Rumor Detection on Social Media with Event Augmentations

Zhenyu He*
University of Electronic Science and Technology of China
Chengdu, China
zhenyu.h@outlook.com

Ce Li*
University of Electronic Science and Technology of China
Chengdu, China
ce.lc@outlook.com

Fan Zhou†
University of Electronic Science and Technology of China
Chengdu, China
fan.zhou@uestc.edu.cn

Yi Yang
Hong Kong University of Science and Technology
Hong Kong, China
imyiyang@ust.hk

ABSTRACT

With the rapid growth of digital data on the Internet, rumor detection on social media has been vital. Existing deep learning-based methods have achieved promising results due to their ability to learn high-level representations of rumors. Despite the success, we argue that these approaches require large reliable labeled data to train, which is time-consuming and data-inefficient. To address this challenge, we present a new solution, **R**umor **D**etection on social media with **E**vent **A**ugmentations (RDEA), which innovatively integrates three augmentation strategies by modifying both reply attributes and event structure to extract meaningful rumor propagation patterns and to learn intrinsic representations of user engagement. Moreover, we introduce contrastive self-supervised learning for the efficient implementation of event augmentations and alleviate limited data issues. Extensive experiments conducted on two public datasets demonstrate that RDEA achieves state-of-the-art performance over existing baselines. Besides, we empirically show the robustness of RDEA when labeled data are limited.

CCS CONCEPTS

• Information systems → Social networking sites; **• Human-centered computing → Collaborative and social computing**; **• Computing methodologies → Artificial intelligence**.

KEYWORDS

Rumor Detection; Contrastive Learning; Event Augmentation

ACM Reference Format:
Zhenyu He, Ce Li, Fan Zhou, and Yi Yang. 2021. Rumor Detection on Social Media with Event Augmentations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3463001

*Equal contribution
†Corresponding author

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463001

1 INTRODUCTION

The wide availability of posting channels in online social platforms such as Twitter, Weibo, Instagram, and Facebook facilitates the propagation of massages [24]. However, this also contributes to the rapid dissemination of unsubstantiated rumors and conspiracy theories which often elicit rapid, large, but naive social responses [18]. Indeed, massive digital misinformation has been designated as a major technological and geopolitical risk by the 2013 report of the World Economic Forum. False rumors have affected our economies, which are not immune to the spread of falsity [19].

A rumor can be understood as an item of circulating information whose veracity status has not been verified at the time of posting. Zubiaga et al. [26] state rumors into two types, "New rumors" that emerge during breaking news and "Long-standing rumors" that are discussed for long periods of time. Hence the most common approaches for rumor detection are trying to exploit the content and propagation structure, which may reflect attractive root features or diffusion patterns.

Traditional detection systems train supervised classifiers, e.g., Decision Tree [3], Random Forest [7] and Support vector Machine (SVM) [22], with hand-crafted features. Recent deep learning based approaches utilize powerful temporal-structural techniques (e.g., LSTM, GRU, RvNN [9, 11]) to capture sequential features from rumor propagation or convolutional neural networks (e.g., CNN [23], Bi-GCN [2]) to learn high-level representations extracted from the propagation and the dispersion of rumors.

Nevertheless, existing methods still suffer from a major challenge: most of them heavily rely on supervised training [2, 9, 11, 12], where large labeled data is necessary but expensive or hard to obtain [15, 25]. It is very time-consuming and labor-intensive to label reliable data, as expert annotators are often required to perform a careful analysis of claims and additional evidence, context, and reports from authoritative sources [16]. Han et al. [4] present a data augmentation technique for rumor detection using contextualized word representations ELMo [13] from the perspective of semantic relatedness, which however ignores the propagation character of rumor events.

Motivated by this, we propose a self-supervised learning framework RDEA, in which three novel event augmentation strategies are designed (i.e., node masking, subgraph, and edge dropping). We permute both content features and propagation structures to generate positive samples for rumor events. Thus the intrinsic data correlation is utilized to derive self-supervision signals and enhance the event representation by contrastive pre-training with the augmented data. Then we fine-tune the model with label to make the final prediction. The main contributions of our work are as follows:

- We design three interpretable augmentation strategies, which are underexplored in rumor event graph-data.
- We present the RDEA framework[1] to learn rumor event representations, which leverages self-supervised pre-training in a contrastive manner. To the best of our knowledge, this is the first study to integrate contrastive SSL in rumor detection.
- Extensive experiments conducted on two public datasets show that RDEA effectively outperforms other supervised methods. Also, we empirically prove the robustness of our model when labeled data are limited.

