Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 10, No. 2, 2022, 233-243.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

# A Hybrid Deep Network Representation Model for Detecting Researchers' Communities

Atefeh Torkaman[1], Kambiz Badie[2], Afshin Salajegheh[1*], Mohammad Hadi Bokaei[3] and Seyed Farshad Fatemi[4]

*1. Department of Computer, South Tehran Branch, Islamic Azad University, Tehran, Iran.*
*2. E-Services and E-Content Research Group, IT Research Faculty, ICT Research Institute, Tehran, Iran.*
*3. Department of Information Technology, ICT Research Institute, Tehran, Iran.*
*4. Sharif University of Technology, Tehran, Iran.*

**Article Info**

**Article History:**
*Received 09 October 2021*
*Revised 06 January 2022*
*Accepted 17 February 2022*

*DOI:10.22044/jadm.2022.11243.2277*

**Keywords:**
*Complex Networks, Network Representation, Deep Learning, Citation Network, Recurrent Neural Network, Long Short-term Memory, Natural Language Processing, Community Detection.*

**Corresponding author A_salajegheh@azad.ac.ir (A. Salajegheh).*

**Abstract**

Recently, network representation has attracted many research works mostly concentrating on representing the nodes in a dense low-dimensional vector. There exist however, some network embedding methods focusing only on the node structure, and some others considering the content information within the nodes. In this paper, we propose a hybrid deep network representation (HDNR) model that uses a triplet deep network architecture considering both the node structure and the content information for network representation. In addition, the author's writing style is also considered as a significant feature in the node content information. Due to the successful application of deep learning in natural language processing (NLP), our model makes use of a deep random walk method in order to exploit the inter-node structures and two deep sequence prediction methods to extract the nodes' content information. The embedding vectors generated in this manner are shown to have the ability of boosting each other for learning the optimal node representation, detecting more informative features, and ultimately a better community detection. The experimental results obtained confirm the efficiency of this model for network representation compared to the other baseline methods.

## 1. Introduction

Complex networks that include various networks such as social networks and citation networks with complex structure and rich node content provide different services, and are applied in various fields like sociology, biology, and economics.

Among several complex networks, citation networks contain paper, book or patent sources, and are linked by the co-citation relationships. The citing relation $R \subseteq U \times U$ in these networks is as "(1)":

$$uRv = v\,cites\,u \tag{1}$$

which determines a citation network $N = (U, R)$, where U is a set of units that includes paper, books, patents, etc.

The citation network analysis is usually applicable in determining the important units and the relations between these units. The complexity of these networks' structure and their huge information pose many challenges for different network analysis tasks.

Generally, the way these networks are represented is a fundamental problem in network analysis.

In order to address this issue, network representation learning or network embedding plays a critical role that aims to project each node into informative low-dimensional and compressed vectors, while preserving its certain microscopic and macroscopic network structure like the community membership, proximity orders, and also inherent networks' information. It should be noted that the representation vectors could be applied in many valuable network analysis tasks like classification, community detection, link prediction, node clustering, recommendation, and data visualization.

The network embedding models often focus on the network structure. In this regard, some of the classical methods are based on the eigenvectors of affinity graphs as feature vectors.

Some others apply graph factorization for the embedding networks. However, they are not scalable for large networks. Also some studies represent the networks as a set of random walks such as LINE, DeepWalk, and Node2Vec. Although, all these methods perform well in comparison to other network representation methods such as spectral clustering, the problem is that they just consider the structural information in the network. However, in the real-world networks, the nodes contain a lot of information.

In order to incorporate the node content information in the network representation, RTM, TADW, Tri_DNR proposed in the recent years are regarded as worthy methods. RTM, however, suffers from obtaining appropriate matrix factorization, TADW is incapable of handling large scale networks, and Tri_DNR is also a supervised algorithm that is simply in charge of coupling two neural networks, and enforces the model with vertex labels. In the real citation networks, in addition to the node content and structural information, the authors' writing style can also be considered as a cognitive feature for learning the network embedding.

Usually, in these networks, papers cite each other, and most of these relations indicate the interaction between the students and their advisors. Usually, the students consider their advisor's writing style in their papers.

Figure 1 shows a toy citation network; each node signifies a paper, and links denote the citation relations. As depicted in Figure 1, papers 8, 10, and 11 include two common phrases in their titles: "Similarity-Search" and "High-Dimensional Spaces". In addition, paper 3 concludes "high dimensional spaces". If we ignore the writing style, they will likely be represented with a similar representation and all of them must be allocated to one community. Although paper 3 is more likely to papers 8, 10, and 11 in terms of shared common phrases and connections, the writing style leads them to two different classes (data mining and artificial intelligence).

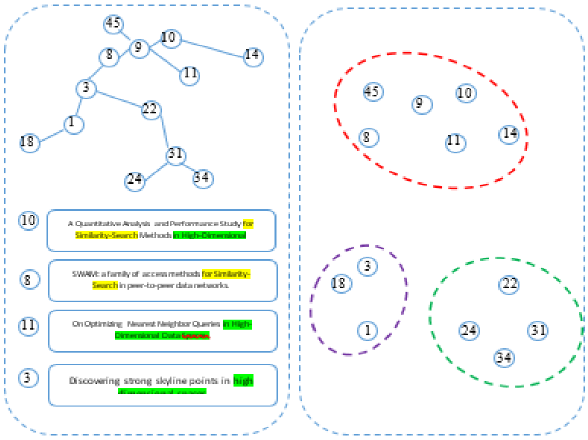


**Figure 1. An example of effect of writing style.**

In general, how to arrange the words, using some punctuation marks between words such as hyphen, underline, etc. using a combination of uppercase and lowercase letters can show the writing style. Regarding this, by investigating the writing style and the way the words are arranged, one can identify their true community. This phenomenon is very intuitive not only in the citation networks but also in other social networks with different languages, cultures, literature, and careers.

