Expert Systems With Applications 202 (2022) 117053

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

GCNFusion: An efficient graph convolutional network based model for information diffusion

Bahareh Fatemi [a,1], Soheila Molaei [b,*,1], Shirui Pan [c], Samira Abbasgholizadeh Rahimi [d,e]

[a] Department of Information Science and Media Studies, University of Bergen, Norway
[b] Department of Network Science and Technologies, University of Tehran, Iran
[c] Faculty of Information Technology, Monash University, Australia
[d] Department of Family Medicine, McGill University, Canada
[e] Mila-Quebec AI Institute, Canada

ARTICLE INFO

Keywords:
Feature selection
Graph embedding
Graph neural networks
Influential nodes
Information diffusion

ABSTRACT

Investigating the dynamics of spreading processes in real-world applications such as pathogen spread prediction, marketing, political events, etc has attracted the attention of researchers from a variety of fields. Influence-based information diffusion is one convincing attempt to solve the information diffusion problem. In this regard, most of the attempts suffer from certain drawbacks such as complexity, dependency on the underlying diffusion model, or low prediction accuracy. We have looked at this problem from a fresh perspective and come up with an innovative solution for solving it. Our hybrid approach falls at the intersection of three research areas: feature selection, graph embedding, and information dissemination. To discover the influential nodes in a network, we develop a method comparable to wrapper methods in feature selection, in which we employ the strength of graph convolutional neural networks (GCNs). The results of our implementation in Python on five datasets Cora, Email, Hamster, Router, and CEnew, under the susceptible–infected–recovered (SIR) model, approved that GCNFusion exceptionally outperforms benchmark methods by respectively around 3%, 5%, 5%, 2%, and 3%. Furthermore, the proposed method is a decent suit for real-world applications on complex networks due to its low computational complexity.

1. Introduction

After the COVID-19 pandemic, the necessity of studying information diffusion is tangible and well-understood. The applications of such studies vary in the wide range of predicting the spread of pathogens (Chinazzi et al., 2020; Ye et al., 2020), ideas (Rehman, Jiang, Rehman, Paul, din et al., 2020), or computer viruses (Chenquan et al., 2020), to diverse applications defined on networks, including link prediction (Singh, Mishra, Kumar, & Biswas, 2020). Information diffusion is a genuinely interdisciplinary topic to the extent that researchers from various disciplines, including computer science, social sciences, political sciences, and medical sciences, etc have been seriously pursuing it (Cao, Han, & Zhu, 2021; Chen, Jiang, Zhang, & Chen, 2021; Chen et al., 2019; Kumar et al., 2021; Moghanian, Saravi, Javidi, & Sheybani, 2020; Qi, Li, Chen, & Xue, 2021; Srinidhi, Ciga, & Martel, 2021; Yao et al., 2020; Zhang, Jiang, & Li, 2021).

Online social networks (OSNs) have enabled people around the world to instantaneously share a variety of information. By analyzing the patterns of the diffused information like «memes» or «tweets» on the network, one can predict the consumer's behavior in purchasing products, just the way companies such as Philips, Microsoft, and HP do, to identify the «influential nodes» of the network, before releasing their products (Libai, Muller, & Peres, 2013). Such analytics have the ability to even predict election outcomes (Cinelli, Cresci, Galeazzi, Quattrociocchi, & Tesconi, 2020; Hughes & Palen, 2009; Padda, 2020).

Real-world networks are commonly large graphs that represent entities (such as individuals) as the «nodes» and the relationship between entities (such as friendship) as the «edges» between them. Through these edges, a piece of information is spread on the graph. In such a setting, for investigating the characteristics of network dynamics, instead of leveraging conventional network analytic tasks with their relatively high computational cost, we can alternatively take advantage of the most recent and efficient methods proposed for learning on graphs.

* Corresponding author.
E-mail addresses: Bahareh.Fatemi@uib.no (B. Fatemi), soheila.molaei@ut.ac.ir (S. Molaei), shirui.pan@monash.edu (S. Pan), samira.rahimi@mcgill.ca (S.A. Rahimi).
[1] Equal Contribution.

https://doi.org/10.1016/j.eswa.2022.117053
Received 8 September 2021; Received in revised form 3 January 2022; Accepted 28 March 2022
Available online 9 April 2022

Predictive diffusion models can broadly be categorized into graph-based models and non-graph-based ones (Jalili & Perc, 2017). Epidemic models fall into the former class. In these models, which are inspired by the models proposed for investigating the spread of pathogens, nodes can adopt particular statuses over time. Changes in these statuses, reflect the diffusion process on the network. Adding more statuses to the model, makes it fits the actual spread better. However, the speed of diffusion will be changed. Thus, more complicated models are slower when they reach their stable state (Li, Wang, Gao and Zhang, 2017). Models in the latter category, study the diffusion at the level of individual nodes. Some piece of information starts propagating from an initial set of nodes and then based on a cascade model gets spread through the whole network. Although these methods are effective in describing the information diffusion, they are greatly dependent on the underlying diffusion model and are not well suited to cascade prediction (Chen et al., 2019).

As opposed to previous methods, our proposed one, not only operates independently from any underlying diffusion model but also is capable of capturing more complex nonlinear structures of the input graph. Moreover, our method has a high generalization power and works regardless of different types of networks and cascades. We propose an innovative graph-based approach that uses the astounding potential of graph neural networks (GCNs). We present our proposed method by combining three research areas: Feature Selection, node Representation Learning, and Information Diffusion. Graph convolutional neural networks are a generalized variant of convolutional neural networks (CNNs) that operate directly on non-euclidean graph data. In the literature, GCNs have mainly been employed for graph representation learning but in this study, we use these networks as a proxy for classifying graph nodes. On the other hand, feature selection algorithms select a subset of relevant features based on predefined criteria, to facilitate the implementation of machine learning algorithms, especially on massive datasets. Our ultimate goal in this study is to find a set of network nodes that have the ability to disseminate information vastly. To this end, by making some changes, we have generalized one of the most up-to-date methods for feature selection to be used specifically on graph data. To the best of our knowledge, no research has ever approached the information diffusion problem in such a framework.

This paper's main contributions are summarized as follows:

- Considering the fact that most graph data in real-world settings is unlabeled, supervised techniques appear to lose their functionality for most graph-based algorithms, to that end, we present an unsupervised model for Information Diffusion on graphs.
- Inspired by wrapper approaches on feature selection, we present a novel method for identifying graph influential nodes, combining two neural network-based methods proposed for node classification and feature selection.
- Eventually by disseminating the information on some real-world graphs, using the Susceptible–Infected–Recovered (SIR) simulation model, we conduct several experiments on several types of real-world graph data to affirm the effectiveness of our information diffusion proposed method.

The rest of the article is organized as follows. First, in Section 2, we provide a general overview of the three topics related to this research and highlight some of the most critical studies performed on each. Later on, we will define the problem and delineate our proposed influence-based approach for solving it in Section 3.1. The experimental results are demonstrated in Sections 4 and 5 includes conclusions and future works.

2. Related works

There exist broad ranges of related studies on feature selection, node representation, and information diffusion in the literature. In this section, we take a general overview of each.