2 METHODOLOGY

In this section, we first formally define the task of rumor detection and then introduce our proposed approach. Figure 1 shows three main components of RDEA, i.e., event graph data augmentation, contrastive pre-training, and model fine-tuning.

Problem Statement. Let $C = [C_1, C_2, \cdots, C_{|C|}]$ be a Twitter rumor detection dataset, where C_i is the i-th event and $|C|$ is the number of events. $C_i = [r_i, x_1^i, x_2^i, \cdots, x^i_{|\mathcal{V}_i|-1}, G_i]$, where r_i is the source post, each x_j^i represents the j-th relevant responsive post, $|\mathcal{V}_i|$ refers to the number of posts in C_i, and $G_i = \langle \mathcal{V}_i, \mathcal{E}_i \rangle$ denotes an event graph of a set of edges $\mathcal{E}$ and nodes $\mathcal{V}$ with r_i being the root node [2, 12, 20]. If x_2^i has a response to x_1^i, there will be an directed edge $x_1^i \rightarrow x_2^i$. If x_1^i has a response to r_i, there will be an directed edge $r_i \rightarrow x_1^i$. We denote the feature matrix and the adjacency matrix as $E \in \mathbb{R}^{|\mathcal{V}| \times d}$ and $A \in \{0,1\}^{|\mathcal{V}| \times |\mathcal{V}|}$.

The goal of rumor detection is to learn a classifier $f : C_i \rightarrow Y_i$, where Y_i takes one of the four finer-grained classes N, F, T, U (i.e., Non-rumor, False Rumor, True Rumor, and Unverified Rumor).

2.1 Event Augmentation

Unlike CV and NLP tasks that treat each data instance as isolated, posts and comments in one rumor event are inherently connected and dependent on each other. Thus, augmentation operators tailored for event representation are necessary. We note that many rumor creators are intended to mislead the public, which also leads to a vulnerable event structure and representation. Moreover, malicious users and naive users are likely to promote the spread of rumors, and the echo chamber effect takes place frequently in social media. To address these problems, we introduce three event augmentation strategies for rumor detection by modifying the graph structure and node attributes but still, maintain the keypoint of the event.

Node masking. The participants in a rumor event can be grouped into (1) malicious users and (2) naive users by their intentions [25]. Malicious users intentionally spread false rumors, driven often by monetary and/or non-monetary benefits (e.g., power and popularity) and naive users are vulnerable normal users who unintentionally engage in false rumor propagation – they mistake false rumors as truth. Consequently, relying too much on participants in a rumor

[1] Source code available at https://github.com/hzy-hzy/RDEA

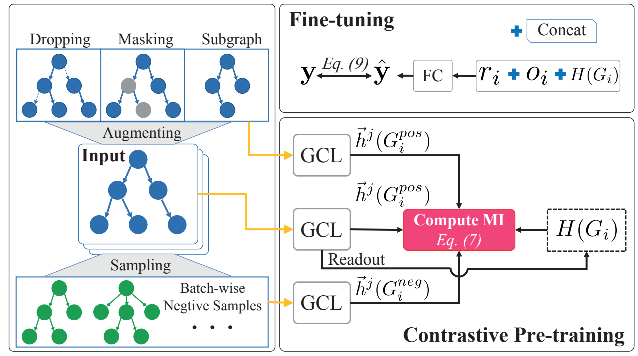


Figure 1: An overview of the proposed RDEA method. Contrastive vector of the input event graph is obtained via contrastive pre-training first. Then, we obtain the event representation by concatenating the pre-trained vector with textual graph vector and source post features. Last, we make predictions via fully connected layers and fine tune the model parameters.

event may cause bad effects. To tackle this problem, We randomly mask node features in the event graph except the root node at each training epoch. Formally, given node feature matrix without root node $E^{-r} \in \mathbb{R}^{(|\mathcal{V}|-1) \times d}$ and the masking rate is p_m, the feature matrix after applying the node masking (E^{-r}_{mask}) is computed as follows:

$$E^{-r}_{mask} = \mathrm{M} \odot E^{-r}, \tag{1}$$

where $\mathrm{M} \in \{0,1\}^{(|\mathcal{V}|-1) \times d}$ is the masking matrix by randomly generating $(|\mathcal{V}|-1) \times p_m$ rows of zero vector.

