Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-023-00953-5

A tale of two roles: exploring topic-specific susceptibility and influence in cascade prediction

Ninghan Chen[1] · Xihui Chen[1] · Zhiqiang Zhong[2] · Jun Pang[1,3]

Received: 10 February 2023 / Accepted: 28 June 2023

Abstract
We propose a new deep learning cascade prediction model CasSIM that can simultaneously achieve two most demanded objectives: *popularity prediction* and *final adopter prediction*. Compared to existing methods based on cascade representation, CasSIM simulates information diffusion processes by exploring users' dual roles in information propagation with three basic factors: users' *susceptibilities*, *influences* and *message contents*. With effective user profiling, we are the first to capture the topic-specific property of susceptibilities and influences. In addition, the use of graph neural networks allows CasSIM to capture the dynamics of susceptibilities and influences during information diffusion. We evaluate the effectiveness of CasSIM on three real-life datasets and the results show that CasSIM outperforms the state-of-the-art methods in popularity and final adopter prediction.

Keywords Cascade prediction · Information diffusion · Popularity prediction · Susceptibility · Influence · Social media

1 Introduction

On social media, people are sharing billions of posts, news and videos with their friends or followers every day. These sharing behaviours lead to the rapid diffusion of unprecedented amounts of information (Chen et al. 2019) in the form of *cascades*. The prevalence of information cascades exposes people to information of their interest faster, and meanwhile also amplifies the damage of false information such as

Responsible editor: Charalampos Tsourakakis.

✉ Jun Pang
jun.pang@uni.lu

[1] Department of Computer Science, University of Luxembourg, Esch-sur-Alzette, Luxembourg

[2] Faculty of Natural Sciences, Aarhus University, Aarhus, Denmark

[3] Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Esch-sur-Alzette, Luxembourg

rumours (Guarino et al. 2021). The COVID-19 pandemic gives us a chance to rediscover the importance of social media not only as networking platforms but also as an information source which can actually interfere with our everyday decisions (Xu et al. 2021). Thus, it is crucial to understand and forecast cascade dynamics to effectively promote useful messages, e.g., for viral marketing (Wang et al. 2015), and proactively control the impact of misinformation (Song et al. 2017). The problem of *cascade prediction* aims to achieve two objectives along this direction: *popularity prediction* and *adopter prediction*. We say that a user *adopts* a message and becomes an active adopter if the user shares the message from at least one of his/her friends. With an observation of early adopters, the goal of popularity prediction is to predict the number of final adopters while adopter prediction is to forecast who will adopt the message at a future time point. Final adopter prediction is required to enforce the real effectiveness of information diffusion in applications such as marketing and the vaccination campaign during the COVID-19 pandemic. In such cases, we need to ensure information reaches as many targeted users as possible in addition to a large number of recipients.

Cascade prediction has garnered attention from both industry and academy over the past decade (Cheng et al. 2014; Yu et al. 2015) and the solutions have evolved from the methods based on *diffusion models* (Panagopoulos et al. 2020) to those based on *cascade representation* (Chen et al. 2019). Diffusion model-based methods characterise the interpersonal influences between users and simulate the diffusion process through social relations. These methods are not scalable for large networks due to the repetitive simulations of diffusion models. Moreover, they rely on some unrealistic assumptions such as independent cascades and uniform influence probabilities between users (Panagopoulos et al. 2020). Therefore, despite their explainability, this class of methods are sub-optimal for cascade prediction. By contrast, the methods based on cascade representation characterise the features of observed early cascades instead of modelling diffusion processes. Machine learning models are employed for downstream predictions. These methods have become state-of-the-art due to their overwhelming prediction performance, especially with the recent success of deep learning. Compared to earlier methods using hand-crafted predictive features, deep learning allows for automatic extraction of cascade representations which capture the heterogeneous information embedded in cascades (Xu et al. 2021). For instance, the application of recurrent neural networks (RNN) and graph node embedding simultaneously captures the temporal rankings of early adopters and the structural properties of their neighbours in social graphs (Yang et al. 2019). Despite their promising performance, deep learning methods confront a few inherent challenges as repeatedly emphasised in the literature such as the imbalanced distribution of cascades (Tang et al. 2021) and cascade graph dynamics (Sun et al. 2022). Moreover, except for FOREST (Yang et al. 2019), they are designed either for popularity prediction or for microscopic prediction which infers the *next adopter*. Without modelling diffusion processes, they are thus sub-optimal for predicting final adopters.

In this paper, we aim to combine the advantages of the two classes of previous studies and apply deep learning to model the diffusion process of information on social media. This approach will allow us to get rid of the inherent challenges in embedding observed cascades, and efficiently achieve the two objectives using a single method. The key to diffusion process modelling is to capture the interpersonal influences between a user and his/her friends before adopting a message. Cao et al. (2020D) conducted the first attempt CoupledGNN by modelling the cascading effect only with users' influences. One shortcoming of this method is that it ignores the double roles simultaneously played by users in information diffusion: distributors and receivers which have been widely accepted in the literature (Panagopoulos et al. 2020; Wang et al. 2015). In this paper, our goal is thus to explore users' profiles of these two roles to improve the performance of cascade prediction. Specifically, we will address the following perspectives on modelling diffusion process which have not been well studied so far:

- A user's decision to forward a message should result from three factors: *message content*, *influences* of active friends and *susceptibility* of the user. Intuitively, a user's influence measures his/her ability to convince another user to share his/her message while susceptibility measures how likely the user gets influenced by other users (Panagopoulos et al. 2020; Wang et al. 2015).
- Users' influences and susceptibilities are not only user-specific but also *topic-specific*. This phenomenon has not been discussed before. Social media users, especially on platforms featured by microblogs such as Twitter and Sina, usually have multiple topics of interest and different sharing patterns. Suppose a sports news reporter with a hobby of pop music. He will be more influential for sports-related tweets than those about music. As an information receiver, the reporter will be more cautious to spread sports news compared to music-related tweets.
- Influences and susceptibilities are *context-dependent* (Wang et al. 2015). In other words, they spread through social relations during diffusion processes. A user will become more susceptible to a message when he/she sees that message shared by a larger number of users. Similarly, when more users have adopted the message a user shared, then the user becomes more influential to his/her friends due to the accumulated trust in the message.

To the best of our knowledge, we are the first to integrate users' topic-specific and context-dependent susceptibilities and influences into cascade prediction. We start by validating our hypothesis that users' influences and susceptibilities are topic-specific with our collected Twitter data. Then we propose a new deep learning cascade prediction model, which leverages the social network structure and simulates the propagation of messages from early adopters through social relations. The model can be effectively trained to achieve the two cascade prediction objectives at the same time. In this model, we explicitly embed users' susceptibility and influence profiles as two representation vectors. With graph neural networks (GNN) (Kipf and Welling 2017), we model the activation of users according to topic-specific susceptibilities and influences and the dynamics of susceptibilities and influences. Through comprehensive experiments with three real-life datasets, we show our model outperforms state-of-the-art baselines in both popularity and adopter prediction with almost all measurements.

2 Related works

2.1 Diffusion model-based methods

This line of methods iteratively run their diffusion models to simulate the information propagation process as viral contamination (Panagopoulos et al. 2020). Two typical diffusion models are used: Independent Cascade (IC) (Song et al. 2017; Wang et al. 2015) or Linear Threshold (LT) (Kempe et al. 2003). Earlier stochastic methods require manual assignment of influence probabilities for each user pair, which is not tractable in practice. To address the deficiency, embedding learning-based methods are proposed such as TIS (Wang et al. 2015) and EMBED-IC (Bourigault et al. 2016) and CELFIE (Panagopoulos et al. 2020). User-specific susceptibility and influence are represented as latent parameters which are estimated according to observed cascades. The activation of a user can thus be determined by his/her susceptibility and influence vectors. One advantage of such methods is to well characterise the diffusion process and output the activation state of each user. However, they suffer from high computation overhead and the strong assumptions on diffusion models make them sub-optimal for cascade prediction (Yang et al. 2019; Sun et al. 2022; Chen et al. 2019).

2.2 Generative methods

With the time stamps of users' sharing behaviours, a cascade of early adopters is abstracted as an event sequence and thus temporal point processes can be applied to simulate the arrivals of events. According to the employed point processes, we have two types of generative methods: the ones based on the Reinforced Poisson process (Shen et al. 2014) and those based on the self-exciting Hawkes process (Cao et al. 2017). Due to the assumption of temporal point processes, this line of methods over-simplify information diffusion and are thus limited in prediction performances.

2.3 Cascade representation-based models

This class of methods extract features of observed cascades as representation vectors and employ machine learning models to infer cascades dynamics. Earlier works rely on manually crafted features from user profiles (Cui et al. 2013) and message contents (Hong et al. 2011). Deep learning overtakes feature-engineering methods recently due to its overwhelming performance. DeepCas (Li et al. 2017) is the first end-to-end deep learning method for popularity prediction. It samples diffusion paths from cascade graphs and makes use of recurrent neural networks (RNN) to embed these sequential paths. Following DeepCas, a number of methods have been proposed by extending RNNs to calculate cascade representations (Wang et al. 2017; Yang et al. 2019; Wang et al. 2018). With social relations between adopters, some studies model cascades as cascade graphs and use various methods to calculate their embedding vectors with more effective sampling methods (Tang et al.

2021) or graph embedding methods (Chen et al. 2019; Sun et al. 2022; Xu et al. 2021). In spite of their promising performance, deep learning cascade prediction faces some inherent challenges as stated in Chen et al. (2019); Tang et al. (2021); Zhou et al. (2021). New methods are continuously developed to address them. For instance, Tang et al. (2021) addressed the impacts of hub structures and deep cascade paths in cascade graphs while (Chen et al. 2019) identified the challenges to combine cascade structures with temporal information. Zhou et al. (2021) studied the impact of the long-tailed distribution of cascade sizes on cascade prediction. In addition, except FOREST (Yang et al. 2019), deep learning based methods focus on either popularity prediction or microscopic prediction, i.e., forecasting the next single adopter, and thus cannot predict popularity and final adopters simultaneously. Cao et al. (2020D) proposed a different approach CoupledGNN by modelling the cascading effects with graph neural networks (GNN) (Kipf and Welling 2017), i.e., users' sharing behaviours are influenced by their neighbours in social networks. However, this method oversimplified the diffusion process by ignoring users' double roles in information diffusion and thus produced sub-optimal prediction performances. Inspired by Cao et al. (2020D), we explore users' dual roles as information receivers and distributors and propose a new deep learning model that can not only predict the size of cascades but also the final adopters.

