Check for updates

Modeling the social influence of COVID-19 via personalized propagation with deep learning

Yufei Liu[1,4] · Jie Cao[2] · Jia Wu[3] · Dechang Pi[4]

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Social influence prediction has permeated many domains, including marketing, behavior prediction, recommendation systems, and more. However, traditional methods of predicting social influence not only require domain expertise, they also rely on extracting user features, which can be very tedious. Additionally, graph convolutional networks (GCNs), which deals with graph data in non-Euclidean space, are not directly applicable to Euclidean space. To overcome these problems, we extended DeepInf such that it can predict the social influence of COVID-19 via the transition probability of the page rank domain. Furthermore, our implementation gives rise to a deep learning-based personalized propagation algorithm, called DeepPP. The resulting algorithm combines the personalized propagation of a neural prediction model with the approximate personalized propagation of a neural prediction model from page rank analysis. Four social networks from different domains as well as two COVID-19 datasets were used to analyze the proposed algorithm's efficiency and effectiveness. Compared to other baseline methods, DeepPP provides more accurate social influence predictions. Further, experiments demonstrate that DeepPP can be applied to real-world prediction data for COVID-19.

1 Introduction

With the explosion of big data, social networks are providing a plethora of information on user interactions [1, 2]. For example, Twitter's average realizable daily active count in the second quarter of 2022 was 237.8 million, a year-on-year increase of 16.6% [3]. However, what we are seeing with the isolation imposed by COVID-19 is social networks playing an even more important role in the everyday interactions between people - particularly when it comes to social influence. The term 'social influence' is usually understood as the process whereby a user's emotions, opinions, or behaviors are shaped by their environment, i.e., the process by which people alter their behavior under the influence of others [3]. With


✉ Jie Cao
  cao_jie@hfut.edu.cn

Extended author information available on the last page of the article.

the globalization of online social networks, social influence analysis has spread to many domains, including marketing [4], behavior prediction [5], recommendation systems [6, 7], influence maximization [8], public opinion guiding [9], communities [10], and graph anomaly detection [11]. We cannot deny that social influence has become ubiquitous and complex in shaping our social decisions. Therefore, there is much interest in developing methods to understand, describe, and identify the mechanisms and evolutions behind the social influence.

The process of predicting influence on a user involves sampling local neighbors, building a local network from the samples, and then learning the potential predictive signals from that network. Matsubara et al. [12] designed a dynamic model of social influence based on a differential equation drawn from classic susceptibility theory. The approach proposed by Li et al. [13] combines RNNs with representation learning to infer the cascading size of features. With both these methods, the aim is to predict the statistical patterns of social influence over time, including cascading size and global patterns. According to Qiu et al. [14], the famous social influence prediction method, DeepInf, has been developed for general social networks scenes. The idea is that we can predict the user's behavior state by taking into account the behavior state and characteristics of their neighbors. However, due to the sparsity, the neighbor information of the COVID-19 network is very limited, so they are invalid for the network-based COVID-19 prediction.

In this paper, to overcome these issues, we extended DeepInf such that our proposed model utilizes personalized propagation to predict social influence at the user level. We built on DeepInf [14] by integrating the transition probability α of the page rank domain with a GCN model. Specifically, our proposed model first uses a graph neural network to learn the latent social representation of users by taking their local network as input. We then replace the GAT/GCN network with PPNP (personalized propagation of neural predictions) [15], a model that enhances influence scores efficiently. Meanwhile, α adjusts the size of the neighborhood influence. **H** is the prediction matrix, and α**H** offers greater flexibility, as shown in Figure 1. Finally, the transition probability α provides us with a way to achieve the ideal balance between maintaining locality (i.e., retaining a close proximity to the root to avoid over-smoothing) and leveraging the information obtained from large neighbors. In experiments, this propagation scheme has been shown to be extremely efficient and has the
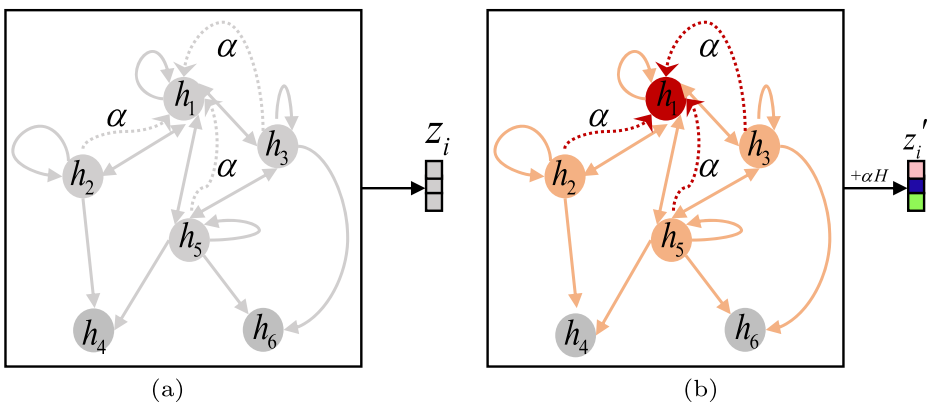
Figure 1 The neural network first predicts the influence of each node according to its own characteristics. Then, the personalized PageRank algorithm is used to propagate influence adaptively. (a) shows the personalized propagation. (b) shows the improved personalized propagation

benefit of being able to use far more propagation steps (an infinite number in fact) without resulting in over-smoothing of the output. In short, we extended DeepInf by leveraging the two algorithms, PPNP and APPNP (approximation to PPNP) [15] extended from a GCN, and transplanting them from page ranking into social influence analysis. Our implementation gives rise to deep learning-based personalized propagation algorithm, called DeepPP. The algorithm predicts how v will behave in the future based on its current behavior by analyzing the state of n neighbors around a user v. An active or inactive node (user) can exist in the system. But, to forecast the state of v at the end of the interval, we need to know v's current state.

We examined four social networks in various domains to evaluate our algorithm's efficiency and effectiveness, including OAG, Digg, Twitter, and Sina Weibo, as well as two COVID-19 datasets (Hubei and Holland). We compared DeepPP with: a conventional GNN* model [16], the advanced PPNP and APPNP [15], and the most famous models, DeepInf [14] and DeepEmLAN [17]. The results of an extensive study demonstrated that DeepPP provided a better $F1 - Measure$ than the current advanced baselines. On the COVID-19 datasets, the DeepPP provided a higher level of precision. It has been demonstrated that DeepPP can predict the social influence of COVID-19 based on excellent performance on various datasets.

The following list summarizes our main contributions.

1. Inspired by DeepInf, we integrated the transition probability α of the page rank domain into a GCN model, thereby extending DeepInf.
2. The resulting DeepPP algorithm repurposes PPNP and APPNP without any additional time complexity from page ranking to social influence analysis.
3. Comprehensive evaluations of model performance show DeepPP to be more accurate than the baseline methods.

Following is the remainder of this paper. We review the status of research on social influence prediction in Section 2. Section 3 introduces the overview of DeepPP and outlines the personalized propagation process based on deep learning for modeling the social influence of COVID-19. Section 4 describes the experiments used to validate DeepPP. Section 5 highlights concluding remarks and next steps.

2 Related work

Our literature review covers the following topics: traditional social influence analysis, deep learning-based social influence prediction, and graph representation learning. The results show that an accurate model for predicting social influence may not have been established.

2.1 Methods of traditional social influence analysis

Social influence analysis is traditionally based on the study of interpersonal communications, where user features are typically extracted by hand, which can be tedious. Further, the analyses are based on domain expertise in sociology or cognitive science and, therefore, extending the results is difficult [14]. A study by Li et al. [18] distinguished two types of social influence analysis models: (1) macro, which assumes all users have equal power to influence; and (2) micro, which explores individual levels of user influence. Of the micro-level influence models, independent cascade and linear threshold are the two most common. Nevertheless, both types of models assume that users will not change their state, i.e., the

probability of influencing others or being influenced [19] using Bayes Theorem [20, 21]. As such, they are not particularly reflective of real social networks.

2.2 Methods of social influence prediction with deep learning

Many fields have benefited from deep learning, and social influence analysis is no exception. However, its application to this field is relatively recent. To date, both micro- and macro-level methods of analysis have been developed. Micro-level models focus on user interactions. The key assumption is that user interactions, e.g., ratings, comments, retweets, affect the behavior of others. One of the most advanced models is DeepInf [14], a model that predicts user-level influence from end to end. It integrates a network embedding [22] with a GAT [23] and a GCN [24]. Both GATs and GCNs are better suited to non-Euclidean spatial data at aggregating features of neighbor vertices onto central vertices than traditional machine learning methods [25]. In fact, experiments with DeepInf show the best predictive performance with multi-headed GATs, even when compared to a GCN [24]. By using dual GNNs instead of a single GNN in recommender systems, Wu et al. [26] designed a deep latent representation of multifaceted social impact.

