TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 6, December 2020, pp. 3331~3338
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i6.16300

New prediction method for data spreading in social networks based on machine learning algorithm

Maytham N. Meqdad[1], Rawya Al-Akam[2], Seifedine Kadry[3]
[1]Al-Mustaqbal University College, Iraq
[2]Koblenz-Landau University, German
[3]Beirut Arab University, Lebanon

Article Info

Article history:

Received Apr 9, 2020
Revised Jun 7, 2020
Accepted Jul 16, 2020

Keywords:

Data spreading
Machine learning
Prediction
Social network

ABSTRACT

Information diffusion prediction is the study of the path of dissemination of news, information, or topics in a structured data such as a graph. Research in this area is focused on two goals, tracing the information diffusion path and finding the members that determine future the next path. The major problem of traditional approaches in this area is the use of simple probabilistic methods rather than intelligent methods. Recent years have seen growing interest in the use of machine learning algorithms in this field. Recently, deep learning, which is a branch of machine learning, has been increasingly used in the field of information diffusion prediction. This paper presents a machine learning method based on the graph neural network algorithm, which involves the selection of inactive vertices for activation based on the neighboring vertices that are active in a given scientific topic. Basically, in this method, information diffusion paths are predicted through the activation of inactive vertices by active vertices. The method is tested on three scientific bibliography datasets: The Digital Bibliography and Library Project (DBLP), Pubmed, and Cora. The method attempts to answer the question that who will be the publisher of the next article in a specific field of science. The comparison of the proposed method with other methods shows 10% and 5% improved precision in DBLP and Pubmed datasets, respectively.

Corresponding Author:

Seifedine Kadry,
Department of Mathematics and Computer Science, Faculty of Science,
Beirut Arab University,
Beirut, Lebanon.
Email: s.kadry@bau.edu.lb

1. INTRODUCTION

Information diffusion in homogenous and heterogeneous networks is a dynamic process of keen interest to researchers. This concept refers to how information like news of events outbreaks etc. spread from a set of origin nodes to other nodes across the network [1-3]. Information diffusion has been studied in many fields ranging from health care [4-6] to social networks [7, 8]. One of the most important tasks of network-based systems is to understand, model, and predict rapidly developing events within the network. After discovering the structure of a network, it is possible to predict the patterns of events including their shape, size and development, which can be described as information diffusion [9]. Over the years, researchers have tried several methods to model information diffusion in homogeneous and heterogeneous networks [10-13].

Formally, a data network is represented by a graph G = (V, E) where V is the set of vertices and E is the set of edges. This graph is called homogeneous if the vertices and their edges are of the same type, and is

called heterogeneous otherwise. Homogeneous networks have been the subject of many studies with the focus being on semantic analysis, communicable disease control [14-17], and link prediction [18-20]. Recently, more attention has been paid to heterogeneous networks, as they could provide a more realistic representation of real-world phenomena [21, 22]. In a study by Watt [23], he investigated the role of threshold values and network structure in information diffusion in these networks. In [10], information diffusion in heterogeneous networks through hyperpaths was studied. This study proposed a method called MLTM-R for analyzing information diffusion power in different hyperpaths. In this method, predictions were made with the PathSim algorithm used to weight the links between each two nodes [24, 25].

Recent years have seen a growing interest in the use of deep learning in heterogeneous networks [26, 27]. In [28], a core deep learning (CDL) framework was used to solve the problem of heterogeneous visual versus near-infrared (VIS-NIR) image matching through topic diffusion in networks. Tang et al. [29] proposed a LINE algorithm for embedding learning that traverses all edge types and samples one edge at a time for each edge type. Chang et al [30] developed a deep architecture for information diffusion prediction through information encoding in heterogeneous networks. In [31], a new algorithm called Metapath2vec was presented for information encoding in heterogeneous networks, where concepts and patterns are mapped by the use of hyperpaths. The review of previous works reveals some strengths and weaknesses in the current approach to information diffusion in heterogeneous networks. The use of deep learning in the study of information diffusion processes such as topic diffusion and information cascades can help avoid the problems of more traditional methods. The major disadvantage of the previous works is that most topic diffusion methods use local similarity and encoding based on neighboring nodes. For large heterogeneous networks, it is time-consuming and difficult to perform local similarity calculations for each two corresponding nodes. As a result, there is a need for a more comprehensive yet less complex automatic method for measuring the similarity of nodes and finding diffusion paths in heterogeneous networks.

