1-2024

Predicting viral rumors and vulnerable users with graph-based neural multi-task learning for infodemic surveillance

Xuan ZHANG
Singapore Management University, xuanzhang.2020@phdcs.smu.edu.sg

Wei GAO
Singapore Management University, weigao@smu.edu.sg

Predicting viral rumors and vulnerable users with graph-based neural multi-task learning for infodemic surveillance

Xuan Zhang *, Wei Gao

a School of Computing and Information Systems, Singapore Management University, Singapore

*Corresponding author

Published in Information Processing and Management, 2024, 61, 103520. DOI: 10.1016/j.ipm.2023.103520

Abstract: In the age of the infodemic, it is crucial to have tools for effectively monitoring the spread of rampant rumors that can quickly go viral, as well as identifying vulnerable users who may be more susceptible to spreading such misinformation. This proactive approach allows for timely preventive measures to be taken, mitigating the negative impact of false information on society. We propose a novel approach to predict viral rumors and vulnerable users using a unified graph neural network model. We pre-train network-based user embeddings and leverage a cross-attention mechanism between users and posts, together with a community-enhanced vulnerability propagation (CVP) method to improve user and propagation graph representations. Furthermore, we employ two multi-task training strategies to mitigate negative transfer effects among tasks in different settings, enhancing the overall performance of our approach. We also construct two datasets with ground-truth annotations on information virality and user vulnerability in rumor and non-rumor events, which are automatically derived from existing rumor detection datasets. Extensive evaluation results of our joint learning model confirm its superiority over strong baselines in all three tasks: rumor detection, virality prediction, and user vulnerability scoring. For instance, compared to the best baselines based on the Weibo dataset, our model makes 3.8% and 3.0% improvements on Accuracy and MacF1 for rumor detection, and reduces mean squared error (MSE) by 23.9% and 16.5% for virality prediction and user vulnerability scoring, respectively. Our findings suggest that our approach effectively captures the correlation between rumor virality and user vulnerability, leveraging this information to improve prediction performance and provide a valuable tool for infodemic surveillance.

Keywords: Infodemic surveillance, Neural multi-task learning, Rumor detection, User vulnerability, Virality prediction

1. Introduction

Online rumors refer to unverified information circulating on the Internet, especially on social media. Rumors with misinformation that distort facts can cause false beliefs and unnecessary panic in public. Those viral rumors, specifically, can result in incredibly harmful repercussions due to their extensive reachability, evidenced as various man-made tragedies sparked by the rumors about COVID-19 during the latest pandemic (Ali, 2020, Islam et al., 2021).

Automatic rumor detection based on machine learning methods is an active research topic in recent years (Bian et al., 2020, Ma et al., 2016, Ma et al., 2017, Ma et al., 2018, Wang et al., 2020). However, the potential for online rumor surveillance is limited by the cost-effectiveness of monitoring all identified rumors and participants indiscriminately, while not all rumors and participants are equally important in surveillance. For example, a widely spreading rumor can have a greater range of impact and tends to be more convincing to users since high virality can serve as a mental shortcut to make a judgment by evoking

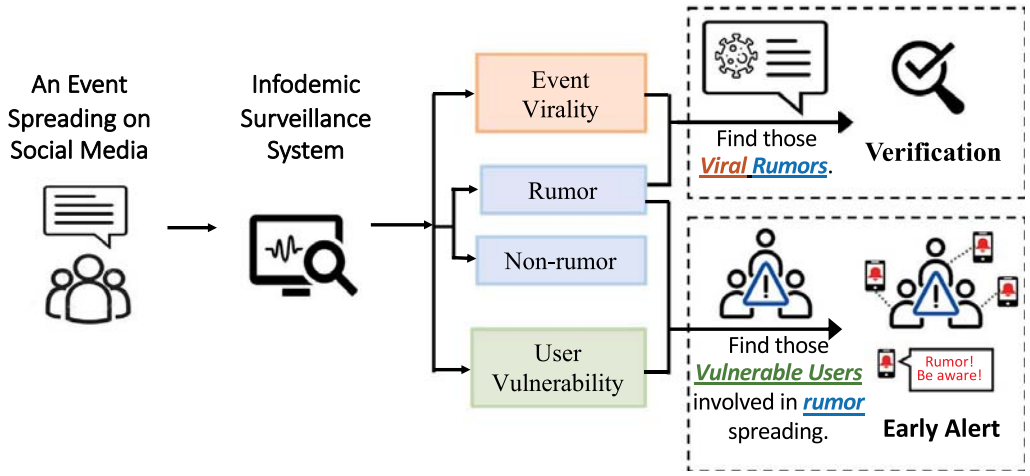

Fig. 1. An application scenario of the infodemic surveillance system that can predict viral rumors and vulnerable users.

a greater perception of social norms (Kim, 2018; Lee & Oh, 2017); meanwhile, a rumor can affect individual recipients differently, depending on how gullible each recipient is Mercier (2017), which may in turn influence how broadly and deeply the rumor diffuses. By accurately estimating information **virality** as well as user **vulnerability** in the context of rumor detection, we can expect to track noteworthy rumors and users more cost-effectively on various social media platforms.

However, such needs currently exceed the capacity of various existing rumor detection tasks as they generally overlook the assessment of the potential reach of rumors and the hazard of their spreaders. Imagine if there were a surveillance system that combined rumor detection, virality prediction, and vulnerability prediction as illustrated in Fig. 1, with such a system, stakeholders would be able to send alert to those at-risk vulnerable users and request the potentially viral rumors to be professionally verified in a timely manner, resulting in more effective information surveillance and early intervention. To meet this goal, it is necessary to bridge a few major gaps.

Previous information diffusion research has investigated the virality prediction of general information. For example, Cheng, Adamic, Dow, Kleinberg, and Leskovec (2014) combined various temporal and structural features with machine learning models to estimate how large an information propagation graph of a source message can reach in the future. Li, Ma, Guo, and Mei (2017) used neural network models to learn feature representations that characterize virality for a similar purpose. However, such general virality models do not consider the distinction between rumors and normal information, and the influence of gullible users on diffusion, who are more likely to be activated by rumors.

Among misinformation related studies, the concept of user vulnerability (i.e., gullibility) is proposed to measure the propensity that motivates a user to participate in spreading rumor or fake news. For example, Shen et al. (2019) attempt to predict user vulnerability to fake news using a classifier that considers features of post content, user status (e.g., number of followers and friends), and network connectivity. However, these works do not further link user vulnerability to rumor's virality. In social network analysis, the susceptibility of users to item adoption is measured along with topology intrinsic factors (Hoang & Lim, 2016), such as the ability of users to spread items and the potential of users to adopt items, which contribute to the wide dissemination of items in the network. Although this may help analyze the relationship between information virality and the strength of users to spread (or adopt) information from a topology perspective, rumor virality is arguably not only related to the network's topological structure but also to the content of the information being spread and user's cognitive state.

Based on public rumor detection datasets, such as TWITTER[1] (Ma et al., 2017), which typically consist of a set of propagation cascades of source posts labeled as rumor and non-rumor, we observe that the virality (i.e., the number of users participating in the spread) can be in some extent related to the interaction between the vulnerability of the involved users (i.e., user's propensity of engaging in rumor spreading)[2] and whether the information propagated is a rumor or a non-rumor. Fig. 2 presents four illustrative examples from the TWITTER dataset. Overall, most users attracted by non-rumors are typically much less vulnerable than in rumors. Notably, rumors tend to spread more virally when the involved users are overall more vulnerable as shown in (a) vs (b). Meanwhile, non-rumors tend to be more viral when the participating users are generally less vulnerable, as illustrated by the distinction between (c) and (d). According to the above observations, it seems that some implicit links exist among the user vulnerability, the virality and the rumor/non-rumor nature of concerned information. If this held true, it would be likely for us to harness their relationship to facilitate three prediction tasks concerning rumor detection, information virality prediction, and user vulnerability estimation altogether. One might typically assume that virality is irrelevant to whether the concerned information is a rumor or not because both rumors and non-rumors could be viral (or not viral). However, we argue that given user vulnerability as a bridge,

[1] We combine the Twitter15 and Twitter16 datasets released by Ma et al. (2017) into one larger dataset, namely TWITTER.
[2] The user's vulnerability is estimated as the proportion of rumors among all the information, in which the user is engaged.

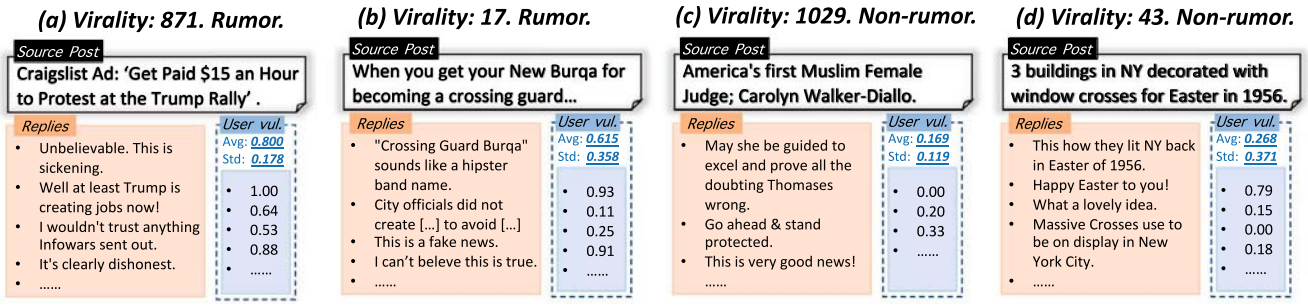

Fig. 2. Example rumors and non-rumors of different virality and reposts to them by users of different vulnerabilities, taken from TWITTER dataset (Ma et al., 2017). Virality is defined as the number of users involved in the spread, and user vulnerability is defined as the fraction of rumor events over all events a user engaged in.

the relationship between the virality and the rumor/non-rumor nature of information could be established implicitly and validated with the performance of the relevant prediction tasks.

To verify our presumption, we propose a unified multi-task learning framework based on Graph Neural Networks (GNN) and hierarchical graph pooling methods for joint learning of rumor detection, virality prediction, and user vulnerability scoring. The framework aims to improve the performance of the three seemingly independent tasks simultaneously to aid information surveillance and early intervention. In particular, we first transform information spreading networks into user interaction networks, which are used to train user embeddings using a network-based approach. The user embeddings are further enhanced by incorporating post content using a user-post cross-attention mechanism. Then we use DiffPool (Ying et al., 2018) to discover latent communities of users that exhibit similar behavior in spreading rumors, and use a Community-enhanced Vulnerability Propagation (CVP) method with reference to the latent communities to refine user embeddings. To mitigate negative transfer among the tasks, we adopt two radically different multi-task training methods, Gradnorm (Chen, Badrinarayanan, Lee, & Rabinovich, 2018) and meta-learning (Buffelli & Vandin, 2020) over the three tasks. Our main contributions can be summarized as follows:

- To our best knowledge, this is the first study on rumor detection, virality prediction, and user vulnerability scoring in a unified framework for infodemic surveillance.
- We use inductive GNNs and hierarchical graph pooling to learn the three tasks jointly. In particular, we pre-train general user embeddings and propose a CVP method to further refine user embeddings
- We train the framework under two different multi-task settings, namely concurrent training with GradNorm and meta-learning, for dealing with task training conflicts, i.e., negative transfer.
- We build two large datasets with virality and user vulnerability annotated based on existing rumor detection datasets, on which our method outperforms strong baselines for all three tasks.[3]

2. Related work

Our study in this paper intersects multiple research topics including rumor detection, information virality prediction and user vulnerability analysis under multi-task learning framework, which are reviewed separately in this section.