3 Problem definition

Let $\mathcal{M}$ be a set of messages. We use the term "*message*" to refer to a piece of information that can be disseminated over social media. It can be a tweet on Twitter or an image on Instagram. In this paper, we focus on textual messages and our approach can be straightforwardly extended to other message types if their representations can be effectively calculated. For any message $m \in \mathcal{M}$, we have the set of *early adopters* that had shared this message up to the time t_0 after the message was first posted, denoted by $C_m^{t_0}$. The observation time t_0 depends on the requirements of downstream applications as well as the popularity of social media platforms. It can be of hours on Twitter and Sina, and years for citation networks. We use $G = (\mathcal{V}, \mathcal{E})$ to denote the social graph recording the social relations between users. Specifically, $\mathcal{V}$ is the set of nodes representing the set of users and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set edges indicating the social relations. The network can be directed or undirected depending on social media platforms. For instance, the following relationships on Twitter are directed while the friendships on Facebook are undirected.

3.1 Popularity prediction

The problem of *popularity prediction* is to predict the final number of adopters, i.e., $n_m^{\infty} = |C_m^{\infty}|$ according to the early adopters in $C_m^{t_0}$ and the social graph. In practical applications, the final time can be determined as a given fixed time t. Formally, given a set of messages $\mathcal{M}$ and their observed cascades $\{C_m^t|m \in \mathcal{M}\}$, the problem of popularity prediction can be solved by minimising the mean relative square error (MRSE) loss:

$$\mathcal{L}_{\mathbf{pop}} = \frac{1}{|\mathcal{M}|} \sum_{m\in\mathcal{M}} \left(\frac{\tilde{n}_m^\infty - n_m^\infty}{n_m^\infty} \right)^2 \tag{1}$$

where $\tilde{n}_m^\infty = f_{\Theta,G}(C_m^{t_0})$. Note that $f_{\Theta,G} : \mathcal{V}^\mathcal{P} \rightarrow \mathbb{Z}$ is the regression function customised to graph G and parameterised by the set of trainable parameters Θ where $\mathcal{V}^\mathcal{P}$ denotes the power set of $\mathcal{V}$. It takes the set of early adopters as input and outputs the predicted final size of the cascade. We select relative error over the absolute error to avoid the potential negative impacts of the various cascade sizes, e.g., exposing unnecessary weights to more popular messages.

3.2 Final adopter prediction

The goal is to predict the set of users who will forward the target message. This is different from the *microscopic cascade prediction* in the literature (Yu et al. 2015; Yang et al. 2019) which aims to predict the next adopter according to the observed ones. The problem of final adopter prediction can be solved by minimising the following loss function:

$$\mathcal{L}_{\mathbf{adp}} = -\frac{1}{|\mathcal{M}|} \sum_{m\in\mathcal{M}} \left(\sum_{v\in C_m^\infty} \log q_{\Theta,G}(C_m^{t_0}, v) + \sum_{v\notin C_m^\infty} \log \left(1 - q_{\Theta,G}(C_m^{t_0}, v)\right) \right)$$

where $q_{\Theta,\mathcal{G}} : \mathcal{V}^\mathcal{P} \times \mathcal{V} \rightarrow [0, 1]$ is the trainable function customised to social graph G and parameterised by Θ that predicts the probability of a specific user adopting the message. In the end, we can select the users with probabilities larger than a predefined threshold as the output set of final adopters. An alternative is to output the top $\tilde{n}_m^\infty$ users with the largest activation probabilities.

4 Topic-specific susceptibility and influence

In this section, we will validate our hypotheses that a user's susceptibility and influence vary according to the topics of messages. This hypothesis actually contains an implicit claim that users adopt messages on multiple topics on social media. In other words, users have their own topic preferences. We start with validating this claim and then examine the dependence of susceptibility and influence on topics. Before our validation, we present our collection of necessary social media data to support our analysis. We conduct our own data collection instead of using existing publicly available datasets because they do not have all the inputs over a sufficiently long period required for our analysis.

4.1 Twitter data collection

We use Twitter as the data source of our analysis because of its popularity and friendly data-sharing interfaces for data analysts. The metadata downloaded with retweeted messages includes the original tweets' IDs which allow us to retrieve the corresponding cascades. To efficiently obtain a sufficiently large set of users, we refer to the Twitter dataset published in Chen et al. (2022). The dataset contains tweets related to COVID-19 vaccination from four Western European countries: Germany, France, Belgium and Luxembourg. In order to obtain the Twitter users, we first crawled the tweets according to the published tweet IDs and extracted the account IDs of the originating users. Then for each originator, we queried and downloaded its followers and followees, according to which we successfully constructed the social network. Specifically, if user v follows user v', then an edge is created from v to v'. We calculated the largest weakly connected subgraph of the social network as the final set of users to eliminate isolated users and ensure interconnectivity between users. In the end, we downloaded the tweets together with their metadata from the remaining users in two time periods each of which spans three months. One period starts from March 1st, 2020 while the other starts from March 1st, 2021. By these two periods separated by 1 year, we can examine the consistency of our empirical analysis over time. We summarise the statistics of the final social networks and tweets in Table 6 in "Appendix A". Note that the numbers regarding tweets count those shared or posted by users. In Twitter, sharing an existing tweet generates a new tweet with a unique ID and metadata containing the ID of the original one.

In this paper, we focus on the texts of retweeted messages and thus remove all other content such as '@', hyperlinks and 'RT' which stands for 'retweet'. For quoted tweets, we only consider the quoted tweets and ignore the comments added by users. In our analysis, we only consider the users who have shared more than 5 different messages in our dataset to ensure the reliability of our analysis. In practice, users' social connections evolve over time by adding new connections or removing existing ones. In the following analysis, we do not consider this dynamic nature of social graphs by assuming that Twitter users do not frequently cancel their following-ships and their topics of interest stay relatively stable. In our collection, the social graph is built according to the data collected in early 2020.

4.2 Users' topic preferences

We validate our observation that users simultaneously participate in discourses of multiple topics on social media.

4.2.1 Topic modelling

Topic modelling has upgraded from traditional LDA method to machining learning methods (Greene and Cunningham 2006). In Zhang et al. (2022), it has been

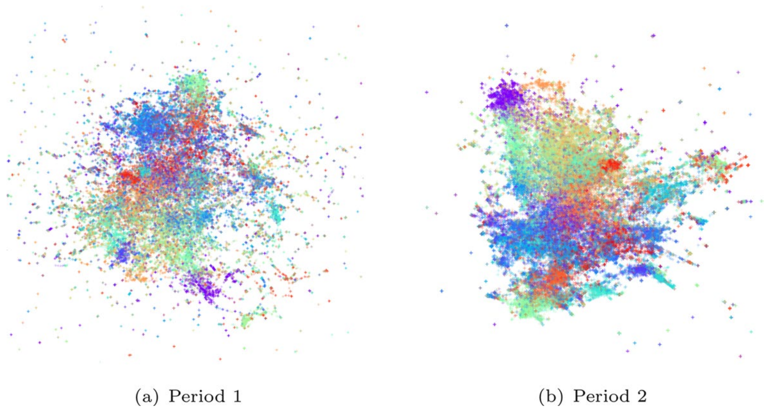
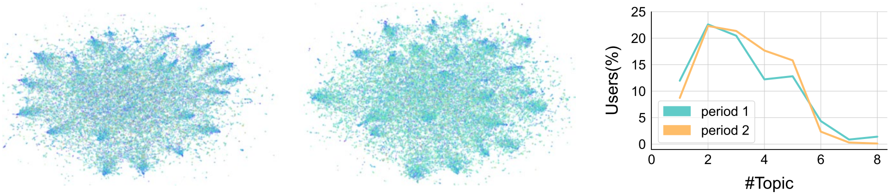


Fig. 1 Clustering retweets into topics

shown that the combination of high-quality text embeddings and clustering methods is more efficient in learning topics of the same quality as complex neural network models. In this paper, we adopt the most effective combination in Zhang et al. (2022), i.e., RoBERTa+UMAP+K-Means, to cluster tweets with similar topics. RoBERTA (Liu et al. 2019) is a pre-trained transformer-based text embedding method and UMAP (McInnes and Healy 2018) is used to conduct dimension reduction of text embeddings while K-Means is one of the most widely used classical clustering methods. Besides textual tweets, the number of topics is required as an input parameter. In our dataset, we only consider the textual content of messages. As a result, original tweets and retweets will have the same embeddings.

We classify the collected tweets in each period with the selected topic modelling method. After several trials, we select 25 topics due to the relatively higher quality of the output clusters. In the end, we have 25 clusters of retweets, the i-th of which is denoted by $\mathcal{S}_i$. In Fig. 1, we depict retweets as data points and layout them according to their text embedding vectors mapped to a 2-D space with UMAP (McInnes and Healy 2018) in the two selected periods. The colours indicate their clusters. With the widely accepted measurements: C_V and Normalised Pointwise Mutual Information (NPMI), we measure the coherence values of extracted topics which are 0.649 and 0.138 for the first period, and 0.704 and 0.142 for the second period. According to the criteria adopted in topic modelling works such as Zhang et al. (2022), these numbers indicate a more than good topic coherence. We extract the representative keywords with their TFIDF rankings, and manually examine the topics of the clusters. We find that in general, the tweets in these clusters are about specific topics such as the death and infection numbers of COVID-19, Black Lives Matter movement and COVID-19 policies (see "Appendix E" for the top 10 keywords and our coarse annotation).



(a) Visualisation (Period 1) (b) Visualisation (Period 2) (c) Distribution of the number of topics of interest.

Fig. 2 User topic preferences and distribution

4.2.2 User topic preference

We represent the topic preferences of user v by a vector $\mathbf{v}$ by counting the proportions of his/her retweets in each topic. Formally, let $\mathcal{R}_v$ be the set of retweets of user v and recall that $\mathcal{S}_j$ is the set of retweets in the j-th topic, then the j-th element of $\mathbf{v}$ is calculated as $\frac{|\mathcal{R}_v \cap \mathcal{S}_j|}{|\mathcal{R}_v|}$. In Fig. 2a, b we layout users as data points according to their preference vectors mapped to 2 dimensions in the two periods with UMAP (McInnes and Healy 2018). We can see that users' vectors scatter all over the space. This confirms the diversity of Twitter users' topics of interest. Another observation is that users cluster naturally, which indicates users group with others with similar interests. Both observations help demonstrate the validity of the users' topic vectors we calculate. We consider a user is interested in the j-th topic if the j-th element of his representation vector is over 0.08 which is double the value of the null model where users have equal preferences over the 25 topics. Figure 2c shows the distributions of the number of topics users prefer in the two periods. We observe that about 86% users actively participate in at least 2 topics. On average, each user is interested in 3 topics. According to the above discussion, we can conclude that users are interested in multiple topics.