In this paper, the problem of predicting the path of information diffusion in a network is mapped to a deep learning problem. Since predicting the new users who will be in the path of information flow is a recognition process, this problem can be solved by machine learning algorithms. As noted in section X, recently, deep machine learning algorithms have been widely used in this field. Also, researchers have developed deep machine learning algorithms that can use graph data in the learning process. This paper presents a machine learning method based on graph neural networks, which involves selecting the inactive node to be activated based on its neighboring active nodes in each scientific topic. In other words, in this method, information diffusion paths are predicted through the activation of inactive nodes by active nodes. To evaluate the proposed method, it is tested on three heterogeneous scientific databases: The Digital Bibliography and Library Project (DBLP), Pubmed, and Cora. The method seeks to answer the question that who will be the publisher of the next article in a particular field of science. The comparison of the proposed method with other methods shows 10% and 5% improvement in precision in DBLP and Pubmed datasets, respectively.

In summary, the most important innovations of the present work are as follows:

- Presenting a deep learning model where the information of a heterogeneous network is encoded in the form of a deep learning graph, which can model the information diffusion path.
- Providing a feature extraction mechanism to find the degree of correlation of neighboring vertices in different graph hyperpaths.
- Testing the method on the heterogeneous datasets DBLP, Pubmed, and Cora, which have real-world applications, in order to demonstrate the applicability of the proposed method.

The remaining sections of the article are organized as follows. Section 2 describes the different components of the proposed method. Section 3 describes the testing procedure and analyzes the results. And Section 4 presents the conclusions.

2. PROPOSED METHOD

Information diffusion is a widely discussed dynamic network process with potential applications in various fields of science. This term refers to the spreading of information or similar concepts such as news, innovation, virus or malware a set of vertices to other vertices across the network. There is a rich body of literature on information diffusion in complex networks, where different models and their interactions with network topology have been analyzed [1]. The previous studies have been mostly focused on heterogeneous networks. An information network like G = (V, E) where V is the set of vertices and E is the set of edges is homogeneous if the edges and vertices are of the same type. Conversely, the networks with more than one type of node or edge are called heterogeneous [8-10]. For example, in DBLP, which is an important computer science bibliography database, the vertices could be authors, articles, and venues (journals/conferences) and edges could be the author-author relationship in the sense that they have worked in the same area, and attended the same conferences.

Here, we model information diffusion and specifically topic diffusion in heterogeneous information networks. To this end, we use a concept called meta-path. The meta-path p on the grid TG = (A, R), where A and R represent vertices and relationships, is defined as follows:

$$A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1} \quad (1)$$

Here, l is an index of the meta-path. The summation relationship between different types of vertices (A1 - Al+1) is given by:

$$R = R_1 \circ R_2 \circ \dots R_l \quad (2)$$

where o is the combination operator. In DBLP, for example, each author-author or author-conference-author relationship is considered a single meta-path.

Figure 1 shows an example of the diffusion of the topic of "data mining" in DBLP, where authors can be linked through different meta-paths. This paper provides a machine learning method based on graph neural networks in which an inactive node is activated by its active neighbors in a particular scientific topic. Given that the prediction of new users who will be in the path of information flow is a recognition process, this problem can be solved by machine learning algorithms.

The general framework of the method consists of two main phases: 1) designing a machine learning scheme (learning machine) for the prediction process, and 2) evaluating the accuracy of the designed scheme (machine) in predicting the flow of information in the dataset of interest. The first step involves training a learning machine, where the input is the data collected from the information network graph and the output is the tag "Yes" or "No", showing whether or not the node specified in the input will be selected as the next path of information diffusion. The purpose of this machine is to create a regression function for optimal mapping between input data and output tags. In the second phase, a test dataset, which is taken from the collected data, is used to test the designed machine. In the testing and accuracy evaluation phase, the classification process is done once randomly and another time with the designed machine. In the end, the quality of the vertices obtained from these two methods is compared.

