Dual Graph Networks with Synthetic Oversampling for Imbalanced Rumor Detection on Social Media

Yen-Wen Lu
National Cheng Kung University
Tainan, Taiwan
ywlu2@illinois.edu

Chih-Yao Chen
University of North Carolina at Chapel Hill, NC, USA
cychen@cs.unc.edu

Cheng-Te Li
National Cheng Kung University
Tainan, Taiwan
chengte@ncku.edu.tw

ABSTRACT

Rumor detection is to identify and mitigate potentially damaging falsehoods, thereby shielding the public from misleading information. However, existing methods fall short of tackling class imbalance, meaning rumor is less common than true messages, as they lack specific adaptation for the context of rumor dissemination. In this work, we propose Dual Graph Networks with Synthetic Oversampling (SynDGN), a novel method that can determine whether a claim made on social media is rumor or not in the presence of class imbalance. SynDGN properly utilizes dual graphs to integrate social media contexts and user characteristics to make accurate predictions. Experiments conducted on two well-known datasets verify that SynDGN consistently outperforms state-of-the-art models, regardless of whether the data is balanced or not.

CCS CONCEPTS

• **Information systems → Data mining**.

KEYWORDS

Rumor Detection, Fake News Detection, Graph Neural Networks, Class Imbalance, Oversampling, Co-attention Mechanism

ACM Reference Format:
Yen-Wen Lu, Chih-Yao Chen, and Cheng-Te Li. 2024. Dual Graph Networks with Synthetic Oversampling for Imbalanced Rumor Detection on Social Media. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3589335.3651494

1 INTRODUCTION

The dissemination of disinformation on social media has become a serious problem that needs to be tackled. To prevent rumors from widely spreading, an effective way to detect rumors is crucial. Automatic rumor detection based on deep learning has made great progress in recent years [23]. Typical approaches learn the truthfulness of a claim (e.g., a tweet) by capturing contextual features [15]. In the real-world applications of rumor detection, the model is expected to learn from a limited number of identified rumors, compared to a large set of non-rumors [3, 16]. That said, the issue of class imbalance needs to be handled specifically.

WWW '24 Companion, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0172-6/24/05...$15.00
https://doi.org/10.1145/3589335.3651494

Table 1: Summary. Abbrev.: class imbalance (CI), GNN-based, user attributes (UA), user comments (UC), propagation structure (PS), auxiliary data (AD), emotion words (EW), knowledge base (KB), knowledge graph (KG), Source (Src) domain.

	CI	GNN	UA	UC	PS	AD
RvNN [11]				✓	✓	
BayesianDL [21]				✓		
AIFN [19]				✓		EW
GEAR [22]		✓				KB
BiGCN [1]		✓	✓	✓	✓	
GCAN [10]		✓	✓			
GERDA [3]	✓(Text Gen.)	✓				KG
EBGCN [18]		✓		✓	✓	
SynDGN (this work)	✓(Embedding Gen.)	✓	✓	✓		

The class imbalance issue on rumor detection is not properly addressed. Three challenges are faced. First, when learning to represent a given thread on social media, we need to incorporate information about the social context of postings [8, 12], such as user comments and user attributes, into the design of mitigating class imbalance. Second, data resampling is an effective strategy to deal with class imbalance only if proper features are adopted [14]. The resampling needs to be aware of elements that can benefit the unfolding of veracity, such as users and their interactions with posts, rather than only the instances of text posts. Third, the supervision signal of the rumor class is very limited. We should develop a mechanism to better utilize the labeled rumor texts, and propagate supervision signals to unlabeled ones. We summarize the most represensitive and relevant studies in Table 1. Most methods have similar settings with UA, UC, and/or PS, and utilize GNNs to better capture the social contexts of rumors.

