Unsupervised Post-Time Fake Social Message Detection with Recommendation-aware Representation Learning

Shao-Ping Hsiao
National Cheng Kung University
Tainan, Taiwan
xxiao414@gmail.com

Yu-Che Tsai
National Taiwan University
Taipei, Taiwan
roytsai27@gmail.com

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

ABSTRACT

This paper deals with a more realistic scenario of fake message detection on social media, i.e., unsupervised post-time detection. Given a source message, our goal is to determine whether it is fake without using labeled data and without requiring user interacted with the given message. We present a novel learning framework, Recommendation-aware Message Representation (RecMR), to achieve the goal. The key idea is to learn user preferences and have them encoded into the representation of the source message through jointly training the tasks of user recommendation and binary detection. Experiments conducted on two real Twitter datasets exhibit the promising performance of RecMR, and show the effectiveness of recommended users in unsupervised detection.

CCS CONCEPTS

• **Information systems → Data mining**.

KEYWORDS

Unsupervised Learning, Fake News Detection, Social Media

ACM Reference Format:
Shao-Ping Hsiao, Yu-Che Tsai, and Cheng-Te Li. 2022. Unsupervised Post-Time Fake Social Message Detection with Recommendation-aware Representation Learning. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3487553.3524259

1 INTRODUCTION

Social media allows users to express themselves, share information, and interact with each other. Various messages, including disinformation, can spread among users [1]. Detecting fake messages is now a crucial task on social media, and a number of supervised learning-based methods had been proposed to deal with the task [20]. While fake messages can be effectively identified by transformer-based language models [8, 17], advanced supervised methods further resort to user comments [18], social networks [16], user attributes [13], and user-message interactions [10], to achieve better performance.

There are still two challenges in detecting fake social messages. First, existing studies (e.g., [4, 5, 10, 13, 16]) utilize user feedbacks to the source message, e.g., response comments, retweets, and endorsement, to learn features that can reflect the veracity. They assume that comment texts, message propagation, and retweeter profiles are available when training and inference. However, if one aims at determining whether a source message is fake or right at its posting, user feedbacks to the message are not generated yet. Second, supervised methods require significant amounts of labeled messages to train effective detection models. It is difficult to obtain high-quality fake and true labels on messages, and the labeling process is labor-intensive and time-consuming. Besides, as new messages on novel topics are always producing [1], it is impractical to instantly have the labeled data on new topics.

The realistic scenario is that we have new-created messages without any veracity labels, and we need to detect fake messages right at their postings, i.e., we cannot wait for user feedbacks. In this work, therefore, we propose to solve a novel but challenging task, *detecting fake messages in an unsupervised manner at their posting time, without using users interacted with the given source message.* Although recent studies have proposed unsupervised methods [5, 12, 19], they still require different user feedbacks, and cannot allow the detection of fake messages at the posting time.

To tackle the task, we present a novel framework, **Rec**ommendation-aware **M**essage **R**epresentation (RecMR). The idea is to find the potential participants or audience via learning user recommendation, and utilize the list of recommended users to enhance the representation learning of the given message. While recent studies have demonstrated the effectiveness of users who interacted with the source message [5, 10, 13], all of them require waiting for the appearance of interacted users. RecMR aims to simulate user interactions with the source message via recommending users at the posting time, instead of using interacted users. In other words, this paper investigates whether recommending users to a source message can benefit the detection of fake messages.

2 RELATED WORK

We summarize and compare the relevant studies on fake message detection in Table 1. Overall speaking, the present work differs from existing studies in three major aspects. First, while only few of past work [4, 5, 16] have leveraged user-message interactions (IM), none of them learn the user recommendation (Rec) for a new message. That said, whether the potential audience of the new-coming message can benefit the veracity classification is not investigated. Second, although existing methods do not rely on learning recommendation, the user propagation sequence and structure (PS) and the response comments (RC) of the given message is usually adopted. While a new message is created, it is infeasible to have PS and RC at the first time. Hence, for the purpose of early detection, we instead

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524259

Table 1: Comparison of related studies. Note: supervised (Sup.) and unsupervised (Unsup.), source message (SM), assumption (AS), response comments (RC), user attribute (UA), propagation structure (PS), graph representation learning (GL), interacted messages (IM), and user recommendation (Rec). For SM, "S" and "L" are short and long texts. For AS, a work assumes that fake "F" and true "T" labels of messages are available at training stage, verified "V" users can be used.