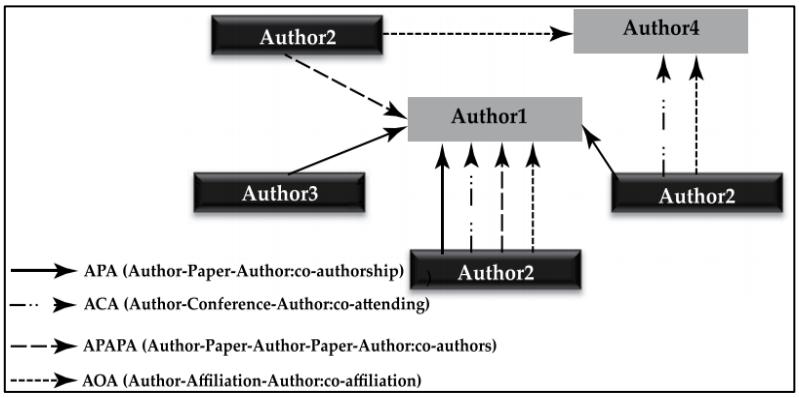

Figure 1. An example of a heterogeneous network [15]

2.1. Graph convolution network algorithm

This section describes the details of graph convolutional network (GCN) and the next section explains how it is used in the learning from the graph data of this study. Graph data can be broken down into two main elements: vertices vij and edges aij. A graph can be described by the following 3-tuple.

$$\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{A}) \quad (3)$$

where $\boldsymbol{V} \epsilon \mathbb{R}_{N \times f}$ is the vertex signal matrix describing N vertices each with f features, $\boldsymbol{A} \epsilon \mathbb{R}_{N \times N}$ is the adjacency matrix which encodes the edges information as described in section 2, and each element A is defined as follows;

$$\text{aij} = \begin{cases} \text{wij , if there is an edge between i and j} \\ 0, \text{otherwise} \end{cases} \quad (4)$$

An example graph and its vertex matrix V and adjacency matrix A are shown in Figure 2.

	P	Q	R	S
P	0	1	1	0
Q	1	1	0	0
R	1	0	0	1
S	0	0	1	0

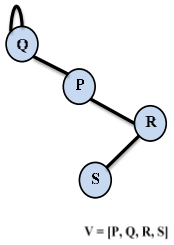

Figure 2. An example of a graph and its adjacency matrix

2.2. Graph convolution

Graph data can provide a brief representation of information in vertices and edges. To process and learn this information, one has to use a convolution filtering method to filter both vertex information and edge information. This is a spatial approach related to the graph convolution method, which uses the local neighborhood graph filtering strategy. The graph convolution operation is based on the polynomials of the adjacency matrix of the graph.

$$H = h0I + h1A^1 + h2A^2 + h3A^3 + \cdots + hkA^k \quad (5)$$

This filter is defined as the k^{th} degree polynomial of the adjacency matrix. The exponent of this polynomial encodes the number of steps from the vertex of interest, which is multiplied by the assumed filter factors. The scalar factor h_i determines how much each neighbor of a vertex contributes to the convolution operation. Therefore, the filter matrix is obtained as $H \in \mathbb{R}_{N \times N}$. The convolution of the vertices V with the filter H is defined as the following matrix multiplication, where $V_{out}, V_{in} \in \mathbb{R}_N$.

$$Vout = HVin \quad (6)$$

This model can be adjusted in three ways. The first way is to avoid A becoming exponentiated and simplify the adjacency polynomial in (2) into the linear form given in (6). The reason behind this approach is that, as shown by VGGNet, a cascade of filters can effectively estimate the receptive field of a large filter.

$$H \approx h0I + h1A \quad (7)$$

The next step is to create the adjacency tensor $\boldsymbol{\mathcal{A}}$. This tensor consists of multiple adjacency matrices A_e, which are the slices of this tensor, each encoding a specific edge feature. Therefore, the linear filter matrix in (6) is defined as a convex combination of adjacency matrices as given in (7). This equation can be simplified into (8).

$$H = h0I + h1A1 + h2A2 + \cdots + hLAL{-}1 \quad (8)$$

$$H \approx \sum_{e=0}^{L} h_e A_e \quad (9)$$

Multiple edge features are encoded by multiple adjacency matrices, each of which encodes a single feature. Also, as shown in Figure 3, the edges are subdivided into multiple matrices. Figure 3 shows the default linear GCN filter in an image application. A filter factor is isotropically applied to all vertices at a given distance. In this case, h_0 is applied to the vertex of the 0^{th} step and h_1 is applied to all adjacent vertices. If this figure is enclosed in another set of pixels, each pixel in that set will be multiplied by the filter factor h_2.