In this paper, we present a novel model, Dual Graph Networks with Synthetic Oversampling (SynDGN) for *imbalanced rumor detection.* This work has two goals. The first is to determine the veracity of claims in a class-imbalanced setting. The second is to effectively incorporate semantic and user information into the detection model. The idea of SynDGN is two-fold. One is leveraging dual GNNs on users and tweets to better utilize the limited supervision signal. The other is a novel synthetic embedding generation module to effectively oversample the minority class on social contexts. While traditional applications of SMOTE [2] operate on the feature space directly, SynDGN uniquely incorporates dual GNNs. The design seeks to delve deeper into both tweet threads and user behaviors, ensuring the synthetic samples are not just duplicates but are contextually relevant and rich in the extracted information from both domains. Furthermore, in SynDGN, the synthetic embedding generation is not a disjoint process. Instead, it runs in tandem with

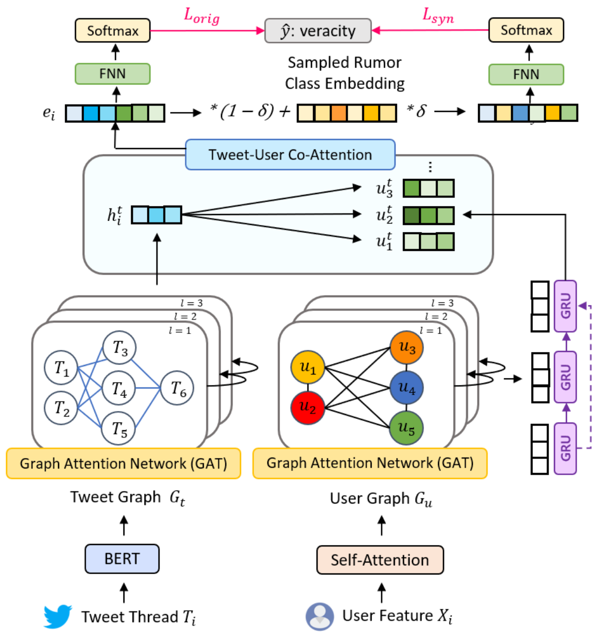

Figure 1: The overview of the SynDGN model.

the joint learning of representations from tweets and users. This integrated learning method allows the generated embeddings to be directly influenced by the interplay between tweet content and user behavior, a layer of complexity beyond the conventional SMOTE application. Extensive experiments conducted on two benchmark datasets show that SynDGN outperforms state-of-the-art models, and SynDGN also shows robustness with the presence of class imbalance. The contribution of this work is as follows: (a) We propose a novel model, SynDGN, with dual graph networks to fuse features from text and user profiles for rumor detection. (b) SynDGN consistently outperforms competing baselines on two public benchmark datasets. (c) Our SynDGN shows robustness when the class distribution is imbalanced.

2 METHODOLOGY

Problem Statement. Given a Twitter thread T_i, encompassing a source tweet and subsequent comments, paired with user features X_i. The goal is to predict the veracity of the source tweet, denoted $\hat{y}_i \in 0, 1$. Besides, we want the model to maintain robust performance even when faced with imbalanced class distribution, i.e., when the number of *rumor* instances is considerably smaller than of *non-rumor* ones in the training data.

The overview of our SynDGN model is exhibited in Figure 1. First, we encode the tweet thread, including the source tweet and its comments, into contextualized text representations, and we construct a tweet graph according to the learnable correlation between text representations. Second, We utilize two graph neural networks (GNNs) to learn the representations of tweet threads and users, respectively. Following that, third, a co-attention mechanism is established to capture the correlation between text representations and user embeddings. Then we interpolate the fused representations to address the data imbalance issue through a proposed synthetic embedding generation module. A feed-forward layer is used to yield the prediction outcome.

