nodes are made with the tweet along with the user nodes in the heterogeneous graph and the edges are the same as the edges between tweet and user nodes in the same graph.

A subgraph attention network framework 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 utilizes a mechanism of attention similar to graph attention networks[12]. The network learns the importance of the neighbors of each node and combines the neighbors' representation with their importance to form the representation of each neighbor.

Subgraph-level Attention: The subgraphs obtained after decomposition each have different information. The information in tweet-word subgraph consists of the global semantic relation 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.

3.2 Rumor Propagation Based Deep Neural Network

The general architecture of the author's proposed tweet-level EDR using RP-DNN consists of four major components: 1) data encoding layers, 2) stacked RNN layers, 3) stacked attention models and 4) classification layer.

The data encoding layers encode raw context inputs of two types. The purpose of these important layers is to convert 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.

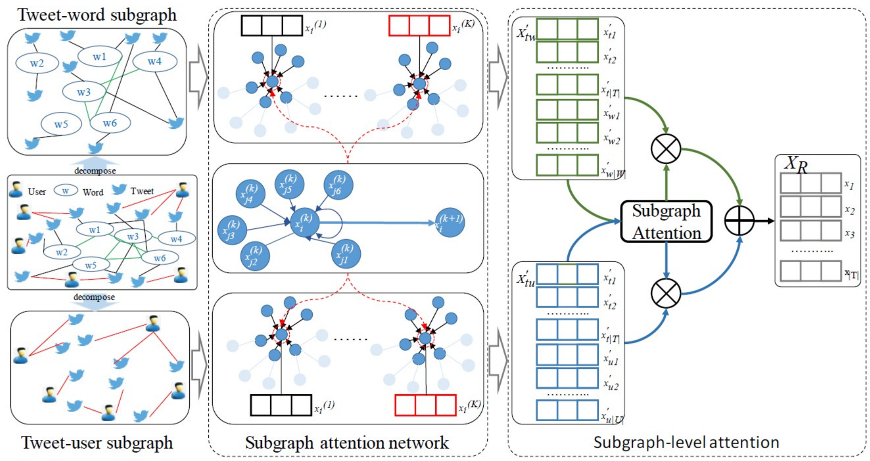

Fig. 1. The architecture of the heterogeneous graph attention networks for rumor detection.

In the stacked RNN layers the authors proposed two concurrent context embeddings to investigate two correlated context inputs, and use two more layers of forward LSTMs to get more information about features which are abstract. 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 provided information about all time steps for context embedding layers for the calculation of attention weight. They employed the concept of hierarchical attention networks (Yang et al., 2016b)[13] and adapted the context-aware model in their networks. In their architecture, the attention mechanism is applied to two layers rather than being computed 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_c^t 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 that was specifically fine-tuned for the task of rumor detection (Han et al., 2019a)[4]. In order to produce the final short-text embeddings, averaging ELMo word vectors is utilized 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.

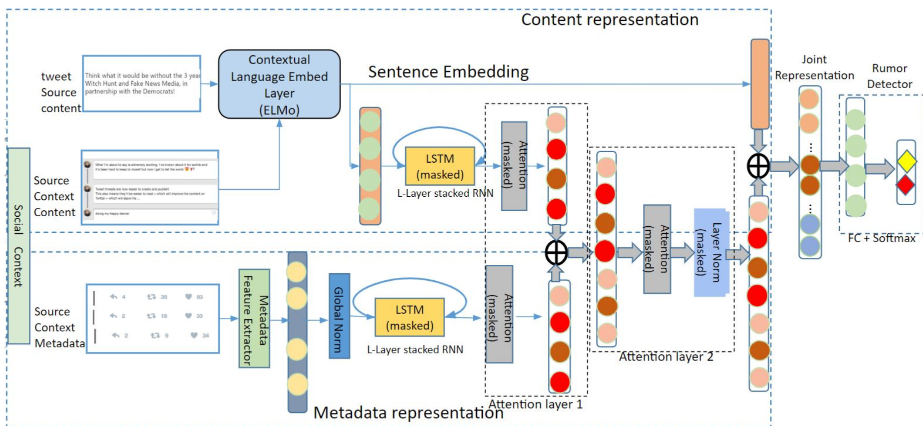

Fig. 2. Overview of the RP-DNN model architecture

3.3 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 detecting rumors, the i-th event is c_i and m denotes the events number. 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 TD-GCN and a BU-GCN. The adjacency matrix for TD-GCN is $A^{TD} = A'$ and the adjacency matrix for BU-GCN is represented as $A^{BU} = A'^{T}$. Both TD-GCN and BU-GCN embrace the identical feature matrix, which is X.