Node embedding

Due to the emerging applications of machine learning on graphs as a result of the huge information amount being produced in the form of complex networks, Graph embedding, in particular, node embedding has attracted great attention recently since it is the means by which machine learning applications can make predictions or discover new patterns (Hamilton, Ying, & Leskovec, 2017). Graph embedding or graph representation methods can be categorized into three main categories, (1) Factorization-based models such as Laplacian eigenmaps (Belkin & Niyogi, 2001), GraRep (Cao, Lu, & Xu, 2015), and HOPE (Ou, Cui, Pei, Zhang, & Zhu, 2016) which are amongst early methods in graph learning that are centered around matrix factorization. These early methods of representation learning obtain the node embedding vectors by decomposing the graph's adjacency information. Besides their high computational complexity, the main limitation of these methods is that they fail to use content information. (2) Random-walk-based methods such as DeepWalk (Perozzi, Al-Rfou, & Skiena, 2014) and node2vec (Grover & Leskovec, 2016) with the main intuition of mapping each node into a space in which the co-occurrence probability of similar nodes remains preserved. These methods overemphasize proximity information and neglect other structural information, furthermore, they are too dependent on hyperparameters. (3) Deep-learning-based approaches, in particular graph convolutional networks (GCNs) which are variants of Convolutional neural networks (CNNs) and currently are regarded as the most efficient methods. GCN (Kipf & Welling, 2017) employs an aggregation on the first-order graph neighborhoods to obtain representation vectors using both topological and feature information, DGI (Velickovic et al., 2019) uses a mutual information maximization strategy to learn node representation vectors, MAPPING (Fatemi, Molaei, Zare, & Pan, 2021) obtains node representation vectors aggregating graph neural networks with a manifold learning algorithm.

Feature selection

Feature selection refers to the task of removing irrelevant and redundant features of data that leads to a reduction of dimensionality as well as an improvement in learning performance. This redundancy elimination can be seen as the intersection of the feature selection and influential nodes identification in the sense of respectively trying to determine the most important features and nodes. Various feature selection approaches in the broad classes of wrapper, filter, embedded, hybrid, and ensemble methods (Zebari, Abdulazeez, Zeebaree, Zebari, & Saeed, 2020) have been proposed. Wrapper methods, using a search algorithm make some subsets of features and then evaluate each set by some learning algorithm such as a classifier. Filter methods, on the contrary, work independently from learning algorithms and for example, take advantage of information theory as a preprocessing to rank the features. Embedded methods leverage a training process as a variable selection using a single learner, without splitting. Deep learning methods have proven their remarkable power in the feature selection domain. For instance, Chang, Rampasek, and Goldenberg (2018) leverages the dropout layer as the strategy for feature ranking purposes. Kasneci and Gottron (2016) estimates a local linear model for each neuron and then propagates and aggregates these models. Borisov, Haug, and Kasneci (2019) proposes to add a new specific layer to the network that can be used for feature ranking and feature selection purposes.

Information diffusion

The information diffusion analysis problem has been addressed with various approaches in the literature, in an overall view, two main branches of the diffusion studies are "diffusion models" and "identification of influential nodes" (Zhang, Luo, & Boncella, 2020). The former

Table 1
Comparison of some research on information diffusion.

Research	Category	Cascading model	Approach
Gomez-Rodriguez, Leskovec, and Schölkopf (2013)	Diffusion models	IC	Survival theory
Lagnier, Denoyer, Gaussier, and Gallinari (2013)	Diffusion models	LT	Probabilistic model
Ohsaka, Sonobe, Fujita, and Kawarabayashi (2017)	Influence-based	IC	–
Zhang, Chen, Dong, and Zhao (2016)	Influence-based	SIR	Node ranking
Molaei, Farahbakhsh, Salehi, and Crespi (2020)	Influence-based	SI	Information entropy
Guo et al. (2020)	Influence-based	SIR	Information entropy
Chen et al. (2019)	Deep learning	–	LSTM+GCN.
Cao et al. (2021)	Deep learning	–	GNN
GCNFusion	**Hybrid**	SIR	**GCN**

one includes the first attempts in the field such as Susceptible–Infected–Susceptible (SIS) or Susceptible–Infected–Recovered (SIR) models that work independently from the graph topological data (Hethcote, 2000; Leskovec, McGlohon, Faloutsos, Glance, & Hurst, 2007). On the contrarily, the Independent Cascade model (IC) (Goldenberg, Libai, & Muller, 2001) or Linear Threshold model (LT) (Yang & Leskovec, 2010), which are built on a directed graph with each node having the ability to be activated or deactivated, leverage graph topological data. The main strategy of identification of influential nodes, on the other hand, is to find a set of nodes that have the capability to initiate vast spreadings due to their advantageous positions in the underlying networks (Kwak, Lee, Park, & Moon, 2010; Watts & Dodds, 2007). A tangible example of this strategy is the identification of the super-spreaders of the coronavirus to be vaccinated first. The spreaders can be considered as the first top-k ranked nodes of the graph based on various criteria such as nodes' local information degree centrality. H-index assigns importance values based on the second-order proximity, LocalRank considers 4th order degrees of the neighborhood, ClusterRank takes a combination of degree centrality and clustering coefficient into consideration. Closeness centrality, Betweenness centrality and Katz centrality suffer from high computational costs due to their shortest-path computations. PageRank as a random-walk-based method, has been also used as ranking criterion. In some studies, information entropy has been used as a criterion of node importance (Chen, Xiao, Zeng, & Zhang, 2014; Guo et al., 2020; Molaei et al., 2020; Qiao, Shan, Yu, & Liu, 2018). Some greedy algorithms such as Chen, Wang, and Yang (2009a) and Kempe, Kleinberg, and Tardos (2003) that suffer from high computational cost, also have been proposed. Deep-learning-based methods have also been proposed for solving cascade prediction problem. Cao, Shen, Cen, Ouyang, and Cheng (2017), Li, Ma, Guo and Mei (2017) design Recurrent Neural Networks for this purpose. Our approach falls into a new category defined as hybrid since it employs deep neural networks and an influence-based strategy jointly. This approach not only benefits from the power of neural networks, such as the capacity to capture complex structures of the data and higher performance, but it also overcomes the drawbacks of prior methods as we stated in the previous section. Table 1 introduces some more research conducted on this topic.

3. Proposed method

In this section, we first formulate the problem and then demonstrate the general framework and components of the proposed technique.

3.1. Problem definition

Input graph G contains N nodes $v_i \in \mathcal{V}, \{i = 1 \dots N\}$ indicating graph entities and $(v_i, v_j) \in \mathcal{E}$ are the edges modeling the connections between these entities. $\mathbf{A} \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix which can be either binary or weighted. Our objective would be to obtain the influential nodes set $E = \{e_1, e_2, \dots, e_m\} \subseteq \mathcal{V}$ from which originating a cascade of information results in a notable outbreak on the network.

3.2. Overall framework

Our goal is to identify the most influential nodes of the graph in such a way that if the diffusion of a piece of information starts from these nodes, we can make sure a great percentage of the nodes will be receiving it. GCNFusion uses a node classification learner to evaluate the node influence values. We design an augmented neural network containing a Graph Convolutional Network as the classifier. The influential nodes are selected in such a way that the classifier retains its predictive power. The rest of this section contains a detailed explanation of GCNFusion, our spreading model (SIR), and the complexity analysis of the proposed method. The implementation phase was carried out via Python programming language and powerful packages built on it such as PyTorch (Paszke et al., 2019). Fig. 1 illustrates the flow chart of the proposed method, which will be examined in the next section more precisely.