The macro-level solutions focus on the patterns of global social influence. For example, DeepCas [13], a method of analyzing macro-social impact models using RNNs, involves a pattern of cascading that includes all aspects of the cascading and their associations with the final cascading size [22]. In this method, as a two-dimensional colored diagram, an end-to-end predictor visualizes all cascades of influence information. Using DeepCas as inspiration, Cao et al. [27] developed a method which represents information cascades using an explainable model of the generation Hawkes process [28]. Their model showed better performance than traditional generation and feature-based models. DeepHawkes [27], as well as DeepInf, is data-driven, which means it can learn from previous cascades and can take advantage of historical information. A data-driven cascading method called LST-MICs, that uses long short-term memory (LSTM), was further developed and extended by Gou et al. [29] to learn sequence features from cascading features and RNN functions. They used a Weibo and a Twitter dataset to predict outbreaks more accurately than previous methods. With Cas2vec [30], it is possible to accurately predict virus cascades without manually extracting the relevant information from the framework, which could be costly to obtain. Time intervals are used instead of the information contained in the events to extract information.

Social influence analysis models are closely related to graph representation learning, and many studies have been conducted on graph representation learning as part of graph mining, e.g., DeepWalk [31], Line [32], Node2vec [33], Metapath2vec [34], NetMF [35], Graph Kernel [36], and the most advanced method PSCN [37]. More recently, researchers have been exploring the notion of graph representation learning with semi-supervised information. Some examples include GCN [24], GraphSage [38], and GAT [23] as the most advanced model.

3 Methodology

Analyzing social influence with deep learning techniques is a problem related to GNN. Because GNN in recommendation systems has other types of applications, such as collaborative filtering and predictive page ranking, drawing inspiration from other fields to

improve our existing research is a highly useful exercise. Figure 2 illustrates personalized propagation based on deep learning for modeling the social influence associated with COVID-19.

3.1 Influence propagating

We extended DeepInf by leveraging two algorithms, PPNP and APPNP. Klicpera et al. [15] proposed them by extending a GCN, where each node has the same effect on its neighbors. In other words, the independent cascade model recalled in Section 2 has the same influence scores and probability of being propagated to its neighbors for each edge. With the transition probability α, information transmission is adjusted so that each node has a different influence on its neighbors. In experiments, PPNP and APPNP algorithms perform better than the GNN algorithm when the transition probability α is set to between 0.05 and 0.2 [15]. This is also compared to the GCN algorithm, which has an average effect on all neighbors. Given that the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ contains $|\mathcal{V}|$ vertices and $|\mathcal{E}|$ edges. $\mathbf{A}$ is the adjacency matrix of $\mathcal{G}$. $\mathbf{I}_n$ is the added self-loops for $\mathcal{V}$. Let $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n$. The transport vector $\mathbf{i}_x$ preserves the local neighborhood of the node [39]. Essentially, the $I(x, y)$ score (i.e. Influence of root node x on non-root node y) is equivalent to the $y^{(th)}$ of a personalized page ranking $\pi_{pr}(\mathbf{i}_x)$. A recursive equation with a transport vector is expressed with a transition probability α as:

$$\pi_{pr}(\mathbf{i}_x) = \alpha((\mathbf{I}_n) - (1 - \alpha)\hat{\mathbf{A}})^{-1}\mathbf{i}_x, \alpha \in (0, 1) \tag{1}$$

The influence score in (1) can be used to generate the prediction objective function.

$$\mathcal{Z} = softmax(\alpha(\mathbf{I}_n - (1 - \alpha)\hat{\mathbf{A}})^{-1}\mathbf{H}), \mathbf{H}_{i,i} = \mathcal{F}_\theta(\mathbf{X}_{i,s}) \tag{2}$$

where $\mathbf{X}_i$ is the eigenmatrix. $\mathbf{H}$ is the prediction matrix. $\mathcal{F}_\theta$ represents a neural network that predicts $\mathbf{H} \in \mathbb{R}^{n \times c}$. (1) and (2) are the PPNP model. However, calculating the time complexity requires a dense matrix $\mathbb{R}^{n \times c}$ with the time complexity $O(n^2)$. APPNP was developed as a means of addressing this shortcoming. More specifically, by using the power method, which has a linear computational complexity, the eigenvalues for diagonalizable matrices are calculated as:
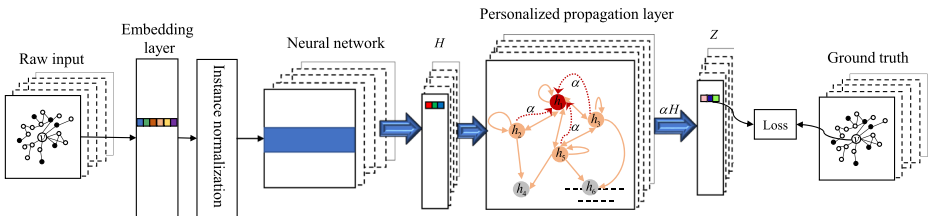
$$Z^{(0)} = \mathbf{H} = f_\theta(\mathbf{X}) \tag{3}$$

$$\mathcal{Z}^{(k+1)} = (1 - \alpha)\hat{\mathbf{A}}\mathcal{Z}^{(k)} + \alpha\mathbf{H} \tag{4}$$

$$\mathcal{Z}^{(k)} = softmax((1 - \alpha)\hat{\mathbf{A}}\mathcal{Z}^{(k-1)} + \alpha\mathbf{H}) \tag{5}$$

where $\mathcal{Z}^{(k)}$ is a transport set that can effectively provide approximate predictions. The integer k defines the number of steps required to perform a power iteration.

Our model is derived from a GCN and a graph attention network model as follows.

Figure 2 Overview of the DeepPP model. The neighbors of node v are sampled with the goal being to predict the state of node v after a number of iterations

3.2 Network encoding with a GCN

GCN, that analyses graph-structured data, is semi-supervised learning [24]. For eigen decomposition, the GCN uses a Laplace graph of the Fourier domain. Specifically, the rule in (6) for a GCN is layer-by-layer propagation, which consists of multiple GCN layers.

$$\mathcal{F}\left(\mathbf{H}^{(l)}, \mathbf{A}\right) = \sigma \left(\mathbf{A}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right) \tag{6}$$

where, $\mathbf{H}^{(0)}$ is the eigenmatrix. When $\mathbf{D}$ is normalized, the adjacency matrix $\mathbf{A}$ benefits from a diagonal node degree matrix by preserving the graph's self-loop. Putting all this together, $\mathbf{A}$ is as follows:

$$\mathbf{A} = \hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}} \tag{7}$$

3.3 Multi-head attention with a GAT

Another neural network model, a GAT [40, 41], processes graphical structured data in the propagation using self-attention techniques. In more detail, the GAT calculates the node states based on the node's neighbors, where each node is assigned an attention factor, calculated using its coefficient of attention α_{ij} $(i \rightarrow j)$.

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\alpha}^T \left[\mathbf{W}\vec{h}_i \,\middle\|\, \mathbf{W}\vec{h}_j\right]\right)\right)}{\sum\limits_{k \in N_i} \exp\left(\text{LeakyReLU}\left(\vec{\alpha}^T \left[\mathbf{W}\vec{h}_i \,\middle\|\, \mathbf{W}\vec{h}_j\right]\right)\right)} \tag{8}$$

By the weight matrix $\left[\mathbf{W}\vec{h}_i \,\middle\|\, \mathbf{W}\vec{h}_j\right]$, $(\mathbf{W} \in \mathbb{R}^{F \times F})$, α as a weight vector. T stands for transposition, $\|$ stands for series operation. By combining the normalized attention coefficient and the linear feature combination of the final output feature, we can obtain the following expression.

$$\vec{h}_i' = \sigma \left(\sum_{j \in N_i} \alpha_{ij} \mathbf{W}\vec{h}_j\right) \tag{9}$$

The characteristic functions can be concatenated using the k independent attention layers in (9) through the multi-head attention technique, which is an appropriate approach for the learning process:

$$\vec{h}_i' = \sigma \left(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in N_i} \alpha_{ij}^k \mathbf{W}^k \vec{h}_j\right) \tag{10}$$

3.4 Modeling the social influence of COVID-19

Through deep learning, DeepInf can automatically identify hidden patterns at the user level and predict their influence. PPNP is a personalized neural prediction propagation algorithm. By combining the PPNP and APPNP algorithms, we improve the DeepInf method to model the social influence of COVID-19. More specifically, we combine the personalized

page ranking algorithms in the DeepInf method. The GAT/GCN network is replaced with a personalized page ranking model, and a novel algorithm named DeepPP is devised.

We use the transfer probability α to adjust the size of the neighborhood influence. Therefore, the regression equation of a stealthy transport vector enhancement is combined with the prediction matrix **H** and the stealthy transfer probability α as (11)

$$\mathcal{Z}_{DeepPP} = softmax(\alpha(\mathbf{I}_n - (1-\alpha)\hat{\mathbf{A}})^{-1}\mathbf{H} + \alpha\mathbf{H}) \tag{11}$$

PPNP's linearity is reduced by adding $\alpha\mathbf{H}$. The flexibility of $\alpha\mathbf{H}$ is apparent in practical applications.