Learning Tweet Representation. We employ the pre-trained BERT [6] as the sentence encoder. Given a word sequence, i.e., source tweet or each of its comments, we feed it into BERT and take the hidden state corresponding to the `[CLS]` token as the sentence representation. To model the correlation between sentences in a tweet thread, we construct a tweet graph $\mathcal{G}_t = (\mathcal{N}_t, \mathcal{E}_t)$ for information propagation, where each node represents a sentence, and edges are initialized to be fully-connected. The initial representation of each node n_j^t, denoted h_j^0, is assigned as the derived sentence representation. The edge weights w_{jk} can be learned via graph attention networks (GAT) [17]. As a result, node representations h_j^l at layer l will be updated via aggregating its neighbors' messages, given by: $h_j^l = \sum_{k \in \mathcal{N}(n_j^t)} w_{jk} h_k^{l-1}$.

Learning User Representations. We utilize GNNs to learn the representations of users that participate in a tweet thread. The idea is to highlight reliable users who contribute the most to determine whether a tweet is a rumor. We follow DTCA [20] to extract user features, such as whether a user is verified, whether allowing geo-positioning or not, and the number of followers/followees/favorites. Then we utilize an MLP layer to convert the original user features into low-dimensional user embeddings. Afterward, the multi-head (4 heads in our setting by default) self-attention is used to learn the user representation. Similar to the tweet graph, we also build a user graph $\mathcal{G}_u = (\mathcal{N}_u, \mathcal{E}_u)$ where each node n_i^u is a user, and each edge represents the correlation between users. Identically, we again use GAT to derive edge weights α_{ij}, and the aggregated user representation can be obtained by: $\hat{u}_i = \sum_{j \in \mathcal{N}(n_i^u)} (\alpha_{ij} W_{out}^j W_{in}^i) \hat{h}_j + b_u$, where W_{out}^j, W_{in}^i and b_u are learnable parameters. In addition, to better encode user features, each node's representation is updated with a GRU module with the residual connection to obtain the final user representation. Meanwhile, to model the latent correlation between users and sentences, we adopt co-attention [10] to combine information from sentence embeddings of tweet graph h_i and user embedding of user graph u_i, and generate the fused representation e_i for a tweet thread T_i.

Synthetic Embedding Generation. To mitigate the data imbalance issue, we propose an augmentation technique called Synthetic Embedding Generation (SEG), inspired by SMOTE [2], to generate synthetic representations for *rumor* instances (i.e., the minority class). SMOTE originally selects the nearest neighboring instances with minor class in the feature space, and creates additional samples by interpolating their features. However, the process of selecting nearest neighbors can be computationally intensive. To overcome this, we adopt a different approach in SEG. Instead of selecting nearest neighbors, we randomly choose embeddings from the rumor instances and perform interpolation to generate synthetic embeddings. This is achieved through the following formula: $h_{syn} = (1-\delta) \cdot e_i + \delta \cdot (e_{rumor})$, where e_i is the representation derived from Section 2.2, and e_{rumor} is the BERT embedding of a randomly sampled rumor instance. In each batch, we always generate h_{syn} with the same number of non-rumor instances. The main advantage of SEG is its efficiency, achieved with a minor trade-off in performance. Moreover, we can dynamically adjust the contribution of rumor class embeddings during the augmentation process

Table 2: Data settings and statistics.

Datasets	Split	Threads	Replies	True	False
RumorEval	Train	177	2530	127	50
	Dev	22	198	10	12
	Test	20	822	8	12
PHEME	Train	1369	15883	826	543
	Dev	160	2068	105	55
	Test	176	2040	136	40

Table 3: Main results. The scores are in percentage.

	RumorEval			PHEME		
	P	R	F1	P	R	F1
SVM	72.20	58.33	64.53	75.68	73.42	74.53
CNN	71.05	54.16	61.47	81.06	78.81	79.92
DeClarE	80.76	79.16	79.95	83.78	76.07	79.74
RvNN-BU	64.73	75.00	69.49	70.36	58.88	64.11
RvNN-TD	80.69	79.13	79.90	66.72	57.42	61.72
BaysienDL	71.53	73.48	72.49	77.17	67.85	72.21
AIFN	73.81	70.00	71.85	71.73	72.73	72.23
GEAR	80.00	81.25	80.62	81.11	78.82	79.95
BiGCN	84.23	84.01	84.12	86.87	86.75	86.81
EBGCN	86.48	86.57	86.52	87.11	87.34	87.22
SynDGN	**93.07**	**88.02**	**90.47**	**88.12**	**87.43**	**87.77**

by δ, where δ is a random variable $\delta \in [0, 1]$ following a uniform distribution.