		SM	AS	RC	UA	PS	GL	IM	Rec
Sup.	RvNN [15]	✓(S)	F/T	✓		✓			
	FakeBERT [8]	✓(L)	F/T						
	dEFEND [18]	✓(L)	F/T	✓					
	GCAN [13]	✓(S)	F/T		✓	✓	✓		
	FANG [16]	✓(S)	F/T	✓	✓	✓	✓	✓	
	RDLNP [10]	✓(S)	F/T	✓		✓	✓		
	UPFD [4]	✓(L)	F/T	✓	✓	✓	✓	✓	
Unsup.	UFNDA [12]	✓(S)	T		✓				
	UFD [19]	✓(L)	V	✓	✓	✓			
	GTUT [5]	✓(L)				✓	✓	✓	
	This work	✓(S)	T		✓		✓	✓	✓

learn to recommend users of interest to the given message. Third, most of existing studies belong to supervised approaches, and few including ours are unsupervised manners [5, 12, 19]. Among them, our assumption is similar with UFNDA that only some true messages are available to train the unsupervised model. That said, this work aims to explore how user recommendation can benefit the unsupervised approach.

3 PROBLEM STATEMENT

Let $\mathcal{S} = \{s_1, s_2, \cdots, s_{|\mathcal{S}|}\}$ be a set of $|\mathcal{S}|$ social messages, and $\mathcal{U} = \{u_1, u_2, \cdots, u_{|\mathcal{U}|}\}$ be a set of $|\mathcal{U}|$ users. Each message $s_i \in \mathcal{S}$ is represented by a short-text document $s_i = \{w_1^i, w_2^i, \cdots, w_m^i\}$, where w_j^i denotes the j-th word in message s_i, and m is the number of words in s_i. Each user $u_j \in \mathcal{U}$ is associate with a user attribute vector $\mathbf{x}_j$. When a new message s_i is posted, we are allowed to collect users who react (e.g., endorse, comment, or retweet) to s_i after a while of the posting. However, these interacted users (e.g., retweeters) and messages (response comments) will not be used in our problem setting. Each message s_i is associated with a binary label $y_i \in \{0, 1\}$, which represents its truthfulness: $y_i = 0$ indicates the message s_i is true and $y_i = 1$ implies s_i is fake. We assume only real messages are accessible at the training stage while there could be real and fake at the testing stage. Moreover, each user $u_j \in \mathcal{U}$ contains a set of historical interacted messages $\mathcal{I} = \{\cdots, (u_j, s_k), \cdots\}$, where each (u_j, s_k) pair indicates user u_j had interacted with message s_k.

We utilize user metadata and profiles to define the user attribute vector $\mathbf{x}_j$ of every user u_j. The extracted features are listed as follows: (1) number of words in a user's self-description, (2) number of words in u_j's screen name, (3) number of users who follows u_j, (4) number of users that u_j is following, (5) number of created stories for u_j, and (6) whether u_j allows the geo-spatial positioning.

Unsupervised Post-Time Fake Message Detection. Given a collection of "real" social messages $\mathcal{S}$, along with the short-text document for each $s_i \in \mathcal{S}$, the user u_j who creates message s_i and her user feature vector $\mathbf{x}_j$, and the set of historical user-message interactions $\mathcal{I}$, the goal is to train an unsupervised detection model $\mathcal{F}(u_j, s_k, \mathcal{I}) \rightarrow \hat{y}_k$ that can accurately predict the binary veracity label $\hat{y}_k$ for a new-coming message $s_k \in \mathcal{S}'$ created by user u_j just at its posting, where $\mathcal{S}'$ is the set of new-coming messages.

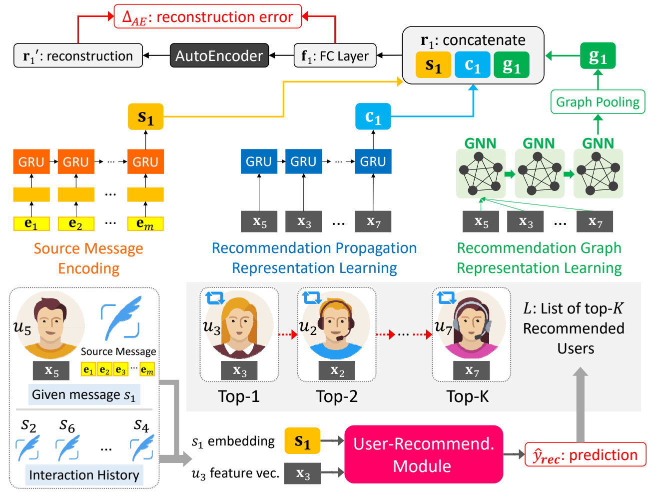

Figure 1: The overview of the proposed RecMR framework.