4.3 Topic-specific susceptibility and influence

Whether a user retweets a message is determined by his/her susceptibility and the influences received from his/her followees who have shared the message. We hypothesise that the interplay between susceptibility and influences is not only user-specific but also *topic-specific*. Many methods have been proposed to learn the latent representations for users' susceptibilities and influences according to past observed cascades (Panagopoulos et al. 2020; Wang et al. 2015). However, we cannot validate our hypothesis by directly comparing the representations extracted from past cascades of different topics. This is because the learning processes on different topics are independent. Therefore, the learned representations do not belong to the same space and are not comparable. Therefore, we select an intuitive approach based on a heuristic utilised in the literature (Bourigault et al. 2016) that if users' susceptibility and influence are topic-specific, we will have two observations:

1. As an information receiver, a user will have different patterns regarding sharing messages from his/her followees between topics;
2. As an information distributor, a user's followers will have different patterns regarding sharing messages retweeted or posted by the user.

If these two differences are present in our dataset, we can infer that the interaction of a user's susceptibility and influence varies between topics. After a user shares a message, the user will have an influence on each follower's decision whether to share the message. According to this intuition, we use the frequency with which a follower forwards messages after the user's sharing to measure the strength of the interplay between the user's influence and the follower's susceptibility. In the following, we first present our measurements for a user's *susceptibility pattern* as a receiver and his/her *influence pattern* as a distributor, and then discuss our analysis of our dataset.

4.3.1 Measuring topic-specific susceptibility

Intuitively, given a topic, we use the relative frequencies with which a user forwards messages from his/her followees to quantitatively capture the user's sharing pattern as an information receiver. Suppose a user v with the set of his/her followees $\mathcal{U}_v^+ = \{v' \in \mathcal{V} | (v, v') \in \mathcal{E}\}$. We assume a pre-defined order between the followees of user v and use v_i to denote the i-th followee. Let $\mathcal{R}_{v,j}$ be the set of tweets that u retweeted about the j-th topic, and $t(m, v)$ to denote the time when m is posted or retweeted by user v. The susceptibility vector of v for topic j is denoted by $\mathbf{s}_{v,j} \in \mathbb{Z}^{|\mathcal{U}_v^+|}$ whose i-th element is the number of messages retweeted by the i-th followee in $\mathcal{U}_v^+$ before v retweets the same message, i.e, $|\{m \in \mathcal{R}_{v,j} \cap \mathcal{R}_{v_i} | t(m, v) > t(m, v_i)\}|$.

4.3.2 Measuring topic-specific influence

We use the frequencies with which a user's followers share his/her retweeted messages to quantify the influence patterns of the user as an information distributor. Suppose a user v with the set of followers $\mathcal{U}_v^- = \{v' \in \mathcal{V} | (v', v) \in \mathcal{E}\}$ ranked according to a pre-defined order. Let $\mathbf{h}_{v,j}$ denote the influence vector of user v of the j-th topic. Then the i-th element is the number of retweets conducted by the follower v_i in $\mathcal{U}_v^-$ after seeing the same message posted or retweeted by user v, i.e., $|\{m \in \mathcal{R}_{v,j} \cap \mathcal{R}_{v_i} | t(m, v) < t(m, v_i)\}|$. Similar to the definition of topic-wise susceptibility similarity, we also consider the top K favourite topics of user v, i.e., $\mathcal{T}_K^v$. Then the *topic-wise influence similarity* of user v is defined as follows:

$$IP@K(v) = \frac{2}{K \cdot (K-1)} \sum_{j,k \in \mathcal{T}_K^v \wedge j<k} \frac{\mathbf{h}_{v,j} \cdot \mathbf{h}_{v,k}}{\|\mathbf{h}_{v,j}\| \cdot \|\mathbf{h}_{v,k}\|}. \tag{3}$$

The domain of $IP@K$ is between 0 and 1. A lower value indicates a users' influence varies more between topics.

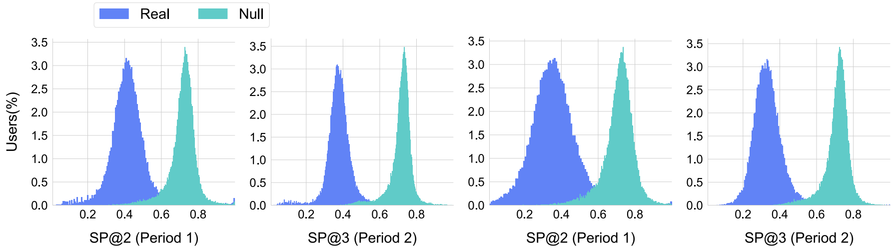


Fig. 3 Distributions of susceptibility pattern $SP@K$

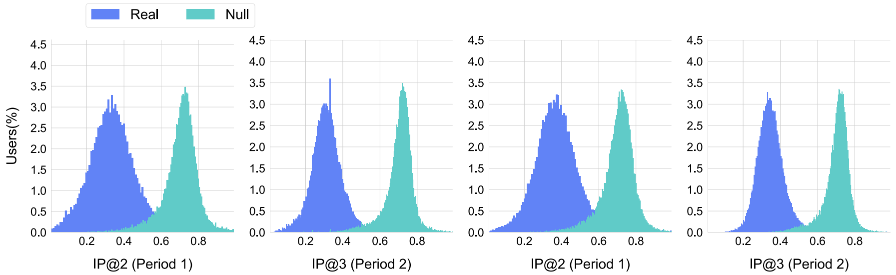


Fig. 4 Distributions of influence pattern $IP@K$

4.3.3 Experimental analysis

We re-use the topics extracted in Sect. 4.2 to analyse the topic dependence of susceptibility and influence. In Figs. 3 and 4, we show the distributions of $SP@K$ and $IP@K$ values over the users when K is set to 2 and 3, respectively. For either measurement, we construct a null model as a reference to capture the distributions when the topic-specific phenomenon is absent. Take susceptibility patterns as an example. For any user v and each topic (e.g., the j-th topic), we construct a null vector $\mathbf{s}'_{v,j}$. Its k-th element is a uniformly sampled random number between 0 and $|\mathcal{R}_{v_k,j}|$ representing the number of messages of the j-th topic of user $v_k \in \mathcal{U}_v^+$ that have ever been shared by user v. A general observation is that users' susceptibility and influence patterns roughly follow normal distributions. The curves of the distributions become narrower and shift right when larger K values are set. This is natural that more topics considered will lead to smaller average mutual similarity. We can see with all the selected K values, users have smaller values for both susceptibility pattern and influence pattern than the null models. On average, users' $IP@K$ and $SP@K$ fall into the range between 0.3 and 0.4 which are only half of those when susceptibilities and influences are not topic-specific. The difference indicates that users' sharing behaviours and their influences on friends differ between the topics of their interest.

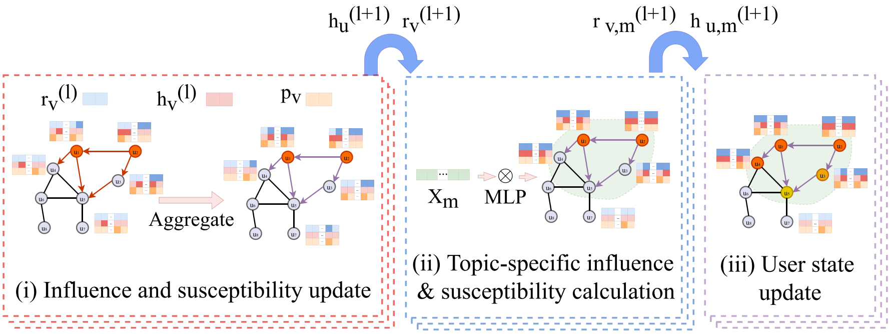


Fig. 5 Framework of the CasSIM model

5 Our CasSIM model

The propagation of a message can be interpreted as a process of multiple sequential generations. In any generation, each user first updates his/her influence and susceptibility according to the current activation states of users. Then the user decides whether to forward the targeted message according to his/her updated susceptibility and the influences of his/her friends who have forwarded the message. Inspired by CoupledGNN (Cao et al. 2020D), we use multi-layered graph neural networks to model this iterative process. We depict our framework in Fig. 5. At each layer, three sequential tasks are accomplished. The first task is to update susceptibility and influence by aggregating the profiles of social network neighbours. This task actually simulates the spread of influence and susceptibility and thus captures their context-dependence property. The second task is to calculate the topic-specific influence and susceptibility according to the user's topic preferences and the target message's content. The last task is to update each user's activation state by aggregating the interplay between his/her susceptibility and the influences of all the active friends. Note that the number of layers can to some extent indicate the depth of propagation simulated in our model. According to the small world property in social networks, a relatively small number of layers is sufficient to cover the major component of the network and enforce accuracy. We also add a self-activation mechanism to simulate that users adopt the message to further overcome the impact of this hyperparameter.

For each user $v \in \mathcal{V}$, we use $State_v \in [0, 1]$ to store his/her activation state indicating the probability that user v is activated. Furthermore, for each generation ℓ, user v is associated with three embedding vectors $\mathbf{r}_v^{(\ell)}$, $\mathbf{h}_v^{(\ell)}$ and $\mathbf{p}_v$ indicating his/her susceptibility, influence and topic preferences, respectively.

5.1 Influence and susceptibility update

As users' influences and susceptibilities propagate through social relations, we make use of a graph neural network to first aggregate the profiles from their friends and then combine the aggregation with their own profiles. We start by describing the update of susceptibility vectors. We use the idea of graph attention networks (Velickovic et al.