Hence, we come up with an idea to learn the embedding vectors from the network structure, content, and the authors' writing style in the form of a triplet deep neural network architecture called HDNR. These vectors mutually enhance each other for an optimal node representation.

The experimental results demonstrate that HDNR outperforms the baselines in both effectiveness and efficiency.

In summary, the following contributions will be made in this paper:

- We proposed HDNR as a novel representation model, which learns the low-dimension embedding for each node, and is applicable for different types of networks that are documented and have rich information such as citation networks and social networks.
- For the first time, our method considers the authors' writing style as an important content feature for embedding the citation networks.
- For the network representation purposes, our method considers the network structure along with the co-occurrence of words and the authors' writing style,

while the previous studies have not considered such items altogether.

The rest of this paper is organized as what follows. In Section 2, the network embedding methods are summarized. The basic concepts are mentioned in Section 3. The proposed model is brought up in Section 4. Analysis of the results are discussed in Section 5. Finally, the concluding remarks and future works are presented in Section 6.

## 2. Related Works

As stated before, network representation attempts to represent the nodes in a lower-dimensional space, which has been very useful in node classification [1], [2], link prediction [3], [4], recommendation [5], [6].

Recently, several representation methods have been proposed in order to learn the appropriate embedding vectors that investigate the networks from different perspectives such as the structural aspects and content aspects..

From the network structure perspective, some network representation methods consider the node relations and their neighbors. In this regard, some of the classical network embedding methods are based on finding the appropriate eigenvectors [7], [8], [9], [10], [11] or graph factorization for the network embedding purposes [12] , [13].

Also there exist some methods functioning on the basis of the statistical properties of the node and its neighborhood to provide the structural features [14].

Recently, due to the prosperous usage of deep learning in NLP [15], the DeepWalk algorithm has been proposed that employs the random walk method to learn a feature vector for each node [16]. In this context, Grover *et al*. have improved the DeepWalk algorithm using the DFS and BFS algorithms to consider the homophily and structural equivalences [17].

There is also a method called LINE that takes into account the first-order proximity observed from the connections between the nodes, while maintaining the second-order proximity for the common neighbors of the node [18]. Within this scope, Wang *et al*. have suggested a semi-supervised deep learning model based on the first-order and second-order proximity [19].

From the network content aspect, some recent methods exist that attempt to convert a text into a vector space such as TFIDF or LDA that employs the bag-of-word method [20]. Within this scope, the Skip-Gram model utilizes an ordinary neural network model for learning the distributed vectors as well [15].

Although the aforementioned approaches perform well in comparison to the other network representation methods, the problem is that they just consider one source of information in the network that may cause a shallow representation. However, in the real-world networks, the nodes contain much various information. In this regard, a recent approach named TADW has also exploited the structural and content information for network representation [21]. The drawbacks of this method are the difficulty in obtaining the matrix factorization as well as handling the large-scale networks and ignoring the order of the nodes. Also Tri_DNR is another model that benefited from both the structure and content, which uses the node labels to enhance the classification accuracy [22].

In addition, an approach called CARE has been proposed that applies local neighborhood and community node information to cover both the local and global social networks structures [23].

In the meantime, NE-FLGC is an approach considering the node content that proposes a network embedding method based on the first-order and second-order proximity observed in the network structure [24].

In this research work, HDNR is proposed which uses a deep triplet architecture to extract information from both the network structure and content. At the content level, in addition to the co-occurrence of word sequences, the author's writing style is also considered as a cognitive feature.

## 3. Basic Concepts

### 3.1. Network embedding problem

Given a network $G=(V,E,D)$, $V=\{v_1,v_2,v_3,...,v_n\}$ is a set of nodes, where n is the number of nodes. $E=\{e_{ij}\}_{i,j=1}^{n}$ consists of the set of edges, where $e_{ij}$ encodes the edge between $v_i$ *and* $v_j$ . $D$ is the set of textual data that relates to each node of $v_i$ . The network embedding problem aims to learn a mapping function: $f:V\rightarrow R^d$ which projects each node to a low-dimensional space to capture the structural and content properties of the network. Thus the nodes with similar network structures or having a similar text content are close in the representation space.

### 3.2 Preliminary: Skip-Gram, DeepWalk, and LSTM

In this sub-section, the basic applied models in our research are introduced.

*Skip-Gram*: One of the most applicable methods in NLP is the Skip-Gram model [15] introduced by Mikolov in 2013, an efficient way that learns the representation vectors of words from the massive amount of text data. The main purpose of this model is to find word representations for predicting the neighbor words in a document.

*paragraph vector model (Doc2Vec)*: Later, in 2014, Le and Mikolov extended the Skip-Gram model as the paragraph vector model (Doc2Vec), an unsupervised framework that learns continuous distributed vector representations in variable-length documents. These vectors are learned to predict the surrounding words in texts sampled in a paragraph [25]

*DeepWalk Model*: This model was proposed by Prozzi *et al*. [16], as the first and most popular embedding method that bridges between network embedding and word embedding by treating the nodes as words and creating short random walks as sentences. DeepWalk transforms the network into collections of the linear sequence using a sampling method known as a truncated random walk. Indeed, the sampled node sequences specify the connections between the nodes in a network [26].