As shown in Figure 3, to create the adjacency tensor, the adjacency matrix can be subdivided into 9 adjacency matrices. Each of these adjacency matrices shows a different relative link (edge feature) to a given vertex. The next step is to apply a unique filter to each adjacency matrix to perform convolution followed by aggregation. This gives a direction to the GCN filter. This is equivalent to a 3×3 FIR filter in traditional GCNs.

All the above-described filters are for a single vertex feature. For the extension to multiple vertex features, each h_l must be in $\mathbb{R}_C$ so that H has a dimension of $\mathbb{R}_{N \times N \times C}$. Therefore, each vertex feature has a filter matrix H of size N×N. Therefore, in (9) can be rewritten as in (10), where H(c) is an N×N slice of H and h(c) is a scalar related to an input feature and a slice of A_l.

$$H^c \approx \sum_{e=0}^{L} h_e^c A_e \quad (10)$$

Based on (c), the vertex signal is described by in (11), where $V_{in}(c)$ is a column of V_{in} where c is the only vertex feature. A bias in the form of $b \in \mathbb{R}$ is also added to the formulation, results in $V_{in} \in \mathbb{R}_N$.

$$Vout = \sum_{c=1}^{C} H^c \mathrm{V}_{in}^c + b \quad (11)$$

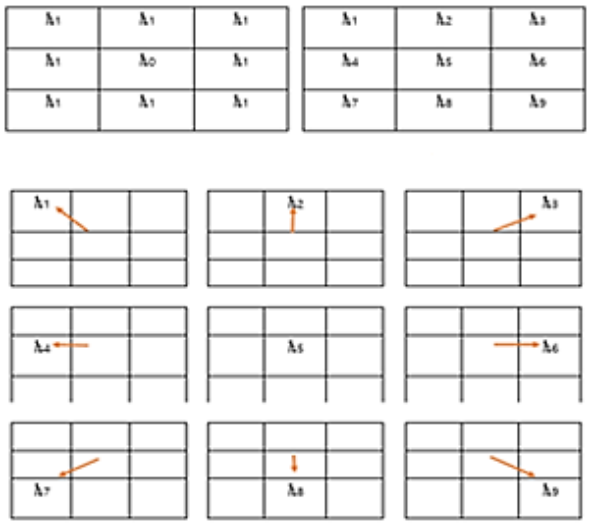

Figure 3. Types of filters that can be defined in the GCN algorithm

2.3. Architecture used for GCN

This architecture of CNN consists of three general steps. The first step is the preparation of the input graph, which could be DBLP for example, so that the active and inactive vertices are equalized and leveled. However, this step also has a separate input in the form of a tag. The output of this step is the graph Z_0, which will be fed to the next step.

The second step is to perform the convolution operation on this graph in several layers. In this step, a filter is applied to each vertex based on its adjacent edge. The input of the first convolution layer Z_1 is fed to the second convolution layer, which produces the output Z_2.

In the third step, the created features are linearized and given to a Softmax layer, which decides which node should be activated next. For this purpose, multiple linear vectors are created by concatenating all the graphs created in the previous steps and their different permutations from $Z_0$to Z_2. Each of these vectors is called a feature vector. The output of these operations is a linear vector that is fed to the fully connected neural network placed in the last layer. Finally, this network, which is known as Softmax, is used to decide on the next active node.

2.4. Implementation of classification algorithm

In the proposed algorithm, a learning machine for image classification in the dataset processed in the classification phase is used to construct a model that produces values that are as close as possible to the expected values. Several different approaches have been developed for this purpose. In this paper, classification algorithms are used to classify the user into different classes in the dataset. The image classification is done based on the feature vector extracted in the previous phase.