2.1. Rumor detection

Several surveys (Cao et al., 2018; Sharma et al., 2019; Xu, Sheng, & Wang, 2021; Zannettou, Sirivianos, Blackburn, & Kourtellis, 2019; Zubiaga, Aker, Bontcheva, Liakata, & Procter, 2018) have comprehensively reviewed the literature on rumor detection. Here we introduce some works most relevant to us. Many studies on rumor detection mainly extract discriminative features from post content, user profiles, and propagation patterns, and learn supervised classifiers for classification (Castillo, Mendoza, & Poblete, 2011; Ma et al., 2017; Zhao, Resnick, & Mei, 2015). Later, representation-learning methods become a dominant approach by exploiting end-to-end neural network models to learn latent representations for classification (Alkhodair, Ding, Fung, & Liu, 2020; Bi, Wang, Zhang, & Gao, 2022; Bian et al., 2020; Dou, Shu, Xia, Yu, & Sun, 2021; Liu & Wu, 2020; Ma et al., 2018; Nguyen, Sugiyama, Nakov, & Kan, 2020; Sun, Zhang, Zheng, & Ma, 2022). For example, Bian et al. (2020) use a Bi-directional Graph Convolutional Network (Bi-GCN) to capture the diffusion and dispersion properties from propagation trees. Sun et al. (2022) propose a Dual-Dynamic Graph Convolutional Networks (DDGCN) model to learn the dynamics of messages in propagation and the dynamics of background knowledge from Knowledge graphs simultaneously. However, given the multitude of rumors online, it is not cost-effective nor necessary to track all the rumors detected. Differing from existing rumor detection methods which aim to distinguish rumors from non-rumors, we allow the systems to be able to distinguish the virality of rumors and non-rumors in terms of the number of *unique* users that they may reach for more cost-effective surveillance.

[3] Data and code are available at https://github.com/jadeCurl/Predicting-Viral-Rumors-and-Vulnerable-Users

2.2. Information virality prediction

To the best of our knowledge, there is little work focusing on predicting the virality of rumors. However, general studies on the virality of information diffusion are abundant (Chen, Zhou, Zhang, & Bonsangue, 2021; Kefato et al., 2018; Li et al., 2017; Tan, Liu, Li, Zhang, & Zhang, 2022; Yu, Cui, Wang, Song, & Yang, 2015; Zhang et al., 2021; Zhao, Erdogdu, He, Rajaraman, & Leskovec, 2015). Existing approaches are commonly based on feature engineering using original post, sharer, and community structure (Cheng et al., 2014; Jenders, Kasneci, & Naumann, 2013; Weng, Menczer, & Ahn, 2014). Based on temporal and structural features, generative models were also designed (Shen, Wang, Song, & Barabási, 2014; Yu et al., 2015). For example, Shen et al. (2014) modeled the information diffusion as a reinforced Poisson process. Those generative approaches made strong assumptions over macroscopic distributions and stochastic processes, which limits their application in practice. For better feature representation, deep neural networks were utilized with promising results for virality prediction (Kefato et al., 2018; Li et al., 2017; Tan et al., 2022; Zhang et al., 2021).

However, these general virality prediction works do not take into account the interaction between vulnerability of involved users and whether the information propagated is a rumor or a non-rumor. To the best of our knowledge, there is no existing research specifically studying the difference between the virality of rumors and that of non-rumor information. In the fake news domain, some works found that fake news spreads faster and wider than true news in general (Vosoughi, Roy, & Aral, 2018). However, no work has been attempted to predict the extent of rumor's propagation directly, and it remains unclear how such extent differs between rumor and non-rumor. Our study does not discriminately predict the virality of rumor and non-rumor either. Instead, our prediction of viral rumors is achieved indirectly by cross-referencing the outputs of the rumor detection task and the virality prediction task, which not only share common features at the graph level, but also have the user vulnerability task as the bridge. As a result, we can better predict the virality of information if the information type (i.e., rumor or non-rumor) and the vulnerability traits of participating users are known. This is useful from an information surveillance point of view because the joint prediction allows us to keep track on those widely circulated rumors and vulnerable users at same time.

2.3. User vulnerability analysis

Existing works have been focused on studying users' susceptibility to various information sources in social media and social networks (Albladi & Weir, 2020; Hoang & Lim, 2016; Lee & Lim, 2015; Wald, Khoshgoftaar, Napolitano, & Sumner, 2013). There have been few previous analyses on the vulnerability of users to rumors and fake news (Bringula, Catacutan-Bangit, Garcia, Gonzales, & Valderama, 2021; Pennycook & Rand, 2019; Rath, Gao, & Srivastava, 2019). Rath et al. (2019) proposed a community health assessment model based on the concept of believability derived from computational trust metrics to calculate the vulnerability of nodes and communities to fake news spread. Bringula et al. (2021) found that technological, internal, and external factors may positively or negatively affect university students' vulnerability to political misinformation. In this paper, we use a unified neural model to learn features representing user vulnerability to rumors and further link user vulnerability to the virality of information, that can help rumor detection and the estimation of extent of rumors' potential reach to network users.

2.4. Graph neural networks (GNNs)

Recently, Graph Convolutional Networks (GCNs) based on matrix factorization have demonstrated state-of-the-art performance in various graph-related tasks (He, Xu, Jameel, Wang, & Chen, 2022; Sun et al., 2022; Veličković et al., 2018; Wang, Xie, Wang, Lee, & Wei, 2023; Warmsley, Waagen, Xu, Liu, & Tong, 2022; Zhai, Yang, & Zhang, 2023; Zhang, Wu, Zhang, Ma, & Yu, 2023). These transductive GCNs work well on fixed graphs, but are difficult to apply to unseen nodes that do not appear at training time (Defferrard, Bresson, & Vandergheynst, 2016). To overcome this weakness, Hamilton, Ying, and Leskovec (2017) proposed an inductive GraphSAGE model which aggregates node features from neighbors. Different kinds of hierarchical pooling processes can be added on top of the GNN models to capture the deeper information from hierarchical graph structures, for strengthening the learned node embedding (Zhang et al., 2023). For example, gPool (Gao, Chen, & Ji, 2019) and SAGPoo (Lee, Lee, & Kang, 2019) select top-K nodes to form induced subgraphs, which may lose useful graph structure and node information. EdgePool (Diehl, 2019) reduces edges in the graph to obtain pooled subgraphs by fixing the number of nodes in the pooled graph to half of the original. The above methods can learn a pooled smaller graph by actively dropping some node and edge information. In contrast, DiffPool (Ying et al., 2018) map the nodes into a set of communities via soft assignments. In this paper, we use GraphSAGE and DiffPool to build our multi-task learning framework.

3. Problem definition

The spread of rumors, e.g., "Bat soup caused Wuhan virus", on social media platforms can cause significant harm. Disseminating such false information may arouse misperception, hatred, and even ethnic conflict. We aim to facilitate the monitoring and early warning of the rampant spread of rumors by utilizing initial propagation information. Specifically, we focus on a series of prediction objectives: (1) differentiate rumors and non-rumors given an early (thus partial) observation of information propagation on social media platforms, (2) predict the scale of propagation of a concerned message given the observation of propagation at initial stages, (3) identify those vulnerable users who appear in the observed propagation and are more likely to disseminate such rumors, and

assign them a vulnerability score, and (4) strengthen the performance of joint prediction on rumor, virality, and user vulnerability through improved representations of task-relevant features.

To address these issues, we define our research problems as follows. Given a rumor dataset, each instance is given as a propagation network corresponding to a specific social event (or claim), which consists of a source post and its cascading messages spreading the event via reposting (e.g., replying and retweeting) behaviors (Bian et al., 2020; Song, Chen, Chang, Weng, & Shuai, 2021). We define an entire post propagation network as graph G, in which the timestamp of the last post is T. Additionally, given the predictive nature of our task, which requires information from the early stages of post propagation, we introduce an observed (i.e., partial) propagation network $\mathcal{G} \subset \mathrm{G}$, where $\mathcal{G} = \{\mathbf{V}, \mathbf{A}\}$ comprises a set of nodes and edges:

- $\mathbf{V} = \{v_1, v_2, \ldots, v_{|\mathbf{V}|}\}$ is the node set corresponding to the posts engaging in the propagation. Each $v_i = (s_i, t_i, u_j)$ indicates that it is the jth user $u_j \in \mathbf{U}$ $(1 \le j \le |\mathbf{U}|)$ who creates the ith post content s_i $(1 \le i \le |\mathbf{V}|)$ at timestamp t_i $(t_i \in [0, t]$, where t is the size of observation time window for $\mathcal{G}$ and $t < T)$, and $\mathbf{U} = \{u_1, u_2, \ldots, u_{|\mathbf{U}|}\}$ is the set of unique users that appear in $\mathcal{G}$.
- $\mathbf{A}^{|\mathbf{V}|\times|\mathbf{V}|}$ is the adjacency matrix indicating the reposting relationships among the posts. Each entry in the matrix is a binary value (0 or 1) that indicates whether there is a reposting relationship between two corresponding posts. Without the loss of generality, we omit the direction of message propagation, i.e., $\mathbf{A}$ is symmetric.

We then formulate three prediction tasks as below, corresponding to our first three research objectives:

1. *Rumor Detection:* Given an observed propagation network $\mathcal{G} \subset \mathrm{G}$, we formulate this task as a two-way graph classification problem of predicting whether the final propagation network G will be a rumor or not, namely the task $\mathcal{T}_1 : \mathcal{G} \to y(\mathrm{G}) \in \{\text{rumor}, \text{non-rumor}\}$, where $y(\mathrm{G})$ is the ground-truth class of G.
2. *Virality Prediction:* Given an observed propagation network $\mathcal{G} \subset \mathrm{G}$, we define this task as a regression problem of forecasting the total number of unique users participating in spreading the entire event G, denoted as $\mathcal{T}_2 : \mathcal{G} \to \log_2 |\mathbf{U}_\mathrm{G}|$, where $\mathbf{U}_\mathrm{G}$ represents the set of unique users that appear in G. Following previous work (Kupavskii et al., 2012; Li et al., 2017; Tsur & Rappoport, 2012), the model will fit the logarithm of ground-truth $|\mathbf{U}_\mathrm{G}|$ by squashing the absolute size of network.
3. *User Vulnerability Scoring:* Given an observed propagation network $\mathcal{G} \subset \mathrm{G}$, we try to infer the vulnerability of each unique user $u_j \in \mathcal{G}$ as how much it is susceptible to rumor spreading, which is denoted as the task $\mathcal{T}_3 : u_j \to [0, 1]$. The ground-truth user vulnerability is defined as the fraction of rumors over all the information in which the user is engaged.

Furthermore, in order to address the fourth research objective, we propose a unified GNN model and adopt multi-task learning frameworks to jointly train the model for the three tasks above. To mitigate training conflicts among these tasks, we leverage the underlying correlations that might suggest their mutual predictability, which allows us to effectively integrate and optimize the prediction of rumor, virality, and user vulnerability within a single model.