Accordingly, the NLP models such as the Skip-Gram model can be trained on these linear sequences (random walks) for obtaining a distributed representation vector for nodes. This model only uses the structural information for learning, and thus ignores the text information augmented with each node.

*Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Model*: Recurrent neural networks (RNN), also known as Auto-associative or feedback network, are a family of neural networks that are well-appropriate for processing the sequential data [27]. These networks are often used in the natural language processing, speech recognition, computer vision, and several other areas.

Usually, RNNs may suffer from the inability to achieve long-term dependencies due to the vanishing gradient problem [28]. In this regard, the LSTM model is a breakthrough in the recurrent neural network (RNN), which has been introduced by the German researchers Hochreiter and Schmidhuber to model long-term dependencies in the sequential data and to solve the vanishing gradient problem [29]. LSTM works well for a variety of learning tasks, previously considered impossible to learn. These tasks include machine translations [30], handwriting recognition [31], speech recognition [32], [33], image captioning [34], [35], and every other task with long-term dependencies [36].

One-to-many, many-to-one, and many-to-many are different types of RNNs. Image captioning and music generation are two important applications of the one-to-many models. The sentiment classification is one of the most common used of the many-to-one model. Named entity recognition, machine translation, and video classification on the frame level are significant applications of the many-to-many model [37].

In this work, for representing the author's writing style information, we use the one-to-many LSTM model, which is represented in Figure 6.

This model supplies one input ($v_1$), and predicts multiple outputs ($h_{1,2,\ldots T}$), and tries to generate sentences based on a single starting word (the node (paper) Id number). The main purpose of this model is as "(2)"

$$h_{1,2,\ldots T} = LSTM(v_t) \tag{2}$$

$h_{1,2,\ldots T}$ are different outputs, and $v_t$ is the only input of this model.

Generally, LSTM includes two important components: states and gates [38].

States: the values that offer the information for output. The hidden states and cell states are the different types of states. The cell state is a memory of the LSTM cell; the hidden state is an output of this cell.

Gates: The values that control the flow of states information. The gates consist of input, output, and forget gates.

Within one-to-many LSTM scope, the formulas of the gates are defined as follow:

Input gate, denoted as $i_t$ that determines which information should be updated in the current situation:

$$i_t = \sigma(W_i \begin{vmatrix} h_{t-1} \\ v_t \\ \hat{y}_{t-1} \end{vmatrix} + b_i) \tag{3}$$

Forget gate: denoted as $f_t$ that filters the cell state and outputs the filtered result:

$$f_t = \sigma(W_f \begin{vmatrix} h_{t-1} \\ v_t \\ \hat{y}_{t-1} \end{vmatrix} + b_f) \tag{4}$$

Output gate: defined as $o_t$ that proposes new values for the cell state:

$$o_t = \sigma(W_o \begin{bmatrix} h_{t-1} \\ v_t \\ \hat{y}_{t-1} \end{bmatrix} + b_o) \qquad (5)$$

Given $g_t$ as a new memory cell candidate:

$$g_t = \tanh(W_g \begin{bmatrix} h_{t-1} \\ v_t \\ \hat{y}_{t-1} \end{bmatrix} + b_g) \qquad (6)$$

and $c_t$ as the final memory cell:

$$c_t = i_t \odot g_t + f_t \odot c_{t-1} \qquad (7)$$

Thus $h_t$ as the final hidden state:

$$h_t = o_t \odot \tanh(c_t) \qquad (8)$$

For the first step, $h_0 = \hat{y}_t = \vec{0}$ , and $v_t$ is the only input that is fed to the model and for the next steps $v_t = \vec{0}$ . $W_i, W_f, W_o, and\, W_g$ are the weights learned from the network. $b_i$ , $b_f$ , $b_o$ , $and\, b_g$ are the bias vectors.

$\sigma$ is the sigmoid activation function, tanh() is the hyperbolic tangent function, and $\odot$ is the element-wise product.

The unnormalize score for step t is computed as follows:

$$S_t = h_t W_s \qquad (9)$$

$S_t$ is inserted into the softmax layer for the probability distribution over the total words in the vocabulary. Thus the distribution for generating the output word in the $t_{th}$ step is as follows:

$$d_t = soft\max(S_t) \qquad (10)$$

The output $\hat{y}_t$ is a sample from $d_t$ and its embedding vector is fed to the next step.

## 4. Proposed Model

In this section, we introduce our model (HDNR), which utilizes the network structure together with the nodes' contents to learn a high-quality embedding vector for every node. The HDNR model consists of 3 parts: DeepWalk, Doc2Vec, and LSTM. Each part represents the node $v_i$ by an embedding vector. At the end, HDNR concatenates these three vectors into a unified representation to consider the structural and content features together.

Given a network G with N nodes, what HDNR does is to try to minimize the following loss:

$$L = L_{DeepWalk} + L_{Doc2Vec} + L_{LSTM} \qquad (11)$$

Figure 2 represents the architecture of our model. At the top of Figure 2, the structural information of a node is extracted as a vector using the DeepWalk method, assuming that the connected nodes are statistically interdependent (e.g. connected nodes might have similar embedding).

With respect to this method, first, a walk is generated from the node $v_t$ , then randomly jumping to other nodes to generate other walks. Next, the Skip-Gram model is applied to these random walks to obtain network embedding.