3.3. Method

As previously indicated, a graph convolutional neural network can be thought of as the generalization of convolutional neural networks on non-euclidean graph-structured data. These networks have been used in a wide range of graph learning applications on top of which, node classification has been addressed in Kipf and Welling (2017), thus we use the concept of GCNs as a node classifier to simulate wrapper methods in feature selection algorithms. We aim to find a set of most influential nodes by minimizing error on the entire graph. (For the sake of simplicity, in case of unfamiliarity with neural networks, GCN (which is described in the following) can be seen as a black box, responsible for node classification).

We consider a combinatorial multi-layer neural network with the following propagation rule motivated by the first-order approximation of localized spectral filters on graphs (Defferrard, Bresson, & Vandergheynst, 2016):

$$H^{(l+1)} = \sigma(\tilde{D}^{\frac{-1}{2}} \tilde{A} \tilde{D}^{\frac{-1}{2}} H^{(l)} W^{(l)}), \tag{1}$$

where $\tilde{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix included self-loops, $\tilde{D} \in \mathbb{R}^{N \times N}$ is the diagonal degree matrix whose non-zero elements are the degree of the relative node, and σ denotes a ReLU activation function. $H^{(l)}$ is the matrix containing integrated node representation vectors in the lth layer. We want the output of the network to be a $N \times k$ matrix (k being the number of node classes) on which applying a row-wise $softmax(.)$ activation function simulates a node classifier. $W^{(l)}$ is the learnable weight matrix parameter that is trained using gradient descent.

We define the input of the network as (2), which can be interpreted as an extra layer inspired by Borisov et al. (2019), added to the network:

$$H^{(0)} = \tilde{A} \odot \sigma(W^{co}), \tag{2}$$

where $W^{co} \in \mathbb{R}^{N}$ is a learnable vector parameter that is practically responsible for canceling out the less important nodes' effect on the classification task. In other words, W^{co} contains node influence ranking values. σ denotes sigmoid activation function. Putting (1) and (2)

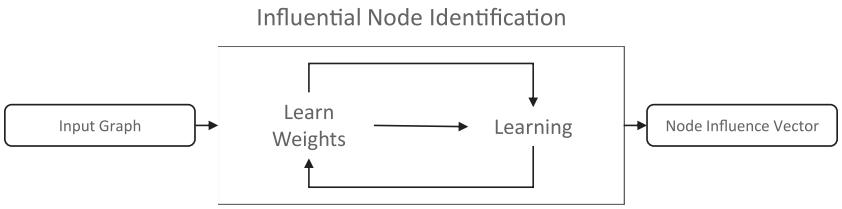


Fig. 1. Flow chart of GCNFusion.

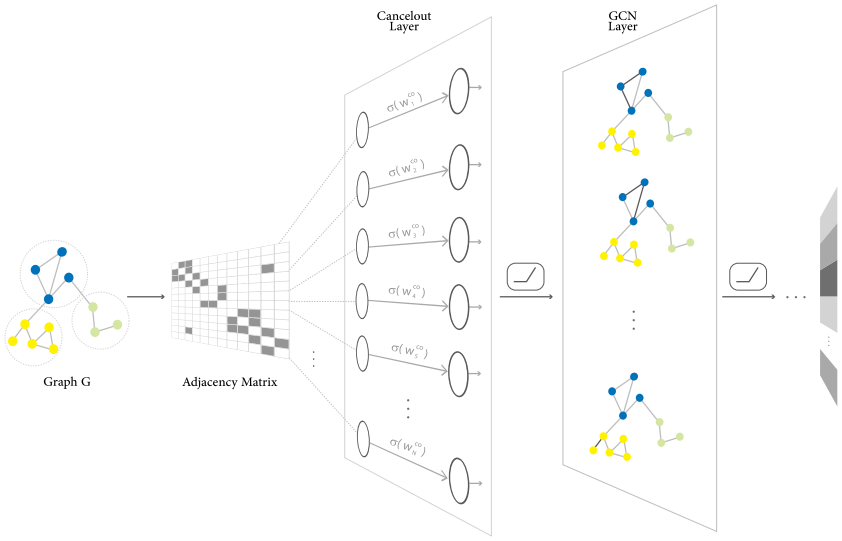


Fig. 2. A high-level overview of the influential node identification by GCNFusion. To simulate wrapper methods in feature selection, The input graph's adjacency matrix is passed through a neural network consisting of two contiguous sections corresponding to the influential node identification (Cancelout Layer) and Node Classification (GCN Layer). Different shades of gray color in the output indicate different class labels.

together, the propagation rule of a three layer GCNFusion network takes the simple form:

$$Z = softmax\big[\tilde{A}ReLU\big[\tilde{A}H^{(0)}W^{(1)}\big]W^{(2)}\big], \tag{3}$$

The overall model is schematically illustrated in Fig. 2. The intuition behind the cancelout layer is as follows: if for instance, w_i^{co} is a negative number, then $\sigma(w_i^{co})$ is equal to 0, then the corresponding node in the graph does not affect the output of the network. In other words, the first layer is simulating an influential node identification process. So once the network is trained by optimizing a cross-entropy loss function using gradient descent and training the parameters, the cancelout weights are set in a way that the nodes are well-classified and the m influential nodes are the ones corresponding to the m highest values of W^{co}. Fig. 3 illustrates a toy example of the intuition behind the cancelout layer, in which the darker colors correspond to higher importance values. As can be seen in the figure, node 4, which has a more important value, also has a higher degree. It can be interpreted that the GCNFusion has the ability to include the degree of nodes in calculating their importance. Algorithm 1 shows the overall steps of GCNFusion.

$$\mathcal{L} = -\sum_{f=1}^{F} Y_f \ln Z_f + \lambda_1 var\left(\frac{W^{co}}{N}\right) + \lambda_2 \left\| \frac{W^{co}}{N} \right\|_1, \tag{4}$$

Eq. (4) shows the loss function of GCNFusion, with F being the number of clusters, Y being the set of labels, and Z being the output of the neural network. Since real-world data is commonly unlabeled, data labels can be constructed using a clustering approach on node representation vectors that are produced by the GCN. In this regard, in our experiments, we perform the Kmeans algorithm to generate Y.

The first term is associated with the node classification task performed by GCN. The regularization term $var(\frac{W^{co}}{N})$ is added to preserve the diversity over the W^{co}'s elements, and the term $\|\frac{W^{co}}{N}\|_1$ is used to keep the W^{co} sparse and its elements small as possible.

Algorithm 1 GCNFusion

Input: adjacency matrix **A**, depth l, weight matrices $W^{(1)}, W^{(2)}, W^{(co)}$
Output: node influence values $w^{(co)}$,

1: Set $H^{(0)} = \tilde{A} \odot \sigma(W^{co})$, as shown in 2
2: Get predictions by forward propagation rule: $H^{(l+1)} = \sigma(\tilde{D}^{\frac{-1}{2}}\tilde{A}\tilde{D}^{\frac{-1}{2}}H^{(l)}W^{(l)})$,
3: Back propagate error values and update parameters by SGD,
4: Return node influence values $w^{(co)}$.