4 THE PROPOSED RECMR FRAMEWORK

The overview of RecMR is presented in Figure 1. RecMR is an end-to-end unsupervised fake message detection framework. Given the user-message interaction history, we first generate the message embeddings via GRU (Sec. 4.1), and have them together with user feature vectors fed into a user recommendation module (Sec. 4.2). A prediction probability will be produced, and is accordingly used to generate the list of top-K recommended users. We learn recommendation-aware representation from propagation and graph perspectives based on the feature vectors of recommended users (Sec. 4.3). Last, we can obtain a recommendation-enhanced representation by concatenating the derived embeddings, feed it into an Autoencoder model, and utilize the reconstruction error of source message to estimate its veracity (Sec. 4.4). We use only true messages for model training, and both true and fake can appear for model evaluation.

4.1 Source Message Encoding

The source message is represented by a word-level encoder. The input is the one-hot vector of each word in message s_i. Since the length of message is different, we perform zero padding by setting a maximum length m. Let $\mathbf{E} = [e_1, e_2, ..., e_m] \in \mathbb{R}^m$ be the input vector of source message, in which e_m is the one-hot encoding of the m-th word. We create a fully-connected layer to generate word embeddings, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m] \in \mathbb{R}^{d \times m}$, where d is the dimensionality. The derivation of $\mathbf{V}$ is given by: $\mathbf{V} = \tanh(\mathbf{W}_w \mathbf{E} + \mathbf{b}_w)$, where $\mathbf{W}_w$ is the matrix of learnable weights, and $\mathbf{b}_w$ is the bias term. Then, we utilize Gating Recurrent Units (GRU) [2] to learn the word sequence representation from $\mathbf{V}$ by taking the output hidden vector. The embedding of source message s_i can be depicted by: $\mathbf{s}_i = GRU(\mathbf{v}_t)$, $t \in \{1, ..., m\}$, where m is the GRU dimensionality.

4.2 User Recommendation

We learn to recommend the potential users to the given source message s_i. Given a user-message pair (u_j, s_k) with the corresponding user feature vector $\mathbf{x}_j$ and message embedding $\mathbf{s}_k$, the purpose of user recommendation is to train to generate a prediction probability $\hat{y}_{rec}$ that indicates the preference and potential of interacting with message s_k by user u_j. We treat the user recommendation as a binary classification task, i.e., predicting whether u_j will interact with s_k. By feeding the vector that concatenates $\mathbf{x}_j$ with $\mathbf{s}_k$ into a certain recommendation module, we can produce a list of recommended users based on the prediction probability $\hat{y}_{rec}$. The user recommendation part can be depicted via: $\hat{y}_{rec} = UserRec([\mathbf{x}_j, \mathbf{s}_k])$. To implement the recommendation module, here we utilize a typical method, DeepFM [6] with its default model setting, to be $UserRec$. Negative sampling is employed to produce non-interacted messages for DeepFM training. We use cross entropy to be the loss function of user recommendation $\mathcal{L}_{rec}$.

4.3 Recommendation-aware Representation

With the user recommendation module, we can obtain a list of top-K recommended users, $L = (\bar{u}_1, \bar{u}_2, ..., \bar{u}_K)$, whose order is determined by the generated probability $\hat{y}_{rec}$ in descending order. We utilize L to simulate how users interact with the source message s_i. Recommended users with higher $\hat{y}_{rec}$ are considered to have higher potential to interact with s_i shortly after its posting. While $\hat{y}_{rec}$ also reflects user preference on s_i, it is natural to take users with higher preference scores to aid the detection of fake messages. To incorporate the recommendation outcomes into the representation learning of source message, below we present two methods: *propagation representation* and *graph representation.*

Propagation Representation regards the list of recommended users as the propagation sequence originated from the user who posts the source message. Given the recommended list L, we use GRU [2] as a sequential model to learn the propagation representation. The output hidden vector, given by $\mathbf{c}_i = GRU(\mathbf{x}_t)$, $t \in \{1, ..., K\}$ and $\mathbf{x}_t$ is the feature vector of recommended user $\bar{u}_t$, is the derived propagation representation for source message s_i.