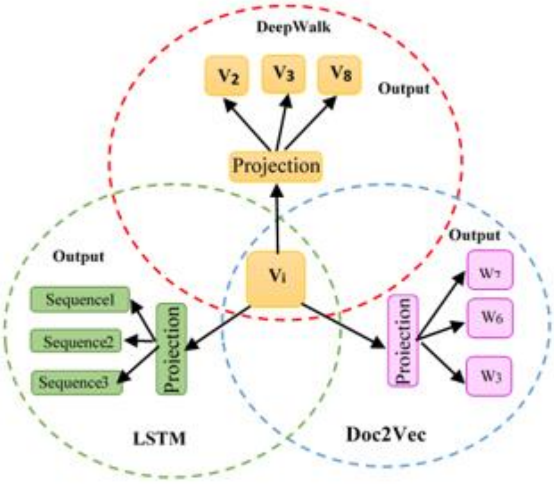


**Figure 2. Architecture of HDNR method.**

Here, each node and the generated random walk are given as a word and its context, respectively. Figure 3 shows the overview of the DeepWalk approach. The vector that is circled around and marked with the letter A is used in the final concatenation.

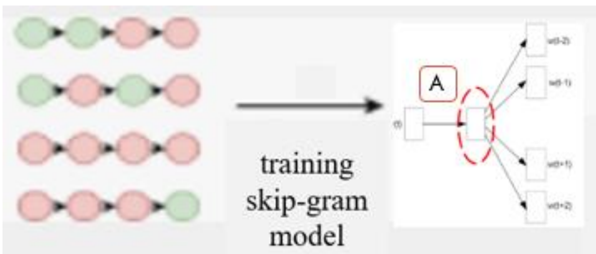


**Figure 3. An overview of DeepWalk approach.**

The goal of the DeepWalk is to enhance the likelihood of nodes around $v_t$ for all random walks $r \in R_i$ through the following loss function:

$$L_{DeepWalk} = -\sum_{i=1}^{N}\sum_{r\in R_i}\sum_{-b\le j\le b,\, j\ne 0}\log P(v_{i+j}\,|\,v_i) \quad (12)$$

$N$ is the total number of nodes, $R_i$ is the random walk sequence, and $b$ is the window size of these sequences. The likelihood of observing the neighbors of the node $v_i$ by using softmax is:

$$\mathrm{P}(v_{i+j}\,|\,v_i) = \frac{\exp(v_{v_i}^{T} v'_{v_{i+j}})}{\sum_{v=1}^{N}\exp(v_{v_i}^{T} v'_{v})} \quad (13)$$

$v_v$ is the input representation vector of node $v$ and $v'_v$ is the output representation vector of node $v$.

In the left lower layer of Figure 2, by utilizing a one-to-many LSTM model, the author's writing style is extracted and then embedded into a vector space.

In this regard, the node (paper) Id numbers are represented with a one-hot vector that is fed as an input ($v_i$) into the one-to-many LSTM. Then the corresponding sequences of words in the papers' titles and abstracts are generated as the outputs (Figure 4).

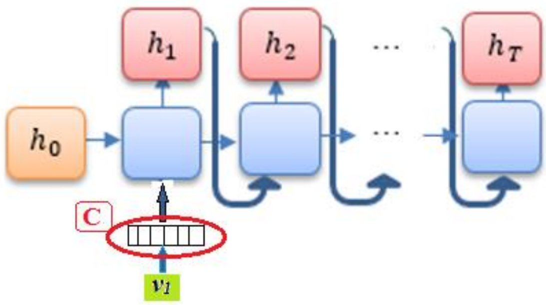


**Figure 4. An overview of one-to-many LSTM method.**

The number of the words in the sequences depends on the window size, which is considered based on the task.

For example, Figure 5 shows one-to-many LSTM with paper number 95049 and 3 text sequences as the output with window size 3.

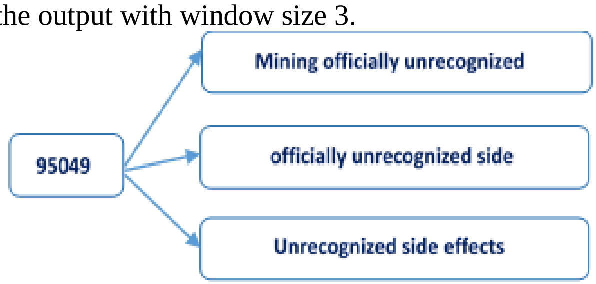


**Figure 5. An example of one-to-many LSTM.**

In fact, in this layer, we try to learn the sequences of the words given a node Id (the author's writing style) through the "(14)" loss function.

$$L_{LSTM} = -\sum_{i=1}^{N}\sum_{s\in S}\log P(s\,|\,v_i) \quad (14)$$

where $s$ is the word sequence given input node $v_i$

The likelihood of observing a given sequence $s$ (style of writing) per node $v_i$ is:

$$P(s\,|\,v_i) = \prod_{j=1}^{t} P(w_j\,|\,v_i, w_0, \dots, w_{j-1}) \quad (15)$$

where:

$$P(w_j\,|\,v_i, w_0, \dots, w_{j-1}) = d_{j,w_j} \quad (16)$$

j is the index of the word $w_j$ in $d_j$ that is defined in "(10)".

Finally, the right lower layer of Figure 2 captures the contextual information, e.g. the correlations of words within a document. In this layer, we use the Doc2vec method to learn the node content vector.

The overview of the Doc2vec method is depicted in Figure 6. The hidden vector (B) is used in the final concatenation.