2018) to take into account the various contributions of friends to the update. Let $\mathcal{N}(v)$ be the neighbours of user v, i.e., $\{v' \in \mathcal{V} | (v, v') \in \mathcal{E}\}$. Formally, the aggregated susceptibility of user v can be calculated as follows:

$$\mathbf{a}_{v,s}^{(\ell)} = \mathbf{W}^{(\ell)} \sum_{v' \in \mathcal{N}(v)} \mathbf{h}_{v'}^{(\ell)} \cdot \mathrm{StateGate}(State_{v'}^{(\ell)}) \cdot \phi_{v,v'}^{(\ell)} \tag{4}$$

where $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d_r^{(\ell)} \times d_r^{(\ell+1)}}$ is the weight matrix and $d_r^{(\ell)}$ defines the dimension of a user's susceptibility vector at the ℓ-th layer, i.e., $\mathbf{r}_v^{(\ell)}$. The function StateGate() is the state-gating mechanism (Cao et al. 2020D) to reflect the non-linearity of activation states. In our implementation, we use a 2-layered MLP. The attention $\phi_{v,v'}^{(\ell)}$ calculates the contribution of the influence of user v's neighbour v'. This attention is determined not only by v''s susceptibility but also by v's influence vector. Formally, the attention is calculated as follows:

$$\phi_{v,v'}^{(\ell)} = \frac{\exp(\mathbf{e}_{v,v'}^{(\ell)})}{\sum_{v'' \in \mathcal{N}(v)} \exp(\mathbf{e}_{v,v''}^{(\ell)})} \tag{5}$$

where $\mathbf{e}_{v,v'}^{(\ell)} = \boldsymbol{\psi}_s^{(\ell)}\left(\mathbf{W}_r^{(\ell)}\mathbf{r}_{v'}^{(\ell)} \parallel \mathbf{W}_h^{(\ell)}\mathbf{h}_v^{(\ell)}\right)$. Note that $\parallel$ is the concatenation function of two vectors, and $\boldsymbol{\psi}_s \in \mathbb{R}^{d_h^{(\ell)} + d_r^{(\ell)}}$ where $d_h^{(\ell)}$ is the dimension of influence vectors at layer ℓ. Moreover, $\mathbf{W}_h^{(\ell)} \in \mathbb{R}^{d_h^{(\ell)} \times d_h^{(\ell)}}$ and $\mathbf{W}_r^{(\ell)} \in \mathbb{R}^{d_r^{(\ell)} \times d_r^{(\ell)}}$ are two weight matrices. In the end, we combine the aggregated susceptibility of neighbours with the user's own susceptibility:

$$\boldsymbol{r}_v^{(\ell+1)} = \mathrm{relu}\left(\mathbf{W}^{(\ell)}\mathbf{h}_v^{(\ell)} + \boldsymbol{a}_{v,s}^{(\ell)}\right) \tag{6}$$

where $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d_r^{(\ell+1)} \times d_s^{(\ell+1)}}$ and relu is the non-linear activation function. The update of user v's influence is similar to that of his/her susceptibility. We first aggregate the influence of his/her friends according to their activation states with attention networks. The aggregated influence $\mathbf{a}_{v,h}^{(\ell)}$ is calculated as follows:

$$\mathbf{a}_{v,h}^{(\ell)} = \mathbf{W}^{(\ell)} \sum_{v' \in \mathcal{N}(v)} \mathbf{h}_{v'}^{(\ell)} \cdot \mathrm{StateGate}(State_v^{(\ell)}) \cdot \lambda_{v,v'}^{(\ell)} \tag{7}$$

where $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d_h^{(\ell)} \times d_h^{(\ell+1)}}$ is the weight matrix. We calculate the attention $\lambda_{v,v'}^{(\ell)}$ as follows:

$$\lambda_{v,v'}^{(\ell)} = \frac{\exp(\mathbf{o}_{v,v'}^{(\ell)})}{\sum_{v'' \in \mathcal{N}(v)} \exp(\mathbf{o}_{v,v''}^{(\ell)})}, \tag{8}$$

where $\mathbf{o}_{v,v'}^{(\ell)} = \boldsymbol{\psi}_h^{(\ell)}(\mathbf{W}_h^{(\ell)}\mathbf{h}_{v'}^{(\ell)} \parallel \mathbf{W}_r^{(\ell)}\mathbf{r}_v^{(\ell)})$. Note that $\boldsymbol{\psi}_h \in \mathbb{R}^{d_h^{(\ell)} + d_r^{(\ell)}}$ and $\mathbf{W}_h^{(\ell)} \in \mathbb{R}^{d_h^{(\ell)} \times d_h^{(\ell)}}$ and $\mathbf{W}_r^{(\ell)} \in \mathbb{R}^{d_r^{(\ell)} \times d_r^{(\ell)}}$ are two weight matrices which are different from those used in updating susceptibility. The influence vector of user v at the layer $\ell + 1$ is calculated as follows:

$$\mathbf{h}_v^{(\ell+1)} = \mathrm{relu}(\mathbf{W}^{(\ell)}\mathbf{h}_v^{(\ell)} + \boldsymbol{a}_{v,h}^{(\ell)}) \tag{9}$$

5.2 Calculating topic-specific influence and susceptibility

We show how to customise a user's susceptibility and influence according to the topic of the message under diffusion in order to capture their topic-specific property. Suppose $m \in \mathcal{M}$ is the message being propagated. We take user $v \notin C_m^{t_0}$ at ℓ-th generation as an example and illustrate how to convert the vectors $\mathbf{r}_v^{(\ell)}$ and $\mathbf{h}_v^{(\ell)}$ into $\mathbf{r}_{v,m}^{(\ell)}$ and $\mathbf{h}_{v,m}^{(\ell)}$. We use $\mathbf{x}_m \in \mathbb{R}^{d_x}$ to denote the embedding vector of message m. As emphasised previously, in this paper, we concentrate on messages in the form of texts and the model can be extended to integrate other formats such as images if their representations can be effectively calculated. In our model, we use the pre-trained RoBERTa model (Liu et al. 2019) to calculate the embedding vectors of textual messages.

As empirically validated in the previous section, most users have multiple topics of interest in social media and their preferences vary between topics. Although the focus of the topics may shift over time as pointed out in Yuan et al. (2020), users' interests remain relatively stable. For instance, a sports news reporter may switch from reporting a local football team to national teams due to the opening of the FIFA World Cup, but the topic still remains around football. Users' topic preferences are extracted from their past sharing behaviours. We use $\mathbf{p}_v \in \mathbb{R}^{d_p}$ to denote the embedding vector for his/her topic preferences. Intuitively, given a targeted message m, we capture its related topic by referring to users' past topic preferences and utilise an MLP module to calculate the adjustments that should be exposed to the user's susceptibility and influence vectors. Starting with susceptibility, we calculate the corresponding topic-specific susceptibility vector as follows:

$$\mathbf{r}_{v,m}^{(\ell)} = \mathbf{W}^{(\ell)}\left(\mathrm{MLP}(\mathbf{p}_v \parallel \mathbf{x}_m) \circ \mathbf{r}_v^{(\ell)}\right) \tag{10}$$

where $\circ$ represents the dot product of two vectors. In addition, MLP is a multi-layer perceptron with an input vector of dimension $d_p + d_x$ and outputs a vector of dimension $d_r^{(\ell)}$. The weight matrix $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d_{r,m}^{(\ell)} \times d_r^{(\ell)}}$ conducts the linear transformation of the susceptibility vector to dimension $d_{r,m}^{(\ell)}$. Similarly, we have the user's topic-specific influence calculated as follows:

$$\mathbf{h}_{v,m}^{(\ell)} = \mathbf{W}^{(\ell)}\left(\mathrm{MLP}(\mathbf{p}_v \parallel \mathbf{x}_m) \circ \mathbf{h}_v^{(\ell)}\right). \tag{11}$$

Note $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d_{h,m}^{(\ell)} \times d_h^{(\ell)}}$ and the output of MLP has the dimension of $d_h^{(\ell)}$.

5.3 User state update

With users' topic-specific susceptibility and influence, we can model their interplay which changes their activation states. The influences of each user's active neighbours are first aggregated as the total amount of topic-specific influences exposed to the user. Then we use an MLP module to capture the likelihood of the user adopting the message only according to the exposed influences, denoted by $\gamma_v^{(\ell)}$. We use

$$\gamma_v^{(\ell)} = \text{sigmoid}\Big(\text{MLP}\Big(\Big(\sum_{v' \in \mathcal{N}(v)} \mathbf{h}_{v,x}^{(\ell)} \cdot State_{v'}^{(\ell)}\Big) \parallel \mathbf{r}_{v,m}^{(\ell)}\Big) + \beta_v\Big). \tag{12}$$

where $\beta_v \in \mathbb{R}$ is a self activation parameter. Intuitively, the probability is dependent on the user's topic preferences. In our model, we use a one-layer MLP followed by a sigmoid function to capture this dependence. In the end, we combine the above activation probability with the user's current activation status into the user's new activation state:

$$State_v^{(\ell+1)} = \begin{cases} 1, & \text{if } v \in C_m^{t_0} \\ \text{sigmoid}\big(\mu_1^{(\ell)} State_v^{(\ell)} + \mu_2^{(\ell)} \gamma_v^{(\ell)}\big), & \text{if } v \notin C_m^{t_0}. \end{cases} \tag{13}$$

Note that $\mu_1^{(\ell)}, \mu_2^{(\ell)} \in \mathbb{R}$ are two weight parameters which are to be trained. The initial state, i.e., $State_v^{(0)}$, is set to 1 if $v \in C_m^{t_0}$ or 0, otherwise. In the end, we calculate the final size of the cascade $\tilde{n}_m^{\infty}$ as $\sum_{v \in \mathcal{V}} State_v$.

5.4 User profiling

From the above discussion, we can see that our model uses three input vectors for each user v at the 0-th layer: $\mathbf{p}_v^{(0)}$, $\mathbf{r}_v^{(0)}$ and $\mathbf{h}_v^{(0)}$. A few methods have been proposed in the literature to learn users' susceptibility and influence embedding from users' sharing history (Wang et al. 2015; Panagopoulos et al. 2020). In this paper, we pre-train a simple but effective model to prepare the three types of initial vectors. Suppose we have the cascades for the past messages in $\mathcal{M}_{hist}$. We interpret them as the ultimate states of users in the corresponding information diffusion processes. In other words, for each $m \in \mathcal{M}_{hist}$, we have the final cascade C_m^{∞}. We set $State_v^{\infty} = 1$ if $v \in C_m^{\infty}$ and 0 otherwise. We calculate the activation state for each user $v \in C_m^{\infty}$ based on his/her topic-specific susceptibility and his/her active friends' influence, which is denoted by $\widetilde{State}_{v,m}$. Formally,

$$\widetilde{State}_{v,m} = \text{sigmoid}\Big(\boldsymbol{\alpha} \cdot \sum_{v' \in \mathcal{N}(v) \cap C_m^{\infty}} \big(\text{MLP}(\mathbf{p}_v^{(0)} \parallel \mathbf{x}_m) \circ (\mathbf{h}_{v'}^{(0)} \parallel \mathbf{r}_v^{(0)})\big)\Big) \tag{14}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{d_r^{(0)} + d_h^{(0)}}$ and MLP outputs a vector of dimension $d_r^{(0)} + d_h^{(0)}$. In the end, $\mathbf{p}_v^{(0)}$, $\mathbf{r}_v^{(0)}$ and $\mathbf{h}_v^{(0)}$ are trained by minimising the objective function:

$$\mathcal{L}_{initial} = -\frac{1}{|\mathcal{M}_{hist}|} \sum_{m \in \mathcal{M}_{hist}} \sum_{v' \in C_m^{\infty}} \log(\widetilde{State}_{v,m}). \tag{15}$$

There may exist users who do not participate in any cascades. For these users, we set the neutral vectors $\mathbf{0}$ to these users as their three profile vectors.