Graph Representation considers that the recommended users can interact with each other (e.g., mention one another, form subthreads) to discuss the source s_i, rather than only reacting to s_i sequentially. We can create a graph between recommended users in L and leverage graph representation learning to capture the latent interactions. We build a fully-connected graph $G = (L, A)$, in which nodes are K users in L, and edges connect all users with weights: $A = [a_{ij}] \in \mathbb{R}^{K\times K}$ and $a_{ij} = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|}$. To implement the graph representation learning, here we adopt typical graph neural networks, graph convolutional network (GCN) [9], to generate the embeddings of user nodes, and utilize self-attention graph pooling (SAGPool) [11] with its default settings to pool user embeddings into the final graph representation for source s_i: $\mathbf{g}_i = SAGPool(GCN(G))$.

4.4 Unsupervised Detection Training

The final recommendation-enhanced representation of source s_i, denoted by $\mathbf{r}_i$, is the concatenation of the derived embeddings from text encoding $\mathbf{s}_i$, propagation representation $\mathbf{c}_i$, and graph representation $\mathbf{g}_i$, i.e., $\mathbf{r}_i = [\mathbf{s}_i, \mathbf{c}_i, \mathbf{g}_i]$. We train the unsupervised model based on Autoencoder. An autoencoder is learned to reconstruct the input true message's embedding $\mathbf{s}_i$ using its recommendation-enhanced embedding $\mathbf{r}_i$. The reconstruction error Δ_i is treated as the score to estimate the veracity of the source message at the testing stage. Messages with higher error Δ_i indicate higher potential of being fake, and thus will be ranked at top positions.

Specifically, the encoder of autoencoder is given by $\mathbf{f}_i = \sigma(\mathbf{W}_1\mathbf{r}_i)$. The decoder can be depicted by $\mathbf{r}'_i = \mathbf{W}_3(GRU(RV(\mathbf{W}_2\mathbf{f}_i)))$, where σ is ReLU activation function, RV is the Repeat Vector that repeats the incoming inputs a specific number of times, and $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{W}_3$ are learnable weights. We use mean squared error (MSE) to be the reconstruction error $\Delta_i = MSE(\mathbf{r}_i, \mathbf{r}'_i)$. The training of autoencoder is to minimize MSE, which is the loss function $\mathcal{L}_{ae}$. At last, we train the entire detection model to jointly recommend users and utilize them to reconstruct message embedding. The final loss function is $\mathcal{L} = \mathcal{L}_{rec} + \alpha\mathcal{L}_{ae}$, where the hyperparameter α balances two tasks ($\alpha = 1$ by default). We use Adam as the optimizer.

5 EXPERIMENTS

Datasets. Two well-known datasets, Twitter15 and Twitter16 [14], are used. Each dataset contains a set of messages, along with their sequences of retweet users. We choose only "true" and "fake" labels as the ground truth. We use user IDs to crawl user attributes via Twitter API. The numbers of messages (Real:Fake) are 742 (370:372) and 412 (205:207), and the numbers of users are 198,799 and 135,837, for Twitter-15 and Twitter-16, respectively.

Settings. For unsupervised detection task, we randomly select 80% "true" messages for training, and the remaining 20% "true" messages together with the same number of "fake" messages for testing. A random 10% of training set is used as the validation set. For user recommendation task, we will generate 5 negative users for each positive user-message interaction. The evaluation metrics include Precision, Recall, and F1 at Q% of fake messages ($Q = 30, 60$). The experiments are repeated 30 times, and the average scores are reported. For hyperparameter settings, the learning rate is searched in $\{0.1, 0.01, 0.001\}$, 2-layer GCN, the dimensionality of all embeddings is $d = 32$, the epoch number is fixed as 50, the pooling rate in SAGPool is 0.5. Besides, we vary the number of $K = \{10, 30, 50\}$ recommended users in the experiments.

Competing Methods. We have four variants of RecMR to answer the three evaluation questions. (1) The full model (RecMR), (2) the model without graph representation learning (RecMR-R), (3) the model using ground-truth users (i.e., real retweeters) interacted with source messages (RecMR/GT), and (4) the model using random recommended users (RecMR/RD). UFNDA [12] is the state-of-the-art baseline on unsupervised detection since it also uses autoencoder and true messages for training.