The proposed algorithm uses a modified version of the standard algorithm described above for classification. Since the output of the proposed method is tagged in two classes, the multi-class version of these algorithms is used. The features included in the feature matrix of each image form an n-dimensional vector, which belongs to one of two classes.

$$X = (x_1, x_2, \cdots, x_n) \quad (12)$$

$$C = (c_1, c_2) \quad (13)$$

$$Data = (X_1, X_2, \cdots, X_L),\ \mathrm{L} = \text{Number of Sample} \quad (14)$$

The algorithm presented in Figure 4 shows how the learning machine is built and tested. In the proposed algorithm, each machine is first trained using the extracted data. As can be seen, each learning machine is trained separately for each dataset extracted in the algorithm. This is done using the "generateML" function in line 5. Then, the folding algorithm is used to test each machine. Therefore, each machine is

repeatedly subdivided and trained and tested by different datasets. More details on the mechanism of the folding algorithm and measurement of the accuracy of the learning machine are provided in the next section.

```
Classification_Algorithm={'GCN'}
dataset={'DBLP',Pubmet','Cora'}
for each dataset Do
    for each fold Do
        for each feature Do
            (Xtrain, Ytrain, Xtest, Ytest) ←makefold(Dataset_i)
            ML ← generateML (Classification_Algorithm,Xtrain, Ytrain)
            accuracy(fold) ← calculateAccuracy(ML,Ytest,)
        end
    end
end
```

Figure 4. Pseudo-code of the classification stage

3. TESTS

3.1. Test preparation

The proposed method was tested on three real datasets, namely DBLP, Pubmed, and Cora, which have been used in numerous empirical studies. DBLP: this is a computer science bibliography database containing the name of major authors, conferences, and publications. In the network used for DBLP, objects represent authors. The meta-paths considered in this network are author-paper-author (APA), author-paper-author-paper-author (APAPA), author-conference-author (ACA), and author-conference-author- conference-author (ACACA). This dataset is typically used to extract different topics and examine the diffusion of information about a specific topic. Information contained in this dataset pertains to the period between 1954 and 2016.

Pubmed: this is a bibliography dataset for the field of medical sciences, which includes authors, conferences, and publications. In this network used for Pubmed, authors are represented by objects and the considered meta-paths are APAPA and APA. Information of this dataset is for the period between 1994 and 2003. Cora: this is another computer science bibliography database. The meta-paths used for this dataset are APAPA and APA. This dataset contains information from 1990 to 2012.

In the evaluation process, the diffusion process was modeled for several topics contained in these datasets, which include data mining, machine learning, social networks, health care, DNA and infectious disease. These particular topics were selected because of their high frequency in the dataset and the considerable amount of data available for comparison and conclusion.

Training and testing operations were performed by the use of the K-Fold method as described earlier in the paper. In this method, data is partitioned into K subsets. Each time, one of these K subsets is used for testing and the other K-1 are used for training. This procedure is repeated k times so that each data is used exactly once for training and once for testing. In the end, the average result of these K tests is reported as a final estimate. In the K-Fold method, the ratio of data of classes in each subset should match this ratio in the main set.

Finally, the performance of the method in predicting topic diffusion was evaluated in terms of the criterion known as Precision. This criterion was calculated using the following definitions:

- TP: if an active node is correctly labeled as active
- TN: if an inactive node is correctly labeled as inactive
- FP: if an active node is incorrectly labeled as inactive
- FN: if an inactive node is incorrectly labeled as active

Table 1 presents the parameters of the GCN algorithm illustrated in Figure 4. All tests of this study were performed with these parameter settings. In this table, hidden1 and hidden2 are the number of nodes in the two convolution layers. Also, early_stopping refers to the early termination condition of the algorithm, which is convergence in less than 10 iterations.

3.2. Comparison with other works

For further evaluation of the proposed method, it was compared with other methods in the field of information diffusion. This comparison was made with two methods, heterogeneous probability model- independent cascade (HPM-IC), heterogeneous probability model-linear threshold (HPM-LT) [15] and multi-relational linear treshold Model-relation level aggregation (MLTM-R) [16], which are based on probabilistic functions. These methods were implemented on the test datasets using the settings recommended in the respective references. The output of these methods was also the information diffusion path for several

scientific topics. In this section, the test results for several topics on DBLP and Pubmed datasets are reported based on precision and recall criteria. Table 2 presents the results of this evaluation in terms of precision.