Final Prediction. We employ an MLP layer to generate the final prediction, given by: $\hat{y} = \text{softmax}(W_s \cdot h_{syn} + b_s)$, where $\hat{y} \in \{0, 1\}$ is the final prediction representing whether a tweet is a rumor, and W_s and b_s are model weights. Similarly, we take e_i after another MLP layer as input to predict veracity, given by: $\hat{y} = \text{softmax}(W_r \cdot e_i + b_r)$, where W_r and b_r are model weights. This yields two losses, corresponding to the orignal (orig) instances and synthesized (syn) instances. We use cross entropy loss for both of them, denoted as $\mathcal{L}_{syn}$ and $\mathcal{L}_{orig}$. The main purpose of $\mathcal{L}_{syn}$ is to align synthetic embeddings to the correct prediction. A hyperparameter β is used to weigh the two objectives, and the final loss is expressed as: $\mathcal{L} = (1 - \beta) \cdot \mathcal{L}_{orig} + \beta \cdot \mathcal{L}_{syn}$, in which we set $\beta = 0.5$ by default.

3 EXPERIMENTAL EVALUATION

Evaluation Settings. We evaluate our model on two well-known datasets, PHEME [24] and RumourEval [5]. We follow the settings of previous studies [19–21] about feature usage and data splitting. The data statistics are presented in Table 2. Both datasets contain Twitter conversation threads. A thread comprises a source tweet (i.e., claim), its tree-structure replies (i.e., comments), and a veracity label (i.e., fake or real). The evaluation metrics include Precision (P), Recall (R), and macro F1. For both datasets, the user feature embedding is set to 32. For RumourEval, we use $5e-5$ for the learning rate, 0.3 for L2 regularization, and 32 for batch size. For PHEME, they are set to $5e-5$, 0.1, and 256, respectively. The number of layers for GNN is 2. We use Adam [9] as the optimizer. We compare SynDGN with the following competitors: **SVM** [7], **CNN** [4], **DeClarE** [13], **RvNN-BU** [11], **RvNN-TD** [11], **BaysienDL** [21], **AIFN** [19], **GEAR** [22], **BiGCN** [1], and **EBGCN** [18]. For the hyperparameters of all baselines, we refer to their original papers and follow their tuning strategies for choosing the best performance.

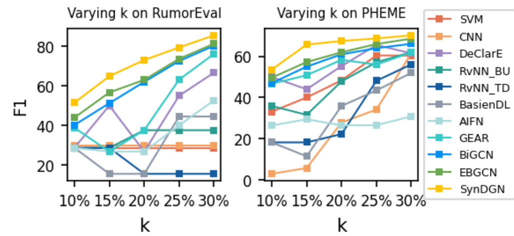

Figure 2: Results on imbalanced data by varing k.

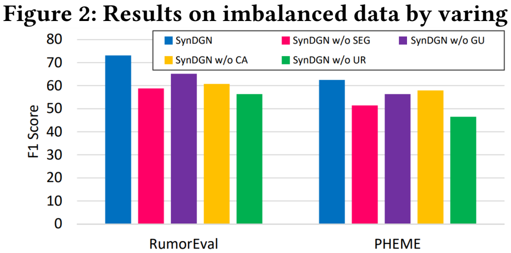

Figure 3: Results on ablation study under $k = 20\%$.

Main Results. As shown in Table 3, the proposed SynDGN significantly outperforms all baselines by a large margin. Even when compared with the state-of-the-art, Bi-GCN and EBGCN, SynDGN consistently leads to higher scores. In addition, we observe that all methods perform better on PHEME. The reason could be that RumourEval contains fewer training instances, and its testing set contains events that are not seen in the training set. Nevertheless, SynDGN substantially closes the gap between datasets, showing that SynDGN generalizes well even when the size of the training set is small.