Subgraph. It is observed that the majority of the users support true rumors and a higher number deny false rumors when the entire life cycle of the rumor is considered; whereas by studying only the early reactions to rumors, it was found that users showed a tendency to support rumors independent of their veracity [15]. Hence, having too much access to the whole life cycle of an event when training the model may hinder its ability to detect rumors in an early stage when inferring. To tackle this problem, we generate the subgraph of an input event graph G_i by starting a random walk from the root node at each training epoch. The walk parallel and iteratively travels to its neighborhood with a probability p_s. When the random walk is over, we obtain the generated subgraph G_{i_sub}.

Edge dropping. DropEdge [14] is a novel technique to alleviate over-fitting and over-smoothing for GCN-based models. It randomly removes edges from input graphs at each training epoch, which helps to augment training data and reduce message passing. In addition, we apply DropEdge in the event graph to reduce the echo chamber effect, which can potentially affect users' stances and viewpoints and increase social polarization and extremism [1, 17]. Formally, given an adjacency matrix A with N_e edges and the dropping rate is p_d, the adjacency matrix after applying DropEdge, A_{drop} , is computed as follows:

$$A_{drop} = A - A', \tag{2}$$

where A' is the constructed matrix by randomly sampling $N_e \times p_d$ edges from the original edges.

2.2 Contrastive Pre-training

In this session, we present how to obtain mutual information via contrastive pre-training among the input and augmented events.

Formally, for the node j of event graph G, the self-supervised learning procedures are as follows:

$$h_j^{(k)} = \mathbf{GCL}(h_j^{(k-1)}), \tag{3}$$

$$h^j = \mathbf{CONCAT}(\{h_j^{(k)}\}_{k=1}^K), \tag{4}$$

$$H(G) = \mathbf{READOUT}(\{h^j\}_{j=1}^{|\mathcal{V}|}), \tag{5}$$

where $h_j^{(k)}$ is the feature vector of node j at the k-th layer, GCL is the graph convolutional encoder [6], h^j is obtained by summarizing feature vectors at all depth of the GCL into a single feature vector that captures patch information at different scales centered at every node, and $H(G)$ is the global representation of the given event graph after applying READOUT. We test with several GNN encoders and pooling methods, and choose GIN [21] as our GCL and mean as our READOUT function. The goal of contrastive pre-training is to maximize the mutual information (MI) over the rumor propagation graph dataset, which is computed as:

$$\begin{aligned} I_\psi(h^j(G); H(G)) :=& \mathbb{E}[-sp(-T_\psi(\vec{h}^j(G_i^{pos}), H(G_i)))] \\ &- \mathbb{E}[sp(T_\psi(\vec{h}^j(G_i^{neg}), H(G_i)))], \end{aligned} \tag{6}$$

where I_ψ is the mutual information estimator modeled by discriminator T_ψ, G_i is an input event graph sample, G_i^{pos} are the positive samples of G_i, G_i^{neg} are the negative samples of G_i, and $sp(z) = \log(1 + e^z)$ is the softplus function. As for positive samples, we use the local patch representations of the input event graph and the generated graph samples in Section 2.1, i.e., $G_i(E_{mask}^{-r})$, G_{i_sub}, and $G_i(A_{drop})$. As for negative samples, we use the local patch representations of other event graphs in a batch. Note that ψ denotes the set of parameters of a neural network.

To better predict the rumor veracity, we develop a strategy to combine textual features, mutual information among event graphs and put emphasis on the source post, then use this combined embedding as the event graph representation.