3.4. Spreading model

In the end, for the evaluation purpose, having the influential nodes detected, we defuse the information using the SIR model which has been widely used in the literature to analyze the spreading process and influential nodes identification methods (e.g. Guo et al. (2020) and Zhang et al. (2016)). This model has the highest capability to capture the dynamics of information diffusion compared to other spreading models. The SIR model divides the population into three categories: Susceptible (nodes that are exposed to the information), Infected (nodes that have received the information), and Removed or Recovered (nodes that for whatever reason will not participate in disseminating information, e.g. they forget or refuse to spread it). At each time step, each infected node randomly infects one of its direct neighbors with probability μ. Meanwhile, an infected node gets recovered with the probability of β and will not be infected again. We set $\lambda = \frac{\mu}{\beta}$ as

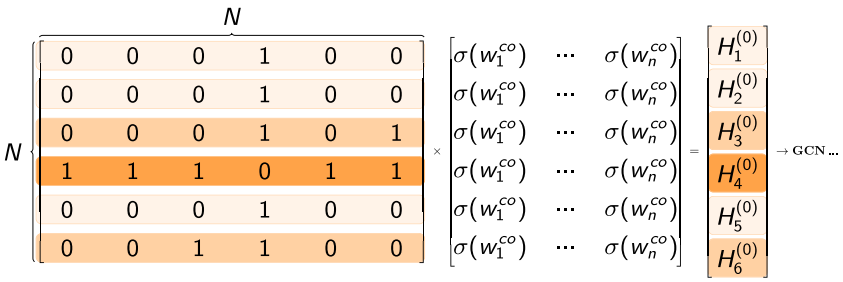


Fig. 3. A toy example of how the CancelOut layer determines the nodes' importance.

the infected rate, of which the crucial effect on the spreading ability of source spreaders is usually investigated in the literature. In our experiments, according to Castellano and Pastor-Satorras (2010) we set $\mu = 1.5\mu_c$, where $\mu = \frac{\langle k \rangle}{\langle k \rangle^2 - \langle k \rangle}$. Since $\mu < \mu_c$, leads to a poor propagation and on the other hand, for $\mu \gg \mu_c$, the information reaches almost the whole network and does not make sense in evaluating a spreading process either.

3.5. Computational complexity analysis

The computational complexity of GCNFusion is restricted to the node classifier and the complexity of the CancelOut layer is $\mathcal{O}(n)$ and can be ignored. The node classifier module (GCN), we leveraged in our proposed method does not impose much computational load on the model. For instance, our one-layer GCN-based model GCNFusion has the overall time complexity $\mathcal{O}(md + nd^2)$ where d is the dimension of a node adjacency vector. By making several approximations and simplifications this complexity can be reduced to $\mathcal{O}(m)$. The space complexity is similarly equal to $\mathcal{O}(nd + d^2)$.

4. Experiments

As previously indicated, using the SIR model, we diffuse the information through the network. We set the influential nodes generated by GCNFusion as the initial spreaders of the SIR model and evaluate several measurements to evaluate our performance on determining the quality of the influential nodes set. We will go through the performance metrics, datasets we utilized, and the final results in this section.

4.1. Datasets

To demonstrate the efficiency of our proposed method, we conducted experiments on five datasets in different research areas ranging from biology to social networks: Cora (Sen et al., 2008) is a citation network containing edges that represent citation relationship between authors, Email (Kunegis, 2013) is a communication network containing edges that represent individual messages between users, Hamster (Rossi & Ahmed, 2015) contains edges representing the hamsterster.com's users family and friendship relationships, Router (Spring, Mahajan, & Wetherall, 2002) is a router level internet topology network, and CEnew (Jeong, Tombor, Albert, Oltvai, & Barabási, 2000) is the metabolic network of the C.elegans worm. Table 2 contains the statistics of each network.

4.2. Performance metrics

The diffusion process can be evaluated by the following measurements which are commonly used in the Literature (Zhang et al., 2016):

- **Infected scale** $F(t)$:

$$F(t) = \frac{n_{I(t)} + n_{R(t)}}{n}, \tag{5}$$

where $n_{I(t)}$ and $n_{R(t)}$ are respectively the number of infected and removed nodes at time t. $F(t)$ indicates the diffusion speed.

Table 2
Dataset statistics. $\langle k \rangle$ and $\langle c \rangle$ are respectively average degree and average clustering coefficient.

Dataset	Type	#Nodes	#Edges	$\langle k \rangle$	$\langle c \rangle$
Cora	Citation network	2708	5429	4	0.24
Email	Communication network	1133	5451	9.6	0.22
Hamster	Social network	2426	16 631	13.7	0.53
Router	Internet network	5022	6258	2.4	0.01
CEnew	Metabolic network	453	2025	8.9	0.39

- **Final affected scale** $F_{(t_c)}$:

$$F_{(t_c)} = \frac{n_{R(t_c)}}{n}, \tag{6}$$

where t_c is the time SIR has reached its stable state, and $n_{R(t_c)}$ is the number of removed nodes at t_c. $F_{(t_c)}$ indicates the final diffusion ability of the initial spreaders.

- **Average shortest path length** L_s:

$$L_s = \frac{1}{|S|(|S|-1)} \sum_{\substack{u,v \in S \\ u \neq v}} L_{u,v}, \tag{7}$$

where S denotes the initial spreaders and $L_{u,v}$ is the average shortest path between nodes u, v. Larger values of L_s indicate more scattered and thus more diffusive spreaders.

4.3. Results

In this section, we intend to examine the impact of various factors on the diffusion phenomenon including the set of initial nodes, the number of initial nodes, the infected rate (λ), and the nature and structure of the network. For the evaluation purpose, we select diffusion methods Adaptive Degree (Chen, Wang, & Yang, 2009b), VoteRank (Zhang et al., 2016), k-shell (Kitsak et al., 2010), EnRenew (Guo et al., 2020), Entropy Rank Method (Molaei et al., 2020) and DIL (Liu, Xiong, Shi, Shi, & Wang, 2016) to compare with GCNFusion.

In Fig. 4 the final affected scales $F_{(t_c)}$ on five datasets are compared. For each dataset, in each iteration, we started the information diffusion with an increasing portion of the influential nodes as the initial spreaders in the SIR model with $\lambda = 1.5$. The horizontal axis P is the ratio of the initial spreaders in the whole network. Larger values of $F_{(t_c)}$ proves that when the spreading process reaches its stable state, the initial spreaders have been more capable of diffusing. As illustrated in this figure, the proposed method, in comparison with other methods, has a higher final affected scale for nearly all values of P, on all datasets. Characteristically, the larger portion of initial spreaders on all baseline methods leads to a broader diffusion, but the fact that the blue line, corresponding to DGNFusion, works as an upper bound indicates its notable performance on all five datasets.