4. Methodology

As shown in Fig. 3, our whole framework consists of four components: (i) user interaction graph construction; (ii) input embedding; (iii) refined embedding; and (iv) output layers.

4.1. User interaction graph construction

To understand the patterns of how users spread rumors and non-rumors, predict the likelihood of a source post going viral, and identify users who are more vulnerable to spreading misinformation, we first need to find the users involved in the propagation and the relationships between them. Similar to existing work (Li et al., 2017; Ratkiewicz et al., 2011), we construct user interaction graphs $\mathcal{G}_u = \{\mathbf{U}, \mathbf{A}_u\}$ where the nodes are individual users based on propagation network $\mathcal{G}$, as shown in Fig. 4. Basically, for any pair of unique users, we create an edge between them in the user interaction network as long as there is a reposting behavior between their posts in any post propagation network. Formally, we denote $\mathbf{A}_u \in \mathbb{R}^{|\mathbf{U}|\times|\mathbf{U}|}$ is the adjacency matrix indicating relationships among the users. Compared to a follower/followee network, the user interaction network structure reflects the truly active interactions between users.

4.2. Input embedding

The input embedding layer is used to capture the primitive characteristics of each user. To achieve this, the layer is divided into two separate modules: the time-aware post embedding and the contrastive learning-based user embedding as shown in Fig. 3(b). The main idea is to use the post information and users' global interactions to improve the user embeddings, enabling a more accurate representation of the users and their behavior.

4.2.1. Time-aware post embedding

Post information can reveal users' inner characteristics (Rissola, Bahrainian, & Crestani, 2019) and opinions (Zhu et al., 2015), which is important in building user representation and can further help tasks like rumor detection (Bian et al., 2020) and virality

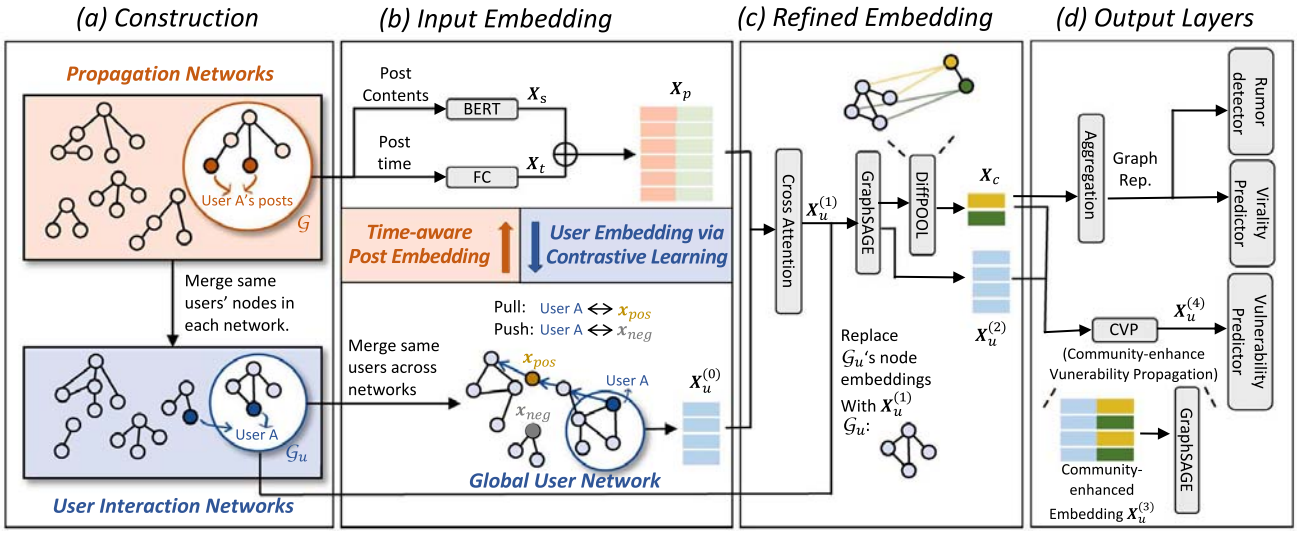

Fig. 3. Overview of the proposed multi-task model. (a) Construction. The user interaction network $\mathcal{G}_u$ is constructed based on its corresponding post propagation network $\mathcal{G}$. (b) Input Embedding. We generate time-aware post embedding $\mathbf{X}_p$, and general user embedding $\mathbf{X}_u^{(0)}$. (c) Refined Embedding. We obtain latent user community information $\mathbf{X}_c$ via Diffpool. (d) Output Layers. The final graph representation for $\mathcal{G}_u$ and user representations $\mathbf{X}_u^{(4)}$ updated via CVP are fed to the corresponding classifiers for our three tasks. *Note that the three tasks become separated only at (d) Output Layer, while the other layers (a) Construction, (b) Input Embedding, and (c) Refined Embedding are all needed for the three tasks as parts of their inputs and representation learning process.*

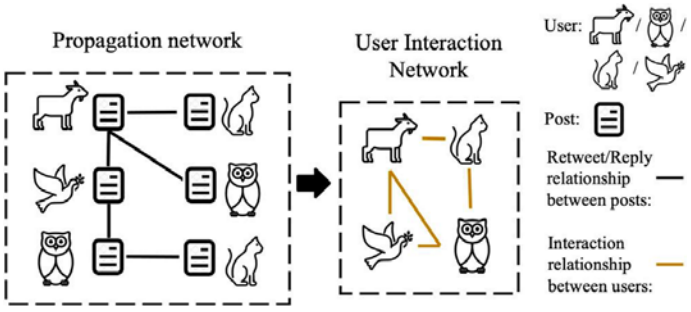

Fig. 4. An illustration of user interaction network construction. For any pair of unique users, we create an edge between them in the user interaction network as long as there is a reposting behavior between their posts in any post propagation network.

prediction (Chen, Kong, Xu, & Mao, 2019). To obtain the representation of each post's content, we use a pre-trained BERT-based encoder (Devlin, Chang, Lee, & Toutanova, 2019). Specifically, we first flatten the propagation network $\mathcal{G}$ in chronological order, which consist of a sequence of posts. This allows each post representation to jointly attend to nodes in different positions for better capturing semantics. Secondly, we insert two special tokens, i.e., [CLS] and [SEP], to the beginning and the end of post content $s_i \in v_i$, where the [CLS] token is intended to represent the semantic meaning of the post following it. The feature of token [CLS] from the last layer is taken to represent s_i, which is denoted as $\mathbf{x}_{s_i} \in \mathbb{R}^d$. To make the model more efficient, we froze BERT parameters when training our own model.

Apart from the content feature, the time of each post is expected helpful in signaling the information type (Nguyen et al., 2020) and future virality (Huang, Wang, & Zhang, 2019). Thus, we make post embedding time-dependent by using a fully connected (FC) layer to convert $t_i \in v_i$ into a vector $\mathbf{x}_{t_i}$ and create the time-aware post embedding as $\mathbf{x}_{p_i} = [\mathbf{x}_{s_i}, \mathbf{x}_{t_i}]$, where $[,]$ is the concatenation operator. Let matrix $\mathbf{X}_p \in \mathbb{R}^{|\mathbf{V}| \times 2d}$ denote embedding of all the posts in propagation network $\mathcal{G}$.

4.2.2. *Pre-train user embedding with contrastive learning*

The information diffusion process naturally reflects the rich proximity relationships between users (Zhang, Lyu, & Zhang, 2018). Mining the deep structure and patterns hidden in such relational networks can help learn the representation of users (Pan & Ding, 2019). In this paper, we first construct a global user network across specific events based on the information diffusion process, and then study the general characteristics of users by their locations and connections in the global user network.

Specifically, we first create a global user graph by merging the same users across different user interaction graphs into one node. We do not assume any user-level annotation is available for it. Then we use self-supervised contrastive learning to learn user embeddings. Given a user u_j, we simulate a random walk of fixed length in the network to get the positive samples from the nodes encountered along the path, and randomly pick negative samples. Due to the large size of the user network, we restrict both the

number of positive and negative samples as one. Let $\mathbf{x}_{u_j}^{(0)} \in \mathbb{R}^d$ denote u_j's embedding, the objective of contrastive learning is as follows:

$$L = \mathbf{x}_{u_j}^{(0)} \mathbf{x}_{pos} - \mathbf{x}_{u_j}^{(0)} \mathbf{x}_{neg}, \tag{1}$$

where $\mathbf{x}_{pos}$ and $\mathbf{x}_{neg}$ are embeddings of u_j's positive and negative samples in the random walk, respectively.

4.3. Refined embedding

For each event, we get both the user embeddings $\mathbf{X}_u^{(0)} \in \mathbb{R}^{|\mathbf{U}| \times d}$ and time-aware post embeddings $\mathbf{X}_p \in \mathbb{R}^{|\mathbf{V}| \times 2d}$ from the input embedding layer. Then we use a user-post cross attention (Vaswani et al., 2017) mechanism by treating $\mathbf{X}_u^{(0)}$ as query, and $\mathbf{X}_p$ as key and value. This basically uses user-specific features in $\mathbf{X}_u^{(0)}$ to guide our model in extracting post features in $\mathbf{X}_p$ relevant to the corresponding users, and thus refine the user embeddings with the post embeddings, which is formulated as:

$$\begin{aligned} Q = \mathbf{X}_u^{(0)} W_q, \quad K = \mathbf{X}_p W_k, \quad V = \mathbf{X}_p W_v, \\ \mathbf{X}_u^{(1)} = \text{softmax}\left(\frac{QK^T}{\sqrt{d}} V \right), \end{aligned} \tag{2}$$

where $\mathbf{X}_u^{(1)}$ represents the updated user embeddings with post information, and $\mathbf{W}_q$, $\mathbf{W}_k$ and $\mathbf{W}_v$ are trainable parameter matrices with dimensions of $d \times d$, $2d \times d$ and $2d \times d$ respectively.

To aggregate features from the local neighborhood users to capture their commonalities for better user representation, we then use GraphSAGE (Hamilton et al., 2017) as follows:

$$\mathbf{X}_u^{(2)} = \text{GraphSAGE}(\mathbf{X}_u^{(1)}, \mathbf{A}_u), \tag{3}$$

where $\mathbf{X}_u^{(2)}$ is the updated user representations containing user interaction information.

In our model, we employ DiffPool (Ying et al., 2018) for graph pooling on top of $\mathbf{X}_u^{(2)}$. DiffPool is used to identify latent clusters of like-minded users who may share similar views and vulnerabilities. These clusters enhance our ability to learn and perform specific tasks more effectively. Specifically, DiffPool is used to coarsen the graph $\mathcal{G}_u$ from $|\mathbf{U}|$ user nodes to $|\mathbf{V}_c|$ community nodes. This is done in two steps of standard DiffPool: (1) compute a soft community assignment matrix $\mathbf{C} \in \mathbb{R}^{|\mathbf{U}| \times |\mathbf{V}_c|}$ using GraphSAGE again as below:

$$\mathbf{C} = \text{softmax}(\text{GraphSAGE}(\mathbf{X}_u^{(2)}, \mathbf{A}_u)), \tag{4}$$

and (2) get the community node embedding matrix $\mathbf{X}_c = \mathbf{C}\mathbf{X}_u^{(2)}$ for the new coarsened graph.

Features of both original user graph and pooled community graph are then fed into output layers for rumor detection, virality and vulnerability prediction tasks, as shown in Fig. 3(c).