After contrastive pre-training on the event graphs, we obtain the pre-trained vector $H(G_i)$ of an input event graph G_i by (5). Then, for an event $C_i = [r_i, x_1^i, x_2^i, \cdots, x_{|\mathcal{V}_i|-1}^i, G_i]$, we get the textual graph vector o_i by averaging the original features of all the relevant responsive posts and the source post, i.e., $o_i = \frac{1}{n_i}(\sum_{j=1}^{|\mathcal{V}_i|-1} x_j^i + r_i)$. To put emphasis on the source post, we simply reuse the source post features r_i. Then, we concatenate the contrastive vector, textual graph vector, and source post features to merge the information as:

$$\mathbf{S}_i = \mathbf{CONCAT}(H(G_i), o_i, r_i). \tag{7}$$

2.3 Fine tuning

After pre-training and utilizing textual features, we obtain the pre-trained event representation which has self-supervised signals derived from the raw data and its augmentation. In the fine-tuning stage, we initialize parameters with pre-trained parameters and train the model with labeled data. Then predictions are made via fully connected layers and a softmax layer:

$$\hat{\mathbf{y}}_i = \text{softmax}(FC(\mathbf{S}_i)). \tag{8}$$

Then we adopt the cross entropy loss of the predictions $\hat{Y}$ and ground truth distributions Y over all events C, which defined as:

$$\mathcal{L}(Y, \hat{Y}) = \sum_{i=1}^{|C|} \mathbf{y}_i \log \hat{\mathbf{y}}_i + \lambda \|\Theta\|_2^2, \tag{9}$$

where $\|\Theta\|_2^2$ is the L_2 regularizer over all the model parameters Θ, λ is a trade-off coefficient.

3 EXPERIMENTS

3.1 Experimental Setup.

Datasets. For experimental evaluation, we use two publicly available Twitter datasets released by [12] namely *Twitter15* and *Twitter16*. In two datasets, nodes refer to users, edges represent retweet or response relationships, and features are the extracted top-5000 words in terms of the TF-IDF values mentioned in Section 2. Each source tweet is annotated with one of the four class labels, i.e., Non-rumor (N), False rumor (F), True rumor (T), and Unverified rumor (U), according to the veracity tag of the article in rumor debunking websites (e.g., snopes.com and Emergent.info) [12].

Baselines. We conduct experiments on several baselines:

- **DTC** [3]: A rumor detection approach applying decision tree that utilizes tweet features to obtain information credibility.
- **SVM-TS** [10]: A linear SVM-based time-series model that leverages handcrafted features to make predictions.
- **RvNN** [11]: A recursive tree-structured model with GRU units that learn rumor representations via the tree structure.
- **PPC_RNN+CNN** [8]: A rumor detection model combining RNN and CNN for early-stage rumor detection, which learns the rumor representations by modeling user and source tweets.
- **Bi-GCN** [2]: using directed GCN, which learns the rumor representations through Bi-directional propagation structure.

Metrics. We use accuracy (Acc.) and F-measure (F_1) as the evaluation protocols. Specifically, Acc. measures the proportion of correctly classified source tweets, and F_1 is the harmonic average of the precision and recall values across four classes.

Parameter settings. We randomly split the datasets into five parts and conduct 5-fold cross-validation to obtain robust results. For both the datasets, we evaluate Acc. over the four categories and F_1 on each class. The parameters of RDEA are updated using stochastic gradient descent, and we optimize the model by Adam algorithm [5]. The dimension of hidden feature vectors is 64. The masking rate in node masking is set to 0.2, the probability in subgraph is 0.4, and the dropping rate in DropEdge is 0.4. The self-supervised pre-training process is iterated upon 25 epochs, while the supervised fine-tuning process is iterated upon 100 epochs, and early stopping is applied when the validation accuracy stops increase by 10 epochs.

3.2 Results and Analysis

Performance Comparison. Table 1 shows the performance of the proposed method and all the baselines on *Twitter 15* and *Twitter 16*, respectively. First, it is obvious that all the deep learning methods

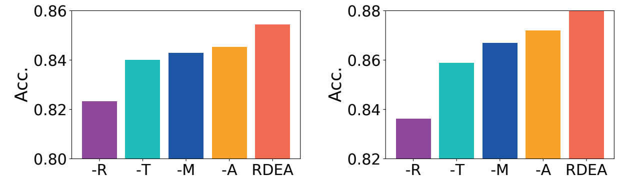
Table 1: Rumor detection results on *Twitter15* and *Twitter16* datasets with 100% label fraction data. Abbrev.: Non-Rumor (N), False Rumor (F), True Rumor (T), Unverified Rumor (U).