Calculation of High-level Node Representations: As the operation of DropEdge is completed, TD-GCN obtains the features of top-down propagation. Where the BU-GCN obtains the 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 enhance the performance of the detection process by root feature enhancement. In this step, each nodes' hidden feature vectors are concatenated with the root node's hidded feature matrix. 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.

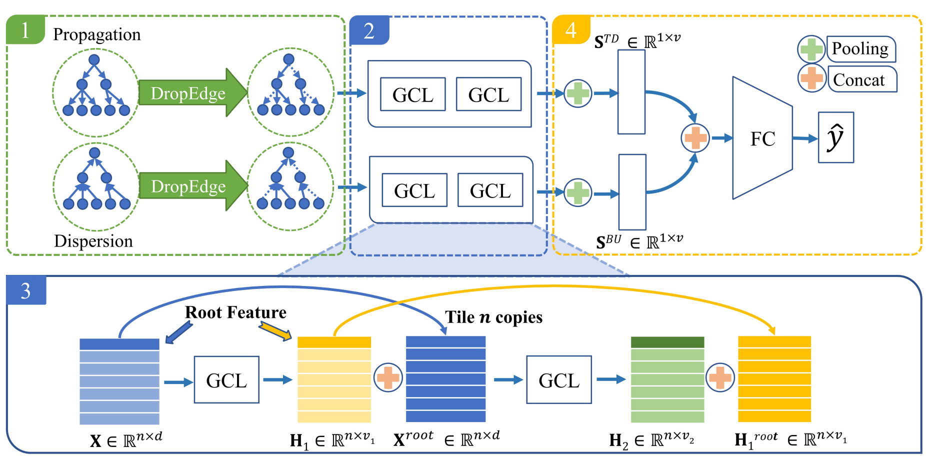

Fig. 3. The Bi-GCN model architecture

4 Experiments

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

4.1 Tasks

For experimenting with the HGAN model, two Twitter datasets which are publicly available are used: Twitter15 and Twitter16. These datasets were collected by Ma et al. [15]. Twitter15 is comprised of 1490 tweets while Twitter16 is made up of 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 set for validation to select models and the remainder is split into training and test sets in the ratio of 3:1. This is done 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 1. Statistics of the HGAN datasets

Statistics	Twitter15	Twitter16
No. of source tweets	1,490	818
No. of users	276,663	173,487
No. of tweets	331,612	204,820
No. of non-rumors	374	205
No. of false-rumors	370	205
No. of true-rumors	372	207
No. 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 source tweets of rumor and non-rumor along with their contexts which are in association with six breaking news events in real-world. Secondly, they used the Twitter 15/16 datasets which also contain source tweets of rumor and non-rumor and their context in propagation tree forms. Lastly, the PHEME (6392078; Kochinka et al., 2018)[19] dataset was used which contains manually labeled source tweets of rumor and non-rumor along with replies of 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 that are informative and popular.

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 top-5000

Table 2. 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

words from extraction on the basis of the TF-IDF values which 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 3. Statistics of the Bi-GCN dataset

Statistic	Weibo	Twitter15	Twitter16
No. of posts	3,805,656	331,612	204,820
No. of users	2,746,818	276,663	173,487
No. of events	4664	1490	818
No. of True rumors	2351	374	205
No. of False rumors	2313	370	205
No. of Unverified rumors	0	374	203
No. of Non-rumors	0	372	205
Avg. time length/event	2,460.7 Hrs	1,337 Hrs	848 Hrs
Avg. No. of posts/event	816	223	251
Max No. of posts/event	59,318	1,768	2,765
Min No. of posts/event	10	55	81

4.2 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)[5], 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 to evaluate the performance of the models the F1 measure of the precision and recall in each category is used.

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 the cross-validation technique Leave one (event) out as a rough 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 SoA 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 basis of Accuracy (Acc.) over the two categories and Precision (Prec.), Recall (Rec.), F1 measure (F1) on each class. For the two Twitter 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.