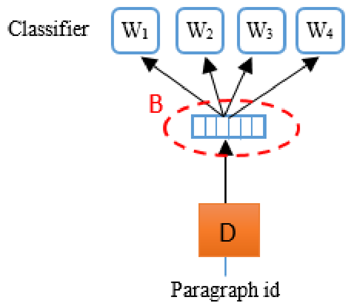


**Figure 6. An overview of Doc2Vec method.**

We attempt to learn the sequences of the words given a paragraph Id through the "(17)" loss function.

$$L_{Doc2Vec} = -\sum_{i=1}^{N}\sum_{-b\le j\le b,\, j\ne 0}\log P(w_j\,|\,v_i) \quad (17)$$

where $b$ is the window size of the sequences and $w_j$ is the j-th word in a Doc2Vec's contextual window. The likelihood of observing the words $w_{-b} : w_b$ given the current node $v_i$ is:

$$\mathrm{P}(w_j\,|\,v_i) = \frac{\exp(v_{v_i}^{T} v'_{w_j})}{\sum_{w=1}^{W}\exp(v_{v_i}^{T} v'_{w})} \quad (18)$$

where $v'_{w_j}$ is the output representation of $w_j$ and W is the total number of separated words in the entire network.

Consequently, we concatenate all the three hidden vectors (A; Figure 3, B; Figure 6, C; Figure 4) according to "(11)" as the final feature vector. These three vectors will boost each other for detecting more informative features and better node representation.

## 5. Analysis of results

In this section, the quality of our method is evaluated.

### 5.1 Real citation networks

We run the proposed method on two real citation networks, and consider the paper title as the node content. In the citation network, each paper cite or is cited by other papers.

DBLP version 4: includes the bibliography data in the computer science. In our experiments, for a fair comparison, a list of conferences from four research areas selected by [22] is considered, which are data mining (ICDM, SDM, PKDD, PAKDD, KDD), database (SIGMOD, ICDE, VLDB, EDBT, PODS, DASFAA, ICDT, SSDBM, CIKM), artificial intelligent (IJCAI, AAAI, NIPS, ICML, ECML, ACML, IJCNN, UAI, ECAI, COLT, ACL, KR), and computer vision (ICCV, ECCV, ACCV, MM, ICPR, ICIP, CVPR, ICME). This network includes 60,744 papers and 52,890 edges.

CiteSeer-M10: is a well-known citation network that consists of 10 distinct scientific research areas: financial, archaeology, biology, computer science, agriculture, economics, material science, petroleum chemistry, physics, industrial engineering, and social science. 10 classes, 10,310 papers, and 77,218 edges exist in this network.

### 5.2 Experimental settings

We perform network visualization and node classification in order to evaluate the HDNR quality compared to different models.

### 5.3 Baselines

In order to compare the result of the proposed model with the other notable approaches, we use some baseline models.

From the Structure Perspective:

LINE [18]: is a network embedding method based on the network structure, which is scalable for huge networks, and easily applicable on arbitrary types of them.

DeepWalk [16]: applying random walk on the network structure to generate linear sequences. Here, the Skip-Gram model is trained on these walks in order to obtain an embedding vector for every node.

From the Content Perspective:

LDA [20]: is a Latent Dirichlet allocation algorithm, a fast and flexible generative probabilistic model, which learns a topic distribution for providing a more powerful and explicit representation for each document.

Doc2Vec [39]: is the Paragraph Vectors algorithm that embeds each document in the distributed dense vectors, and can easily train the model by predicting the words in a document.

Tri_DNR [22]: is a triparty deep representation algorithm that exploits the network structure, label, and node content to learn the representation for every node in a network.

In this paper, we train our model and baseline methods with the structure and text embedding dimensions $k = 100$, and adjust the other parameters like the baselines [22], [16], [18], [20], [39]. The experiments are repeated 10 times for every method, and the results obtained are reported.

### 5.4 Performance on community detection

A community is a set of entities that are close to each other within their own group rather than to the entities outside it. The community recognition is one of the most fascinating research topics that has received a lot of attention in several fields such as statistics and computer science as well. The problem of community detection is NP-hard [40].

In this work, in order to find the researchers' communities and experiment with different algorithms in the detecting processes, a linear SVM is applied. Indeed, compared to the non-linear or complicated relational classifiers, this linear classifier decreases the effect of complicated learning on the performance of the classification. Regarding this, the node classification is conducted separable on the DBLP and Citeseer datasets. In order to measure the performance of our algorithm, we consider the percentages of the training samples between 10% to 70%. The results are reported in Table 1 Table 2.

According to these tables, we have the following observation:

The methods like DeepWalk and LINE that are only based on the structural characteristics, due to limited information and sparsity, perform poorly on these two datasets. LSTM that is only learned node content (document or sentences) embedding is much better than the structure-based methods, and Doc2Vec performs suitable on the DBLP set.

In order to improve the performance of these methods, the concatenations of the structural and content methods are considered. In this way, the embedding of the DeepWalk and Doc2Vec, and LSTM are concatenated. The results of these combinations are better than using them individually. HDNR outperforms DeepWalk, Doc2Vec, a combination of them, LINE, and LDA. However, HDNR performs better than Tri_DNR, although we do not use the label information. Tri_DNR uses a simple coupled neural network model; our model applies LSTM, which is capable of learning long-term dependencies and remembering information more easily than the simple neural networks, and does not suffer from vanishing gradients in the sequential data.
In order to summarize, HDNR remarkably outperforms its counterparts. Specially, under p = 70%, HDNR (0:790) beats DeepWalk (0:428), and Tri_DNR (0:744) by 84:6% and 6:76% on the DBLP dataset.