Results. The results are presented in Table 2 and Table 3. We can obtain four insights. First, the proposed full RecMR consistently outperforms the strong baseline UFNDA and all of its variants across all settings. Such results verify the promising usefulness of effectively learning user recommendation for unsupervised fake message detection. We can simulate user-message interactions to better identify fake messages just when they get posted. Second, it is interesting to find that the performance of ground-truth users who interact with the source messages (i.e., RecMR/GT) are not as good as the recommended users, and close to random users (i.e.,

Table 2: Experimental results on Twitter15 data. The best-performed method for each K is highlighted in bold.

		Precision		Recall		F1	
		@30%	@60%	@30%	@60%	@30%	@60%
	UFNDA (no users)	0.7249±0.0400	0.6895±0.0451	0.3575±0.0197	0.4534±0.0296	0.4788±0.0264	0.5471±0.0358
K=10	RecMR/GT	0.7206±0.0410	0.6992±0.0471	0.3554±0.0202	0.4598±0.0310	0.4760±0.0271	0.5548±0.0374
	RecMR/RD	0.7220±0.0414	0.7008±0.0435	0.3561±0.0204	0.4608±0.0286	0.4769±0.0273	0.5560±0.0345
	RecMR-G	0.7393±0.0456	0.7127±0.0565	0.3646±0.0225	0.4686±0.0371	0.4883±0.0301	0.5655±0.0448
	RecMR	**0.7483±0.0376**	**0.7202±0.0535**	**0.3690±0.0186**	**0.4836±0.0352**	**0.4943±0.0249**	**0.5786±0.0424**
K=30	RecMR/GT	0.7348±0.0263	0.7039±0.0393	0.3624±0.0129	0.4628±0.0259	0.4854±0.0173	0.5585±0.0312
	RecMR/RD	0.7272±0.0245	0.7006±0.0418	0.3506±0.0121	0.4607±0.0275	0.4704±0.0162	0.5559±0.0331
	RecMR-G	0.7351±0.0329	0.7142±0.0429	0.3625±0.0162	0.4696±0.0282	0.4856±0.0217	0.5667±0.0340
	RecMR	**0.7454±0.0385**	**0.7211±0.0404**	**0.3761±0.0170**	**0.4742±0.0332**	**0.5000±0.0228**	**0.5721±0.0300**
K=50	RecMR/GT	0.7257±0.0363	0.6957±0.0427	0.3579±0.0179	0.4574±0.0280	0.4794±0.0240	0.5520±0.0338
	RecMR/RD	0.7213±0.0377	0.6945±0.0435	0.3557±0.0186	0.4567±0.0286	0.4765±0.0249	0.5510±0.0345
	RecMR-G	0.7346±0.0420	0.7103±0.0499	0.3623±0.0207	0.4670±0.0328	0.4853±0.0277	0.5635±0.0396
	RecMR	**0.7406±0.0290**	**0.7183±0.0437**	**0.3717±0.0143**	**0.4723±0.0311**	**0.4945±0.0192**	**0.5699±0.0375**

Table 3: Experimental results on Twitter16 data. The best-performed method for each K is highlighted in bold.

		Precision		Recall		F1	
		@30%	@60%	@30%	@60%	@30%	@60%
	UFNDA (no users)	0.8125±0.0479	0.7642±0.0509	0.3963±0.0234	0.4846±0.0323	0.5328±0.0314	0.5931±0.0395
K=10	RecMR/GT	0.8131±0.0566	0.7597±0.0484	0.3965±0.0276	0.4818±0.0307	0.5331±0.0371	0.5897±0.0375
	RecMR/RD	0.8077±0.0555	0.7540±0.0503	0.3940±0.0271	0.4781±0.0319	0.5296±0.0364	0.5852±0.0391
	RecMR-G	0.8305±0.0518	0.7949±0.0595	0.4051±0.0252	0.5041±0.0377	0.5446±0.0339	0.6169±0.0461
	RecMR	**0.8508±0.0468**	**0.8047±0.0511**	**0.4150±0.0228**	**0.5103±0.0324**	**0.5579±0.0307**	**0.6246±0.0397**
K=30	GT Users	0.8140±0.0527	0.7615±0.0558	0.3971±0.0257	0.4829±0.0354	0.5338±0.0346	0.5910±0.0433
	RecMR/RD	0.8155±0.0587	0.7583±0.0582	0.3978±0.0286	0.4809±0.0369	0.5348±0.0385	0.5886±0.0452
	RecMR-G	0.8277±0.0782	0.7847±0.0734	0.4037±0.0381	0.4976±0.0466	0.5427±0.0513	0.6091±0.0570
	RecMR	**0.8428±0.0543**	**0.7951±0.0619**	**0.4111±0.0265**	**0.5042±0.0392**	**0.5527±0.0356**	**0.6171±0.0480**
K=50	RecMR/GT	0.8047±0.0591	0.7551±0.0604	0.3925±0.0288	0.4789±0.0383	0.5277±0.0387	0.5861±0.0469
	RecMR/RD	0.8065±0.0657	0.7546±0.0638	0.3934±0.0320	0.4785±0.0404	0.5289±0.0431	0.5857±0.0495
	RecMR-G	0.8288±0.0658	0.7928±0.0546	0.4043±0.0321	0.5028±0.0346	0.5435±0.0432	0.6153±0.0424
	RecMR	**0.8403±0.0514**	**0.7956±0.0589**	**0.4099±0.0251**	**0.5046±0.0374**	**0.5510±0.0337**	**0.6175±0.0457**