To demonstrate the algorithm, Figure 3 shows two examples from the Digg dataset. Social influence prediction aims to predict the behavior state of target node v from neighboring nodes. User v represents the target node. The solid nodes represent the active state '1', and the hollow nodes represent the inactive state '0'. Figure 3(a) shows a social influence prediction with an active node, while Figure 3(b) shows a prediction with an inactive node. Our experiments show that DeepPP predicts the ground truth more accurately than DeepInf.
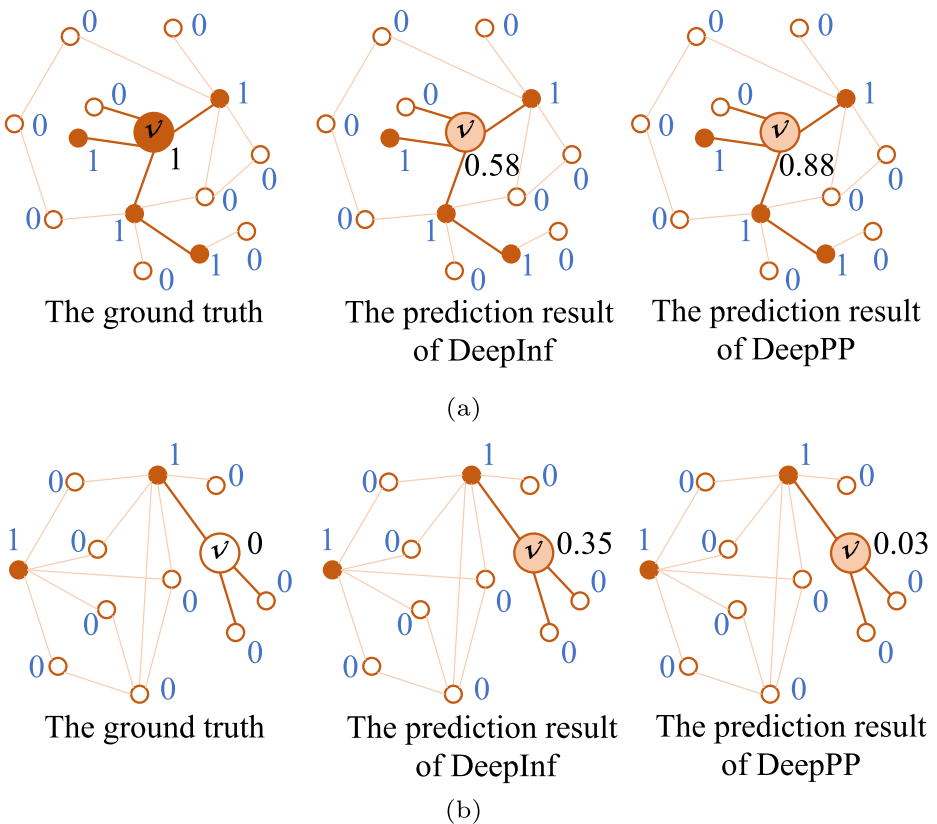
Figure 3 Two samples from the Digg dataset. (a) Social influence prediction with an active node. (b) Social influence prediction with an inactive node

4 Experiments

In our experiments, we compared the DeepPP model with the conventional models GNN* [16], PPNP and APPNP [15], and the most advanced available models DeepInf [14] and DeepEmLAN [17]. To avoid confusion with GNN in a broader sense, the particular method we used has been named GNN*. GNN* is an extension to a common framework that includes information diffusion, relaxation mechanisms, and a random walk model. Its input can be a cyclic graph, a directed graph, an undirected graph, or a mixed graph. DeepEm-LAN integrates different types of attributes and topologies into a single semantic space in a seamless manner, while retaining different types of attributes and topologies to the extent possible. The learning rate for all baselines was set to 0.1, and 1000 epochs for mean square error loss.

4.1 Datasets

To train the classification models, three categories of features were considered for ego-user:(a) vertex features; (b) pretrained network embeddings (in DeepWalk [31]); and (c) hand-crafted features. They are listed in Table 1.

4.1.1 Four social networks in different domains

We conducted the experiment with four datasets from different fields. Table 2 presents the statistical information. $|\mathcal{V}|$ and $|\mathcal{E}|$ represent the number of nodes and edges in graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, respectively. There are N observable instances.

OAG . Microsoft Academic Graph and Aminer are linked in this dataset. [47]. We chose 20 major conferences in the areas of computer science and artificial intelligence, such as SIG-COMM, SIGMOD, AAAI, NeurIPS, similar to the approach described in [48]. The social networks were defined as co-author networks, and social behavior was defined as citation behavior where one researcher cites a paper presented at the conference. Specifically, we want to know how collaborators influence citation behavior.

Digg , a social news site, is based in the United States. There are two features, digging and burying, based on whether people agree with the story. In this dataset, we provide data on the stories that appeared on the front page of major newspapers in 2019.

Table 1 List of features for ego-user

Feature	Description
Vertex	PageRank [15], eigenvector centrality [12], reciprocal of ego user's degree [14], coreness [42], clustering coefficient [43], hub score and authority score [44].
Embedding	Embedding of pre-trained networks
Ego	Active neighbors/ratio [45], the density of subnetworks and the number of connected components formed by active neighbors [46]

Table 2 Statistics for the datasets

symbol	OAG	Digg	Twitter	Weibo
$|\mathcal{V}|$	953,675	71,367	456,626	1,776,950
$|\mathcal{E}|$	4,151,463	1,731,658	12,508,413	308,489,739
N	499,848	27,323	499,160	779,164

Twitter . The Twitter dataset was built from propagations of the announcement of the discovery of the elusive Higgs boson on Twitter in July 2012. This is a friendship network where social behaviors are mapped from retweets in Twitter.

Weibo [14]. Sina Weibo is the most widely used microblogging service in China. It's a social media platform based on user relationships. At present, the monthly active users are 523 million, and the daily active users are 229 million. This dataset contains target tracking networks and tweets during Sept. 28-Oct. 29., 2012.

4.1.2 COVID-19 Datasets

Hubei (see Figure 4). This is a dataset of infected cases provided by the Hubei Provincial Health Committee [1] [49]. In December 2019, COVID-19 first appeared in Wuhan, China. On January 21st, 2020, Hubei Provincial Health Committee reported the first case outside Wuhan. Then from February 15, 2020, the diagnostic policy was changed , resulting in a sharp increase in the number of recorded infection cases. Therefore, this dataset is limited to the period from January 21 to February 14, 2020.

Holland (see Figure 5). This dataset contains data on infection cases collected by the Dutch National Institute for Public Health and the Environment [2] [50]. The first infection, diagnosed on February 27, 2020, went to Italy a week ago, after which the number of cases grew rapidly. Reported cases reached their peak at the end of March, followed by a downward trend in daily reported cases. As reported cases in Holland increased more slowly than that in Hubei, the overall infection period was longer and there are more data points. The gradually increasing number of infected cases is conducive to the accuracy of prediction.
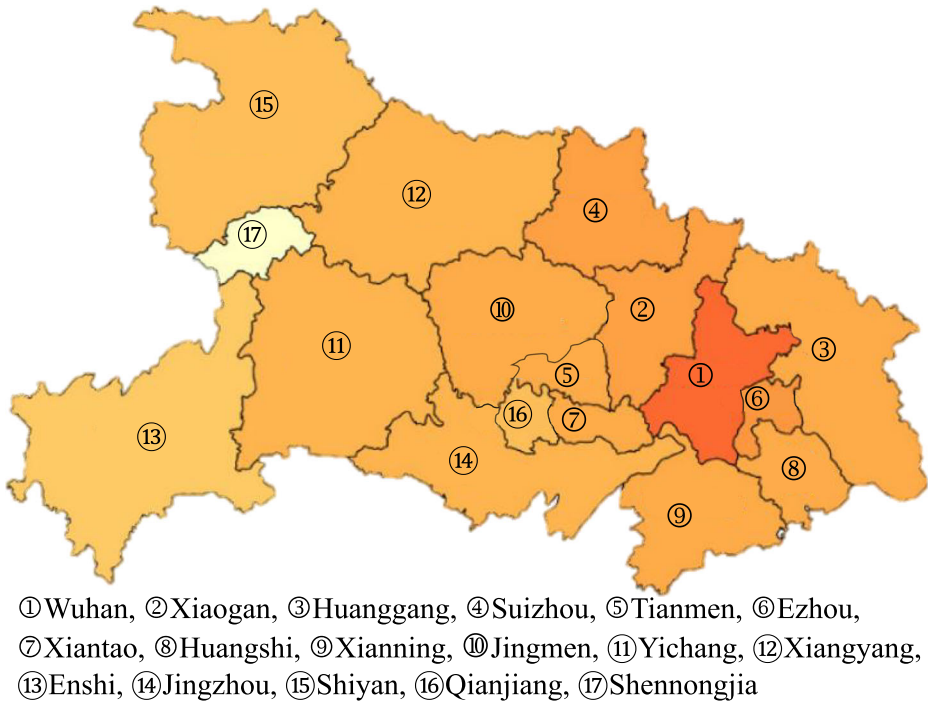
4.2 Evaluation

The following metrics were used to measure performance in the experiments: AUC, $Precision$, $Recall$, and $F1 - Measure$ [51].