4.4. Output layers

4.4.1. Graph classification for rumor detection and virality prediction

For rumor detection and virality prediction, we try different pooling methods over the pooled graph including sum, mean, max, and an additional DiffPool to further pool the user community graph into a single graph representation. We find sum pooling performs best, which might be because the sum operation is more sensitive to the size of the graph. After obtaining the final graph representation, we use multi-layer perceptrons (MLPs) for rumor detection and virality prediction at the graph level.

4.4.2. Node classification for user vulnerability prediction

Community information is expected to help improve the completeness of user information because users within a community often have similar behaviors and interests. For vulnerability prediction, we propose a method named **Community-enhanced Vulnerability Propagation** (CVP) to exploit the latent communities to refine user node representation. The basic idea is to concatenate node representation with its corresponding community representation.

With the soft community assignment matrix $\mathbf{C}$ and community node embedding matrix $\mathbf{X}_c$, we get an embedding matrix $\mathbf{X}_u^c = \mathbf{C}\mathbf{X}_c \in \mathbb{R}^{|\mathbf{U}| \times d}$, where each row is the community embedding of the corresponding node in $\mathbf{X}_u^{(2)}$. Then, we get the updated user embedding by concatenating the original node embedding with its corresponding community embedding $\mathbf{X}_u^{(3)} = [\mathbf{X}_u^{(2)}, \mathbf{X}_u^c]$. Finally, we pass the enhanced node representations through a GraphSAGE again to get the final user representations $\mathbf{X}_u^{(4)} = \text{GraphSAGE}(\mathbf{X}_u^{(3)}, \mathbf{A}_u)$, so that embedding of neighboring nodes can influence each other explicitly.

After obtaining the final node representations, we use an MLP to predict the vulnerability of each user.

4.5. Training strategies

During training, we use the mean squared error (MSE) loss function for user vulnerability prediction and virality prediction, and cross-entropy loss for rumor detection. We train the joint learning framework under two different multi-task settings to explore their ability to mitigate training conflict of multiple tasks: (1) concurrent training, and (2) meta-training based on meta-learning.

Algorithm 1 Our meta-learning-based multi-task algorithm

Input: A set of graphs $\mathcal{G}$; Parameters $\theta = \{\theta_b, \{\theta_{h_k}\}\}$.

1: Initialize parameters θ randomly.
2: **for** *iter* from 1 to *maxIter* **do**
3: $outer\_loss \leftarrow 0$
4: **for** each task $\mathcal{T}_k$ $(k \in \{0, 1, 2\})$ **do**
5: $\theta'_{h_k} \leftarrow \theta_{h_k}$
6: Compute inner loss: $inner\_loss = \mathcal{L}_{\mathcal{T}_k}\left(f_\theta\right)$
7: Update task-specific parameters with gradient descent: $\theta'_{h_k} \leftarrow \theta_{h_k} - \nabla_{\theta_{h_k}} inner\_loss$
8: Update outer loss: $outer\_loss \leftarrow outer\_loss + \mathcal{L}_{\mathcal{T}_k}\left(f_{\theta_b, \theta'_{h_k}}\right)$
9: **end for**
10: $\theta \leftarrow \theta - \nabla_\theta outer\_loss$
11: **end for**
12: Return θ

Table 1
Datasets statistics.

Dataset	Type	# instances	Average # posts	Average # users	Average Vulnerability	Average Virality
TWITTER	**Rumor**	1,560	277.2	271.7	0.843	340.1
	Non-Rumor	579	503.7	498.3	0.196	621.2
WEIBO	**Rumor**	2,311	720.7	701.2	0.904	876.4
	Non-Rumor	2,351	577.9	568.7	0.124	711.0

4.5.1. Concurrent training

To deal with the potential training conflict between three tasks and further improve the model performance, we combine the three loss functions using Gradnorm (Chen et al., 2018) which dynamically adjusts their weights so that the gradient magnitudes of tasks are close and the tasks learn at a similar rate.

4.5.2. Meta-training

Inspired by Buffelli and Vandin (2020), we use a meta-learning strategy to mitigate training conflicts. Let us denote the proposed method as f_θ, where θ represents its parameters. The goal is to find a set of parameters θ that can perform well on the three tasks with only a few steps of gradient descent on each task. Specifically, θ is partitioned into the backbone parameters θ_b and the head parameters $\{\theta_{h_k}\}$ where the backbone of our model are the input and refined embedding layers, and the three heads correspond to specific tasks. As shown in Algorithm 1, the meta-learner tries to find the best configuration via inner and outer loops. For each task, the inner loop updates the head layer parameters $\{\theta_{h_k}\}$ in step 7 through a few steps of gradient descent on a task-specific inner loss $\mathcal{L}_{\mathcal{T}_k}\left(f_\theta\right)$. After updating head parameters, each task-specific loss $\mathcal{L}_{\mathcal{T}_k}\left(f_{\theta_b, \theta'_{h_k}}\right)$ is used to compute the outer loss.

5. Experimental evaluation

5.1. Datasets and setup

5.1.1. Datasets

We construct our data based on two public datasets namely TWITTER (Ma et al., 2017) and WEIBO (Ma et al., 2016). The original datasets were designed for rumor detection with only rumor/non-rumor annotations at the graph level. We need to derive ground-truth virality labels and vulnerability labels. We establish the *virality* label for each graph based on the count of unique users in the entire propagation process of each event in our datasets. Our goal is to predict the virality of a propagation network using its earlier propagation state as input (e.g. when the observation percentage of the propagation $\frac{t}{T}$ equals to 20%, 40%, ..., etc.). Following Liao et al. (2019), our main tables (Tables 2, 3, 4 and 5) show the prediction results obtained when $\frac{t}{T} = 80\%$. The prediction results corresponding to other proportions as observations are provided in Fig. 8. For users, we use the proportion of rumors among all events in which the user is involved as their gold *vulnerability*. Only users engaged in more than one propagation graph are labeled, constituting 15.5% users on TWITTER and 16.7% users on WEIBO. Unlabeled users are not considered in performance evaluation.

We split the dataset into training, validation, and test sets by $80\%, 10\%$, and 10%, respectively. A key priority is to eliminate user overlaps between the training set and the validation/test sets, thereby ensuring no data leakage. To accomplish this, we began by examining each graph within our datasets. If a graph exclusively contains users not found in any other graphs, it is designated as a

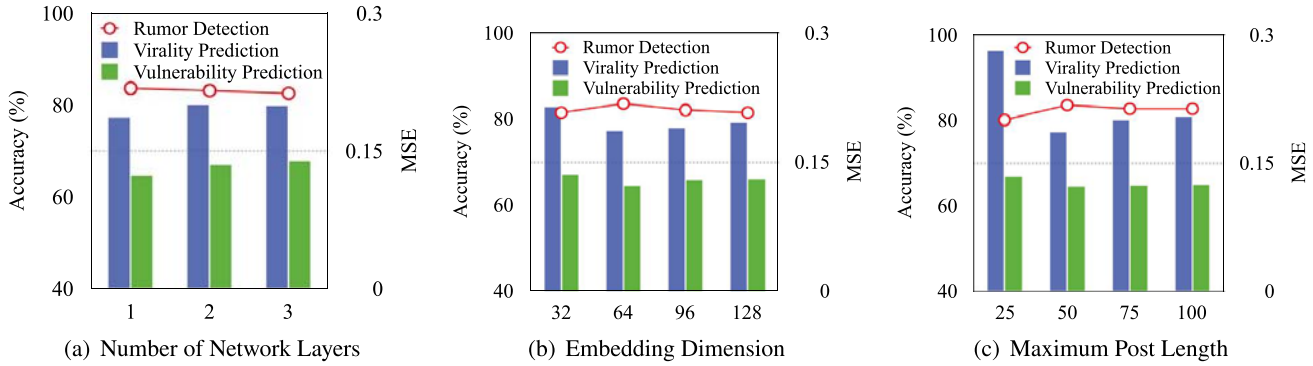

Fig. 5. The impact of hyper-parameters that are related to model's complexity based on the validate set.

non-overlapping graph. Notably, over 20% of the graphs in our datasets are non-overlapping with any other graphs. We then randomly sample from these non-overlapping graphs, so that both the validation and test sets comprise these sampled non-overlapping graphs, with each of the two sets representing 10% of the entire dataset. Finally, the rest of the non-overlapping graphs and all other graphs are put in the training set. This ensures that users from the users in the validation/test sets remain unseen during the training process. Table 1 gives the statistics of our datasets.

5.1.2. Metrics

For the rumor detection task, since the accuracy of the classification results can be affected by the class imbalance, we further use precision, recall, and macro-averaged F1 score (MacF1) score in addition to accuracy to provide a more comprehensive evaluation of performance. Meanwhile, for the virality prediction and vulnerability prediction tasks, we use MSE, and mean squared logarithmic error (MSLE) to evaluate the regression results. In addition, we also use normalized Discounted Cumulative Gain (nDCG) (Järvelin & Kekäläinen, 2000) to measure the models' ranking performance based on the predicted virality and vulnerability scores because nDCG takes into account the relevance and ranking position of each predicted item, which is particularly important for virality and vulnerability prediction tasks where the order of importance of the predicted items can significantly affect the overall effectiveness of the model.

5.1.3. Parameter settings

We implement all GNN-based models with DGL,[4] and all deep learning based models with Pytorch.[5] Our method is based on GraphSAGE (Hamilton et al., 2017) and DiffPool (Ying et al., 2018) models. In our experiments, we employ Random Search (Bergstra & Bengio, 2012) for hyper-parameter tuning since it is more efficient than Grid Search. We set the number of iterations in Random Search as 100. Our focus was on a number of key hyper-parameters: (1) the number of layers in both GraphSAGE and CVP, drawn from $\{1, 2, 3\}$; (2) the pool size for DiffPool, adjusted with the choices in $\{25, 50, 75, 100\}$; (3) the embedding dimension, selected from $\{32, 64, 96, 128\}$; (4) batch size, selected from $\{2, 4, 6, 8, 10\}$; (5) the learning rate, selected from the set $\{1e-2, 5e-3, 1e-3\}$; (6) the dropout rate, sampled from the set $\{0.1, 0.2, 0.4, 0.6\}$; and (7) the maximum post length, selected from the set $\{25, 50, 75, 100\}$. We examine the influence of these hyper-parameters on the validate set. In particular, three of them directly determine the model's complexity, i.e., the number of layers, the embedding dimension, and the maximum post length. We specifically demonstrate their influence on model performance in Fig. 5. Random Search led us to select one layer for both GraphSAGE and CVP, a DiffPool pool size of 50, an embedding dimension of 64, a batch size of 8, a learning rate of 5*e*-3, a dropout rate of 0.2, and a maximum post length of 50.

During training, we update model parameters using stochastic gradient descent and optimize the model by Adam algorithm. The experimental results are averaged over five independent runs with different seeds.

5.2. Results of rumor detection

Given our multi-task models based on concurrent training (MT-CON) and meta-learning (MT-META), we make comparisons with the following baseline models:

- GCNFN (Monti, Frasca, Eynard, Mannion, & Bronstein, 2019): A rumor detection model by exploiting geometric deep learning.
- Bi-GCN (Bian et al., 2020): A rumor detection method based on bidirectional GCN.
- RoBERTa (Pelrine, Danovitch, & Rabbany, 2021): A baseline directly fine-tuning the pre-trained RoBERTa Model, which achieves surprising performance on rumor detection tasks.