5.5 Model training

In order to achieve the two objectives of cascade prediction: popularity and final adopter prediction, we aggregate the two corresponding objective functions into our final loss function to guide the parameter optimisation: $\mathcal{L} = \theta_1 \mathcal{L}_{adp} + \theta_2 \mathcal{L}_{pop} + \theta_3 \mathcal{L}_{reg}$

where θ_1, θ_2 and θ_3 are hyperparameters. The $\mathcal{L}_{reg}$ is added for the purpose of regularisation as an L2 norm for all model parameters.

6 Experimental evaluation

6.1 Datasets

We leverage four real-life datasets in our experiments: Sina, AMINER, Twitter2012 and Twitter2020. Sina and AMINER are publicly available and widely exploited in the validation of previous works related to cascade prediction (Li et al. 2017; Cao et al. 2020D). The Twitter2020 dataset is an extension of our collection described in Sect. 4 while Twitter2012 is a public Twitter dataset collected in 2012 (Weng et al. 2013). Each dataset has two components: a social graph and a text dataset consisting of diffused messages. We select these datasets to ensure a comprehensive evaluation that covers as many practical scenarios as possible. Sina and Twitter represent the social media platforms characterised by microblogs. The users of the Sina dataset are more densely connected. This dense social graph will benefit cascade prediction models with a more complete view of the sources of influences. AMINER is a citation network instead of social media and stores the citation relations between academic authors. We use AMINER to test whether our CasSIM model can also predict the cascades in more general settings. Moreover, in order to check the performance of our model for different lengths of observation periods, i.e., t_0, for each dataset, we construct three sets of cascades by cutting the cascades according to thr ee given time periods. For Twitter and Sina, due to their fast propagation speed, the observation periods are set to 1 h, 2 h and 3 h. For AMINER, we select 1 year, 2 years and 3 years. More details can be found in "Appendix B" and the detailed statistics are summarised in Table 7 in "Appendix B".

6.2 Baselines

Considering the large number of methods for macroscopic and microscopic prediction, we select *representative* methods for comparison. A method is representative if it is typical for a class of methods or claims strong performances. For instance, we use SEISMIC (Zhao et al. 2015) and feature-based methods (Cao et al. 2020D) as representative baselines for machine learning methods without deep learning. We reuse the implementation of these models whenever they are accessible and conduct our own implementation otherwise. A brief description of our baselines can be found in "Appendix C".

6.3 Experimental settings

6.3.1 Evaluation measurements

We use three widely adopted measurements to evaluate the prediction performances regarding popularity. MSLE (Mean square log-transformed error) is a standard evaluation metric (Chen et al. 2019) defined as: $MSLE = \frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}}(\log n_m^\infty - \log \tilde{n}_m^\infty)^2$. We use mean absolute percentage error (MAPE) and wrong percentage error (WroPerc) which is introduced and used in Cao et al. (2020D) to evaluate prediction performance in terms of relative errors. MAPE measures the average relative errors and is defined as: $MAPE = \frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}}\frac{|\tilde{n}_\infty^m - n_\infty^m|}{n_\infty^m}$. WroPerc measures the average percentage of cascades that are poorly predicted and is defined as: $WroPerc = \frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}}\mathbb{1}\left(\frac{|\tilde{n}_\infty^m - n_\infty^m|}{n_\infty^m} \geq \epsilon\right)$. We set the threshold to 0.5 in our experiments. Note that $\mathbb{1}(*)$ is an indication function which outputs 1 when the input proposition is true or 0 otherwise. For each measurement, a lower value indicates better prediction performance.

With regard to evaluating the prediction performance of final adopters, we use the standard metrics: precision, recall and F1 score.

6.3.2 Hyperparameter settings

For each of the three datasets, we randomly split it into training, validation and testing sets according to the ratio 8:1:1. For the text embedding model RoBERTa, we utilise the implementation XLM-RoBERTa (Conneau et al. 2020). We set the maximum size of input strings to 128, and the length of text embedding is 768. For all models including the benchmark models, we tune their hyperparameters to obtain the best performance on validation sets. Early stopping is employed for tuning when validation errors do not decline for 20 consecutive epochs. The learning rate and L2 coefficient are chosen from $10^{-1}, 10^{-2}, \ldots, 10^{-8}$. The hidden units for MLPs are chosen from {32, 64}. The batch size is 32. We train our model for 500 epochs and utilise Adam (Kingma and Ba 2015) for optimisation. We use the first three months' cascades in Sina and Twitter2020 dataset to pre-train users' initial profiles: their topic preference, susceptibilities and influences. For the Twitter2012 dataset, the first week's cascades are used. For the AMINER dataset, the first 2 years' cascades are used. All hyperparameters remain the same as they are recommended in the original papers or the published source codes.

6.4 Overall prediction performance

We compare the performance of our CasSIM model to the baselines for both the two cascade objectives: popularity prediction and final adopter prediction. As discussed previously, not all baselines can achieve these two objectives simultaneously. As a result, for each objective, we compare with the baselines that can achieve the

Table 1 Popularity prediction performance on Sina dataset

Model	1 hour			2 hours			3 hours		
	MSLE	MAPE	WroPerc (%)	MSLE	MAPE	WroPerc (%)	MSLE	MAPE	WroPerc (%)
SEISMIC	5.774	–	50.93	5.688	–	47.84	5.223	–	42.00
Feature-based	4.672	0.359	41.53	4.165	0.315	38.48	4.052	0.308	31.96
DeepCas	3.578	0.291	32.26	3.421	0.288	28.74	3.139	0.270	18.58
DeepHawkes	2.894	0.289	26.21	2.551	0.280	25.89	2.240	0.268	17.57
CasCN	2.749	0.285	27.36	2.442	0.283	25.56	2.181	0.279	17.23
CoupledGNN	2.289	0.242	23.60	2.254	0.236	17.96	2.037	0.223	14.27
CasSeqGCN	2.281	0.252	23.96	2.282	0.239	18.43	2.048	0.224	13.54
FOREST	*2.156*	0.238	20.05	*2.136*	0.235	18.14	1.995	0.230	13.49
CasFlow	2.248	0.239	20.68	2.195	*0.221*	16.79	1.982	0.215	12.10
TempCas	2.290	*0.226*	**18.23**	2.208	0.229	**14.73**	1.960	*0.209*	*11.26*
CasSIM	**2.148**	**0.221**	*19.46*	**2.126**	**0.217**	*14.94*	**1.919**	**0.202**	**11.04**

Table 2 Popularity prediction performance on AMINER dataset

Model	1 year			2 years			3 years		
	MSLE	MAPE	WroPerc (%)	MSLE	MAPE	WroPerc (%)	MSLE	MAPE	WroPerc (%)
SEISMIC	5.496	–	48.24	5.132	–	41.68	4.720	–	32.88
Feature-based	4.069	0.485	37.76	4.004	0.426	32.30	3.523	0.353	28.71
DeepCas	2.031	0.293	28.33	1.916	0.260	22.69	1.908	0.227	21.39
DeepHawkes	2.400	0.294	27.42	1.148	0.252	22.47	1.735	0.191	20.73
CasCN	2.007	0.285	27.49	1.959	0.283	20.28	1.876	0.183	20.99
CoupledGNN	1.970	0.288	25.90	1.798	0.282	20.16	1.430	0.165	19.63
CasSeqGCN	1.953	0.285	25.32	1.773	0.306	20.84	1.458	0.168	19.43
FOREST	1.359	0.293	25.11	1.175	0.298	19.40	1.495	0.154	18.88
CasFlow	1.822	0.256	26.44	1.086	*0.233*	17.01	1.416	0.136	14.83
TempCas	*1.308*	*0.242*	*24.66*	*1.073*	0.236	*16.87*	*1.384*	*0.130*	*14.84*
CasSIM	**1.272**	**0.231**	**24.51**	**1.063**	**0.225**	**16.26**	**1.376**	**0.126**	**14.09**

objective. We independently train each model 5 times and report the average results on testing sets.

6.4.1 Popularity prediction

We outline the performance of all the benchmarks and our CasSIM model on the selected datasets in Tables 1, 2, 3 and 8. Due to the limited space, we put the results about Twitter2012, i.e., Table 8, in "Appendix D" We do not consider DyHGCN in this comparison since it can only conduct microscopic prediction, i.e., predicting the next single adopter. Our objective is to examine whether our CasSIM model

Table 3 Popularity prediction performance on Twitter2020

Model	1 hour			2 hours			3 hours		
	MSLE	MAPE	WroPerc (%)	MSLE	MAPE	WroPerc (%)	MSLE	MAPE	WroPerc (%)
SEISMIC	14.394	–	45.17	13.353	–	39.35	12.631	–	33.33
Feature-based	13.440	0.642	41.78	12.110	0.586	37.72	11.461	0.557	33.90
DeepCas	12.897	0.614	39.73	11.145	0.579	36.02	11.677	0.547	30.13
DeepHawkes	10.705	0.623	36.25	10.499	0.617	35.83	9.188	0.553	25.28
CasCN	10.640	0.592	35.81	9.207	0.552	34.63	9.048	0.550	25.62
CoupledGNN	9.400	0.497	34.49	9.122	0.477	32.86	9.045	0.452	22.55
CasSeqGCN	9.320	0.494	34.82	9.127	0.489	32.98	8.928	0.453	22.43
FOREST	8.799	0.489	33.01	8.469	0.463	30.25	8.147	0.454	21.46
CasFlow	8.916	0.478	31.59	*8.114*	0.458	28.94	8.081	0.446	16.33
TempCas	*8.756*	*0.461*	*28.25*	8.251	*0.442*	*26.67*	*8.070*	*0.426*	*15.38*
CasSIM	**8.569**	**0.443**	**27.53**	**8.046**	**0.437**	**25.43**	**8.032**	**0.422**	**14.76**

outperforms the baselines in different scenarios. If not, we analyse the possible causes so as to understand the scenarios where our model works the best. In general, we can observe that FOREST, CasFlow and TempCas are the best baselines in terms of popularity prediction. In addition, the prediction becomes more accurate when observation periods are longer. These two observations are consistent with the experimental evaluation in the literature (Tang et al. 2021; Chen et al. 2019). We highlight the best performance in bold numbers and italic the second best.