Table 1. Initial parameters of GCN algorithm

Value	Parameter
0.01	learning_rate
200	epochs
16	hidden1
16	hidden1
0.5	dropout
10000	weight_decay
10	early_stopping

Table 2. Results of comparison of the proposed method with other methods for different topics on DBLP using the precision criterion

Accuracy (%)				Subject
MTLM-R	HPM-LT	HPM-IC	GCN	
%56	%55	%60	%75	Data Mining
%37	%32	%48	%50	Machine Learning
%39	%38	%40	%50	Social Network
%56	%55	%62	%75	Medical Care
%11	%12	%14	%15	DNA
%22	%20	%21	%25	infectious disease
%10	%20	%25	%30	Software Engineering
%14	%21	%22	%25	Big Data
%16	%19	%21	%25	Network
%33	%40	%50	%75	Genetic
%10	%11	%37	%50	Biology
%20	%21	%22	%25	Neural etwork

As these results indicate, the proposed method achieved a significant improvement ranging between 10% and 20% in all comparisons. In the DBLP dataset, for example, the proposed method has a 10% higher precision than other methods. The reason for this improvement could be that other methods are purely based on probability functions and calculation of probability between neighboring vertices. This means that these methods have no such thing as feature learning or intelligent processes and only repeat a constant set of calculations. In contrast, as explained in the description of the architecture, the proposed method uses different learning operations for each segment. For example, the convolution function is designed to determine the relation of each node to its neighbors through a learning process. These operations are learned intelligently during the evolution of the GCN algorithm. It should also be noted that in learning algorithms, the entire problem space can be easily explored, whereas, in probability function-based methods, only a part of the problem space can be searched.

4. CONCLUSION

This paper presented a machine learning method based on the graph neural network algorithm, which involves the selection of inactive vertices based on their neighboring active vertices in each scientific topic. Basically, in this method, information diffusion paths are predicted through the activation of inactive vertices by active vertices. Since predicting the new users who will be in the path of information flow is a recognition process, this problem can be solved by machine learning algorithms. The proposed method was tested on three real datasets, DBLP, Pubmed, and Cora, which are extensively used in the empirical studies. The evaluation process involved modeling the diffusion process for several topics contained in these datasets, including data mining, machine learning, social networks, health care, DNA and infectious disease.

Test results showed that the proposed method outperforms other methods in this area. As a potential idea for future studies, the proposed system can be implemented in a parallel platform or with the extraction and combination of other features to reach a stronger system. The use of more robust machine learning concepts may also enhance the quality of the method. The methods with possible benefits in this area include feature reduction and feature learning. The feature reduction method is particularly useful for reducing the overall complexity of the recognition method. Feature learning is a process involving the transfer of the data processing from the original feature space to a new space with higher feature resolution.

REFERENCES

[1] E. Bakshy, I. Rosenn, C. Marlow, L. Adamic, "The role of social networks in information diusion*," in: Proceedings of the 21st international conference on World Wide Web, ACM*, pp. 519, 2012.