[4] https://www.dgl.ai/
[5] https://pytorch.org/

Table 2
Experiment results. $*$ ($**$): Significant improvement over the best baseline with $p < 0.05$ (0.01). Bold denotes the best performance in each task.

(a) Rumor detection

Model	TWITTER				WEIBO			
	Acc.	Pre.	Recall	MacF1	Acc.	Pre.	Recall	MacF1
GCNFN	0.772	0.664	0.714	0.731	0.902	0.885	0.930	0.908
Bi-GCN	0.790	0.736	0.763	0.716	0.913	0.893	0.942	0.921
RoBERTa	0.791	0.749	0.772	0.740	0.917	0.907	0.940	0.923
UPFD	0.815	0.836	0.783	0.805	0.921	0.905	0.940	0.933
DDGCN	0.813	0.821	0.790	0.811	0.918	0.911	0.937	0.925
Us-DeFake	0.819	0.839	0.803	0.821	0.919	0.910	0.931	0.924
MT-CON (ours)	0.820	0.854*	0.814*	0.833	0.946*	0.929*	0.949*	0.943*
MT-META (ours)	**0.826***	**0.879****	**0.826***	**0.845***	**0.954***	**0.941****	**0.955***	**0.952****

(b) Virality prediction

Model	TWITTER			WEIBO		
	MSE	MSLE	nDCG	MSE	MSLE	nDCG
DeepHawkes	2.011	0.067	0.521	0.994	0.048	0.783
NPP	1.199	0.020	0.867	0.867	0.022	0.984
DeepBlue	0.918	0.016	0.894	0.824	0.017	0.985
BERT	0.960	0.019	0.891	0.835	0.018	0.989
CasSeqGCN	0.648	0.009	0.990	0.799	0.013	0.988
TCAN	0.588	0.008	0.992	0.792	0.017	0.989
MT-CON (ours)	0.320**	0.006	0.996	0.716**	0.013	**0.995**
MT-META (ours)	**0.197****	**0.005**	**0.999**	**0.603****	**0.008****	**0.998***

(c) Vulnerability prediction

Model	TWITTER			WEIBO		
	MSE	MSLE	nDCG	MSE	MSLE	nDCG
LING-GAT	0.121	0.062	0.989	0.168	0.074	0.972
GraphRfi	0.151	0.075	0.985	0.179	0.079	0.963
IMP-GCN	0.140	0.074	0.975	0.181	0.079	0.964
U-BERT	0.124	0.065	0.983	0.166	0.084	0.974
PinnerFormer	0.126	0.067	0.985	0.167	0.082	0.969
CLUE	0.127	0.068	0.987	0.164	0.077	0.972
MT-CON (ours)	0.112	0.054	0.991	0.153	0.061*	0.981
MT-META(ours)	**0.104***	**0.035****	**0.995**	**0.137***	**0.047****	**0.988***

- UPFD (Dou et al., 2021): A GCN-based model that considers user preference to help rumor detection.
- DDGCN (Sun et al., 2022): A Dual-Dynamic GCN based model, which can model the dynamics of propagation networks as well as the dynamics of the background knowledge from knowledge graphs.
- Us-DeFake (Su, Yang, Wu, & Zhang, 2023): A rumor detection method by learning the propagation features and the user interaction features.

As shown in Table 2, our proposed joint learning models generally outperform all the baselines on both datasets. We have the following observations:

(1) *GCN with proper design is one promising way in rumor detection.* GCN-based models can effectively identify the propagation patterns of rumors and non-rumors, and are widely used as base models in rumor detection. The integration of prior knowledge through special design can further enhance the models' performance in this task. For example, by taking user preference characteristics into account, On average, UPFD surpasses GCNFN and Bi-GCN in terms of accuracy by 3.8% and 2.0%, respectively.

(2) *Content in posts and their responses provides valuable information for detecting rumors.* Although ignoring the complex graph structures and only exploiting the content information, fine-tuning the pre-trained RoBERTa model can also achieve comparable behavior with GCN-based GCNFN and Bi-GCN models. For example, RoBERTa surpasses GCNFN in terms of accuracy by 2.1%. This suggests the importance of content information unleashed by the strong presentation ability of the pre-trained model for rumor detection.

(3) *Hierarchical graph pooling can help utilize more useful information.* Node representations learned from both basic GCNs are flat since they cannot encode the graph structure in a hierarchical way by just propagating information through nodes and edges. Our proposed models MT-CON and MT-META take advantage of user representation and hierarchical pooling in joint learning, thus beating all the baselines. For example, MT-META surpasses the best baseline Us-DeFake in terms of accuracy by 2.4% on average.

(4) *Mitigating training conflicts of multiple tasks can further improve performance.* MT-META outperforms MT-CON in terms of accuracy by 0.8% on average, suggesting the meta-learning strategy behaves better in mitigating training conflicts for our tasks compared to concurrent training.

5.3. Results of virality prediction

We make comparisons with the following baseline models:

- DeepHawkes (Cao, Shen, Cen, Ouyang, & Cheng, 2017): A Hawkes process-based deep learning model for virality prediction.
- NPP (Chen et al., 2019): A prediction model that learns embeddings for popularity and virality, taking into account time, user, and content factors.
- DeepBlue (Zhang et al., 2021): A popularity and virality method based on bi-layered LSTM, which takes into account historical information such as user reputation and tweet-related features.
- BERT (Tan et al., 2022): A Bert-based model to exploit content features for popularity and virality prediction. The original paper exploits features from other modalities (e.g. images). Here we only use its text processing method for comparison.
- CasSeqGCN (Wang, Wang, Ran, Michalski, & Jia, 2022): A popularity and virality prediction method that employs GCN for network structure features and LSTM for temporal dynamics.
- TCAN (Sun, Zhou, Liu, & Wei, 2023): An explicit time embedding based popularity and virality method, which employs a graph attention encoder and a sequence attention encoder to learn the representation of propagation networks.

As shown in Table 2, MT-CON and MT-Meta generally outperform all the baselines on both datasets. We have the following observations:

(1) *Information contained in propagation structures can aid in virality prediction but may not be sufficient.* DeepHawkes performs the worst, with 39.2% higher in MSE than the second worst model (i.e., NPP), as it only uses the network structure to model the discussion process. Models considering more information like context and user perform clearly better.

(2) *Incorporating user and event features can improve performance in virality prediction.* DeepBlue outperforms NPP by 13.3% lower in MSE on average, as it considers not only individual tweets but also incorporates useful user reputation and post-related features learned from historical tweets. Additionally, our model considers the information type of event (i.e., rumor/non-rumor) and the vulnerability of the users involved in its propagation to aid in virality prediction.

(3) *Content in posts and their responses can signal the future virality of the current event.* Although only exploiting content features, the pre-trained BERT model achieves comparable behavior to DeepBlue, showing only a slight increase in MSE by 3.3%. This suggests the significant role of text in the virality prediction task as well as the strengths of the pre-trained model.

(4) *Incorporating hierarchical pooling can lead to a better understanding of the graph's structure, thus improving the accuracy of virality prediction.* Our models MT-CON and MT-META utilize a combination of different feature representations such as time, user, and post content and information on event type and user vulnerability, which are then hierarchically pooled, leading to improved performance compared to all baselines. For example, MT-META exhibits a lower MSE than TCAN by 49.3% on average.

(5) *Reducing training conflicts between multiple tasks can enhance performance.* Similar to the observation in rumor detection results, MT-META performs better than MT-CON in mitigating training conflicts with the help of a meta learning-based training strategy. On average, MT-META shows a lower MSE than MT-CON by 37.5%.

5.4. Result of user vulnerability prediction

For this task, we make comparisons with the following baseline models:

- LING-GAT (Del Tredici, Marcheggiani, im Walde, & Fernández, 2019): A user representation method using Bi-LSTM layers to capture linguistic features and a transductive graph attention network (GAT) to model a user's social relationships.
- GraphRfi (Zhang et al., 2020): A hand-crafted feature based user representation learning framework using GCN and neural random forest.
- IMP-GCN (Liu, Cheng, Zhu, Gao, & Nie, 2021): A GCN-based user interest-aware representation learning model.
- U-BERT (Qiu, Wu, Gao, & Fan, 2021): A pre-trained user embedding model inspired by the success of the BERT model.
- PinnerFormer (Pancha, Zhai, Leskovec, & Rosenberg, 2022): A user representation learning method based on a sequence of users' recent actions. To make it adapt to the vulnerability prediction task, we treat users' posts and comments as their actions.
- CLUE (Shin et al., 2023): A contrastive learning based user representation learning method for general purpose.

To adapt existing baseline models for the regression task of predicting user vulnerability, we modified them by removing their task-specific final layers and replacing them with a regression layer. This allowed us to use the same model architectures while repurposing them for the specific task of user vulnerability prediction.

As shown in Table 2, our proposed models outperform all the baselines on both datasets for vulnerability prediction. We have some specific observations:

(1) *Hand-crafted features are limited in capturing latent features and deep correlations.* GraphRfi performs the worst with 7.1% higher in MSE than the second worst model (i.e., IMP-GCN), due to the generally weak generalizability of hand-crafted features.

(2) *The content of user posts can reveal their inner vulnerability.* IMP-GCN performs slightly better than GraphRfi by 6.6% lower in MSE, as it uses content information that can convey user propensity. However, it still performs worse than another GCN-based model LING-GAT by 16.5% higher in MSE, as LING-GAT deeply exploits post content information with Bi-LSTM layers.

(3) *Pre-training and contrastive learning are promising ways in user representation learning.* U-BERT outperforms all other baselines, suggesting the usefulness of pre-trained models for user classification respectively. Interestingly, PinnerFormer and CLUE, which do

Table 3
Ablation study results on TWITTER.

#	Input Emb.			Refined Emb.		CVP		Task					
								Rumor		Virality		Vulnerability	
	User Emb.	Cnt. Emb.	Time Emb.	Sum Pool	Hier. Pool	Comm. Enh.	Graph-SAGE	Acc	MacF1	MSE	nDCG	MSE	nDCG
1		✓	✓		✓	✓	✓	0.802	0.757	0.233	0.995	0.138	0.983
2	✓				✓	✓	✓	0.777	0.742	0.391	0.990	0.159	0.988
3	✓		✓		✓	✓	✓	0.784	0.759	0.373	0.990	0.152	0.986
4	✓	✓			✓	✓	✓	0.791	0.771	0.381	0.992	0.143	0.984
5	✓	✓	✓	✓		✓	✓	0.777	0.730	0.405	0.985	0.160	0.984
6	✓	✓	✓		✓			0.821	0.817	0.211	0.994	0.152	0.986
7	✓	✓	✓		✓		✓	0.821	0.809	0.204	0.998	0.147	0.979
8	✓	✓	✓		✓	✓		0.822	0.814	0.204	0.998	0.134	0.984
9	✓	✓	✓		✓	✓	✓	**0.826**	**0.845**	**0.197**	**0.999**	**0.104**	**0.995**

not utilize propagation network structures, demonstrate impressively good performance. For instance, CLUE achieves an average MSE that is 12.6% and 10.7% lower than GraphRfi and IMP-GCN, respectively, which consider more information. This points to the untapped potential of contrastive learning, a technique employed by both PinnerFormer and CLUE. Our model also pre-trains a general user embedding via contrastive learning to help capture user features.