Fig. 5 shows the diffusion speed or infected scale $F(t)$ varying with time. The spreading model is SIR with $\lambda = 1.5$. It can be seen that GCNFusion, not only outperforms baseline methods but also is faster, which means our influential nodes can spread the information through the network much faster. Naturally, for all benchmark methods, over

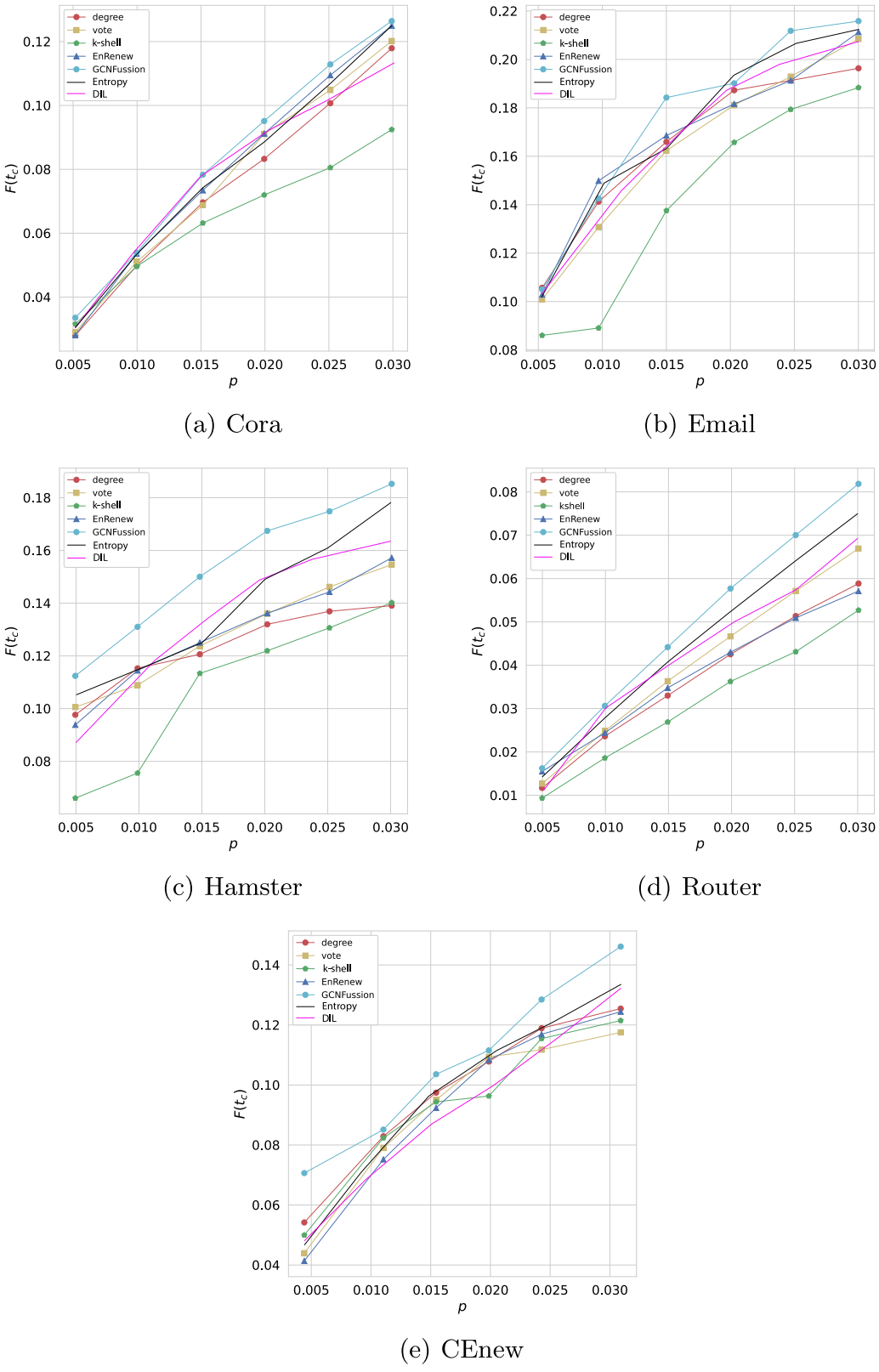
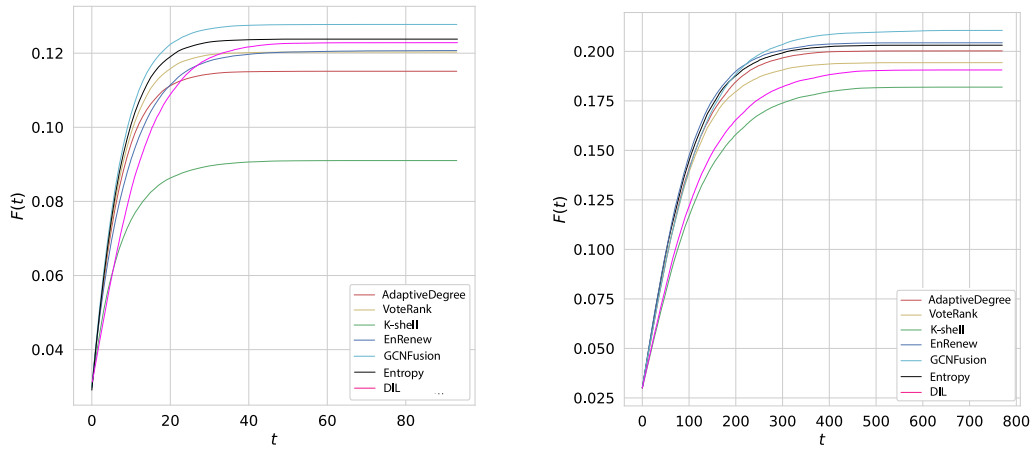


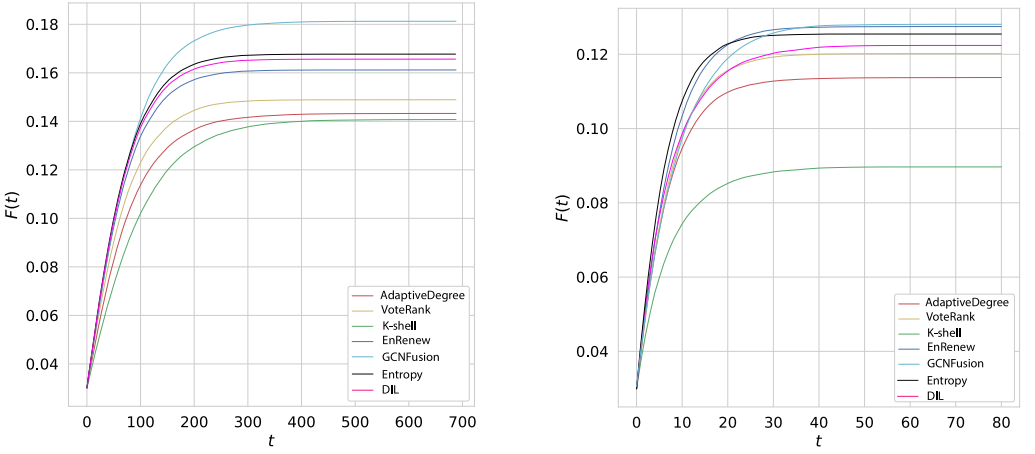
Fig. 4. Comparison of the average final affected scale ($F_{(t_c)}$) of 10 runs, with respect to the ratio of the initial spreaders P.

time, the number of infected and consequently recovered nodes are increasing, however, the position of the blue line, relative to other lines, indicates its notable diffusion speed on all five datasets.

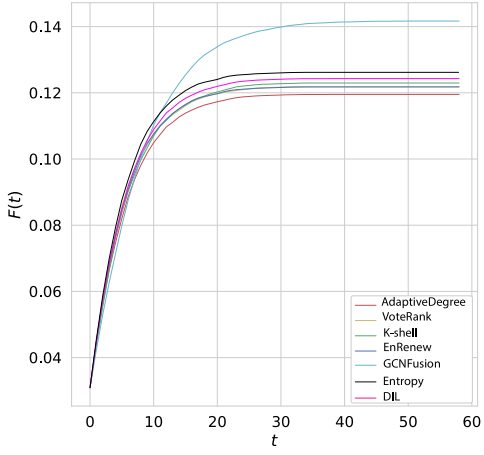
We also conducted some experiments in order to investigate the effect of λ on the diffusion process. The results can be seen in Fig. 6. We examined the final affected scale in the range $1 \leq \lambda \leq 2$ to gain



(a) Cora (b) Email



(c) Hamster (d) Router



(e) CEnew

Fig. 5. The infected scale ($F_{(t)}$) of 10 runs, with respect to time t.

a meaningful insight of the methods since smaller values, regardless of initial nodes, are not effective enough in the spreading process and larger values accelerate the diffusion of information too much. As shown in the figure, on all datasets, almost for all λ values, GCNFusion surpasses all other methods. In this experiment, the superiority of GCNFusion compared to other methods suggests its robustness in terms of infected rate.