Results on Class Imbalance. We compare the performance under imbalanced data situations. We define the number of instances in the rumor/fake class as k percentage of the amount of non-rumor data, $k \in \{10\%, 15\%, 20\%, 25\%, 30\%\}$. The goal is to simulate the real-world scenario that the number of real claims is far more than the number of rumors. All experiments are run 10 times with randomly sampled rumor data, and the performance in terms of F1 score is shown in Fig 2. SynDGN consistently outperforms all competitors when the data exhibits various class imbalances, indicating that even though minority class data is insufficient, SynDGN is still able to generalize well.

Ablation Study. We conduct an ablation study to examine the effectiveness of each component of the proposed SynDGN . We analyze four variants of our model: SynDGN without Synthetic Embedding Generation (w/o SEG), without Gated recurrent unit's Updating on user representations (w/o GU), without Co-Attention (w/o CA), and without the entire User Representation learning (w/o UR). The ablation study is performed on the class imbalance scenario with $k = 20\%$, and F1 scores are reported. The results are exhibited in Figure 3. We can find that every design does contribute positively to the prediction performance. Among these components, the generation of synthetic embeddings is the most effective when tackling the class imbalance issue. More importantly, the synthesis of embeddings in SynDGN is intrinsically linked to the representations of tweets and users participating in the target thread, rather than functioning as an isolated mechanism. In essence, the efficacy of our synthetic embedding generation, especially in the context of imbalanced rumor detection, hinges on accurate portrayal of user-tweet interactions. Any iterations of our model that omit updating

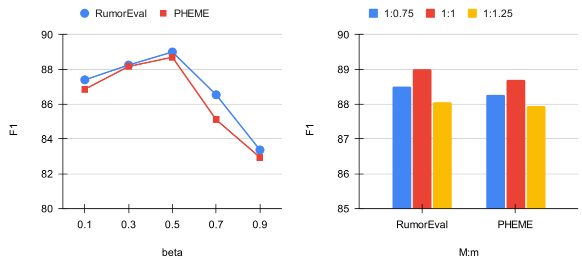

Figure 4: Results on hyperparameter sensitivity for the balancing factor β and the synthetic ratio M:m.

user representations (w/o GU) or exclude the co-attention mechanism (w/o CA) exhibit diminished performance. This underscores that our synthetic embedding generation is intertwined with, and indeed vital to, the nuanced demands of rumor detection.

Hyperparameter Sensitivity. We study the sensitivity of SynDGN to the hyperparameters β and $M : m$, which play pivotal roles in addressing class imbalance through instance synthesis. By varying $\beta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and $M : m = 1 : 0.75, 1 : 1, 1 : 1.25$, we report the performance in F1 in Figure 4. The hyperparameter β is instrumental in modulating the trade-off between the classification loss functions of original and synthesized instances. As β increases, the model's emphasis on the loss function of synthesized instances is augmented, ostensibly to enhance the model's acumen in identifying rumors. The declining trend with increasing β values suggests that an excessive focus on synthesized instances may detract from overall classification performance, likely due to overfitting to the characteristics of synthetic data. The hyperparameter $M : m$ dictates the ratio of original to synthesized instances of rumors, aiming to furnish a more balanced training dataset. The bar chart indicates a nuanced response to different $M : m$ ratios. We can see the synthesis of rumor instances leads to better results when oversampling to the balance status (i.e., $M : m = 1 : 1$), which is the default setting of SynDGN. Over-emphasizing the synthesized rumor instances (e.g., $M : m = 1 : 1.25$) may hurt the performance.