(4) *Decreasing training conflict of multiple tasks leads to improved performance.* This observation is similar to the other two tasks. In the vulnerability prediction task, MT-META achieves a lower MSE than MT-CON by 8.9% on average.

5.5. Analysis

5.5.1. Ablation study

To examine the impact of each key component in our proposed joint learning approach, we perform an ablation study based on our best full model MT-META on the TWITTER dataset. Table 3 shows our experiments. We summarize various aspects and highlight the most interesting findings below.

Input embedding. We first show the effect of each component in the input embedding layer in the first four rows. We can see that there is a moderate degradation in the model performance when the pre-trained user embeddings (i.e., User Emb. in Table 3) are not used, which suggests the importance of mining deep structure and patterns hidden in the constructed global user graph for all our three tasks. In addition, in our model, both content and time information in post embedding play indispensable roles, as demonstrated by the performance drop after the removal of content, time, and both types of embeddings.

Refined embedding. If we replace hierarchical pooling with sum pooling, which means that we directly sum over the node representations to get the graph representations, the performance drops. This is consistent with our hypothesis that learning the hierarchical structure of the graphs (i.e., through DiffPool in our model) can improve the model's performance.

CVP. Both community enhancement and GraphSAGE in CVP play important roles in user vulnerability prediction. This further confirms our hypothesis that CVP can directly help predict user vulnerability by utilizing the shared features among users within latent communities to refine node representation.

5.5.2. Effects of different task loss configurations

To assess the contribution of each task loss in our proposed joint learning method, we conduct an ablation experiment by varying the loss configuration and using our most effective model MT-META on the TWITTER dataset, as shown in Table 4. We have some specific observations:

- *The direct connection between virality and rumorous class (i.e., rumor or non-rumor) is weak.* When only the virality loss is taken into account, the performance of rumor detection is the lowest. Similarly, the performance of virality prediction is the lowest when only the rumor detection loss is considered.
- *Task training conflict can cause performance degradation.* Inappropriate joint learning may result in a certain level of training conflict, as it attempts to balance the learning process among different task objectives. Take the rumor detection task as an example, when the rumor detection loss is jointly optimized with the virality prediction loss, the performance drops by 5.1% in accuracy and 0.6% in MacF1 than only the rumor detection loss being optimized; when jointly optimized with the vulnerability prediction loss, the accuracy slightly decreases from 0.806 to 0.801 while MacF1 slightly increases from 0.796 to 0.8. This might suggest that rumor prediction is more directly related to user vulnerability than virality.
- *Joint learning across all three tasks is essential to prevent training conflict.* This is because given user vulnerability as a bridge, the relationship between the virality and the rumor/non-rumor nature of information could be established implicitly. In the rumor detection task, we observe a 2.5% and 6.2% increase in accuracy and MacF1 respectively in the rumor detection task when jointly optimizing all three losses compared to only rumor detection loss. A similar extent of improvement can be observed in the other two tasks. This implies that joint learning across all three tasks can prevent training conflict, probably due to the bridge effect with more tasks.

Table 4
Effect of different task loss configurations on TWITTER. Underline denotes the performance of a specific task when its corresponding loss was optimized during training. Bold denotes the best performance in each task.

#	Loss			Task					
	Rumor Detection	Virality Prediction	Vulnerability Prediction	Rumor		Virality		Vulnerability	
				Acc	MacF1	MSE	nDCG	MSE	nDCG
1	✓			0.806	0.796	321.766	0.145	23.262	0.005
2		✓		0.673	0.641	0.209	0.990	1.977	0.187
3			✓	0.739	0.704	168.336	0.275	0.122	0.972
4		✓	✓	0.748	0.710	0.201	0.998	0.137	0.961
5	✓		✓	0.801	0.800	9.513	0.479	0.207	0.954
6	✓	✓		0.765	0.742	0.249	0.990	0.518	0.872
7	✓	✓	✓	**0.826**	**0.845**	**0.197**	**0.999**	**0.104**	**0.995**

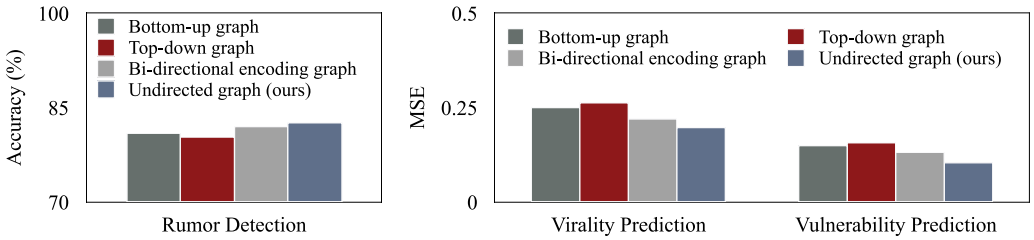

Fig. 6. Effect of edge direction.

5.5.3. *Effects of the edge direction*

To investigate the impact of edge direction in propagation networks in our proposed method, we explore four distinct edge direction settings: (1) A bottom-up graph where edges follows the direction of information being referenced (Bian et al., 2020; Ma et al., 2018), (2) A top-down directed graph, where edges follow the direction of information flow (Bian et al., 2020; Ma et al., 2018), (3) An bi-directional encoding graph, where two GraphSage encoders work simultaneously to represent the features along with the two different directions (i.e., top-down and bottom-up) in the graph. The embeddings produced by these encoders are concatenated to form a comprehensive node representation containing bi-directional information, similar as Bi-GCN (Bian et al., 2020), and (4) Our undirected graph, where each edge, although without direction, indicates a two-way relationship and the adjacency matrix of the graph unequivocally represents a bidirectional graph as each edge is reciprocated. As shown in Fig. 6, our findings indicate that both top-down and bottom-up directed graphs exhibit worse performance than the two bi-directional variants in (3) and (4). This may suggest that feature representation with the underlying model considering both directions is better than considering only a single direction regardless of the actual direction of propagation. Moreover, the undirected treatment in (4) outperforms the bi-directional case in (3). This might be because the undirected graph integrates information from both directions more directly with the symmetric adjacency matrix, rather than a separate learning and then combining them. Such direct integration helps the model better understand the relationships between nodes.

5.5.4. *Effects of the GNN encoder*

To investigate the influence of the GNN encoder within our proposed method, we tested different GNN encoders to derive node representations. In Fig. 7, we display the performance using six distinct encoders, namely GCN (Defferrard et al., 2016), GraphSAGE (Hamilton et al., 2017), GAT (Veličković et al., 2018), GIN (Xu, Hu, Leskovec, & Jegelka, 2018), and a case without a GNN encoder. We observe that the variants with a GNN encoder consistently outperform those without it, indicating the advantage of GNNs in updating node features. Moreover, we notice that GraphSAGE performs slightly better than other GNN encoders. We conjecture that this could be due to GraphSAGE's inductive learning and sampling mechanism, which might be particularly suitable for social network graphs with abundant noisy information.

5.5.5. *Effects of the observation percentage*

For clarity, we only compare our models with the three best-performing baselines for each task. Fig. 8 demonstrates the changes in rumor detection accuracies and the MSEs for virality and vulnerability predictions as the observation percentage $\frac{t}{T}$ changes from 20% to 80%. As the percentage increases, the size of the observed graphs also increases, resulting in improved performance in all three tasks. From the figure, we can see that even only using early propagation information (e.g., 20%), our proposed models MT-CON and MT-META outperform the baseline models with clear margins on the three tasks. In particular, with only 40% propagation as observation, the MT-META can achieve similar rumor detection accuracy as the three baselines using 80% propagation (see Fig. 8(a)).

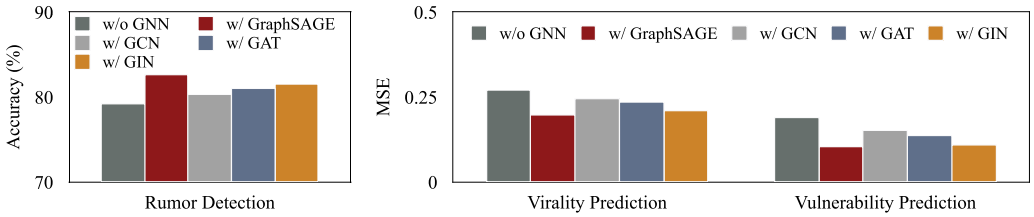

Fig. 7. Effect of the GNN encoder.

Table 5
Effect of training strategies on the TWITTER dataset. STL/MTL: Single-/Multi-task learning.

Setting		Rumor		Virality		Vul.	
		Acc	MacF1	MSE	nDCG	MSE	nDCG
STL	–	0.806	0.796	0.209	0.990	0.122	0.972
MTL	Basic	0.793	0.784	0.228	0.981	0.126	0.973
	CON	0.820	0.833	0.320	0.996	0.112	0.991
	Meta	**0.826**	**0.845**	**0.197**	**0.999**	**0.104**	**0.995**

5.5.6. *Effect of training strategies*

While there is a correlation among tasks, training conflicts leading to negative transfer could occur if an appropriate training strategy is not selected, and our training method can mitigate this issue. Table 5 shows that when we train all three tasks with the basic multi-task training by linearly combining the losses of three tasks, the performance is lower than training them individually, suggesting a training conflict. Gradnorm, used by concurrent training, obviously improves the basic training by adjusting the gradients to balance the learning rates of the different tasks. However, it only improves the performance over the single-task setting marginally. In contrast, the meta-learning strategy outperforms both the basic training and the Gradnorm approach on all three tasks, indicating it mitigates training conflict substantially. This superior performance might be attributed to the fact that in meta-learning, training conflicts have less chance to occur since only θ_b is shared by the different tasks.

When the training conflict is alleviated, that is, after finding a suitable training method, we find that the joint learning results are better than training them individually. Importantly, these results suggest that the proposed approach effectively captures the correlation between rumor virality and user vulnerability and utilizes this information to improve prediction performance. By identifying these correlations, the model can better predict how misinformation might spread and which users are more susceptible to it. Consequently, this allows for targeted interventions and strategies to prevent the spread of misinformation, thereby enhancing the overall effectiveness of combating fake news and rumors in online communities.

5.5.7. *User community visualization*

We visualize the three largest user communities using t-SNE (Van der Maaten & Hinton, 2008) generated by the pooling layer of MT-META. As our model utilizes a soft assignment, we consider each user is assigned to the community with which they have the highest similarity. Fig. 9(a) shows that our model can effectively pull users with similar embeddings together, which facilitates further utilization of information implied by the communities (e.g., the shared patterns and characteristics of users within each community) to help our tasks.

5.5.8. *Case study*

To better illustrate how our model leverages learning from the three tasks and utilizes the relationships between them to improve performance, we show an example in Fig. 9(b) using the graph of a viral rumor to interpret the prediction results.

We can see that our model predicts the event as a *rumor* which appears consistent with its user vulnerability prediction, as most of the reposts are from vulnerable or moderate-vulnerable users. In addition, the event is unappealing to users that are moderately or not vulnerable (i.e., vulnerability score lower than 0.5). The high user vulnerability scores highlight that people are easily swayed by the rumor and may readily accept and spread it without verification, which can explain why the rumor gets virally spread. Compared to vulnerable users, the less vulnerable ones (e.g., with the post marked in orange color) seem to be more critical towards the claim as they notice the controversies of the source post instead of simply agreeing or repeating what others narrate. This observation conforms to the definition of user vulnerability: lower user vulnerability means the user is engaged more often in non-rumors (i.e., verified news) than rumors (i.e., false or unverified information), suggesting that this kind of user is more credible or rational.