Method	Acc.	F_1			
		N	F	T	U
Dataset: *Twitter15*					
SVM-TS	0.544	0.796	0.472	0.404	0.483
DTC	0.454	0.733	0.355	0.317	0.415
RvNN	0.723	0.682	0.758	0.821	0.654
PPRC_RNN+CNN	0.697	0.689	0.760	0.696	0.645
Bi-GCN	0.836	0.791	0.842	0.887	0.801
RDEA	**0.855**	**0.831**	**0.857**	**0.903**	**0.816**
Dataset: *Twitter16*					
SVM-TS	0.574	0.755	0.420	0.571	0.526
DTC	0.465	0.643	0.393	0.419	0.403
RvNN	0.737	0.662	0.743	0.835	0.708
PPRC_RNN + CNN	0.702	0.608	0.711	0.816	0.664
Bi-GCN	0.864	0.788	0.859	0.932	0.864
RDEA	**0.880**	**0.823**	**0.878**	**0.937**	**0.875**

perform significantly better than those that use handcrafted features, demonstrating that deep learning methods are able to learn high-level representations of rumors for detection. Second, the proposed method outperforms other deep learning methods in terms of all the performance measures, which indicates the effectiveness of RDEA. As for RvNN, it only uses the hidden feature vectors of all the leaf nodes, which is heavily influenced by the latest posts, losing much information of the former posts. As for PPC_RNN+CNN, it just views the propagation structure as flat time series, which loses much structural information. As for Bi-GCN, although more powerful than the other two baselines, it still faces the problem that representations are vulnerable to noisy interactions and requires large labeled data to train. Finally, as for our proposed RDEA, it obtains mutual information via contrastive pre-training between different rumor events, which captures the intrinsic correlation of rumor propagation. Moreover, by root feature enhancement and textual graph, RDEA emphasizes the importance of source post and the original content information.

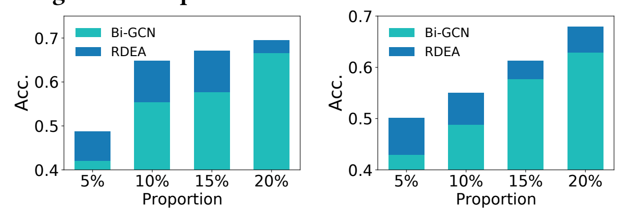
Ablation study. To analyze each variant of RDEA, we compare the proposed method with -R, -T, -A, and -M. The empirical results are summarized in Figure 2. -R, -T, -A, and -M represent our model without root feature enhancement, textual graph, event augmentation, and mutual information, respectively. We have the following two observations. First, the performance of RDEA decreases without any one of the four parts, which indicates that they are all vital to RDEA. Second, the performance of -R decreases the most, which indicates that root features are indispensable as it represents the source of propagation structure. 其实在表达 网络应该有侧重性质

Limited labeled data. Figure 3 shows the performance as label fraction varies. We observe that RDEA is more label-efficient for both datasets than Bi-GCN. Furthermore, the fewer the labels the larger the improvements, which suggesting the robustness and data-efficiency of RDEA.



(a) *Twitter15* (b) *Twitter16*

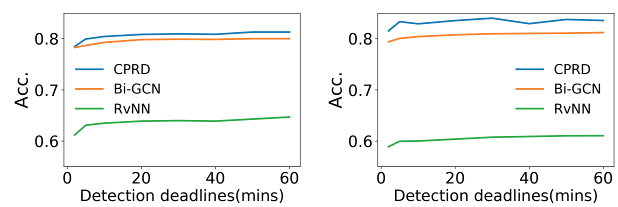
Figure 2: The performance of the RDEA and variants



(a) *Twitter15* (b) *Twitter16*

Figure 3: Fine-tuning results with different label fractions

Early Rumor Detection. Detecting rumors at the early stage of propagation is essential to effectively prevent rumors from spreading and influencing people. For an early detection task, we choose a series of detection deadlines and only use the posts released before the deadlines (i.e., we keep observation times of all rumor events the same) to evaluate the performance in terms of accuracy. Figure 4 shows the results of our RDEA versus Bi-GCN and RvNN at various deadlines for the *Twitter15* and *Twitter16* datasets. We can also observe that the performance of all three methods is almost fixed at an early time (slightly improves over time).