As mentioned before, larger values of L_s indicate more scattered initial spreaders and thus a more diffusive set of spreaders, which can be seen as a metric for measuring the «goodness » of an algorithm. The

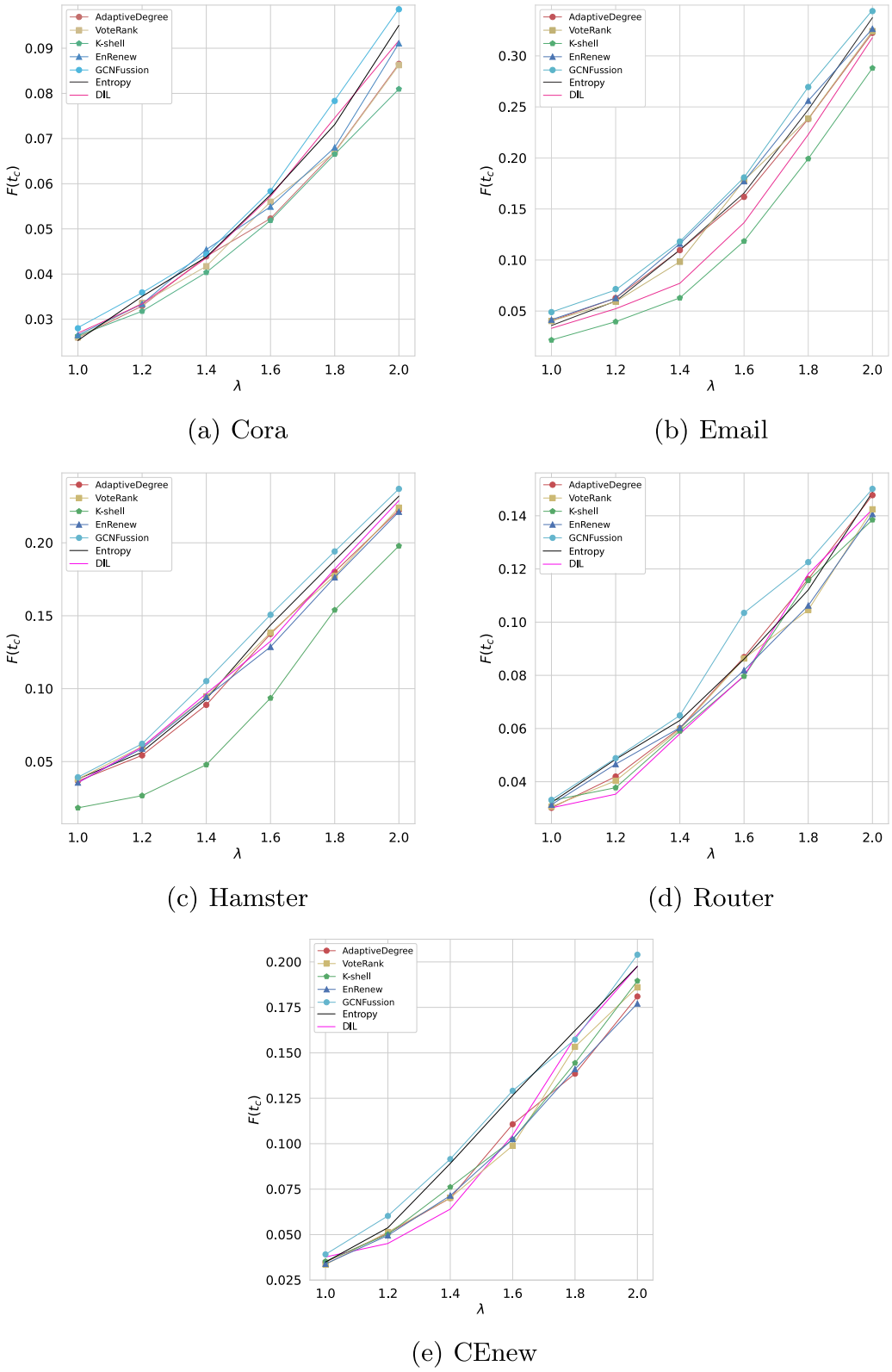


Fig. 6. The effect of parameter infected rate (λ) on final affected scale $F_{(t_c)}$.