### 5.5 Network visualization

One of the main usages of network representation is creating visualizations that layout a network in a 2D or 3D space. We visualize the Citeseer M10 citation network by mapping the embedding vectors onto a 2D space in Figure 7. The papers in different areas are shown in different colours.
As shown in Figure 7(a), in the visualization of Doc2Vec, the nodes with the same colours are not clustered together because it neglects the network structure. In Figure 7(b), the Tri-DNR model performs better than Doc2Vec but some classes are not clear. In Figure 7(c), HDNR is clear enough and has a meaningful layout for each community.

**Table 1. Average Macro fl score values on Citeseer-M10.**

| %P | DeepWalk | Doc2Vec | DeepWalk + Doc2Vec | LINE | LDA | Only LSTM | LSTM + DeepWalk | Tri_DNR | HDNR |
|---|---|---|---|---|---|---|---|---|---|
| 70% | 0.434 | 0.503 | 0.628 | 0.589 | 0.589 | 0.645 | 0.727 | 0.777 | 0.780 |
| 50% | 0.425 | 0.494 | 0.614 | 0.581 | 0.581 | 0.611 | 0.689 | 0.753 | 0.761 |
| 30% | 0.411 | 0.477 | 0.586 | 0.569 | 0.549 | 0.598 | 0.656 | 0.715 | 0.720 |
| 10% | 0.354 | 0.432 | 0.495 | 0.531 | 0.458 | 0.513 | 0.511 | 0.626 | 0.702 |

**Table 2. Average Macro fl score values on DBLP.**

| %P | DeepWalk | Doc2Vec | DeepWalk + Doc2Vec | LINE | LDA | Only LSTM | LSTM + DeepWalk | Tri_DNR | HDNR |
|---|---|---|---|---|---|---|---|---|---|
| 70% | 0.428 | 0.623 | 0.690 | 0.439 | 0.654 | 0.671 | 0.75 | 0.744 | 0.790 |
| 50% | 0.426 | 0.620 | 0.686 | 0.438 | 0.655 | 0.611 | 0.689 | 0.738 | 0.773 |
| 30% | 0.423 | 0.617 | 0.681 | 0.438 | 0.652 | 0.598 | 0.656 | 0.727 | 0.740 |
| 10% | 0.398 | 0.605 | 0.653 | 0.427 | 0.644 | 0.511 | 0.654 | 0.687 | 0.710 |

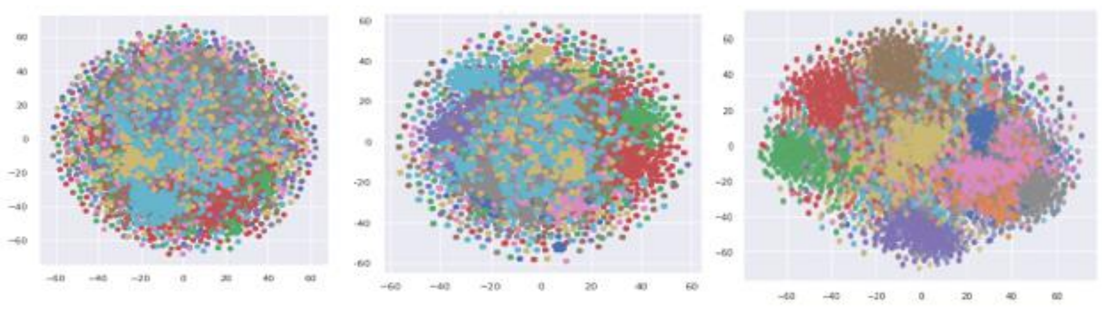


**Figure 7. Visualization on Citeseer M10 dataset.**

## 6. Concluding remarks

In this research work, we proposed a hybrid deep representation (HDNR) model, which considers the node structure and the node content information for network representation purposes. In addition, we discussed that the authors' writing style should be considered as an important feature for embedding the citation networks.
We performed some experiments on two real datasets, CiteSeer-M10 and DBLP version 4,

where the experimental results stand for the high efficiency and high effectiveness of HDNR compared to the other several state-of-the-art baseline algorithms.
The significant items with regard to the performance of our approach are the triplet architecture, vectors sizes, learning rate, and size of the training window.
In the future, we intend to investigate the Node2Vec model, instead of the DeepWalk model, in order to find the first-order and the second-order proximity of the network structure at the same time, something that is very effective in finding more accurate communities. Also we would plan to extend our approach to the heterogeneous networks.

## References


[1] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node Classification in Social Networks, In: Aggarwal C. (eds)," in *Social Network Data Analytics. Springer*, Boston, MA, 2011.

[2] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine,* Vol. 29, No. 3, pp. 93-106, 2008.

[3] Gao, Fei, Katarzyna Musial, Colin Cooper, and Sophia Tsoka., "Link prediction methods and their accuracy for different social networks and network metrics," *Scientific programming* , 2015 .

[4] D. L.-N. a. J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology,* Vol. 58, No. 7, pp. 1019-1031, 2007.

[5] Jamali, Mohsen and Martin Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010.

[6] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: A heterogeneous information network approach," *WSDM,* pp. 283-292, 2014.

[7] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Advances in neural information processing,* pp. 585-591, 2002.

[8] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science,* Vol. 290(5500), pp. 2319–2323, 2000.

[9] S. T. Roweis and L. K. Saul, "Non-linear dimensionality reduction by locally linear embedding," *Science,* Vol. 290(5500), pp. 2323, 2326, 2000.

[10] T. F. Cox and M. A. Cox, "Multi-dimensional scaling," *In: Handbook of data visualization, Springer,* pp. 315-347, 2000.