We first analyzed the overall performance of methods. We searched each parameter in the threshold space across the six datasets and calculated the corresponding $F1$ to get the most appropriate value, i.e., the best $F1 - Measure$ ($F1_{best}$). Figure 6 shows the $F1_{best}$ score and the total average of the datasets. On $F1_{best}$, DeepPP was on average 2.01% higher than DeepEmLAN, 2.80% higher than APPNP, 3.78% higher than PPNP, 6.26% higher

---

[1] http://wjw.hubei.gov.cn/fbjd

[2] https://www.rivm.nl

①Wuhan, ②Xiaogan, ③Huanggang, ④Suizhou, ⑤Tianmen, ⑥Ezhou, ⑦Xiantao, ⑧Huangshi, ⑨Xianning, ⑩Jingmen, ⑪Yichang, ⑫Xiangyang, ⑬Enshi, ⑭Jingzhou, ⑮Shiyan, ⑯Qianjiang, ⑰Shennongjia

**Figure 4** Infection during COVID-19 in Hubei, China

than DeepInfGAT, and 8.73% higher than DeepInf-GCN. Hence, DeepPP yielded superior results to the baselines.

To determine the effects of hyper-parameters, the transfer probability α was adjusted from $0.2 \leq \alpha \leq 1.0$, in 0.2-step increments. We then ran 1024 mini batches over 1000 epochs. Moreover, we used dropout technology with a dropout rate of 0.2. We compared predictions at $\alpha = 0.2, 0.4, 0.6,$ and 0.8. These results, as detailed in Table 3, clearly indicate that DeepPP outperformed the other six method in terms of AUC, Precision, Recall, and $F1\_Measure$. In DeepInf, DeepInf-GAT had the higher predictive performance than DeepInf-GCN. According to Table 3, DeepPP, DeepEmLAN, APPNP and PPNP were competitive. However, DeepInf-GAT was especially good with regard to Precision on the Weibo, Hubei, and Holland datasets.

Some models based on deep learning were proposed to analyze the social influence, such as the famous DeepInf. We designed a novel model to improve the prediction, which draws on DeepInf, PPNP and APPNP as inspiration. However, compared to the baseline methods, DeepPP has: more flexibility in practical applications, a more convenient way to adjust the parameters; and better performance with large and small datasets.

4.3 Parameters analysis

In terms of the gauging the influence of the other parameters, there are three categories to test: 1) the basic training parameters, which include the window size and the z-variable
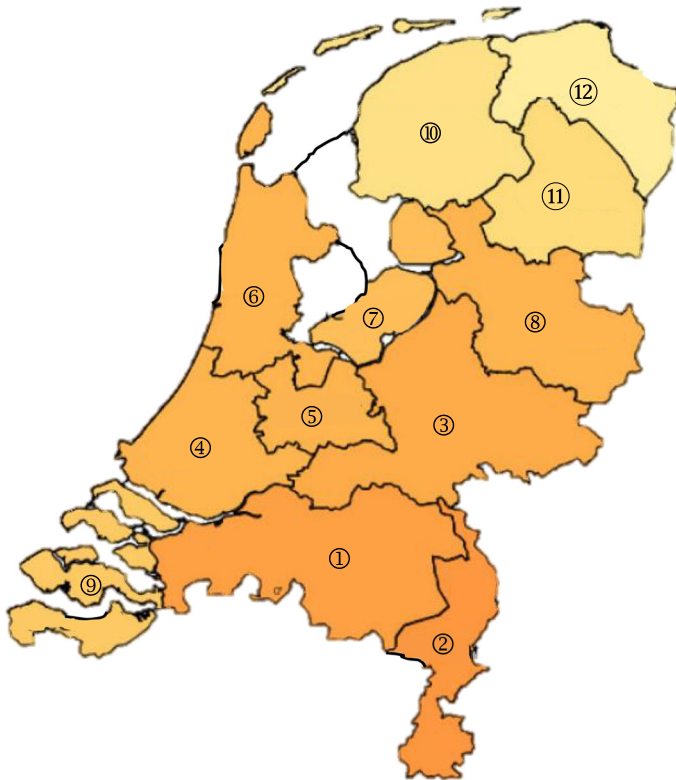
①Noord-Brabant, ②Limburg, ③Gelderland, ④Zuid-holland, ⑤ Utrecht,
⑥Noord-holland, ⑦Flevoland, ⑧Overijssel, ⑨Zeeland, ⑩ Friesland,
⑪Drenthe, ⑫Groningen

Figure 5 Infections during COVID-19 in Holland

dimension of the potential space; 2) the temporal convolution network (TCN) unit param-
eters, i.e., the filter size and the TCN levels; and 3) the score attention parameters. For all
these experiments, we set α to 0.8.

First, we studied the effects of changing the windows size. This directly affects the
length of the time-dependencies in the historical data. The larger the window, the more data
dependencies that can be captured. However, as the window size increases, so does the com-
puting power required, which in turn affects the detection speed. The first row of Figures 7
and 8 shows that five window sizes were tested, i.e., 5, 20, 50, 100, 300. OAG and Weibo
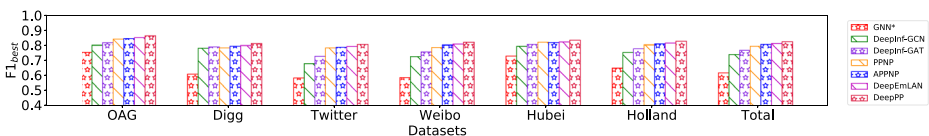
Figure 6 A comparative analysis of DeepPP and baseline methods in F1$_{best}$ across six datasets

Table 3 Comparison of different methods with respect to AUC, Precision, Recall, and $F1 - Measure$, at varying values for α

Data	Model	α=0.2				α=0.4				α=0.6				α=0.8			
		AUC	Prec.	Rec.	F1	AUC	Prec.	Rec.	F1	AUC	Prec.	Rec.	F1	AUC	Prec.	Rec.	F1
OAG	GNN* [16]	62.99	28.16	76.56	43.02	63.60	30.02	74.25	42.09	64.49	31.07	71.74	42.86	65.40	30.08	60.73	43.51
	DeepInf-GCN[14]	60.74	28.16	71.86	40.54	61.65	28.36	72.74	41.65	62.76	29.05	73.28	42.31	63.54	30.30	74.37	43.01
	DeepInf-GAT [14]	63.63	30.62	58.88	43.06	65.12	32.50	59.05	43.80	66.24	33.21	59.99	44.16	66.05	33.31	60.97	44.36
	PPNP [15]	63.17	29.97	76.02	42.97	64.15	30.61	73.79	43.22	65.07	31.62	71.25	43.62	65.90	33.48	64.59	44.19
	APPNP [15]	64.94	30.91	73.87	43.64	65.47	32.42	69.31	44.02	65.78	33.03	66.35	44.18	65.96	33.36	65.51	44.23
	DeepEmLAN [17]	64.01	31.15	69.94	42.59	64.61	32.24	66.43	43.17	65.00	32.56	66.32	44.16	65.62	33.02	65.86	44.29
	DeepPP	68.51	38.36	73.16	46.00	69.58	39.24	68.70	46.97	70.66	40.00	66.70	47.75	71.80	40.79	66.08	48.87
Digg	GNN* [16]	83.90	51.85	70.26	62.87	84.19	61.18	69.32	61.47	84.28	65.36	68.09	64.16	84.81	65.46	69.61	70.07
	DeepInf-GCN [14]	81.06	56.28	64.44	59.36	82.18	57.15	65.65	60.22	83.27	58.04	66.54	61.76	84.16	58.73	67.62	62.89
	DeepInf-GAT [14]	84.89	57.06	67.38	63.52	87.47	63.55	68.64	66.89	88.59	65.24	69.51	71.08	90.17	66.80	70.48	72.19
	PPNP [15]	81.92	52.41	68.33	59.02	85.23	61.75	66.87	63.53	87.51	66.26	67.91	67.06	89.05	68.32	71.53	69.78
	APPNP [15]	85.77	59.50	71.19	64.36	86.81	63.25	68.75	65.52	87.92	68.70	65.25	66.59	89.08	68.39	71.43	69.36
	DeepEmLAN[17]	83.38	56.74	70.49	62.16	86.25	62.36	69.41	65.44	88.10	67.12	71.85	68.76	89.26	70.03	72.46	70.05
	DeepPP	86.32	63.59	71.90	69.23	87.67	64.31	71.25	70.15	89.17	69.24	71.19	69.98	90.64	72.35	71.15	71.38
Twitter	GNN* [16]	74.26	41.76	68.31	50.15	74.90	43.09	67.59	50.94	75.81	44.48	66.72	52.03	76.92	44.39	66.00	53.13
	DeepInf-GCN [14]	74.04	45.16	64.04	50.35	74.93	46.02	65.23	51.16	75.84	46.24	66.00	52.18	76.61	44.41	66.75	53.27
	DeepInf-GAT [14]	75.11	41.75	65.25	54.21	76.20	45.17	67.47	55.13	76.86	46.43	68.36	55.02	77.21	47.43	69.18	56.92
	PPNP [15]	73.98	42.25	64.59	51.01	75.02	42.51	66.83	52.02	76.53	44.61	66.58	53.39	77.32	45.52	66.58	54.71
	APPNP [15]	75.97	43.48	67.28	52.76	76.36	44.62	66.25	53.35	77.22	46.32	65.26	54.20	77.69	46.47	65.26	54.84
	DeepEmLAN [17]	73.28	42.09	68.32	52.61	74.19	43.36	67.13	53.76	75.48	44.09	68.91	54.88	76.09	45.62	68.54	55.56
	DeepPP	74.55	45.45	67.27	51.53	76.38	46.15	65.57	53.46	77.74	46.77	67.02	56.22	78.66	47.73	66.73	55.61