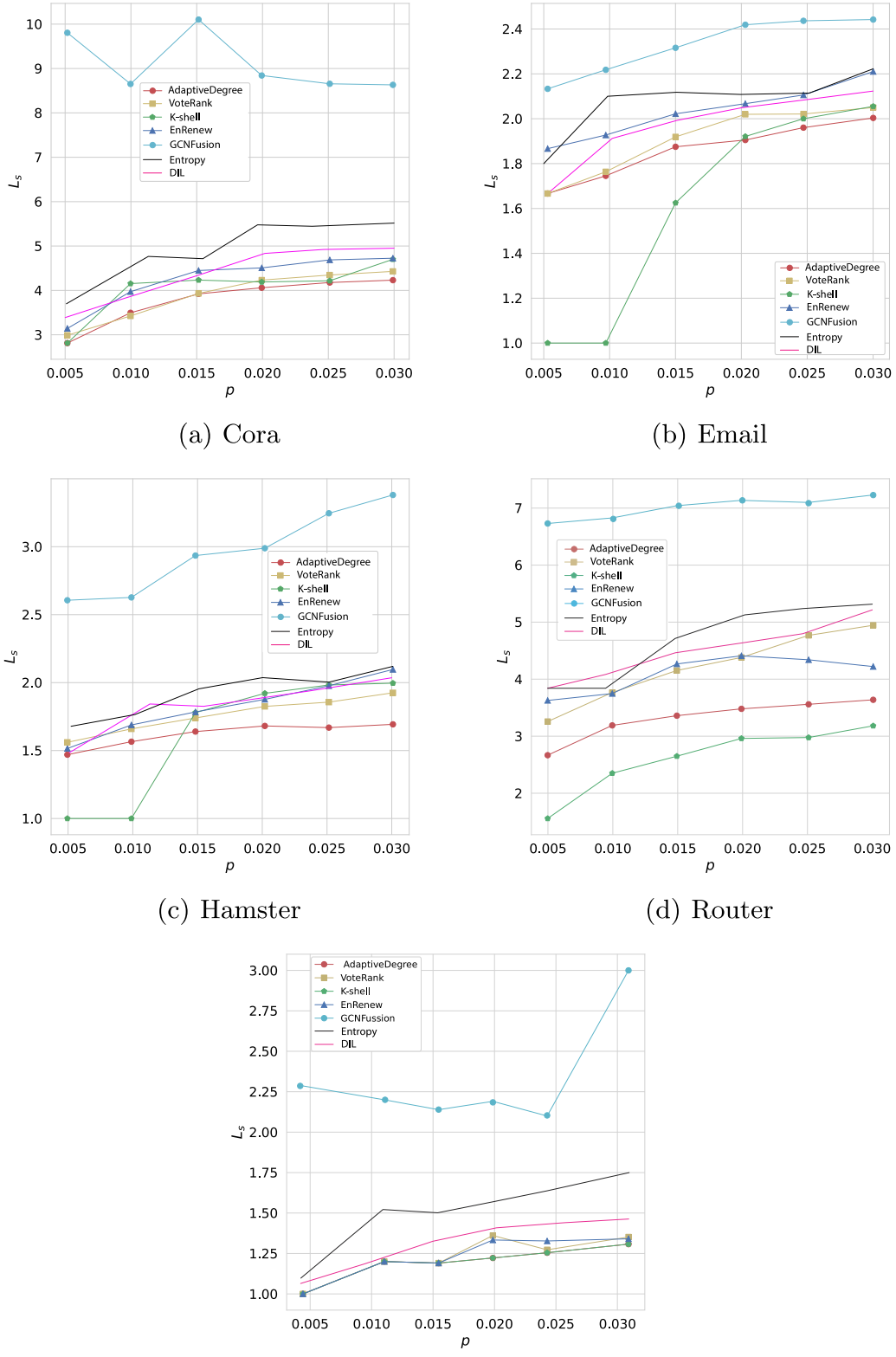


Fig. 7. Average shortest path length (L_s) of different portions of initial spreaders. The spreading model is SIR with $\lambda = 1.5$.

results of the investigation on this property, are shown in Fig. 7. On all datasets, GCNFusion performs as an upper bound, far better than the other methods.

5. Conclusions and future works

5.1. Conclusions

In this paper, we introduced a novel approach for solving the problem of influence-based information diffusion. We borrowed intuition from three cutting-edge research areas: graph representation learning, feature selection, and information diffusion. To each node of the input graph, an influence value is assigned using a graph neural network augmented by a variant of a feature selection layer. These values can be leveraged to control the spreading process on networks such as infectious diseases, rumors, etc. To evaluate the algorithm we adopted the SIR model and proved that the proposed method significantly outperforms benchmark approaches on information diffusion. In summary, the results of the experiments on five datasets of Cora, Email, Hamster, Router, and CEnew, indicated that for different portions of initial spreaders, our proposed method is respectively 3%, 5%, 5%, 2%, 3% more superior in information diffusion compared to the benchmark methods. GCNFusion is easily implemented and computationally efficient and can be used for either labeled or unlabeled graphs of any kind, topology, and size.

5.2. Limitations and future works

Our proposed method of disseminating information has shown satisfactory results on homogeneous networks. However, it is not yet applicable to heterogeneous networks. This will be achievable by making some modifications. In the continuation of this research, we intend to address this issue. On the other hand, we anticipate that having a macroscopic view of a network, in addition to a microscopic one can aid the diffusion process. Therefore, in future work, we intend to provide a method that, while preserving the local information of the input graph, also takes the global information into consideration. Although GCN networks have proved their strength, they still can have a deficiency. For more dense input graphs, these networks may show some disruption. However, we do not see this as a very significant weakness because of the sparse nature of most real-world networks.

CRediT authorship contribution statement

Bahareh Fatemi: Method, Analysis, Writing – review & editing. **Soheila Molaei:** Method, Analysis, Writing – review & editing. **Shirui Pan:** Method, Analysis, Writing – review & editing. **Samira Abbasgholizadeh Rahimi:** Method, Analysis, Writing – review & editing.

Declaration of competing interest

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

References

Belkin, M., & Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Nips, Vol. 14* (pp. 585–591).

Borisov, V., Haug, J., & Kasneci, G. (2019). Cancelout: A layer for feature selection in deep neural networks. In *International conference on artificial neural networks* (pp. 72–83). Springer.

Cao, Z., Han, K., & Zhu, J. (2021). Information diffusion prediction via dynamic graph neural networks. In *2021 IEEE 24th international conference on computer supported cooperative work in design (CSCWD)* (pp. 1099–1104). IEEE.

Cao, S., Lu, W., & Xu, Q. (2015). Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management* (pp. 891–900).

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 2017 ACM on conference on information and knowledge management* (pp. 1149–1158).

Castellano, C., & Pastor-Satorras, R. (2010). Thresholds for epidemic spreading in networks. *Physical Review Letters*, *105*, Article 218701. http://dx.doi.org/10.1103/PhysRevLett.105.218701, Retrieved from https://link.aps.org/doi/10.1103/PhysRevLett.105.218701.

Chang, C.-H., Rampasek, L., & Goldenberg, A. (2018). Dropout feature ranking for deep learning models. Unpublished results.

Chen, W., Jiang, M., Zhang, W.-G., & Chen, Z. (2021). A novel graph convolutional feature based convolutional neural network for stock trend prediction. *Information Sciences*, *556*, 67–94.

Chen, W., Wang, Y., & Yang, S. (2009a). Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 199–208).

Chen, W., Wang, Y., & Yang, S. (2009b). Efficient influence maximization in social networks. In *KDD '09*, *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 199–208). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/1557019.1557047, Retrieved from https://doi.org/10.1145/1557019.1557047.

Chen, D.-B., Xiao, R., Zeng, A., & Zhang, Y.-C. (2014). Path diversity improves the identification of influential spreaders. *EPL (Europhysics Letters)*, *104*(6), 68006.

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

Chenquan, G., Qingdong, F., Qingyi, Z., Zufan, Z., Zhang, Y., & Xiang, Y. (2020). Analysis of computer virus propagation behaviors over complex networks: a case study of Oregon routing network. *Nonlinear Dynamics*, *100*(2), 1725–1740.

Chinazzi, M., Davis, J. T., Ajelli, M., Gioannini, C., Litvinova, M., Merler, S., y Piontti, A. P., Mu, K., Rossi, L., Sun, K., Viboud, C., Xiong, X., Yu, H., Halloran, M. E., Longini, I. M., & Vespignani, A. (2020). The effect of travel restrictions on the spread of the 2019 novel coronavirus (COVID-19) outbreak. *Science*, *368*(6489), 395–400. http://dx.doi.org/10.1126/science.aba9757.

Cinelli, M., Cresci, S., Galeazzi, A., Quattrociocchi, W., & Tesconi, M. (2020). The limited reach of fake news on Twitter during 2019 European elections. *PLoS One*, *15*(6), Article e0234689.

Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems*, *29*, 3844–3852.

Fatemi, B., Molaei, S., Zare, H., & Pan, S. (2021). Manifold approximation and projection by maximizing graph information. In K. Karlapalem, H. Cheng, N. Ramakrishnan, R. K. Agrawal, P. K. Reddy, J. Srivastava, & T. Chakraborty (Eds.), *Advances in knowledge discovery and data mining* (pp. 128–140). Cham: Springer International Publishing.

Goldenberg, J., Libai, B., & Muller, E. (2001). Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, *12*(3), 211–223.

Gomez-Rodriguez, M., Leskovec, J., & Schölkopf, B. (2013). Modeling information propagation with survival theory. In S. Dasgupta, & D. McAllester (Eds.), *Proceedings of machine learning research*: *vol. 28*, *Proceedings of the 30th international conference on machine learning* (pp. 666–674). Atlanta, Georgia, USA: PMLR, Retrieved from https://proceedings.mlr.press/v28/gomez-rodriguez13.html.