[11] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE transactions on pattern analysis and machine intelligence,* Vol. 29, No. 1, pp. 40–51, 2007.

[12] A. Ahmed, N. Shervashidze, S. Narayanamurthy V. Josifovski, and A. J. Smola, "Distributed large-scale natural graph factorization," in *Proceedings of the 22nd international conference on World Wide Web*, 2013.

[13] Ou. M, Cui. P, Pei. J. Zhang. Z, and Zhu. W, "Asymmetric transitivity preserving graph embedding," in *In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining,* 2016.

[14] Henderson, K., Gallagher, B., Li, L., Akoglu, L., Eliassi-Rad, T., Tong, H., and Faloutsos, C., "It’s who you know: graph mining using recursive structural features.," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining.*, 2011.

[15] T. Mikolov, K. Chen, G.Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *arXiv preprint arXiv:1301.3781*, 2013.

[16] B. Perozzi, R. Al-Rfou, and S. Skiena, "“Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, (KDD, 2014),* 2014.

[17] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," in *Proceedings of the 22nd ACM SIGKDD international conference (KDD ’16),* San Francisco, 2016.

[18] Tang J, Qu M, Wang M, Zhang M, Yan J, and Mei Q, "LINE : Large-scale Information Network Embedding Categories and Subject Descriptors," *In: Proceedings of the 24th international conferenceon World Wide Web,* pp. 1067–1077, 2015.

[19] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2016),* 2016.

[20] David M Blei, Andrew Y Ng, and Michael Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research,* Vol. 3, pp. 993–1022, 2003.

[21] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang, "Network representation learning with rich text information," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[22] Pan, Shirui, Jia Wu, Xingquan Zhu, Chengqi Zhang, and Yang Wang, "Tri-party deep network representation," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial*

*Intelligence (IJCAI-16),* 2016.

[23] Keikha, Mohammad Mehdi, Maseud Rahgozar, and Masoud Asadpour, "Community aware random walk for network embedding," *Knowledge-based Systems,* Vol. 148, pp. 47-54, 2018.

[24] Hongyan Xu, Hongtao Liu, Wenjun Wang, Yueheng Sun, NE-FLGC: Network Embedding based on Fusing Local (First-Order) and Global (Second-Order) Network Structure with Node Content, Advances in Knowledge Discovery and Data Mining, 2018.

[25] Quoc V Le and Tomas Mikolov, "Distributed representations of sentences and documents," in *InternationalConference on Machine Learning*, 2014.

[26] Chen, Haochen, Bryan Perozzi, Rami Al-Rfou, and Steven Skiena, "A tutorial on network embeddings.," *arXiv preprint arXiv :1808.02590,* 2018.

[27] Abbas Ahmed and Josef Holmberg, "Information extraction from short text messages," in *LU-CS-EX 2019-18*.

[28] Kozma, Robert, Cesare Alippi, Yoonsuck Choe, and Francesco Carlo Morabito, eds., Artificial Intelligence in the Age of Neural networks and Brain computing, Academic Press, 2018..

[29] Taheri, Aynaz, Kevin Gimpel, and Tanya Berger-Wolf, "Sequence-to-sequence modeling for graph representation learning," *Applied Network Science 4,* Vol. 1, No. 68, 2019.

[30] Cui, Yiming, Shijin Wang, and Jianfeng Li, "LSTM neural reordering feature for statistical machine translation," *arXiv preprint arXiv,* 2015.

[31] Greff, Klaus, Rupesh K. Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber, "LSTM: A search space odyssey," *IEEE transactions on neural networks and learning systems,* vol. 28, No. 10, pp. 2222-2232, 2016.

[32] Graves, Alex, Navdeep Jaitly, and Abdel-rahman Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *IEEE workshop on automatic speech recognition and understanding, IEEE*, 2013.

[33] Han, Song, Junlong Kang, Huizi Mao, Yiming Hu, Xin Li, Yubin Li, Dongliang Xie *et al*., "Ese: Efficient speech recognition engine with sparse lstm on fpga.," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2017.

[34] Chen, Minghai, Guiguang Ding, Sicheng Zhao, Hui Chen, Qiang Liu, and Jungong Han, "Reference-based LSTM for image captioning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[35] Aneja, Jyoti, Aditya Deshpande, and Alexander G. Schwing, "Convolutional image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5561-5570,* 2018.

[36] Staudemeyer, Ralf C. , Eric Rothstein Morris, "Understanding LSTM-a tutorial into Long Short-Term Memory Recurrent Neural Networks.," *arXiv preprint arXiv,* p. 1909.09586, 2019.

[37] Afshine Amidi and Shervine Amidi, "Recurrent Neural Networks cheatsheet," [Online]. Available: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks.

[38] Wang, Haohan, and Bhiksha Raj, "On the origin of deep learning," *arXiv preprint arXiv:1702.07800,* 2017.

[39] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," *In Advances in neural information processing systems,* pp. 3111-3119, 2013.

[40] S. Cavallari, V. W. Zheng, H. Cai, K. C.-C. Chang, and E.Cambria, "Learning community embedding with community detection and node embedding on graphs," in *the 26th ACM International Conference on Information and Knowledge Management, (CIKM '17)*, 2017.

[41] Jonathan Chang and David M Blei, "Relational topic models for document networks," in *International Conference on Artificial Intelligence and Statistics*, 2009.

[42] A. Reihanian, M. R. Feizi-Derakhshi, and H. S. Aghdasi, "Overlapping community detection in rating-based social networks through analyzing topics, ratings and links," in *Pattern Recognition*, 2018.