Table 3 (continued)

Data	Model	α=0.2				α=0.4				α=0.6				α=0.8			
		AUC	Prec.	Rec.	F1	AUC	Prec.	Rec.	F1	AUC	Prec.	Rec.	F1	AUC	Prec.	Rec.	F1
Weibo	GNN* [16]	76.36	45.04	64.99	53.15	77.01	43.89	64.18	52.74	77.86	45.93	65.56	53.88	78.82	47.06	66.18	55.00
	DeepInf-GCN [14]	76.01	45.66	64.14	53.57	77.17	46.73	64.02	53.85	78.21	47.24	65.14	54.63	78.99	47.52	66.81	55.05
	DeepInf-GAT [14]	77.82	46.08	64.56	55.39	78.26	46.96	64.23	56.26	80.34	48.21	65.18	56.75	81.52	48.99	68.11	57.42
	PPNP [15]	75.98	45.89	64.34	53.76	77.25	46.81	64.18	55.21	78.49	47.78	65.03	55.89	80.09	48.68	67.51	56.49
	APPNP [15]	77.96	46.45	64.15	54.61	78.48	47.33	64.01	55.36	79.23	47.96	65.26	56.08	80.15	48.83	67.49	56.63
	DeepEmLAN [17]	75.99	44.46	63.01	53.78	77.65	45.71	63.96	54.47	78.44	46.68	65.83	55.58	81.43	47.58	66.92	56.65
	DeepPP	76.49	46.42	64.32	55.57	78.40	47.39	64.05	56.19	79.99	48.56	65.73	56.36	82.13	49.04	67.75	57.29
Hubei	GNN*[16]	77.89	40.03	71.85	61.04	78.57	40.85	70.59	61.67	80.03	41.26	70.34	62.85	81.50	42.03	69.88	63.09
	DeepInf-GCN [14]	78.90	40.02	64.64	62.60	80.01	40.95	65.47	63.56	80.67	41.48	66.38	64.37	81.05	42.51	67.26	65.42
	DeepInf-GAT [14]	78.68	40.27	67.04	66.45	80.29	41.73	68.43	67.59	81.08	42.86	69.85	68.36	81.67	43.95	70.91	69.71
	PPNP [15]	76.24	41.26	66.33	67.54	78.60	42.10	68.04	68.72	80.52	42.86	69.51	69.85	82.00	43.98	70.85	70.69
	APPNP [15]	79.49	42.58	68.72	68.23	79.99	43.46	69.25	69.08	80.31	43.77	70.81	70.26	82.21	44.59	71.72	71.05
	DeepEmLAN [17]	78.05	42.13	73.86	68.72	79.14	43.27	73.06	69.98	80.46	44.13	72.59	70.59	81.85	44.83	71.69	71.08
	DeepPP	79.85	43.76	69.03	68.16	80.57	44.14	71.02	69.95	82.18	44.53	71.66	70.47	83.01	45.76	72.56	71.58
Holland	GNN* [16]	77.01	37.27	63.39	60.54	78.24	38.10	64.71	61.01	79.01	39.46	65.46	61.88	79.90	40.24	66.58	62.97
	DeepInf-GCN [14]	77.44	37.12	64.85	61.32	78.25	37.59	65.74	62.00	79.09	38.78	66.62	62.77	79.91	39.34	67.45	63.48
	DeepInf-GAT[14]	77.53	39.34	67.00	61.86	78.26	40.45	67.57	62.93	79.14	41.03	68.41	63.85	79.96	41.87	69.52	64.63
	PPNP[15]	76.17	39.26	66.27	66.63	78.33	40.03	67.36	67.57	79.41	41.10	68.11	68.13	79.62	41.93	68.94	68.95
	APPNP [15]	77.75	40.01	66.32	66.74	78.71	40.74	67.49	68.25	79.80	41.68	69.01	69.68	79.84	42.71	69.86	70.36
	DeepEmLAN [17]	77.84	40.15	72.30	66.48	78.25	41.93	71.99	68.31	79.04	42.72	71.28	69.66	79.79	42.92	70.17	70.13
	DeepPP	77.26	40.52	64.36	66.91	78.78	41.76	66.43	68.65	80.18	42.44	68.31	69.74	80.39	43.38	70.11	70.59
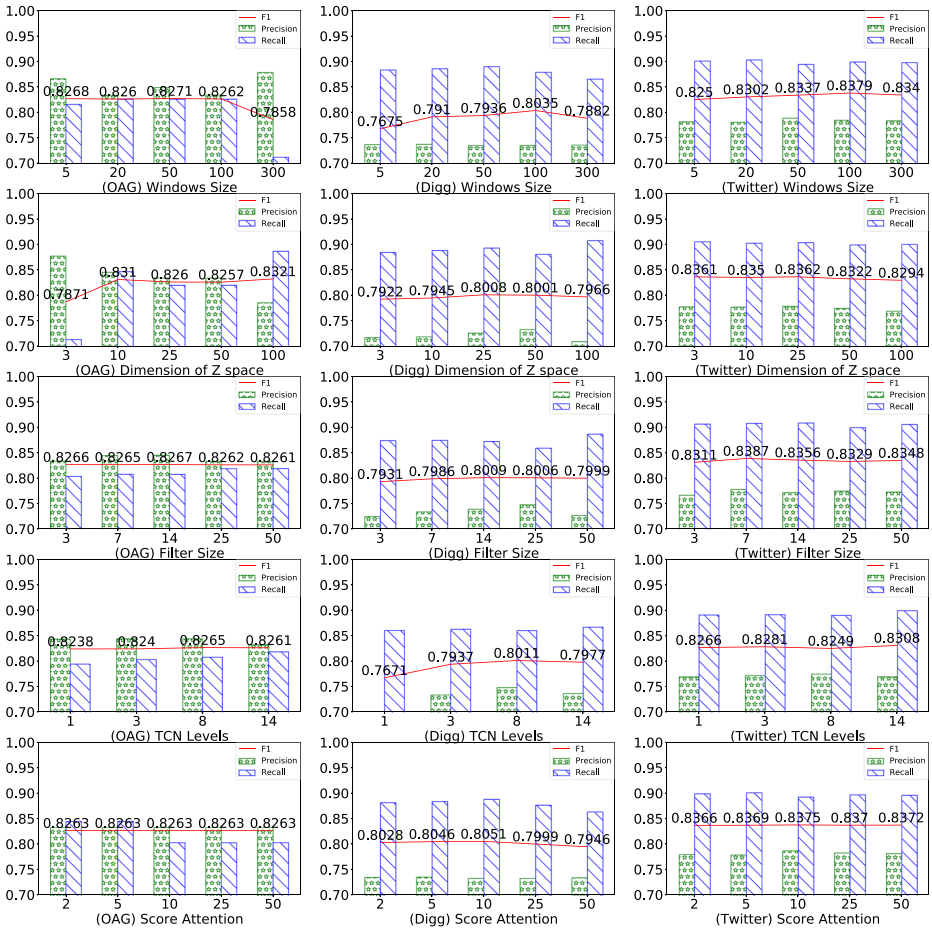
Figure 7 The effect of parameters on OAG, Digg and Twitter

reached maximum $F1 - score$ at a window size of 50, while Digg, Twitter, Hubei and Holland reached their maximums at a window size of 100. From this, we determined that the optimal window size relates to the composition of the dataset. A small window expresses better performance for datasets with weak time correlations. However, with time-dependent datasets, small windows cannot capture long-term dependencies. Additionally, observing the six datasets, all showed a performance degradation with a window size of 300. This indicates that, if the length of the data is too long, the generalization ability of the model will decrease. In addition, a too-large window will lead to a rapid increase in the required computing power, a larger model scale, and a slower training speed.