RecMR/RD). This finding informs us that modeling user preference on the source message based on text content and user attributes can collect more and stronger evidence of veracity. Third, removing graph representation (i.e., RecMR-G) slightly hurts the performance. This shows the usefulness of modeling how users interact with each other to discuss the source message.

6 CONCLUSIONS AND DISCUSSION

This work finds that user recommendation can benefit fake message detection on social media in unsupervised learning. We present RecMR to achieve the goal, and empirically find that modeling user preferences (i.e., simulating user interactions) on source message can bring performance improvement without using any interacted users for unsupervised fake message detection. The current RecMR does not utilize fancy models to be its implementation modules. We can use pre-trained BERT [3] for text encoding and LightGCN [7] for recommendation. RecMR is now built upon the unsupervised setting, it can be extended to the supervised scenario by replacing autoencoder with some neural network prediction layers.

ACKNOWLEDGMENTS

This work is supported by the Ministry of Science and Technology (MOST) of Taiwan under grants 110-2221-E-006-136-MY3, 110-2221-E-006-001, and 110-2634-F-002-051.

REFERENCES

[1] Meeyoung Cha, Wei Gao, and Cheng-Te Li. 2020. Detecting Fake News in Social Media: An Asia-Pacific Perspective. *Commun. ACM* 63, 4 (2020).

[2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *NIPS Workshop on Deep Learning*.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*.

[4] Yingtong Dou, Kai Shu, Congying Xia, Philip S. Yu, and Lichao Sun. 2021. User Preference-Aware Fake News Detection. In *Proceedings of ACM SIGIR*.

[5] Siva Charan Reddy Gangireddy, Deepak P, Cheng Long, and Tanmoy Chakraborty. 2020. Unsupervised Fake News Detection: A Graph-Based Approach. In *Proceedings of ACM Hypertext and Social Media*.

[6] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine Based Neural Network for CTR Prediction. In *Proceedings of IJCAI*.

[7] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of ACM SIGIR*.

[8] Rohit Kumar Kaliyar, Anurag Goswami, and Pratik Narang. 2021. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimedia Tools and Applications* 80, 8 (2021).

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

[10] An Lao, Chongyang Shi, and Yayi Yang. 2021. Rumor Detection with Field of Linear and Non-Linear Propagation. In *Proceedings of the Web Conference*.

[11] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-Attention Graph Pooling. In *Proceedings of ICML*.

[12] Dun Li, Haimei Guo, Zhenfei Wang, and Zhiyun Zheng. 2021. Unsupervised Fake News Detection Based on Autoencoder. *IEEE Access* 9 (2021).

[13] 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 ACL*.

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

[15] Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor detection on twitter with tree-structured recursive neural networks. In *Proceedings of ACL*.

[16] 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 ACM CIKM*.

[17] Kellin Pelrine, Jacob Danovitch, and Reihaneh Rabbany. 2021. The Surprising Performance of Simple Baselines for Misinformation Detection. In *Proceedings of the Web Conference*.

[18] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. DEFEND: Explainable Fake News Detection. In *Proceedings of ACM KDD*.

[19] Shuo Yang, Kai Shu, Suhang Wang, Renjie Gu, Fan Wu, and Huan Liu. 2019. Unsupervised Fake News Detection on Social Media: A Generative Approach. In *Proceedings of AAAI*.

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