6. Implications

Rumors are a constantly evolving phenomenon that requires collaborative efforts to mitigate their negative impact. For the technical contribution, we provide a new perspective, which combines three tasks of rumor detection, virality prediction, and user vulnerability scoring for infodemic surveillance. In this scenario, it proposes a mechanism that leverages the power of GNN to

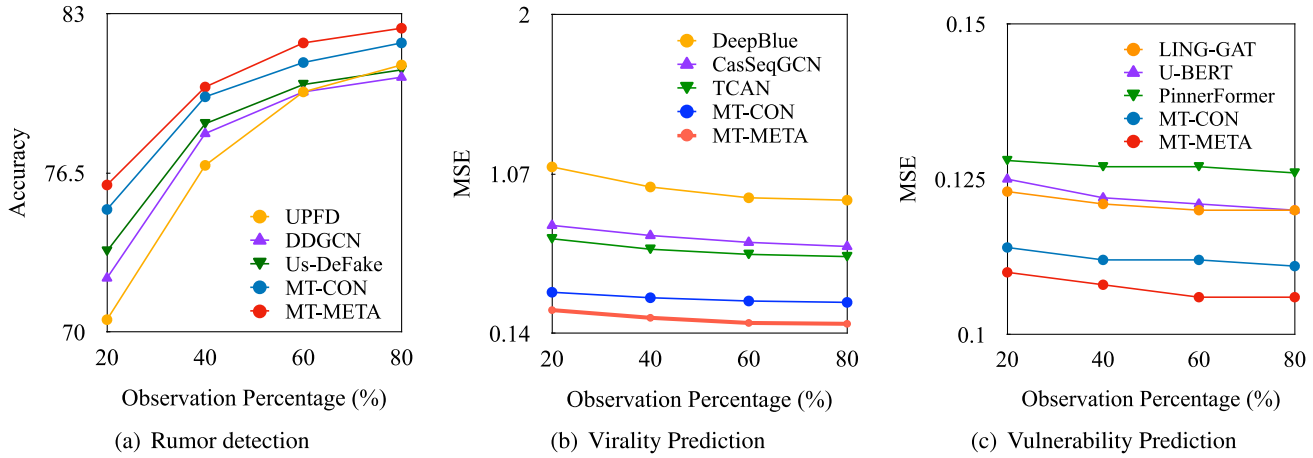

Fig. 8. Impact of the observed fraction of cascades on the performance of three tasks.

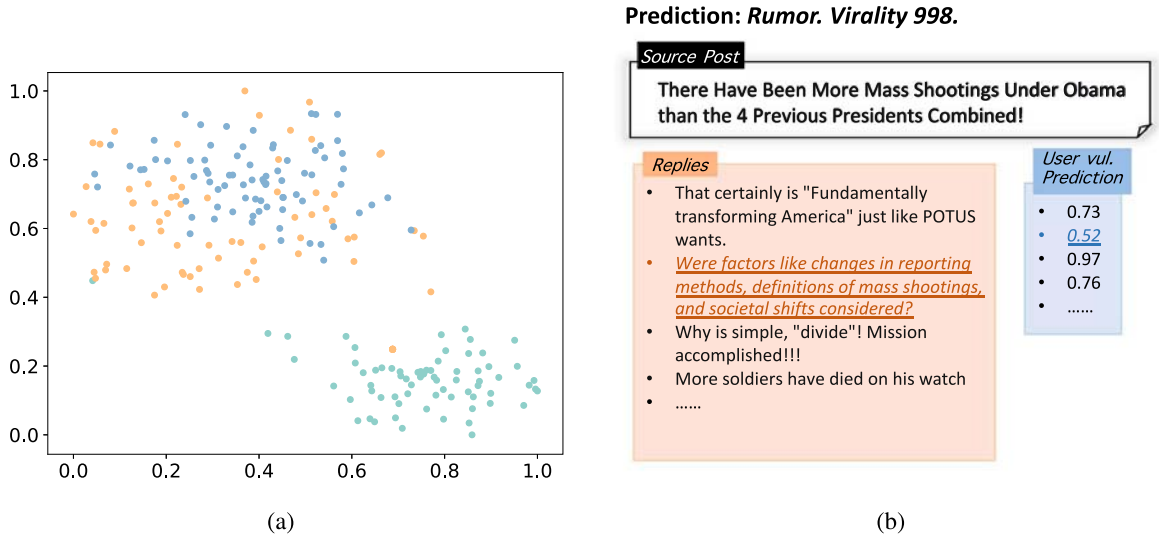

Fig. 9. (a) User community visualization, where each node is one user, and communities are colored differently. (b) An example of a predicted viral rumor with user vulnerability scores by our model.

simultaneously learn these three tasks that were previously learned independently and captures the potential correlations among them. This not only improves the performance of individual tasks but also facilitates timely and effective information infodemic surveillance, that is, *providing prompt and accurate responses to surveillance demands with limited information available*. Our framework also provides a new avenue for future research on infodemic surveillance.

For practical contribution, we provide a feasible strategy for infodemic surveillance in online social media platforms. The surveillance system first detects events and their spread on social media. The detected viral rumors are then sent to fact-checkers for verification, and vulnerable users involved in rumors' spreading are also alerted to make them aware. In this way, firstly, vulnerable users can be precisely protected without disturbing the experience of other users as much as possible. Social media platforms are a crucial source of information and communication for many people. Disrupting the experience of other users could lead to confusion and mistrust, which could exacerbate the problem of infodemics. Therefore, our approach is designed to be precise in identifying and protecting vulnerable users. This targeted approach enables us to provide protection to those who need it most while minimizing any negative impact on the broader user base.

Secondly, the burden of verification can be reduced, resulting in more effective information surveillance and early intervention. The sheer volume of information that is circulated on social media platforms makes it challenging to verify rumors and misinformation manually. Our unified prediction framework, which combines rumor detection, virality prediction, and user vulnerability scoring, enables a more comprehensive and accurate analysis of online information. This means that we can predict the most impactful rumors and vulnerable users simultaneously, which reduces the need for time-consuming manual verification, and has the potential to assist authorities in allocating resources more effectively.

Thirdly, one of the most significant implications of our unified prediction framework is its ability to detect rumors and predict their virality at an early stage, even when propagation information is limited. This is a crucial function of the model as it allows

for timely intervention and prevents rumors from spreading further, potentially causing harm. Early detection of rumors and their potential impact enables authorities and organizations to respond quickly and effectively, thereby minimizing the risk of social unrest, political instability, and other adverse consequences.

7. Conclusion and future work

We propose a joint learning method for detecting rumors, and predicting their virality and user vulnerability in a unified multi-task framework based on graph neural networks. By leveraging the latent correlations of these tasks, our method can forecast rumors that potentially go viral and help find credulous users with a high propensity of spreading rumors, for timely and effective infodemic surveillance. The evaluation confirms that our method outperforms state-of-the-art baselines on all three tasks using two datasets with the ground truth of rumorous class, event virality, and user vulnerability constructed based on existing rumor detection corpora.

In the future, we plan to use a better ranking algorithm to replace the regression for further boosting nDCG. We will also develop approaches for embedding deeper user traits to better reflect users' internal states (maybe at a psychological level) towards rumors for more in-depth user vulnerability analysis.

CRediT authorship contribution statement

Xuan Zhang: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. **Wei Gao:** Conceptualization, Investigation, Methodology, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request

Acknowledgments

This research is supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier-1 grant (Grant No. 19-C220-SMU-013). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of funding agency.

References

Albladi, S. M., & Weir, G. R. (2020). Predicting individuals' vulnerability to social engineering in social networks. *Cybersecurity*, *3*(1), 1–19.
Ali, I. (2020). The COVID-19 pandemic: Making sense of rumor and fear: Op-ed. *Medical Anthropology*, *39*(5), 376–379.
Alkhodair, S. A., Ding, S. H., Fung, B. C., & Liu, J. (2020). Detecting breaking news rumors of emerging topics in social media. *Information Processing & Management*, *57*(2), Article 102018.
Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(2).
Bi, B., Wang, Y., Zhang, H., & Gao, Y. (2022). Microblog-HAN: A micro-blog rumor detection model based on heterogeneous graph attention network. *PLoS One*, *17*(4), Article e0266598.
Bian, T., Xiao, X., Xu, T., Zhao, P., Huang, W., Rong, Y., et al. (2020). Rumor detection on social media with bi-directional graph convolutional networks. In *AAAI* (pp. 549–556).
Bringula, R. P., Catacutan-Bangit, A. E., Garcia, M. B., Gonzales, J. P. S., & Valderama, A. M. C. (2021). Who is gullible to political disinformation?: Predicting susceptibility of university students to fake news. *Journal of Information Technology & Politics*, 1–15.
Buffelli, D., & Vandin, F. (2020). A meta-learning approach for graph representation learning in multi-task settings. In *NIPS workshop on meta-learning*.
Cao, J., Guo, J., Li, X., Jin, Z., Guo, H., & Li, J. (2018). Automatic rumor detection on microblogs: A survey. arXiv preprint arXiv:1807.03505.
Cao, Q., Shen, H., Cen, K., Ouyang, W., & Cheng, X. (2017). Deephawkes: Bridging the gap between prediction and understanding of information cascades. In *CIKM* (pp. 1149–1158).
Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on twitter. In *WWW* (pp. 675–684).
Chen, Z., Badrinarayanan, V., Lee, C.-Y., & Rabinovich, A. (2018). Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML* (pp. 794–803).
Chen, G., Kong, Q., Xu, N., & Mao, W. (2019). NPP: A neural popularity prediction model for social media content. *Neurocomputing*, *333*, 221–230.
Chen, X., Zhou, F., Zhang, F., & Bonsangue, M. (2021). Catch me if you can: A participant-level rumor detection framework via fine-grained user representation learning. *Information Processing & Management*, *58*(5), Article 102678.
Cheng, J., Adamic, L., Dow, P. A., Kleinberg, J. M., & Leskovec, J. (2014). Can cascades be predicted? In *WWW* (pp. 925–936).
Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS* (pp. 3844–3852).
Del Tredici, M., Marcheggiani, D., im Walde, S. S., & Fernández, R. (2019). You shall know a user by the company it keeps: Dynamic representations for social media users in NLP. In *EMNLP* (pp. 4707–4717).
Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL* (pp. 4171–4186).
Diehl, F. (2019). Edge contraction pooling for graph neural networks. arXiv preprint arXiv:1905.10990.
Dou, Y., Shu, K., Xia, C., Yu, P. S., & Sun, L. (2021). User preference-aware fake news detection. In *SIGIR* (pp. 2051–2055).