Next, we studied the effects of the second basic parameter, being the variable z in the m-dimension. The second row of Figures 7 and 8 shows the results for five z-variable dimensions. What we see is poor performance with the OAG dataset with a small z-dimension. This is because the dimension is mapped to a small potential variable space, which results in a large amount of information loss in the encoder stage. In turn, the decoder is unable to recover, resulting in performance degradation. We also observed that changes in the
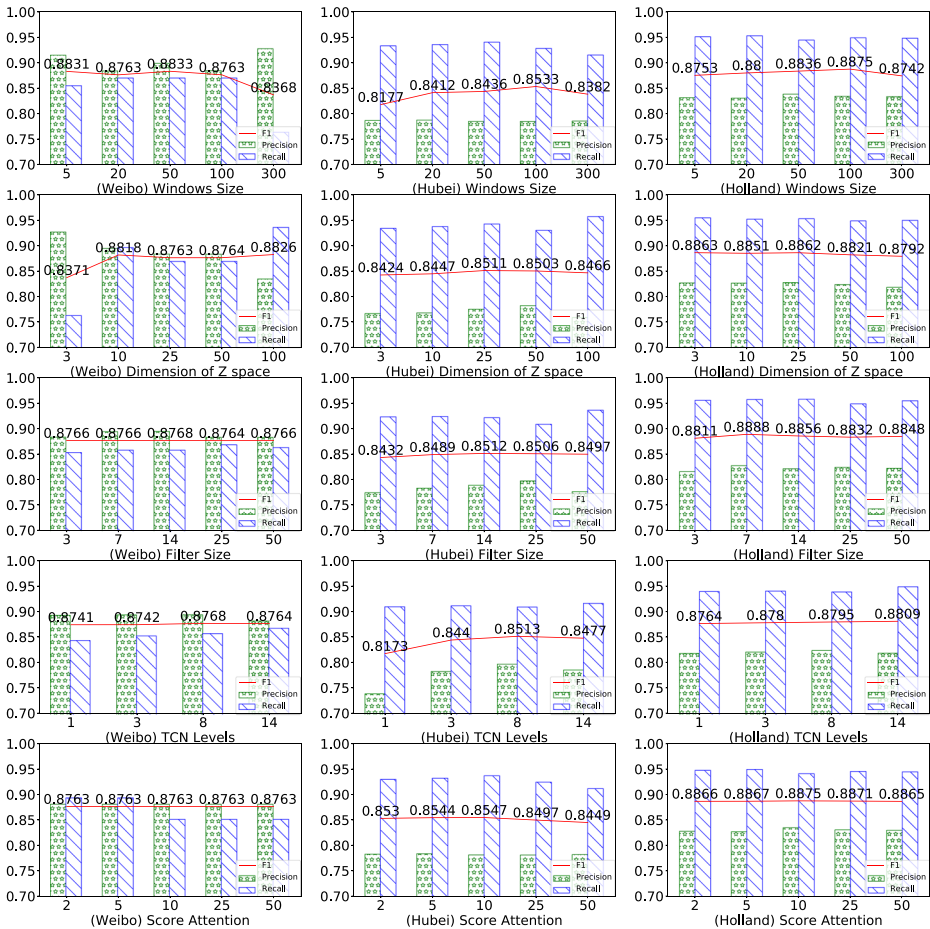
Figure 8 The effect of parameters on Weibo, Hubei and Holland

z-dimension had little effect on performance with the Hubei and Holland datasets. As *z* increased, the loss in the training process changed greatly. However, many iterations helped the models to stabilize.

The TCN filter size was next. Here, we fixed the expansion factor *d*, so only the filter size needed to be adjusted to change the field size. The third row of Figures 7 and 8 show the results for five filter sizes. As shown, the optimal filter size for the smaller Digg dataset was 7, while for the Hubei and Holland datasets, the optimal size was 14. This indicates that the optimal filter size is determined by the size of the dataset. However, due to the fixed expansion coefficient, a model's performance is relatively unaffected by the choice of filter size.

In terms of TCN levels, we found that changes in the TCN level directly affected the scale of the DeepPP model. The fourth row of Figures 7 and 8 shows the results for four TCN levels. From a data perspective, OAG and Digg (group training) have a smaller data scale, and so we saw better performance when using a smaller TCN level. Conversely, for the large-scale Hubei and Holland datasets, representation performance was greatly reduced

at a TCN level of 1. Moreover, the smaller models could not capture the time dependencies effectively. From a model point of view, performance was excellent with a TCN level of 8. However, at a TCN level of 14, although the Weibo dataset showed a slight improvement, the size of the model almost doubled, which is unworkable in practice.

Next, we looked at score attention, varying the parameter's value across 2, 5, 10, 25, 50. The role of this mechanism is to improve the accuracy of abnormal data that is close to normal. The results, appearing in the fifth row of Figures 7 and 8, show that varying this parameter has little effect on the results. This therefore warrants further attention in our ablation experiments.

Test Loss Figure 9 shows a common trend, namely, that the final test loss decreases with an increase in α. In fact, the results show that the larger α is, the less the test loss and the better the performance. From a comparison of four models with different datasets, as shown in Figure 9, we see that DeepPP had better performance than any of the other algorithms.

4.4 Spread analysis of the COVID-19

To illustrate the advantages and disadvantages of our method, we chose a COVID-19 dataset from two different regions, Hubei and Holland. Although these two areas do not represent complete representations of COVID-19's spread, they can reflect the spread of infectious diseases in general.

4.4.1 Hubei

Figure 10 shows the change in prediction accuracy of different prediction algorithms over time. The dates are displayed along the horizontal axis. From January 22, we used all the available information to make COVID-19 day-ahead prediction. For example, in Figure 10(a), the rightmost point displays the results from January 22 to February 13 to predict what will happen on February 14.

Over time, the absolute percentage mean error (APME) error has a tendency to decrease as the amount of data available increases. A rapid increase in infection cases was followed
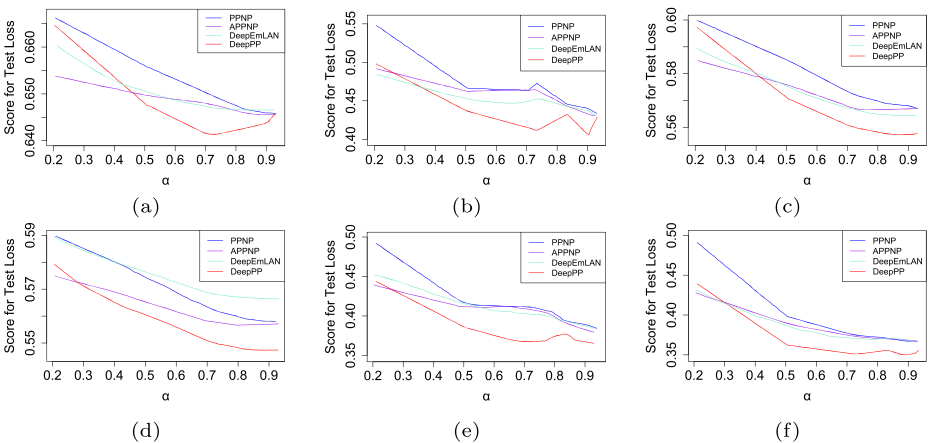
Figure 9 Test loss with $\alpha \in (0.2, 1.0)$. (a) Test loss on OAG, (b) Test loss on Digg, (c) Test loss on Twitter, (d) Test loss on Weibo, (e) Test loss on COVID-19 in Hubei, (f) Test loss on COVID-19 in Holland
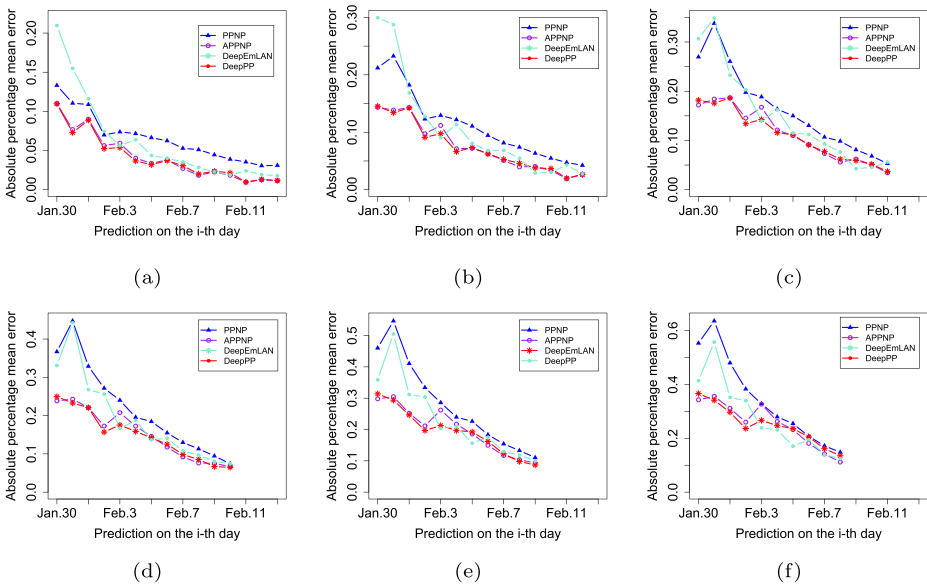
Figure 10 COVID-19 day-ahead predictions from Hubei China. Predictions are given over a 1-to-6-day interval (see subfigures (a) to (f))

by a more gradual trend, displaying a sub exponential rise in daily infections. The APME error will decrease with sub-exponential growth since it is a measure of relative error. Also, as the prediction horizon extends, prediction accuracy decreases rapidly. As shown in Figure 10(e) and (f), it is not possible to accurately predict the number of cases five or six days before and after February 1.