We have three main observations. First, we observe that our CasSIM model outperforms almost all the baselines according to the three measurements in the four datasets. Tempcas only marginally outperforms CasSIM when the observation periods are set to 1 h and 2 h. This may be caused by the relatively large variances of cascade lengths in the Sina dataset. The performance improvements show that our model can accurately predict the final size of cascades on both social media and citation networks where the cascading phenomenon exists. Second, compared to CoupleGNN, our CasSIM model can produce overwhelmingly more accurate predictions, especially when measured by WroPerc. For instance, on the Sina dataset, the increase is larger than 17%. The improvement can even reach 35% in our Twitter2020 dataset. This means the performance of CasSIM is more stable than CoupledGNN. We can also infer that the consideration of users' dual roles in information diffusion is necessary and our CasSIM model effectively captures the interactions between users' susceptibilities and influences. Last, the improvement of our CasSIM model is more significant when observation periods are shorter. For instance, for the Sina dataset, CasSIM increases the performance measured by MLSE by 6% compared to Tempcas when observation periods are set to 1 h. The increase drops to 3% for 2-h observation periods and further decreases to 2% when observation periods are 3 h. We infer that this should result from our consideration of users' topic preferences and message contents in CasSIM. When shorter observation periods are set,

Table 4 Final adopter prediction performance

Model	Twitter2012			Sina			AMINER			Twitter2020		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
DeepDiffuse	0.306	0.306	0.306	0.317	0.297	0.307	0.328	0.317	0.321	0.348	0.364	0.352
TopoLSTM	0.314	0.312	0.313	0.336	0.342	0.341	0.337	0.342	0.340	0.355	0.351	0.354
SNIDSA	0.403	0.408	0.406	0.428	0.377	0.386	0.354	0.339	0.346	0.363	0.374	0.371
FOREST	0.416	0.408	0.411	0.436	0.379	0.393	0.398	0.344	0.392	0.404	0.416	0.413
DyHGCN	0.454	0.441	0.443	0.449	0.393	0.394	0.413	0.388	0.410	0.441	0.448	0.442
CoupledGNN	0.370	0.366	0.367	0.371	0.323	0.332	0.318	0.297	0.262	0.315	0.300	0.302
CoupledGNN + 10%	0.382	0.338	0.346	0.392	0.348	0.352	0.325	0.307	0.319	0.335	0.324	0.329
CoupledGNN + 20%	0.393	0.360	0.357	0.418	0.361	0.365	0.334	0.311	0.327	0.356	0.335	0.342
CoupledGNN + 30%	0.400	0.367	0.372	0.431	0.372	0.378	0.370	0.359	0.364	0.371	0.387	0.385
CoupledGNN + 40%	0.418	0.387	0.397	0.437	0.379	0.381	0.376	0.364	0.370	0.380	0.391	0.387
CoupledGNN + 50%	0.411	0.381	0.388	0.422	0.364	0.370	0.366	0.341	0.351	0.350	0.332	0.341
CasSIM	0.423	0.428	0.425	0.443	0.390	0.394	0.409	0.397	0.412	0.417	0.436	0.425
CasSIM + 10%	0.431	0.429	0.430	0.447	0.404	0.405	0.412	0.408	0.406	0.421	0.437	0.423
CasSIM + 20%	0.448	0.446	0.447	0.452	0.411	0.414	0.422	0.416	0.420	0.432	0.439	0.433
CasSIM + 30%	0.465	**0.468**	**0.467**	0.474	0.423	0.428	0.437	0.433	0.435	0.440	0.442	0.441
CasSIM + 40%	**0.466**	0.463	0.465	**0.477**	**0.441**	**0.451**	**0.439**	**0.435**	**0.436**	**0.443**	**0.446**	**0.444**
CasSIM + 50%	0.451	0.453	0.452	0.449	0.431	0.440	0.423	0.420	0.421	0.429	0.432	0.431

the baselines which only rely on early adopters' co-occurrences in cascades do not have sufficient information for prediction.

Final adopter prediction

In the literature, only FOREST can predict the final adopters while predicting the popularity. It uses a microscopic prediction module to calculate the probability distribution over users to be the next activated user. FOREST iteratively samples the next adopters until a special virtual user named by 'STOP' is sampled. Compared to FOREST, CoupledGNN and our CasSIM model assign an activation probability for each user. As both the models can predict the number of final adopters, i.e., $\tilde{n}_m^\infty$, we can use the $\tilde{n}_m^\infty$ users with the largest activation probabilities as the set of final adopters. Considering the inevitable prediction errors, we use a tolerant parameter η to add a certain percentage of extra adopters. It may be argued that microscopic models can also be applied to predict final adopters by iteratively predicting the next adopters which is similar to FOREST. However, different from FOREST, such models do not have the mechanisms to terminate the sampling. In order to ensure the comprehensiveness of our validation, we manually add an *unfair* terminating condition, that is, the true number of final adopters are sampled. We use the state-of-the-art microscopic models such as DyHGCN and TopoLSTM as representatives. Note that η only works for CoupledGNN and CasSIM since they are introduced to counter the potential errors of their predicted popularity. In Table 4, we list the performance regarding final adopter prediction when observation periods are 3 h for Twitter and Sina, and 3 years for AMINER. For the tolerant parameter η, we use 10%, 20%, 30%, 40% and 50% in our experiments.

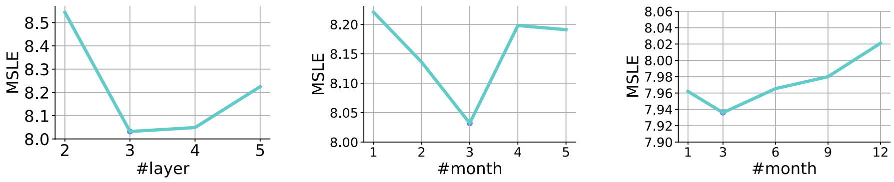
We can see that CasSIM already perform better than all the baselines except for DyHGCN with the original predicted popularity with η set to 0. DyHGCN only performs slightly better than CasSIM when applied on the Sina dataset and Twitter2012. Although the improvement is a bit marginal compared to FOREST, CasSIM has a much better performance than CoupledGNN. With the relatively high-quality cascades in Sina and Twitter2012, CasSIM increases the three measurements by about 18%. The improvement can reach more than 30% on AMINER and Twitter2020. With positive η values set, we can observe an obvious performance increase for both CoupledGNN and CasSIM. It can be expected that too large η will eventually compromise the performance. In our experiments, we can achieve the best performance when η equals 30% or 40% and the performance started to fall when η is 50%.

6.4.2 Discussion

From the above analysis, we can see our CasSIM model produce promising performance for both popularity prediction and final adopter prediction. Moreover, it effectively models the two roles of users in information diffusion. The integration of message contents into our model also helps improve the prediction of popularity when observation periods are short.

Table 5 Ablation study of popularity prediction performance on all datasets

Dataset	Model	1 h/1 year (AMINER)			2 h/2 years (AMINER)			3 h/3 years (AMINER)		
		MSLE	MAPE	WroPerc (%)	MSLE	MAPE	WroPerc(%)	MSLE	MAPE	WroPerc(%)
Sina	CasSIM	**2.148**	**0.221**	**19.46**	**2.126**	**0.217**	**14.94**	**1.919**	**0.202**	**11.04**
	CasSIM-h/r	2.323	0.241	23.09	2.243	0.228	16.73	1.992	0.216	13.73
	CasSIM-up	2.253	0.234	21.81	2.223	0.224	15.04	1.939	0.210	12.97
	CasSIM-x	2.214	0.224	21.74	2.230	0.229	16.43	1.973	0.213	13.36
AMINER	CasSIM	**1.272**	**0.231**	**24.51**	**1.063**	**0.225**	**26.26**	**1.376**	**0.126**	**14.09**
	CasSIM-h/r	1.736	0.284	24.61	1.403	0.278	29.63	1.466	0.139	15.82
	CasSIM-up	1.337	0.247	22.46	1.370	0.231	27.17	1.409	0.139	16.18
	CasSIM-x	1.585	0.259	24.95	1.355	0.246	28.12	1.527	0.140	15.74
Twitter2012	CasSIM	**6.440**	**0.419**	**22.26**	**4.739**	**0.388**	**22.64**	**3.903**	**0.309**	**13.57**
	CasSIM-h/r	6.717	0.448	23.93	4.948	0.417	25.54	4.186	0.339	14.90
	CasSIM-up	6.468	0.425	22.71	4.748	0.395	22.76	3.916	0.313	13.88
	CasSIM-x	6.657	0.436	23.55	4.883	0.406	23.57	3.985	0.320	14.15
Twitter2020	CasSIM	**8.569**	**0.443**	**27.53**	**8.046**	**0.437**	**25.43**	**8.032**	**0.422**	**14.76**
	CasSIM-h/r	9.283	0.488	33.64	9.040	0.473	30.80	8.577	0.448	21.07
	CasSIM-up	8.891	0.473	28.80	8.490	0.451	27.85	8.267	0.436	18.54
	CasSIM-x	8.907	0.478	29.46	9.160	0.464	28.85	8.352	0.452	19.79



(a) Number of GNN layers.
(b) User profiling period.
(c) Prediction period.

Fig. 6 The influence of hyperparameters

6.5 Ablation study

We examine the contributions of the components which are implemented in our CasSIM model and missing in previous works. As we emphasised previously, the novelty of CasSIM is the diffusion process modelling which considers users' profiles as two roles, message contents and topic-specific susceptibilities and influences. We design three variants of CasSIM to study the components related to these factors:

- *CasSIM-h/r* We do not distinguish users' dual roles in diffusion and use the same vectors for users' susceptibilities and influences.
- *CasSIM-up* We remove the pre-training process for the initial user profiles and use random assignments.
- *CasSIM-x* We remove users' topic preference vectors, e.g., $\mathbf{p}_v$ and do not consider the content of messages under diffusion, e.g., $\mathbf{x}_m$.

Table 5 outlines the performance comparison between CasSIM and its variants in terms of popularity prediction. We have three major observations: i) CasSIM performs considerably better than its variants; ii) ignoring users' two roles in information diffusion consistently leads to the largest damage to the prediction performance; iii) except for Sina, message content consistently ranks the second most influential component.

6.6 Hyperparameter test

We examine the influence of three important hyperparameters of CasSIM. The first parameter is the number of GNN layers which can intuitively be interpreted as the number of diffusion generations. The other two parameters relate to the pre-trained user profiles. In CasSIM, we assume that users' profiles are stable over a sufficiently long time, especially for users' topic profiles, e.g., $\mathbf{p}_v$. In the previous experiments, we use the first three months' retweets in our Twitter dataset to pre-train users' susceptibility and influence vectors, and stick to them to conduct following cascade predictions. We would like to test whether this is reasonable in practice and when user profiles should be retrained. We take our Twitter dataset as an example in our investigation. We start with examining how many months in advance are needed in this pre-training process and then track the performance changes when predicting cascades in different periods after the user profile training. Figure 6 shows the results. We vary the number GNN layers z from {2,3,4,5}. We can see that MSLE curve drops to the bottom when $z = 3$, then slowly climbs up when larger numbers of layers are implemented. We vary the number of months whose retweets are used for user profiles from $\{1, 2, 3, 4, 5\}$ and the result shows that the periods for user profiling can be neither too short nor too long. On our Twitter dataset, three months work the best for popularity prediction. To test the effectiveness of pre-trained user profiles, we train and test our CasSIM model on tweets in the 1, 3, 6, 9 and 12 months after the tweets used for user profile training. We can see the popularity prediction performance decreases when the trained user profiles are used to predict cascades later than 3 months. However, a closer look will reveal that the range of the change is rather small. This is consistent with our expectation that user preferences and interests are relatively stable in spite of the vast changes in social news trending.