Gao, H., Chen, Y., & Ji, S. (2019). Learning graph pooling and hybrid convolutional operations for text representations. In *WWW* (pp. 2743–2749).
Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *NIPS* (pp. 1025–1035).
He, L., Xu, G., Jameel, S., Wang, X., & Chen, H. (2022). Graph-aware deep fusion networks for online spam review detection. *IEEE Transactions on Computational Social Systems*.
Hoang, T.-A., & Lim, E.-P. (2016). Virality and susceptibility in information diffusions. In *ICWSM* (pp. 146–153).
Huang, Z., Wang, Z., & Zhang, R. (2019). Cascade2vec: Learning dynamic cascade representation by recurrent graph neural networks. *IEEE Access*, *7*, 144800–144812.
Islam, M. S., Kamal, A.-H. M., Kabir, A., Southern, D. L., Khan, S. H., Hasan, S. M., et al. (2021). COVID-19 vaccine rumors and conspiracy theories: The need for cognitive inoculation against misinformation to improve vaccine adherence. *PLoS One*, Article e0251605.
Järvelin, K., & Kekäläinen, J. (2000). IR evaluation methods for retrieving highly relevant documents. In *SIGIR* (pp. 41–48).
Jenders, M., Kasneci, G., & Naumann, F. (2013). Analyzing and predicting viral tweets. In *WWW* (pp. 657–664).
Kefato, Z. T., Sheikh, N., Bahri, L., Soliman, A., Montresor, A., & Girdzijauskas, S. (2018). Cas2vec: Network-agnostic cascade prediction in online social networks. In *SNAMS* (pp. 72–79).
Kim, J. W. (2018). Rumor has it: The effects of virality metrics on rumor believability and transmission on Twitter. *New Media & Society*, *20*(12), 4807–4825.
Kupavskii, A., Ostroumova, L., Umnov, A., Usachev, S., Serdyukov, P., Gusev, G., et al. (2012). Prediction of retweet cascade size over time. In *CIKM* (pp. 2335–2338).
Lee, J., Lee, I., & Kang, J. (2019). Self-attention graph pooling. In *ICML* (pp. 3734–3743). PMLR.
Lee, R. K.-W., & Lim, E.-P. (2015). Measuring user influence, susceptibility and cynicalness in sentiment diffusion. In *ECIR* (pp. 411–422).
Lee, H., & Oh, H. J. (2017). Normative mechanism of rumor dissemination on Twitter. *Cyberpsychology, Behavior, and Social Networking*, *20*(3), 164–171.
Li, C., Ma, J., Guo, X., & Mei, Q. (2017). Deepcas: An end-to-end predictor of information cascades. In *WWW* (pp. 577–586).
Liao, D., Xu, J., Li, G., Huang, W., Liu, W., & Li, J. (2019). Popularity prediction on online articles with deep fusion of temporal process and content features. In *AAAI. Vol. 33. No. 01* (pp. 200–207).
Liu, F., Cheng, Z., Zhu, L., Gao, Z., & Nie, L. (2021). Interest-aware message-passing GCN for recommendation. In *WWW* (pp. 1296–1305).
Liu, Y., & Wu, Y.-F. B. (2020). Fned: A deep network for fake news early detection on social media. *ACM Transactions on Information Systems (TOIS)*, *38*(3), 1–33.
Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K.-F., et al. (2016). Detecting rumors from microblogs with recurrent neural networks. In *IJCAI* (pp. 3818–3824).
Ma, J., Gao, W., & Wong, K.-F. (2017). Detect rumors in microblog posts using propagation structure via kernel learning. In *ACL* (pp. 708–717).
Ma, J., Gao, W., & Wong, K.-F. (2018). Rumor detection on twitter with tree-structured recursive neural networks. In *ACL* (pp. 1980–1989).
Mercier, H. (2017). How gullible are we? A review of the evidence from psychology and social science. *Review of General Psychology*, *21*(2), 103–122.
Monti, F., Frasca, F., Eynard, D., Mannion, D., & Bronstein, M. M. (2019). Fake news detection on social media using geometric deep learning. In *ICLR*.
Nguyen, V.-H., Sugiyama, K., Nakov, P., & Kan, M.-Y. (2020). Fang: Leveraging social context for fake news detection using graph representation. In *CIKM* (pp. 1165–1174).
Pan, S., & Ding, T. (2019). Social media-based user embedding: A literature review. In *IJCAI* (pp. 6318–6324).
Pancha, N., Zhai, A., Leskovec, J., & Rosenberg, C. (2022). PinnerFormer: Sequence modeling for user representation at pinterest. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 3702–3712).
Pelrine, K., Danovitch, J., & Rabbany, R. (2021). The surprising performance of simple baselines for misinformation detection. In *WWW* (pp. 3432–3441).
Pennycook, G., & Rand, D. G. (2019). Lazy, not biased: Susceptibility to partisan fake news is better explained by lack of reasoning than by motivated reasoning. *Cognition*, *188*, 39–50.
Qiu, Z., Wu, X., Gao, J., & Fan, W. (2021). U-BERT: Pre-training user representations for improved recommendation. In *AAAI. Vol. 35. No. 5* (pp. 4320–4327).
Rath, B., Gao, W., & Srivastava, J. (2019). Evaluating vulnerability to fake news in social networks: A community health assessment model. In *ASONAM* (pp. 432–435).
Ratkiewicz, J., Conover, M., Meiss, M., Gonçalves, B., Patil, S., Flammini, A., et al. (2011). Truthy: Mapping the spread of astroturf in microblog streams. In *WWW* (pp. 249–252).
Rissola, E. A., Bahrainian, S. A., & Crestani, F. (2019). Personality recognition in conversations using capsule neural networks. In *WI* (pp. 180–187).
Sharma, K., Qian, F., Jiang, H., Ruchansky, N., Zhang, M., & Liu, Y. (2019). Combating fake news: A survey on identification and mitigation techniques. *TIST*, *10*(3), 1–42.
Shen, T. J., Cowell, R., Gupta, A., Le, T., Yadav, A., & Lee, D. (2019). How gullible are you? Predicting susceptibility to fake news. In *WebSci* (pp. 287–288).
Shen, H., Wang, D., Song, C., & Barabási, A.-L. (2014). Modeling and predicting popularity dynamics via reinforced poisson processes. In *AAAI* (pp. 291–297).
Shin, K., Kwak, H., Kim, S. Y., Ramström, M. N., Jeong, J., Ha, J.-W., et al. (2023). Scaling law for recommendation models: Towards general-purpose user representations. In *Proceedings of the AAAI conference on artificial intelligence. Vol. 37. No. 4* (pp. 4596–4604).
Song, Y.-Z., Chen, Y.-S., Chang, Y.-T., Weng, S.-Y., & Shuai, H.-H. (2021). Adversary-aware rumor detection. In *ACL* (pp. 1371–1382).
Su, X., Yang, J., Wu, J., & Zhang, Y. (2023). Mining user-aware multi-relations for fake news detection in large scale online social networks. In *Proceedings of the sixteenth ACM international conference on web search and data mining* (pp. 51–59).
Sun, M., Zhang, X., Zheng, J., & Ma, G. (2022). DDGCN: Dual dynamic graph convolutional networks for rumor detection on social media. In *AAAI. Vol. 36. No. 4* (pp. 4611–4619).
Sun, X., Zhou, J., Liu, L., & Wei, W. (2023). Explicit time embedding based cascade attention network for information popularity prediction. *Information Processing & Management*, *60*(3), Article 103278.
Tan, Y., Liu, F., Li, B., Zhang, Z., & Zhang, B. (2022). An efficient multi-view multimodal data processing framework for social media popularity prediction. In *ACM MM* (pp. 7200–7204).
Tsur, O., & Rappoport, A. (2012). What's in a hashtag? Content based prediction of the spread of ideas in microblogging communities. In *WSDM* (pp. 643–652).
Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(11).
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *NIPS. Vol. 30*.
Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *ICLR*.
Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. *science*, *359*(6380), 1146–1151.
Wald, R., Khoshgoftaar, T. M., Napolitano, A., & Sumner, C. (2013). Predicting susceptibility to social bots on twitter. In *IRI* (pp. 6–13). IEEE.
Wang, Y., Qian, S., Hu, J., Fang, Q., & Xu, C. (2020). Fake news detection via knowledge-driven multimodal graph convolutional networks. In *ICMR* (pp. 540–547).
Wang, Y., Wang, X., Ran, Y., Michalski, R., & Jia, T. (2022). CasSeqGCN: Combining network structure and temporal sequence to predict information cascades. *Expert Systems with Applications*, *206*, Article 117693.
Wang, J., Xie, H., Wang, F. L., Lee, L.-K., & Wei, M. (2023). Jointly modeling intra-and inter-session dependencies with graph neural networks for session-based recommendations. *Information Processing & Management*, *60*(2), Article 103209.
Warmsley, D., Waagen, A., Xu, J., Liu, Z., & Tong, H. (2022). A survey of explainable graph neural networks for cyber malware analysis. In *2022 IEEE international conference on big data* (pp. 2932–2939). IEEE.
Weng, L., Menczer, F., & Ahn, Y.-Y. (2014). Predicting successful memes using network and community structure. In *ICWSM* (pp. 535–543).

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks? In *International conference on learning representations*.
Xu, F., Sheng, V. S., & Wang, M. (2021). A unified perspective for disinformation detection and truth discovery in social sensing: A survey. *CSUR*, *55*(1), 1–33.
Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., & Leskovec, J. (2018). Hierarchical graph representation learning with differentiable pooling. In *NIPS. Vol. 31*.
Yu, L., Cui, P., Wang, F., Song, C., & Yang, S. (2015). From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics. In *ICDM* (pp. 559–568).
Zannettou, S., Sirivianos, M., Blackburn, J., & Kourtellis, N. (2019). The web of false information: Rumors, fake news, hoaxes, clickbait, and various other Shenanigans. *JDIQ*, *11*(3), 1–37.
Zhai, P., Yang, Y., & Zhang, C. (2023). Causality-based CTR prediction using graph neural networks. *Information Processing & Management*, *60*(1), Article 103137.
Zhang, Z., Bu, J., Ester, M., Zhang, J., Li, Z., Yao, C., et al. (2023). Hierarchical multi-view graph pooling with structure learning. *TKDE*, *35*(1), 545–559.
Zhang, Y., Lyu, T., & Zhang, Y. (2018). Cosine: Community-preserving social network embedding from information diffusion cascades. In *AAAI. Vol. 32. No. 1*.
Zhang, F., Wu, J., Zhang, P., Ma, R., & Yu, H. (2023). Detecting collusive spammers with heterogeneous graph attention network. *Information Processing & Management*, *60*(3), Article 103282.
Zhang, S., Yin, H., Chen, T., Hung, Q. V. N., Huang, Z., & Cui, L. (2020). Gcn-based user representation learning for unifying robust recommendation and fraudster detection. In *SIGIR* (pp. 689–698).
Zhang, Z., Yin, Z., Wen, J., Sun, L., Su, S., & Philip, S. Y. (2021). DeepBlue: Bi-layered LSTM for tweet popularity estimation. *TKDE*, *34*(10), 4737–4752.
Zhao, Q., Erdogdu, M. A., He, H. Y., Rajaraman, A., & Leskovec, J. (2015). Seismic: A self-exciting point process model for predicting tweet popularity. In *SIGKDD* (pp. 1513–1522).
Zhao, Z., Resnick, P., & Mei, Q. (2015). Enquiring minds: Early detection of rumors in social media from enquiry posts. In *WWW* (pp. 1395–1405).
Zhu, L., Gao, S., Pan, S. J., Li, H., Deng, D., & Shahabi, C. (2015). The pareto principle is everywhere: Finding informative sentences for opinion summarization through leader detection. In *Recommendation and search in social networks* (pp. 165–187).
Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., & Procter, R. (2018). Detection and resolution of rumours in social media: A survey. *CSUR*, *51*(2), 1–36.