DeepEmLAN performed better, but it did not find an accurate prediction before and after January 31. The time series in the leftmost part of Figure 10 is the shortest, so there was less data available to train DeepEmLAN. In these cases with short time series, the prediction accuracy of this pure machine learning algorithm was lower than the other methods.

4.4.2 Holland

Prediction accuracy with the Holland dataset is shown in Figure 11. The COVID-19 situation in Holland was essentially the same as in Hubei prior to April 1, 2020. Here, DeepPP proved to be the best method but with large deviations in prediction accuracy. All compared algorithms have been roughly the same since April 1. Also, whether the network is initially static or dynamic appears to have little effect on prediction accuracy. The DeepPP algorithm is trained on more and more infection data over time. The DeepInf-GCN and DeepInf-GAT separation methods performed best in the whole cycle, while PPNP and DeepEmLAN performed worst.

The prediction accuracy of DeepPP and APPNP is comparable. One possible reason is that the transmission of COVID-19 was primarily inter provincial interaction. The COVID-19 spread mainly in the provinces after the end of March. We then compared the spread of COVID-19 across all seven algorithms. The errors are listed in Table 4, obtained by averaging all APME prediction errors over a 1-to-6-day prediction interval. Table 4 clearly illustrates that the prediction error of DeepPP algorithm is smaller than the other algorithms,
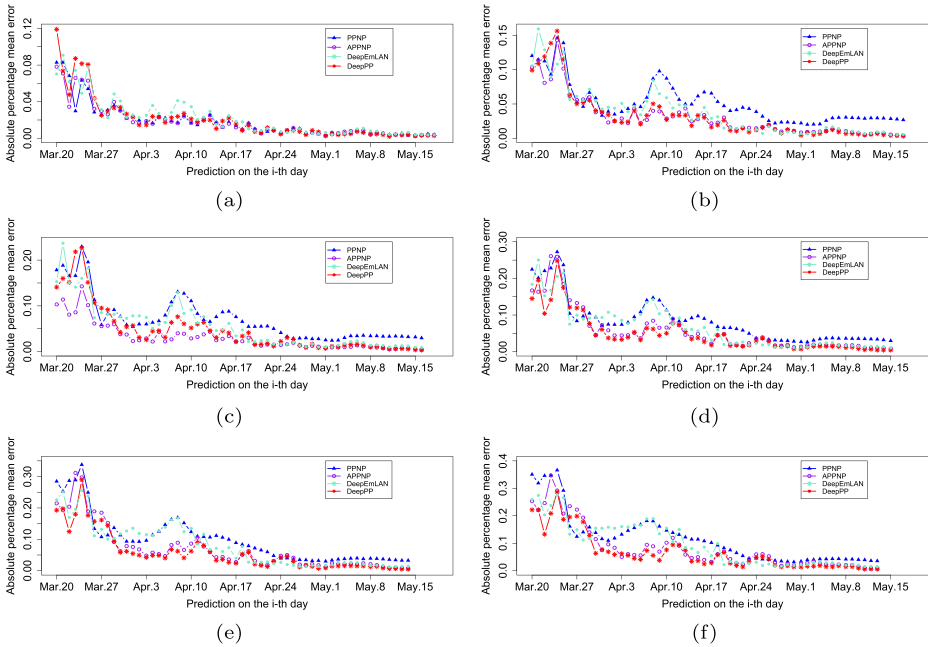
Figure 11 COVID-19 day-ahead predictions from Holland. Predictions are given over a 1-to-6-day interval (see subfigures (a) to (f))

because DeepPP takes into account cities interact with each other. The prediction errors for DeepPP in each city are equal to that of DeepEmLAN. In conclusion, the network-based method offers better prediction accuracy.

4.4.3 Ablation experiments

In the ablation experiments, we compared multiple variants of DeepPP: DeepPP-RNN, DeepPP-noPNF, DeepPP-noScoreAttention, DeepPP-noPointAdjust, using the DeepPP prototype as the control model. DeepPP-RNN replaces the TCN unit with an RNN. DeepPP-noPNF replaces the transformation process in the potential space with a Gaussian distribution $N(0, 1)$. DeepPP-noScoreAttention omits the score attention mechanism and,

Table 4 Error comparison of different methods

Algorithm	Error in Hubei	Error in Holland	Bias
GNN*	0.15	0.043	under
DeepInf-GCN	0.14	0.042	over
DeepInf-GAT	0.13	0.044	
PPNP	0.14	0.051	over
APPNP	0.14	0.043	
DeepEmLAN	0.16	0.057	under
DeepPP	0.13	0.038	
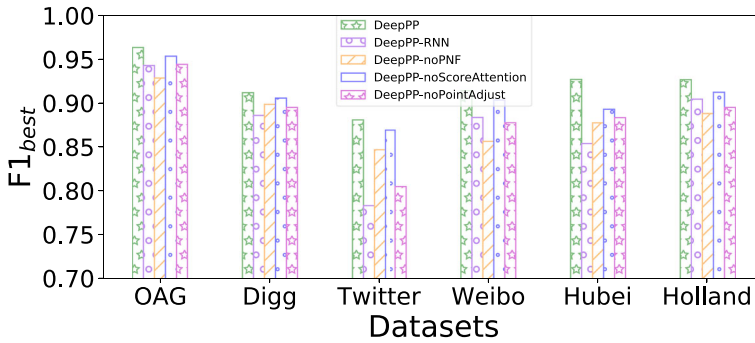
Figure 12 A comparative analysis of ablation experiment in $F1_{best}$ on six datasets

similarly, DeepPP-noPointAdjust omits the point adjust method. For fairness, we also eliminated the peaks-over-threshold (POT) mechanism in our model and used $F1_{best}$ as the evaluation indicator.

As can be seen from Figure 12, DeepPP-RNN did not perform as well as DeepPP. This is because a simple RNN cannot capture the long-term dependencies in time series data. Moreover, DeepPP resulted in a much smaller model that took far less time to train compared to DeepPP-RNN. Hence, training DeepPP would be much easier. From the perspective of the impact of latent space, PNF can capture complex latent variable data patterns and help build and generate latent space variables. From DeepPP-noPNF and DeepPP-RNN in Figure 12, there is evidence that the noPNF variant has a lesser impact on the model compared to the RNN variant, while the RNN variant had a greater impact with the Digg dataset. This is because the scale of packet data in the Digg dataset is very small, so the RNN was able to capture the time dependencies.

In general, the impact of DeepPP-noScoreAttention and DeepPP-noPointAdjust on model organization was smaller than DeepPP-RNN and DeepPP-noPNF. After DeepPP-noScoreAttention eliminates the abnormal scoring mechanism, the gap between DeepPP-noScoreAttention and DeepPP is very small. This is because the score attention mechanism has abnormal data close to normal data. The detection results have an impact, but the impact is much smaller than with the overall model. DeepPP-noPointAdjust had the greatest impact with the Twitter dataset. This is because point adjustment can effectively detect continuous anomaly types. Compared to abnormally scattered datasets and massive data, the impact is much greater.

5 Conclusion

We presented a deep learning-based personalized propagation algorithm, referred to as DeepPP. By extending DeepInf, this algorithm integrates the transition probability of the page rank domain with a GCN. The method can adjust the size of a neighbor's influence. It also has greater flexibility. A variety of social networks (OAG, Digg, Twitter, and Weibo) as well as two COVID-19 datasets (Hubei and Holland) were studied in extensive experiments. As demonstrated by the results, DeepPP yielded better $F1 - Measure$ than the current advanced baselines. The proposed method performed better on the COVID-19 datasets in precision. The excellent performance of DeepPP on various datasets proves that it can be effectively applied to general real-world scenarios for predicting the social influence of

COVID-19. However, although we used a transition probability to achieve greater flexibility in our modeling, we did not consider any user-specified constraints. Further exploring and incorporating these into the model is a worthy future research direction. Another exciting direction would be to use reinforcement learning to combine sampling and learning for modeling the social influence of COVID-19.

Declarations

Competing interests 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.