[43] Laurens Van Der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research,* Vol. 9, pp. 2579–2605, 2008.

[44] Günce Keziban Orman, Vincent Labatut, and Hocine Cherifi, "Qualitative Comparison of Community Detection Algorithms," in *International conference on digital information and communication technology and its applications.s. Springer*, Berlin, 2011.

[45] Batagelj, Vladimir, " Efficient algorithms for citation network analysis," *arXiv preprint cs/0309023,* 2003.

[46] Garfield E, ": From Computational Linguistics to Algorithmic Historiography," in *at the Symposium in Honor of Casimir Borkowski at the University of Pittsburgh School of Information Sciences,*, Pittsburgh , 2001.

[47] Waltman, Ludo, and Erjia Yan., "PageRank-related methods for analyzing citation networks," *In Measuring scholarly impact, pp. 83-100. Springer, Cham*, 2014.

[48] Mariani, Manuel Sebastian, Matúš Medo, and François Lafond., "Early identification of important patents: Design and validation of citation network metrics," *Technological forecasting and social change*

, vol. 146, pp. 644-654., 2019.

[49] Baggio, Jacopo A., Katrina Brown, and Denis Hellebrandt., "Boundary object or bridging concept? A citation network analysis of resilience.," *Ecology and Society* , Vol. 20, No. 2, 2015.

[50] Zarezade, M., E. Nourani, and Asgarali Bouyer. "Community detection using a new node scoring and synchronous label updating of boundary nodes in social networks." Journal of AI and Data Mining 8, no. 2, pp. 201-212., 2020.

.

# ارائه یک مدل نمایش شبکه ژرف تلفیقی جهت آشکارسازی هم‌پویه‌های پژوهشگران

**عاطفه ترکمن[۱]، کامبیز بدیع[۲]، افشین سلاجقه [۱،*]، محمد هادی بکایی[۲] و سید فرشاد فاطمی[۳]**

[۱] گروه کامپیوتر، دانشگاه آزاد اسلامی واحد تهران جنوب، تهران، ایران.

[۲] پژوهشکده فناوری اطلاعات، پژوهشگاه ارتباطات و فناوری اطلاعات، تهران، ایران.

[۳] دانشکده مدیریت و اقتصاد، دانشگاه صنعتی شریف، تهران، ایران.

ارسال ۲۰۲۱/۱۰/۰۹؛ بازنگری ۲۰۲۲/۰۱/۰۶؛ پذیرش ۲۰۲۲/۰۲/۰۷

**چکیده:**

در سال‌های اخیر، بازنمایی شبکه، نظر بسیاری از فعالیت‌های تحقیقاتی را به خود جلب کرده است که عمدتاً بر بازنمایی گره‌ها در یک بردار متراکم کم-بُعد تمرکز دارند. طبق مطالعات انجام شده، برخی از روش‌های بازنمایی شبکه فقط بر ساختار گره تمرکز دارند و برخی دیگر افزون بر آن، اطلاعات محتوایی درون گره‌ها را درنظر می‌گیرند. در این مقاله، مدل بازنمایی شبکه عصبی ژرف تلفیقی (HDNR) که از یک معماری سه‌بخشی شبکه ژرف بهره می‌برد، پیشنهاد می‌شود که ساختار گره‌ها و اطلاعات محتوایی را به‌طور توامان برای بازنشانی شبکه درنظر می‌گیرد. علاوه بر این، سبک نگارش نویسندگان نیز به‌عنوان یک ویژگی قابل توجه برای اطلاعات محتوایی گره‌ها در نظر گرفته شده است. شایان ذکر است، با توجه به کاربرد موفقیت‌آمیز یادگیری ژرف در پردازش زبان طبیعی (NLP)، مدل پیشنهادی بر پایه یک روش گام‌زنی تصادفی ژرف به‌منظور بهره‌برداری از ساختارهای بین گره‌ای، دو روش پیش‌بینی توالی ژرف برای استخراج اطلاعات محتوایی گره‌ها بنا نهاده شده است. بردارهای بازنمایی ای که با این روش تولید می‌شوند، یکدیگر را، در مسیر یادگیری بهینه بازنمایی گره‌ها، شناسایی ویژگی‌های اطلاعاتی بیشتر و در نهایت تشخیص بهتر هم‌پویه‌ها، تقویت می‌نمایند. نتایج تجربی به‌دست‌آمده، کارایی این مدل را برای بازنمایی شبکه در مقایسه با سایر روش‌های پایه تأیید می‌کند.

**کلمات کلیدی:** شبکه‌های پیچیده، بازنمایی شبکه، یادگیری ژرف، شبکه استنادی، شبکه عصبی بازگشتی، حافظه کوتاه مدت طولانی، پردازش زبان طبیعی، آشکارسازی هم‌پویه.

Recently, network representation has attracted many research works mostly concentrating on representing the nodes in a dense low-dimensional vector. There exist some network embedding methods focusing only on the node structure, and some others considering the content information within the nodes. In this paper, we propose a hybrid deep network representation (HDNR) model that uses a triplet deep network architecture that considers both the node structure and the content information for network representation. In addition, the author's writing style is also considered as a significant feature in the node content information. Due to the successful application of deep learning in natural language processing (NLP), our model utilizes a deep random walk method in order to exploit the inter-node structures and two deep sequence prediction methods to extract the nodes' content information. The embedding vectors generated in this manner are shown to have the ability to boost each other for learning the optimal node representation, detecting more informative features, and ultimately a better community detection. The experimental results obtained confirm the efficiency of this model for network representation compared to the other baseline methods.