7 Conclusion

In this paper, we proposed a new deep learning model CasSIM which can simultaneously achieve the two most demanded cascade prediction objectives: popularity prediction and final adopter prediction. Compared to previous models, CasSIM explores the dual roles of users in diffusion processes as both receivers and distributors and models the three basic factors in users' decision to become active: susceptibilities, influences and message contents. With effective user profiling, CasSIM successfully models the topic-specific property of susceptibilities and influences. In addition, the introduction of GNN allows CasSIM to capture the dynamics of susceptibilities and influences during information diffusion. With extensive experiments on three real-life datasets, we validated the effectiveness of CasSIM in predicting popularity and final adopters. The results showed that CasSIM outperforms the state-of-the-art methods, especially when shorter cascades are observed, in both social media and other scenarios where cascades are also present.

We identify a few limits of our CasSIM model which can be addressed in the future. First, we focused on messages in the form of texts and only consider their topics. Second, CasSIM does not consider the temporal ranks between the early adopters. It is interesting to extend and test CasSIM in cascade prediction by combining other types of information in messages such as images and quotations, considering other semantic features such as sentiments, and improve the performance by integrating the time stamps of early adopters.

Appendix A: Statistics of our data collection

We summarise the statistics of our data collection used for the analysis in Sect. 4 in Table 6.

Table 6 The statistics of our Twitter dataset

Social network	#node		5, 808, 938
	#edge		12, 511, 698
	average degree		2.15
Timeline tweets	Period 1(1/3/2020-30/6/2020)	#tweet	7,855,186
		#retweet	4,245,618
		#tweet per user	627.51
		#retweet per user	339.16
	Period 2 (1/3/2021-30/6/2021)	#tweet	3,591,664
		#retweet	1,883,128
		#tweet per user	303.76
		#retweet per user	150.43

Appendix B: Description of datasets in experimental evaluation

Sina

The dataset is collected by Zhang et al. (2013) from Sina, a popular Chinese social media platform and spans from September 28, 2012 to October 29, 2012. For each user, the dataset includes 1,000 additional most recent microblogs. We only keep the cascades with more than 15 users.

AMINER AMINER (Tang et al. 2008) is constructed with data collected from the DBLP computer science bibliography.[1] The social graph describes authors and their citation relations between 1992 and 2002. A cascade corresponds to a paper and tracks the researchers who ever co-authored the paper or cited the paper. We construct the social graph with the papers between 1992 and 2002 and those between 2005 and 2009 for model training, testing and validation.

Twitter2012 This dataset is collected by Weng et al. (2013) from Twitter. It comprises public tweets posted between March 24 and April 25, 2012. Following the approach of Xu et al. (2021), we consider hashtags and their adopters as distinct information cascades. The social network is established with a variety of relationships including followings, retweeting and mentions.

Twitter2020 We collected this dataset with the same social graph constructed based on the following-ships in May 2020 as used in Sect. 4. Specifically, in addition to the six-month tweets, this dataset contains the retweets spanning from March

[1] https://dblp.org/

Table 7 Statistics of Sina, AMINER, and Twitter datasets

	Social network		Ave. # user per ascade			#cascades
	#node	#edges	Observation 1	Observation 2	Observation 3	
Sina	1,776,950	308,489,739	28.36	34.79	38.22	34,897
AMINER	131,415	842,542	14.71	15.71	18.80	30,106
Twitter2012	490,474	1,903,230	15.88	17.28	20.22	88,440
Twitter2020	5,808,938	12,511,698	27.64	30.99	33.76	81,331

1, 2020 to October 30, 2021 for almost 2 years. Our experiments use the cascades with a size larger than 15. Table 7 lists the statistics of the datasets.

Appendix C: Description of baselines

Feature-based method

This is a linear regression model with crafted features with L2 regularisation. We use the same features in Cao et al. (2020D).

SEISMIC (Zhao et al. 2015). SEISMIC uses the Hawkes self-activation point process to approximate the impact of cascading effect for popularity prediction.

DeepCas (Li et al. 2017). DeepCas is the first end-to-end representation learning-based method for popularity prediction. It represents cascades as cascade graphs and uses Bi-GRU to embed a cascade graph with random walk paths.

DeepHawkes (Cao et al. 2017). DeepHawkes integrates deep learning into Hawkes process for popularity prediction. It treats cascades as a temporal series of events. It combines user embedding vectors and cascades encoding by RNNs.

CasCN (Chen et al. 2019). Compared to models like DeepCas sampling random diffusion paths, CasCN samples sub-graphs from cascade graphs into a series of sequential sub-cascades. Then it proposes a dynamic GCN model (Bruna et al. 2014) to learn the cascade representation for popularity prediction.

CoupledGNN (Cao et al. 2020D). CoupledGNN leverages two GNNs to simulate the cascading effects of information diffusion. One updates users' activation states and the other captures the spread of users' influence. Despite being designed for popularity prediction, the simulation of cascading effect also allows it to predict final adopters.

CasSeqGCN (Wang et al. 2022) Similar to CasCN, CasSeqGCN also processes cascade graphs into a sequence of sub-cascades with network topology for popularity prediction. It uses a dynamic route aggregation-based GCN to capture the representations of sub-cascades, and an LSTM to extract temporal information.

FOREST (Yang et al. 2019). FOREST uses an enforcement learning framework to endow a microscopic prediction model with popularity prediction. It can predict both popularity and final adopters. The proposed microscopic model uses

GRU neural networks to embed and use the representation of the last early adopter to predict the next adopter.

CasFlow (Xu et al. 2021). CasFlow uses graph signal processing, graph representation techniques and variational auto-encoder to capture node-level and cascade-level diffusion uncertainty. It is designed for popularity prediction.

TempCas (Tang et al. 2021). TempCas is designed for popularity prediction and claims better performance than previous methods. It considers the temporal changes of cascade graphs and calculates the representation of a cascade graph based on previous snapshots of the cascade graph.

DeepDiffuse (Islam et al. 2018). DeepDiffuse is an attention-based RNN model designed for predicting final adopters. It captures the state transitions of users, leveraging network embeddings and attention models to predict the timing of the next adopter's infection.

TopoLSTM (Wang et al. 2017). TopoLSTM is an RNN model for final adopter prediction. It enhances the standard LSTM by exploiting the hidden states in a directed acyclic graph, which is obtained from the social graph.

SNIDA (Wang et al. 2018). SNIDSA is an RNN model that incorporates structure attention for predicting final adopters. It utilises RNNs to handle sequential data while introducing an attention mechanism to trace the structural dependency among users. Additionally, it employs a gating strategy to integrate both sequential and structural information.

DyHGCN (Yuan et al. 2020). DyHGCN is designed for final adopter prediction. It extracts structural information from both social graphs and cascade graphs to generate dynamic user embeddings through a heterogeneous GCN.

Appendix D: Performance of popularity prediction on the Twitter2012 dataset

See Table 8.

Table 8 Popularity prediction performance on Twitter2012

Model	1 h			2 h			3 h		
	MSLE	MAPE	WroPerc (%)	MSLE	MAPE	WroPerc (%)	MSLE	MAPE	WroPerc (%)
SEISMIC	9.720	–	41.33	8.634	–	40.88	8.355	–	37.13
Feature-based	16.944	0.523	49.38	14.563	0.456	45.16	11.321	0.435	39.78
DeepCas	8.422	0.582	29.16	6.304	0.650	31.71	5.610	0.411	33.97
DeepHawkes	7.071	0.553	29.53	5.326	0.509	26.52	4.765	0.368	20.70
CasCN	7.091	0.498	28.82	5.207	0.485	26.20	4.645	0.349	20.52
CoupledGNN	7.145	0.509	26.30	5.185	0.481	25.97	4.475	0.339	19.79
CasSeqGCN	7.027	0.548	26.76	5.193	0.488	25.64	4.357	0.336	19.36
FOREST	6.856	0.510	25.90	5.169	0.491	24.85	4.198	0.332	18.74
CasFlow	6.692	0.505	24.10	*4.884*	0.397	22.82	4.043	0.337	14.57
TempCas	*6.469*	*0.423*	*22.86*	4.896	*0.349*	**22.10**	*3.969*	*0.318*	*13.91*
CasSIM	**6.440**	**0.419**	**22.26**	**4.739**	**0.338**	*22.64*	**3.903**	**0.309**	**13.57**

Appendix E: Topic annotations of clustered topics

We coarsely classify them into three categories. Label 'COVID-19' encapsulates all discourse about COVID-19, such as infection and death numbers, vaccines, and policies. The label 'BLM' corresponds to discussions around the Black Lives Matter related activities. Topics that do not fall under these two types are preliminarily classified under label 'Other'.

See Tables 9 and 10.

Table 9 Keywords and annotations of tweets in Period 1

Topic	Top 10 words	label
1	Fr tube chine phone live covid france virus coronavirus jours	COVID-19
2	Covid positivo hospitales presidente mands magnifique nombre morts fr virus	COVID-19
3	Corona mssen einfach innen deutschland pandemie knnen innen wre de	COVID-19
4	Love good day time people life today feel mme photo	Other
5	Faut tait monde temps jour jamais temps medecin sante	COVID-19
6	Wow wait benzema moment bonne mesdames messieurs submitted lne enfin	Other
7	Usa china biden trump antifa gop america black game yall	BLM
8	Saint luxembourg vraument mme veux mieux franais neiegkeeten infektioun krank	COVID-19
9	Ms gente pra vida gobierno estn quiero lasso cmo gracias	Other
10	Deutschland gesicherte bei eine gesicherte covid-19 auslosen hat WHO	COVID-19
11	Hate black life gun covid asia jacobblake today year	COVID-19
12	Blm georgefloyd justice trump shot video cop gun ferguson protest	BLM
13	Publiziert soeben resto sketch mars cours daily wonte influencer labonnement	Other
14	Zuhause sicher gut helfen aku hause abstand wre maske virus	COVID-19
15	Happy violences tapes partiels think dankeschn hope cours jespre bir	Other
16	Policy home temporarily media learn resonance account honored healthy hospital	COVID-19
17	Bir ve bu iin gibi mood daha olan kadar yok	Other
18	Policiers ignoraient rbellion participes tweet yeah tabass suffit miff ms	Other
19	Macron allocution million suivi avait avril president qui nation mdrr	Other
20	Love omg mood stadsarchief coup amsterdam collectie commente fotograaf tapi	Other
21	Macron us racist privacy app phone coronavirus black government vaccine	BLM
22	Time shit trump day tweet people lmao fake love year	Other
23	Lockdown flattenthecurve borderline hinausgehen reise arbeit home wash maske ber	COVID-19
24	Death who life china work virus temps cdu year time	COVID-19
25	Confinement start qqun rouvrir peson mourront animal botes jlui follow	Other