[2] M. S. Granovetter, "The strength of weak ties," In Social networks, Academic Press. pp. 347-367, 1977.
[3] Y. Hu, R. J. Song, M. Chen, "Modeling for Information Di usion in Online Social Networks via Hydrodynamics," *IEEE Access,* vol. 5, 2017.
[4] K. Ikeda, *et al.,* "Multi-agent information diffusion model for twitter," *In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)- IEEE Computer Society* vol. 1, pp. 21-26, 2014.
[5] T. Kipf, N. Thomas, and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," arXiv preprint arXiv:1609.02907, 2016.
[6] Y. Moreno, R. Pastor-Satorras, A. Vespignani, "Epidemic outbreaks in complex heterogeneous networks," *The European Physical Journal B*, vol. 26, no. 4, 2002.
[7] R. Yang, B.-H. Wang, J. Ren, W. J. Bai, Z. W. Shi, W. X. Wang, T. Zhou, "Epidemic spreading on heterogeneous networks with identical infectivity," *Physics Letters A,* vol. 364, pp. 3-4, 2007.
[8] M. Salehi, R. Sharma, M. Marzolla, M. Magnani, P. Siyari, D. Montesi, "Spreading processes in multilayer networks," *IEEE Transactions on Network Science and Engineering,* vol. 2, no. 2, 2015.
[9] L. Wang, G. Z. Dai, "Global stability of virus spreading in complex heterogeneous networks," *Siam Journal on Applied Mathematics,* vol. 68, no. 5, 2008.
[10] M. Nadini, K. Sun, E. Ubaldi, M. Starnini, A. Rizzo, N. Perra, "Epidemic spreading in modular time-varying networks," arXiv preprint arXiv:1710.01355, 2017.
[11] G. Demirel, E. Barter, T. Gross, "Dynamics of epidemic diseases on a growing adaptive network," *Scientic reports,* vol. 7, 2017.
[12] P. Sermpezis, T. Spyropoulos, "Information diffusion in heterogeneous networks: The conguration model approach," *in: Proceedings-IEEE INFOCOM*, pp. 3261, 2013.
[13] Y. Zhou, L. Liu, "Social in uence based clustering of heterogeneous information networks," *in: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM,* pp. 338, 2013.
[14] S. Molaei, S. Babaei, M. Salehi, M. Jalili, "Information spread and topic diffusion in heterogeneous information networks," *Scientic Reports,* vol. 8, no. 1, 2018.
[15] S. Molaei, *et al.,* "Information Spread and Topic Diffusion in Heterogeneous Information Networks," *Sci. Rep.,* vol. 8, 2018.
[16] H. Gui, Y. Sun, J. Han, G. Brova, "Modeling Topic Diffusion in Multi-Relational Bibliographic Information Networks," *In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14,* pp. 649-658, 2014.
[17] T. Mikolov, K. Chen, G. Corrado, J. Dean, "Ecient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.
[18] W. Cheng, C. Greaves, M. Warren, "From n-gram to skipgram to concgram," *International journal of corpus linguistics,* vol. 11, no. 4, 2006.
[19] B. Perozzi, R. Al-Rfou, S. Skiena, "Deepwalk: Online learning of social representations," *in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM,* pp. 70, 2014.
[20] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, 1997.
[21] Q. Cao, H. Shen, K. Cen, W. Ouyang, X. Cheng, "Deephawkes: Bridging the gap between prediction and understanding of information cascades," *in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM*, 2017.
[22] Y. LeCun, Y. Bengio, *et al.*, "Convolutional networks for images, speech, and time series," The handbook of brain theory and neural networks, vol. 3361, no. 10, 1995.
[23] F. J. Ordonez, D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, 2016.
[24] Citation Network Dataset, Available [online], URL http://konect.uni-koblenz.de/networks/subelj_cora
[25] P. Kumaran, S. Chitrakala, "Community formation based influence node selection for information diffusion in online social network," *In 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE, 2016),* pp. 1-6, 2016.
[26] M. Lahiri and M. Cebrin, "The Genetic Algorithm as a General Diffusion Model for Social Networks," *Proc. of the 24th AAAI Conference on Artificial Intelligence*, 2010.
[27] L. Li, S. Li, X. Chen, "A new genetics-based diffusion model for social networks," *In 2011 International Conference on Computational Aspects of Social Networks (CASoN, 2011),* pp. 76-81, 2011.
[28] H. Zhu, C. Huang, H. Li, "Information diffusion model based on privacy setting in online social networking services," *The Computer Journal*, vol. 58, no. 4, pp. 536-548, 2014.
[29] L. Liu, *et al*., "Modelling of information diffusion on social networks with applications to WeChat," *Physica A: Statistical Mechanics and its Applications,* vol. 496, pp. 318-329, 2018.
[30] Ayman Madi, Oussama Kassem Zein, Seifedine Kadry, "On the improvement of cyclomatic complexity metric," *International Journal of Software Engineering and Its Applications*, vol. 7, no. 2, pp. 67-82, 2013.
[31] Seifedine Kadry, Rafic Younès. "Etude Probabiliste d'un Systeme Mecanique a Parametres Incertains par une Technique Basee sur la Methode de transformation," *Proceeding of CanCam. Canada,* 2005.