4.3 Result and Analysis

In the case of the HGAN method, it is better performing than the entirety of the baselines on two datasets as evident from Table-4. The framework proposed achieves 91.1% and 92.4% accuracy respectively, further rising by 2.1% and 2.2% compared to the highest scoring baseline. The result clearly shows that the framework can successfully capture the global semantic relations residing in the text contents of rumors, which is helpful for detecting rumors. The baselines based on traditional machine learning methods do not perform as well as the deep learning-based methods. The authors observe that the best performance was achieved by GLAN among the others because it captures not only the local semantics but also the global structure information of the source tweet propagation of rumors in contrast to others which only fractionally captures this information.

The authors' method performs splendidly on ERD tasks as well. Their framework outperforms PPC in under 1 hour or less than 10 retweets. In this regard, HGAN method proposed by the author outperforms GLA.

Table 4. HGAN results on Twitter15 and Twitter16 dataset

Twitter15						Twitter16				
Method	Acc.	N	F	T	U	Acc.	N	F	T	U
		F1	F1	F1	F1		F1	F1	F1	F1
DTR	0.414	0.394	0.273	0.630	0.344	0.414	0.394	0.273	0.630	0.344
DTC	0.465	0.643	0.393	0.419	0.403	0.465	0.643	0.393	0.419	0.403
RFC	0.585	0.752	0.415	0.547	0.563	0.585	0.752	0.415	0.547	0.563
SVM-TS	0.574	0.755	0.420	0.571	0.526	0.574	0.755	0.420	0.571	0.526
SVM-HK	0.511	0.648	0.434	0.473	0.451	0.511	0.648	0.434	0.473	0.451
SVM-TK	0.662	0.643	0.623	0.783	0.655	0.662	0.643	0.623	0.783	0.655
GRU-RNN	0.633	0.617	0.715	0.577	0.527	0.633	0.617	0.715	0.577	0.527
BU-RvNN	0.718	0.723	0.712	0.779	0.659	0.718	0.723	0.712	0.779	0.659
TD-RvNN	0.737	0.662	0.743	0.835	0.708	0.737	0.662	0.743	0.835	0.708
PPC	0.863	0.843	0.898	0.820	0.837	0.863	0.843	0.898	0.820	0.837
GLAN	0.902	0.921	0.869	0.847	0.968	0.902	0.921	0.869	0.847	0.968
HGAN	**0.924**	**0.935**	**0.913**	**0.947**	0.899	**0.924**	**0.935**	**0.913**	**0.947**	0.899

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

Table 5. 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	–

In the case of the Bi-GCN model, the performance data of comparison between the Bi-GCN method and all the other mentioned methods on the Twitter datasets are shown in Table-6. 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, with regard to 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 6. Bi-GCN results on Twitter15 and Twitter16 dataset

Twitter15						Twitter16				
Method	Acc.	N	F	T	U	Acc.	N	F	T	U
		F1	F1	F1	F1		F1	F1	F1	F1
DTC	0.454	0.415	0.355	0.733	0.317	0.473	0.254	0.080	0.190	0.482
SVM-RBF	0.318	0.225	0.082	0.455	0.218	0.553	0.670	0.085	0.117	0.361
SVM-TS	0.544	0.796	0.472	0.404	0.483	0.574	0.755	0.420	0.571	0.526
SVM-TK	0.750	0.804	0.698	0.765	0.733	0.732	0.740	0.709	0.836	0.686
RvNN	0.723	0.682	0.758	0.821	0.654	0.737	0.662	0.743	0.835	0.708
PPC RNN+CNN	0.477	0.359	0.507	0.300	0.640	0.564	0.591	0.543	0.394	0.674
Bi-GCN	**0.886**	**0.891**	**0.860**	**0.930**	**0.864**	**0.880**	**0.847**	**0.869**	**0.937**	**0.865**

Therefore, analyzing the output from several papers we may conclude 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.

5 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 methods based on machine learning and deep learning. Furthermore, early detection of rumor 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 users' social relationship into the heterogeneous tweet-word-user graph in order to help the completion process of early rumor detection improve can be considered an emerging research problem.

Through the analysis of the RP-DNN model, we highlighted the fact that in the very early development stages of social media rumors the SoA performance achieving model is able to address the task of message-level ERD, even in cases where limited information is available. This model also provides further research directions such as incorporating social network structure, examining 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, it inherits all the built-in features of GCN models which allows the proposed model to process structures such as, graphs and trees and also, learn higher-level representations which improves the process of detecting rumors. Considering both accuracy and efficiency, we can see that the GCN-based approaches notably outperform state-of-the-art baselines from the experimental output that we get from our three datasets. Particularly, the proposed model performs the highest result both the causal features of rumor propagation 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 Conference 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.