4 CONCLUSIONS AND DISCUSSION

In this work, we propose a novel model, SynDGN, for imbalanced rumor detection on social media. SynDGN is developed based on the tweet graph and the user graph. SynDGN first obtains sentence embeddings and user feature embeddings via two graph encoders and then exerts a co-attention mechanism to fuse those embeddings. We extend SMOTE to synthesize representations for the minority class (i.e., rumors) to overcome the imbalance data issue. Experimental results confirm the effectiveness of the proposed SynDGN regardless the class distribution is balanced or not.

ACKNOWLEDGMENTS

This work is supported by the National Science and Technology Council (NSTC) of Taiwan under grants 110-2221-E-006-136-MY3, 112-2628-E-006-012-MY3, and 112-2634-F-002-006.

REFERENCES

[1] 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. *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (Apr. 2020), 549–556.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16 (June 2002), 321–357.

[3] Xiangyan Chen, Duoduo Zhu, Dazhen Lin, and Donglin Cao. 2021. Rumor knowledge embedding based data augmentation for imbalanced rumor detection. *Information Sciences* 580 (2021), 352–370.

[4] Yi-Chin Chen, Zhao-Yang Liu, and Hung-Yu Kao. 2017. IKM at SemEval-2017 Task 8: Convolutional Neural Networks for stance detection and rumor verification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. 465–469.

[5] Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. arXiv:1704.05972 [cs.CL]

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4171–4186.

[7] Omar Enayet and Samhaa R. El-Beltagy. 2017. NileTMRG at SemEval-2017 Task 8: Determining Rumour and Veracity Support for Rumours on Twitter.. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. 470–474.

[8] Shao-Ping Hsiao, Yu-Che Tsai, and Cheng-Te Li. 2022. Unsupervised Post-Time Fake Social Message Detection with Recommendation-Aware Representation Learning. In *Proceedings of the Web Conference 2022 (WWW '22)*. 232–235.

[9] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference for Learning Representations (ICLR)*.

[10] Yi-Ju Lu and Cheng-Te Li. 2020. GCAN: Graph-aware Co-Attention Networks for Explainable Fake News Detection on Social Media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 505–514.

[11] Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor Detection on Twitter with Tree-structured Recursive Neural Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 1980–1989.

[12] Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. 2020. FANG: Leveraging Social Context for Fake News Detection Using Graph Representation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM '20)*. 1165–1174.

[13] Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 22–32.

[14] Ismael Ramos-Pérez, Álvar Arnaiz-González, Juan J Rodríguez, and César García-Osorio. 2022. When is resampling beneficial for feature selection with imbalanced wide data? *Expert Systems with Applications* 188 (2022), 116015.

[15] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2931–2937.

[16] Qiang Sheng, Juan Cao, Xueyao Zhang, Rundong Li, Danding Wang, and Yongchun Zhu. 2022. Zoom Out and Observe: News Environment Perception for Fake News Detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. 4543–4556.

[17] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[18] Lingwei Wei, Dou Hu, Wei Zhou, Zhaojuan Yue, and Songlin Hu. 2021. Towards Propagation Uncertainty: Edge-enhanced Bayesian Graph Convolutional Networks for Rumor Detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*. 3845–3854.

[19] Lianwei Wu and Yuan Rao. 2020. Adaptive Interaction Fusion Networks for Fake News Detection. In *The 24th European Conference on Artificial Intelligence*.

[20] Lianwei Wu, Yuan Rao, Yongqiang Zhao, Hao Liang, and Ambreen Nazir. 2020. DTCA: Decision Tree-based Co-Attention Networks for Explainable Claim Verification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 1024–1035.

[21] Qiang Zhang, Aldo Lipani, Shangsong Liang, and E. Yilmaz. 2019. Reply-Aided Detection of Misinformation via Bayesian Deep Learning. *The World Wide Web Conference* (2019).

[22] Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based Evidence Aggregating and Reasoning for Fact Verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 892–901.

[23] Xinyi Zhou and Reza Zafarani. 2020. A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. *ACM Comput. Surv.* 53, 5, Article 109 (sep 2020).

[24] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads. *PLOS ONE* 11, 3 (2016).