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

Guo, C., Yang, L., Chen, X., Chen, D., Gao, H., & Ma, J. (2020). Influential nodes identification in complex networks via information entropy. *Entropy*, *22*(2), 242.

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin*, *40*(3), 52–74, Retrieved from http://sites.computer.org/debull/A17sept/p52.pdf.

Hethcote, H. W. (2000). The mathematics of infectious diseases. *SIAM Review*, *42*(4), 599–653.

Hughes, A. L., & Palen, L. (2009). Twitter adoption and use in mass convergence and emergency events. *International Journal of Emergency Management*, *6*(3–4), 248–260.

Jalili, M., & Perc, M. (2017). Information cascades in complex networks. *Journal of Complex Networks*, *5*(5), 665–693.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., & Barabási, A.-L. (2000). The large-scale organization of metabolic networks. *Nature*, *407*(6804), 651–654.

Kasneci, G., & Gottron, T. (2016). Licon: A linear weighting scheme for the contribution ofinput variables in deep artificial neural networks. In *Proceedings of the 25th ACM international on conference on information and knowledge management* (pp. 45–54).

Kempe, D., Kleinberg, J., & Tardos, E. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 137–146).

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *ICLR '17*, *Proceedings of the 5th international conference on learning representations*. Retrieved from https://openreview.net/forum?id=SJU4ayYgl.

Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., & Stanley, H. E. (2010). Identification of influential spreaders in complex networks. *Nature Physics*, *6*(11), 888–893.

Kumar, A., Vashishtha, G., Gandhi, C., Zhou, Y., Glowacz, A., & Xiang, J. (2021). Novel convolutional neural network (NCNN) for the diagnosis of bearing defects in rotary machinery. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–10.

Kunegis, J. (2013). Konect: the koblenz network collection. In *Proceedings of the 22nd international conference on world wide web* (pp. 1343–1350).

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is Twitter, a social network or a news media? In *Proceedings of the 19th international conference on world wide web* (pp. 591–600).

Lagnier, C., Denoyer, L., Gaussier, E., & Gallinari, P. (2013). Predicting information diffusion in social networks using content and user's profiles. In *European conference on information retrieval* (pp. 74–85). Springer.

Leskovec, J., McGlohon, M., Faloutsos, C., Glance, N., & Hurst, M. (2007). Patterns of cascading behavior in large blog graphs. In *Proceedings of the 2007 SIAM international conference on data mining* (pp. 551–556). SIAM.

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* (pp. 577–586).

Li, M., Wang, X., Gao, K., & Zhang, S. (2017). A survey on information diffusion in online social networks: Models and methods. *Information*, *8*(4), 118.

Libai, B., Muller, E., & Peres, R. (2013). Decomposing the value of word-of-mouth seeding programs: Acceleration versus expansion. *Journal of Marketing Research*, *50*(2), 161–176.

Liu, J., Xiong, Q., Shi, W., Shi, X., & Wang, K. (2016). Evaluating the importance of nodes in complex networks. *Physica A: Statistical Mechanics and its Applications*, *452*, 209–219. http://dx.doi.org/10.1016/j.physa.2016.02.049, Retrieved from https://www.sciencedirect.com/science/article/pii/S0378437116002156.

Moghanian, S., Saravi, F. B., Javidi, G., & Sheybani, E. O. (2020). GOAMLP: Network intrusion detection with multilayer perceptron and grasshopper optimization algorithm. *IEEE Access*, *8*, 215202–215213.

Molaei, S., Farahbakhsh, R., Salehi, M., & Crespi, N. (2020). Identifying influential nodes in heterogeneous networks. *Expert Systems with Applications*, *160*, Article 113580. http://dx.doi.org/10.1016/j.eswa.2020.113580, Retrieved from https://www.sciencedirect.com/science/article/pii/S0957417420304048.

Ohsaka, N., Sonobe, T., Fujita, S., & Kawarabayashi, K.-i. (2017). Coarsening massive influence networks for scalable diffusion analysis. In *SIGMOD '17*, *Proceedings of the 2017 ACM international conference on management of data* (pp. 635–650). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3035918.3064045, Retrieved from https://doi.org/10.1145/3035918.3064045.

Ou, M., Cui, P., Pei, J., Zhang, Z., & Zhu, W. (2016). Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1105–1114).

Padda, K. (2020). Fake news on Twitter in 2016 US presidential election: a quantitative approach. *The Journal of Intelligence, Conflict, and Warfare*, *3*(2), 1–24.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., & Antiga, L. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*, 8026–8037.

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

Qi, T., Li, G., Chen, L., & Xue, Y. (2021). ADGCN: An asynchronous dilation graph convolutional network for traffic flow prediction. *IEEE Internet of Things Journal*.

Qiao, T., Shan, W., Yu, G., & Liu, C. (2018). A novel entropy-based centrality approach for identifying vital nodes in weighted networks. *Entropy*, *20*(4), 261.

Rehman, A., Jiang, A., Rehman, A., Paul, A., din, S., & Sadiq, M. T. (2020). Identification and role of opinion leaders in information diffusion for online discussion network. *Journal of Ambient Intelligence and Humanized Computing*, http://dx.doi.org/10.1007/s12652-019-01623-5.

Rossi, R., & Ahmed, N. (2015). The network data repository with interactive graph analytics and visualization. In *Twenty-ninth AAAI conference on artificial intelligence*.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., & Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine*, *29*(3), 93.

Singh, S. S., Mishra, S., Kumar, A., & Biswas, B. (2020). CLP-ID: Community-based link prediction using information diffusion. *Information Sciences*, *514*, 402–433.

Spring, N., Mahajan, R., & Wetherall, D. (2002). Measuring ISP topologies with rocketfuel. *ACM SIGCOMM Computer Communication Review*, *32*(4), 133–145.

Srinidhi, C. L., Ciga, O., & Martel, A. L. (2021). Deep neural network models for computational histopathology: A survey. *Medical Image Analysis*, *67*, Article 101813.

Velickovic, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., & Hjelm, R. D. (2019). Deep graph infomax. *ICLR (Poster)*, *2*(3), 4.

Watts, D. J., & Dodds, P. S. (2007). Influentials, networks, and public opinion formation. *Journal of Consumer Research*, *34*(4), 441–458.

Yang, J., & Leskovec, J. (2010). Modeling information diffusion in implicit networks. In *2010 IEEE international conference on data mining* (pp. 599–608). IEEE.

Yao, P., Wu, H., Gao, B., Tang, J., Zhang, Q., Zhang, W., Yang, J. J., & Qian, H. (2020). Fully hardware-implemented memristor convolutional neural network. *Nature*, *577*(7792), 641–646.

Ye, Y., Zhang, Q., Ruan, Z., Cao, Z., Xuan, Q., & Zeng, D. D. (2020). Effect of heterogeneous risk perception on information diffusion, behavior change, and disease transmission. *Physical Review E*, *102*(4), Article 042314.

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, *1*(2), 56–70.

Zhang, J.-X., Chen, D.-B., Dong, Q., & Zhao, Z.-D. (2016). Identifying a set of influential spreaders in complex networks. *Scientific Reports*, *6*, 27823.

Zhang, H., Jiang, L., & Li, C. (2021). Cs-resnet: cost-sensitive residual convolutional neural network for PCB cosmetic defect detection. *Expert Systems with Applications*, *185*, Article 115673.

Zhang, L., Luo, M., & Boncella, R. J. (2020). Product information diffusion in a social network. *Electronic Commerce Research*, *20*(1), 3–19.