References

1. Lu, W.-X., Zhou, C., Wu, J.: Big social network influence maximization via recursively estimating influence spread. Knowl.-Based Syst. **113**, 143–154 (2016)
2. Wang, H., Wu, J., Pan, S., Zhang, P., Chen, L.: Towards large-scale social networks with online diffusion provenance detection. Comput. Netw. **114**, 154–166 (2017)
3. Li, N., Guo, B., Liu, Y., Yao, L., Liu, J., Yu, Z.: Askme: joint individual-level and community-level behavior interaction for question recommendation. World Wide Web **25**(1), 49–72 (2022)
4. Du, H., Zhou, Y.: Nostradamus: a novel event propagation prediction approach with spatio-temporal characteristics in non-euclidean space. Neural Netw. **145**, 386–394 (2022)
5. Daud, N.N., Ab Hamid, S.H., Saadoon, M., Sahran, F., Anuar, N.B.: Applications of link prediction in social networks: a review. J. Netw. Comput. Appl. **166**, 102716 (2020)
6. Liu, C.-Y., Zhou, C., Wu, J., Hu, Y., Guo, L.: Social recommendation with an essential preference space. In: Thirty-second AAAI conference on artificial intelligence (2018)
7. Gao, L., Wu, J., Qiao, Z., Zhou, C., Yang, H., Hu, Y.: Collaborative social group influence for event recommendation. In: Proceedings of the 25th ACM international on conference on information and knowledge management. CIKM '16, pp. 1941–1944 (2016)
8. Raj, C., Meel, P.: Arcnn framework for multimodal infodemic detection. Neural Netw. **146**, 36–68 (2022)
9. Yang, Z., Li, Q., Xie, H., Wang, Q., Liu, W.: Learning representation from multiple media domains for enhanced event discovery. Pattern Recogn. **110**, 107640 (2021)
10. Liu, F., Xue, S., Wu, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Yang, J., Yu, P.S.: Deep learning for community detection: Progress, challenges and opportunities. In: Proceedings of the Twenty-Ninth international joint conference on artificial intelligence, IJCAI, pp. 4981–4987 (2020)
11. Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q.Z., Xiong, H., Akoglu, L.: A comprehensive survey on graph anomaly detection with deep learning. IEEE Transactions on Knowledge and Data Engineering (2021)

12. Matsubara, Y., Sakurai, Y., Prakash, B.A., Li, L., Faloutsos, C.: Rise and fall patterns of information diffusion: model and implications. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 6–14 (2012)

13. Li, C., Ma, J., Guo, X., Mei, Q.: Deepcas: an end-to-end predictor of information cascades. In: Proceedings of the 26th international conference on world wide web, pp. 577–586 (2017)

14. Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K., Tang, J.: Deepinf: social influence prediction with deep learning. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 2110–2119 (2018)

15. Klicpera, J., Bojchevski, A., Günnemann, S.: Personalized embedding propagation: combining neural networks on graphs with personalized pagerank. CoRR. 1810.05997 (2018)

16. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Netw. **20**(1), 61–80 (2009)

17. Zhao, Z., Zhou, H., Li, C., Tang, J., Zeng, Q.: Deepemlan: deep embedding learning for attributed networks. Inform. Sci. **543**, 382–397 (2021)

18. Li, K., Zhang, L., Huang, H.: Social influence analysis: Models, methods, and evaluation. Engineering **4**(1), 40–46 (2018). Cybersecurity

19. Koskinen, J., Daraganova, G.: Bayesian Analysis of Social Influence. arXiv (2020)

20. Wu, J., Pan, S., Zhu, X., Cai, Z., Zhang, P., Zhang, C.: Self-adaptive attribute weighting for naive bayes classification. Expert Syst. Appl. **42**(3), 1487–1502 (2015)

21. Wu, J., Cai, Z.: Attribute weighting via differential evolution algorithm for attribute weighted naive bayes (wnb). Journal of Computational Information Systems **7**(5), 1672–1679 (2011)

22. Liao, L., He, X., Zhang, H., Chua, T.-S.: Attributed social network embedding. IEEE Trans. Knowl. Data Eng. **30**(12), 2257–2270 (2018)

23. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: WWW, pp. 2022–2032 (2019)

24. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: International conference on machine learning, pp. 6861–6871. PMLR (2019)

25. Zhang, Q., Wu, J., Zhang, P., Long, G., Zhang, C.: Salient subsequence learning for time series clustering. IEEE Trans. Pattern Anal. Mach. Intell. **41**(9), 2193–2207 (2018)

26. Wu, Q., Zhang, H., Gao, X., He, P., Weng, P., Gao, H., Chen, G.: Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In: The world wide web conference, pp. 2091–2102 (2019)

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

28. Menon, A., Lee, Y.: Proper loss functions for nonlinear hawkes processes. In: Proceedings of the AAAI conference on artificial intelligence, vol. 32 (2018.)

29. Gou, C., Shen, H., Du, P., Wu, D., Liu, Y., Cheng, X.: Learning sequential features for cascade outbreak prediction. Knowl. Inf. Syst. **57**(3), 721–739 (2018)

30. Kefato, Z.T., Sheikh, N., Bahri, L., Soliman, A., Montresor, A., Girdzijauskas, S.: Cas2vec: Network-agnostic cascade prediction in online social networks. In: 2018 Fifth international conference on social networks analysis, management and security (SNAMS), pp. 72–79. IEEE (2018)

31. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 701–710 (2014)

32. Wang, X., Cui, P., Wang, J., Pei, J., Zhu, W., Yang, S.: Community preserving network embedding. In: AAAI, pp. 203–209 (2017)

33. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 855–864 (2016)

34. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 135–144 (2017)

35. Liu, X., Murata, T., Kim, K.-S., Kotarasu, C., Zhuang, C.: A general view for network embedding as matrix factorization. In: Proceedings of the Twelfth ACM international conference on web search and data mining, pp. 375–383 (2019)

36. Siglidis, G., Nikolentzos, G., Limnios, S., Giatsidis, C., Skianis, K., Vazirgiannis, M.: Grakel: a graph kernel library in python. J. Mach. Learn. Res. **21**(54), 1–5 (2020)

37. Niepert, M., Ahmed, M., Kutzkov, K.: Learning convolutional neural networks for graphs. In: International conference on machine learning, pp. 2014–2023. PMLR (2016)

38. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: Proceedings of the 31st international conference on neural information processing systems. NIPS'17, pp. 1025–1035 (2017)
39. Ksantini, R., Ziou, D., Colin, B., Dubeau, F.: Weighted pseudometric discriminatory power improvement using a bayesian logistic regression model based on a variational method. IEEE transactions on pattern analysis and machine intelligence **30**(2), 253–266 (2007)
40. Song, W., Xiao, Z., Wang, Y., Charlin, L., Zhang, M., Tang, J.: Session-based social recommendation via dynamic graph attention networks. In: Proceedings of the Twelfth ACM international conference on web search and data mining, pp. 555–563 (2019)
41. Martin, A., Becker, S.I., Pegna, A.J.: Attention is prioritised for proximate and approaching fearful faces. Cortex **134**, 52–64 (2021)
42. Batagelj, V., Zavernik, M.: Cores decomposition of networks. recent trends in graph theory,algebraic combinatorics, and graph algorithms, pp. 24–27 (2001)
43. Schank, T., Wagner, D.: Approximating clustering coefficient and transitivity. Journal of Graph Algorithms and Applications **9**(2), 265–275 (2005)
44. Zhang, J., Tang, J., Li, J., Liu, Y., Xing, C.: Who influenced you? predicting retweet via social influence locality. ACM Transactions on Knowledge Discovery from Data (TKDD) **9**(3), 1–26 (2015)
45. Li, H., Liu, X., Li, T., Gan, R.: A novel density-based clustering algorithm using nearest neighbor graph. Pattern Recogn. **102**, 107206 (2020)
46. Ugander, J., Backstrom, L., Marlow, C., Kleinberg, J.: Structural diversity in social contagion. Proc. Natl. Acad. Sci. **109**(16), 5962–5966 (2012)
47. Tang, J.: Aminer: Toward understanding big scholar data. In: Proceedings of the Ninth ACM international conference on web search and data mining. WSDM '16, pp. 467. Association for Computing Machinery (2016)
48. Zhuang, C., Ma, Q.: Dual graph convolutional networks for graph-based semi-supervised classification. In: Proceedings of the 2018 world wide web conference, pp. 499–508 (2018)
49. Zhang, Z., Yao, W., Wang, Y., Long, C., Fu, X.: Wuhan and hubei covid-19 mortality analysis reveals the critical role of timely supply of medical resources. J. Infect. **81**(1), 147–178 (2020)
50. Guo, J., Carli, P.D., Lodder, P., Bakermans-Kranenburg, M.J., Riem, M.M.E.: Maternal mental health during the covid-19 lockdown in china, italy, and the netherlands: a cross-validation study. Psychol. Med, pp 1–44 (2021)
51. Wu, J., Cai, Z.-h., Ao, S.: Hybrid dynamic k-nearest-neighbour and distance and attribute weighted method for classification. Int. J. Comput. Appl. Technol. **43**(4), 378–384 (2012)

Affiliations

Yufei Liu[1,4] · Jie Cao[2] · Jia Wu[3] · Dechang Pi[4]

Yufei Liu
liuyufei@nuaa.edu.cn

Jia Wu
jia.wu@mq.edu.au

Dechang Pi
dc.pi@nuaa.edu.cn

[1]　College of Information Engineering, Nanjing University of Finance and Economics, Nanjing, 210023, China

[2]　School of Management, Hefei University of Technology, Hefei, 230009, China

[3]　School of Computing, Macquarie University, Sydney, 2109, NSW, Australia

[4]　College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, 211106, China