Table 10 Keywords and annotations of tweets in Period 2

Topic	Top 10 words	Label
1	Covid vaccine effect hour day people good week last first	COVID-19
2	Impfstoff maske dose immunitat wirksamkeit jahr covid impfen jetzt wirkung	COVID-19
3	Ms pra gente casca estn mundo quiero mme gobierno vida	Other
4	Vaccin cote heure jour dexieme fievre symptome douleur effect fait	COVID-19
5	De ausreisetst abriegelung corona zahlen landesgrenzen dicht balkanroute mit hause	COVID-19
6	Yes fr million rentez year gagner mdrrr ete mood rebellion	Other
7	Fr variant delta virulent boom mondiale aggrave jamais vaccin	COVID-19
8	Germany luxembourg country merkel dead europe case person patient listen	COVID-19
9	Teilen nicely france lithuania slovakia latvia bulgaria estonia policy mme	COVID-19
10	Cristiano mdr lt mme mdrr wsh gt ptn rn pq	Other
11	Fuck person automatically check photo kind twitter pardon monsta instabil	Other
12	Resonance pt2 true nct day 21st pt 2nd diagonales actu2	Other
13	Kes tweet participes suffit miff rgale rt gagnent commentaires bir	Other
14	Fin crise france retraite masion personnel deconfinement sante deces cas	COVID-19
15	Yang aku ada ini yg nak orange ke kita boleh	Other
16	Neerheylissem brabant publicar votes 10 wow acaba lmao yok ang	Other
17	Belgien voir mme vraiment monde faut jamais jour nouvelle journe	COVID-19
18	Actu japonais diagonales weeknd lundi devierg volont bavure caissires repas	Other
19	Imbattable person comrades plage douche transition restaurant open	COVID-19
20	Day symphorien person dirait oubli escortes couvre jour qun kkkk	COVID-19
21	Championne lol himer pulvris jepkosgei joyciline km barbec influenceuse follow	Other
22	Ttrangle essaye saute dfendre gardin paix mdrr extinct bonjour passer	Other
23	Pfizer vaccin moderna biotech covid autorise qu commander doses chine	COVID-19
24	Fuck vcut wib droit dcembre retweet januari absolutely wsh mme	Other
25	Danke wtf voil requinqu trolleuse jours nope family ms pra	Other

Acknowledgements This research was funded in whole, or in part, by the Luxembourg National Research Fund (FNR), grant reference C21/IS/16281848 (HETERS) and PRIDE17/12252781 (DRIVEN). For the purpose of open access, the author has applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission.

References

Bourigault S, Lamprier S, Gallinari P (2016) Representation learning for information diffusion through social networks: an embedded cascade model. In: Proceedings of the 9th ACM international conference on web search and data mining (WSDM), ACM, pp 573–582

Bruna J, Zaremba W, Szlam A, LeCun Y (2014) Spectral networks and locally connected networks on graphs. In: Proceedings of the 2nd international conference on learning representations, (ICLR)

Cao Q, Shen H, Cen K, Ouyang W, Cheng X (2017) DeepHawkes: Bridging the gap between prediction and understanding of information cascades. In: Proceedings of the ACM on conference on information and knowledge management (CIKM), ACM, pp 1149–1158

Cao Q, Shen H, Gao J, Wei B, Cheng X (2020) Popularity prediction on social platforms with coupled graph neural networks. In: Proceedings of the 13th ACM international conference on web search and data mining (WSDM), ACM, pp 70–78

Chen N, Chen X, Pang J (2022) A multilingual dataset of COVID-19 vaccination attitudes on Twitter. Data Brief 44:108503

Cheng J, Adamic L, Dow PA, Kleinberg JM, Leskovec J (2014) Can cascades be predicted? In: Proceedings of the 2014 International conference on World Wide Web (WWW), pp 925–936

Chen X, Zhou F, Zhang K, Trajcevski G, Zhong T, Zhang F (2019) Information diffusion prediction via recurrent cascades convolution. In: Proceedings of the 35th IEEE international conference on data engineering (ICDE), IEEE Computer Society, pp 770–781

Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, Grave E, Ott M, Zettlemoyer L, Stoyanov V (2020) Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th annual meeting of the association for computational linguistics (ACL), Virtual, pp 8440–8451

Cui P, Jin S, Yu L, Wang F, Zhu W, Yang S (2013) Cascading outbreak prediction in networks: a data-driven approach. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining (KDD), ACM, pp 901–909

Greene D, Cunningham P (2006) Practical solutions to the problem of diagonal dominance in kernel document clustering. In: Proceedings of the 23rd international conference on machine learning, vol. 148. USA, pp 377–384

Guarino S, Pierri F, Giovanni MD, Celestini A (2021) Information disorders during the COVID-19 infodemic: the case of Italian facebook. Online Soc Netw Med 22:100124

Hong L, Dan O, Davison BD (2011) Predicting popular messages in twitter. In: Proceedings of the 20th international conference on World Wide Web (WWW), ACM, pp 57–58

Islam MR, Muthiah S, Adhikari B, Prakash BA, Ramakrishnan N (2018) Deepdiffuse: Predicting the 'who' and 'when' in cascades. In: Proceedings of the 2018 international conference on data mining (ICDM), IEEE Computer Society, pp 1055–1060

Kempe D, Kleinberg JM, Tardos É (2003) Maximizing the spread of influence through a social network. In: Getoor L, Senator TE, Domingos PM, Faloutsos C (eds) Proceedings of the 9th ACM international conference on knowledge discovery and data mining (SIGKDD), ACM, pp 137–146

Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: Proceedings of the 2015 international conference on learning representations (ICLR), p. 0. OpenReview.net

Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th international conference on learning representations, (ICLR). OpenReview

Li C, Ma J, Guo X, Mei Q (2017) DeepCas: An end-to-end predictor of information cascades. In: Proceedings of the 26th international conference on World Wide Web (WWW), pp 577–586

Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) RoBERTa: a robustly optimized BERT pretraining approach. CoRR abs/1907.11692

McInnes L, Healy J (2018) UMAP: uniform manifold approximation and projection for dimension reduction. CoRR abs/1802.03426

Panagopoulos G, Malliaros FD, Vazirgiannis M (2020) Influence maximization using influence and susceptibility embeddings. In: Proceedings of the 14th international AAAI conference on web and social media (ICWSM), AAAI Press, pp 511–521

Shen H, Wang D, Song C, Barabási A (2014) Modeling and predicting popularity dynamics via reinforced Poisson processes. In: Proceedings of the 28th AAAI conference on artificial intelligence (AAAI), AAAI Press, pp 291–297

Song C, Hsu W, Lee M (2017) Temporal influence blocking: Minimizing the effect of misinformation in social networks. In: Proceedings of the 33rd IEEE international conference on data engineering (ICDE), IEEE Computer Society, pp 847–858

Sun L, Rao Y, Zhang X, Lan Y, Yu S (2022) MS-HGAT: memory-enhanced sequential hypergraph attention network for information diffusion prediction. In: Proceedings of the 36th AAAI conference on artificial intelligence (AAAI), AAAI Press, Virtual, pp 4156–4164

Tang X, Liao D, Huang W, Xu J, Zhu L, Shen M (2021) Fully exploiting cascade graphs for real-time forwarding prediction. In: Proceedings of the 35th AAAI conference on artificial intelligence (AAAI), Virtual, pp 582–590

Tang J, Zhang J, Yao L, Li J, Zhang L, Su Z (2008) Arnetminer: extraction and mining of academic social networks. In: Proceedings of the 14th international conference on knowledge discovery and data mining, ACM, pp 990–998

Velickovic P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y (2018) Graph attention networks. In: Proceedings of the 6th international conference on learning representations (ICLR). OpenReview

Wang Y, Wang X, Ran Y, Michalski R, Jia T (2022) Casseqgcn: combining network structure and temporal sequence to predict information cascades. Expert Syst Appl 206:117693

Wang Z, Chen C, Li W (2018) A sequential neural information diffusion model with structure attention. In: Proceedings of the 27th ACM international conference on information and knowledge management (CIKM), ACM, pp 1795–1798

Wang Y, Shen H, Liu S, Cheng X (2015) Learning user-specific latent influence and susceptibility from information cascades. In: Proceedings of the19th AAAI conference on artificial intelligence (AAAI), AAAI Press, pp 477–484

Wang J, Zheng VW, Liu Z, Chang KC (2017) Topological recurrent neural network for diffusion prediction. In: Proceedings of the 2017 IEEE international conference on data mining (ICDM), IEEE Computer Society, pp 475–484

Weng L, Menczer F, Ahn Y-Y (2013) Virality prediction and community structure in social networks. Sci Rep 3(1):1–6

Xu X, Zhou F, Zhang K, Liu S, Trajcevski G (2021) CasFlow: exploring hierarchical structures and propagation uncertainty for cascade prediction. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2021.3126475

Yang C, Tang J, Sun M, Cui G, Liu Z (2019) Multi-scale information diffusion prediction with reinforced recurrent networks. In: Proceedings of the 28th international joint conference on artificial intelligence (IJCAI), pp 4033–4039

Yuan C, Li J, Zhou W, Lu Y, Zhang X, Hu S (2020) DyHGCN: a dynamic heterogeneous graph convolutional network to learn users' dynamic preferences for information diffusion prediction. In: Proceedings of the 2020 machine learning and knowledge discovery in databases—European Conference, Springer, vol. 12459, pp 347–363

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: Proceedings of the 2015 IEEE international conference on data mining (ICDM), pp 559–568

Zhang Z, Fang M, Chen L, Namazi Rad MR (2022) Is neural topic modelling better than clustering? An empirical study on clustering with contextual embeddings for topics. In: Proceedings of the 2022 NAACL, association for computational linguistics, pp 3886–3893

Zhang J, Liu B, Tang J, Chen T, Li J (2013) Social influence locality for modeling retweeting behaviors. In: Proceedings of the 23rd international joint conference on artificial intelligence (IJCAI), pp 2761–2767

Zhao Q, Erdogdu MA, He HY, Rajaraman A, Leskovec J (2015) Seismic: a self-exciting point process model for predicting tweet popularity. In: Proceedings of the 2015 international conference on knowledge discovery and data mining (KDD), pp 1513–1522

Zhou F, Yu L, Xu X, Trajcevski G (2021) Decoupling representation and regressor for long-tailed information cascade prediction. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, ACM, Virtual, pp 1875–1879