(a) *Twitter15* (b) *Twitter16*

Figure 4: Result of rumor early detection on two datasets

4 CONCLUSIONS

This study proposes a novel model that innovatively integrates three augmentation strategies to extract meaningful rumor propagation patterns and capture intrinsic representations of user engagement. GNN-based encoder and contrastive learning mechanism give the proposed model the ability to dig information from the complex rumor propagation structure and learn mutual information between propagation event and its augmentation. The experimental results on two real-world datasets demonstrate that RDEA outperforms state-of-the-art baselines in terms of both accuracy and F_1 score. We also evaluate RDEA's performance in detecting rumor in an early stage. Lastly, our model is robust when labeled data are limited.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 62072077) and the National Key R&D Program of China (2019YFB1406202).

REFERENCES

[1] Pablo Barberá, John T. Jost, Jonathan Nagler, Joshua A. Tucker, and Richard Bonneau. 2015. Tweeting From Left to Right: Is Online Political Communication More Than an Echo Chamber? *Psychological Science* (2015).

[2] Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks. In *AAAI*.

[3] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *WWW*.

[4] Sooji Han, Jie Gao, and Fabio Ciravegna. 2019. Data augmentation for rumor detection using context-sensitive neural language model with large-scale credibility corpus. In *ICLR*.

[5] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[6] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[7] Sejeong Kwon, Meeyoung Cha, K. Jung, Wei Chen, and Y. Wang. 2013. Prominent Features of Rumor Propagation in Online Social Media. In *ICDM*.

[8] Yang Liu and Yi-fang Brook Wu. 2018. Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks. In *AAAI*.

[9] J. Ma, Wei Gao, P. Mitra, Sejeong Kwon, Bernard J. Jansen, K. Wong, and Meeyoung Cha. 2016. Detecting Rumors from Microblogs with Recurrent Neural Networks. In *IJCAI*.

[10] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. 2015. Detect Rumors Using Time Series of Social Context Information on Microblogging Websites. In *CIKM*. 1751–1754.

[11] Jing Ma, Wei Gao, and K. Wong. 2018. Rumor Detection on Twitter with Tree-structured Recursive Neural Networks. In *ACL*.

[12] Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning. In *ACL*.

[13] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *NAACL-HLT*. 2227–2237.

[14] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2020. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *ICLR*.

[15] Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu. 2019. Combating Fake News: A Survey on Identification and Mitigation Techniques. *ACM Transactions on Intelligent Systems and Technology* (2019).

[16] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter* 19, 1 (2017), 22–36.

[17] H Akin Unver. 2017. Digital Challenges to democracy: politics of automation, attention, and engagement. *Journal of International Affairs* (2017).

[18] Michela Del Vicario, A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. Stanley, and Walter Quattrociocchi. 2016. The spreading of misinformation online. *PNAS* 113 (2016), 554 – 559.

[19] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* 359, 6380 (2018), 1146–1151.

[20] Ke Wu, Song Yang, and Kenny Q. Zhu. 2015. False rumors detection on Sina Weibo by propagation structures. In *ICDE*. 651–662.

[21] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *ICLR*.

[22] Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. 2012. Automatic detection of rumor on Sina Weibo. In *MDS*.

[23] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, Tieniu Tan, et al. 2017. A Convolutional Approach for Misinformation Identification.. In *IJCAI*. 3901–3907.

[24] Fan Zhou, Xovee Xu, Goce Trajcevski, and Kunpeng Zhang. 2021. A survey of information cascade analysis: Models, predictions, and recent advances. *Comput. Surveys* 54, 2, Article 27 (2021), 36 pages. https://doi.org/10.1145/3433000

[25] Xinyi Zhou and Reza Zafarani. 2020. A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. *Comput. Surveys* 53, 5, Article 109 (2020), 40 pages. https://doi.org/10.1145/3395046

[26] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and Resolution of Rumours in Social Media: A Survey. *Comput. Surveys* 51, 2, Article 32 (2018), 36 pages. https://doi.org/10.1145/3161603