SRTTU

Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

JECEI

Research paper

Centrality and Latent Semantic Feature Random Walk (CSRW) in Large Network Embedding

***M. Taherparvar*[1], *F. Ahmadi Abkenari* [1,2,]*, *P. Bayat*[1]**

[1]Department of Computer Engineering, Rasht Branch, Islamic Azad University, Rasht, Iran.

[2]Faculty of Computer Engineering and Information Technology, Payam Noor University, Tehran, Iran.

Article Info

Article History:
Received: 03 October 2022
Reviewed: 10 November 2022
Revised: 30 December 2022
Accepted 21 January 2023

Keywords:
BTM topic modelling
Centrality criteria
Deep learning
Network embedding
Social network analysis

*Corresponding Author's Email Address:
Fateme.Abkenari@pnu.ac.ir

Abstract

Background and Objectives: Embedding social networks has attracted researchers' attention so far. The aim of network embedding is to learn a low-dimensional representation of each network vertex while maintaining the structure and characteristics of the network. Most of these existing network embedding methods focus on only preserving the structure of networks, but they mostly ignore the semantic and centrality-based information. Moreover, the vertices selection has been done blindly (greedy) in the existing methods.
Methods: In this paper, a comprehensive algorithm entitled CSRW stands for centrality, and a semantic-based random walk is proposed for the network embedding process based on the main criteria of the centrality concept as well as the semantic impact of the textual information of each vertex and considering the impact of neighboring nodes. in CSRW, textual analysis based on the BTM topic modelling approach is investigated and the final display is performed using the Skip-Gram model in the network.
Results: The conducted experiments have shown the robustness of the proposed method of this paper in comparison to other existing classical approaches such as DeepWalk, CARE, CONE, COANE, and DCB in terms of vertex classification, and link prediction. And in the criterion of link prediction in a Subgraph with 5000 members, an accuracy of 0.91 has been reached for the criterion of closeness centrality and is better than other methods.
Conclusion: The CSRW algorithm is scalable and has achieved higher accuracy on larger datasets.

Introduction

Life without social media is unimaginable in today's world. The media in different forms play a prominent role in our vitality. In fact, media is one of the influential pillars of information constitution and attainment. In the meantime, social networks have become the foundation of borderless communication while their remarkable impacts on people's lives and continuance are immeasurable. One source of big data with its three mainstream characteristics of volume, velocity, and variety originates from social networks. Moreover, in the shadow of social behaviors, these data may include human collaborations, interactions, and communications that are categorized as highly nonlinear and complex problems.

A graph as a structure of vertices and edges is an adequate tool for representing high-dimensional data such as networks of human collaborations, wireless sensors, research paper citations, etc. The scale of complex networks ranges from hundreds to billions of vertices, making the efficient network analysis process

Doi: 10.22061/jecei.2023.9279.600

challenging. A very suitable solution for this problem is the concept of network embedding, in which each node is mapped to a low-dimensional vector while the global and local characteristics of the network will be maintained [1], [2].

Also, hidden information related to vertices should be extracted based on each network's characteristics. This information could indicate the specifications that highly affect the correct analysis of the network. For example, the neighborhood information and semantics of vertices (in different orders) could help to properly advance the random walk in the network embedding process. Therefore, the vector representations of nodes could be employed in tasks such as network analysis and classification, node clustering, and link prediction [3]-[5]. Considering the high potential of network embedding, there are two main challenges in this domain. First, the scale of real-world networks is big data. Hence, the learning task may take months or fail. Second, network data are often complex and high-dimensional, which makes it very challenging to design a suitable model with the aim of preserving the network structure. The network embedding process consists of two steps: First, sampling the network data as a corpus and then embedding it using the Word2vec approach [6]. Meanwhile, DeepWalk is a pioneering method among network embedding approaches [1].

In networks that have been constructed based on real-world scenarios, most vertices have a low degree with only a few nodes of a high degree. So, criteria such as the degree or different centrality approaches of each node alone are not optimal indicators for selecting a vertex in the random walk process.

Recently, many efforts have been devoted to the development of network embedding algorithms. Early research mainly emphasized reducing the dimensions of the network based on the feature extraction process. However, the high cost of calculating the adjacency matrix of large-scale networks is a major challenge in these approaches. Recently, inspired by the success of Word2Vec, interesting research endeavors have been conducted in network embedding frameworks that result in DeepWalk [1], Node2vec [2], and LINE [7] approaches. These classical methods have shown promising performance in many machine learning applications.

The DeepWalk method creates random walks for each vertex and uses them as background information to learn the representations of the vertices [1]. Node2Vec extends DeepWalk by utilizing two predefined parameters to control the random walk method, which is a trade-off between breadth-first and depth-first search traversal approaches [2]. DeepWalk and Node2Vec face the problem of insufficient sampling in dense networks, so some local patterns are not reflected in these perspectives. In addition, some research focus on the examination and extraction of vertices' features, such as text [8]or labels [10].

Our proposed approach of centrality and semantic-aware random walk (CSRW) in this paper for network embedding, employs one criterion among the set of degree, degree centrality, load centrality, closeness centrality, and eigenvector centrality. After loading the corresponding value of each node, the random walk generation process will begin in the following manner: The first node is randomly selected to start the process then the next node is selected from the neighbours of the previous vertex in such a way that the average selected criterion of that node and the nodes of its next hops is more optimal than the rest of the neighboring nodes (the number of hops is set from one to four).

As depicted in Fig. 1, if we consider the number of orders to be equal to 4, node No. 3 has a higher selection priority than nodes No. 2 and No. 4 in terms of the average degree criterion.

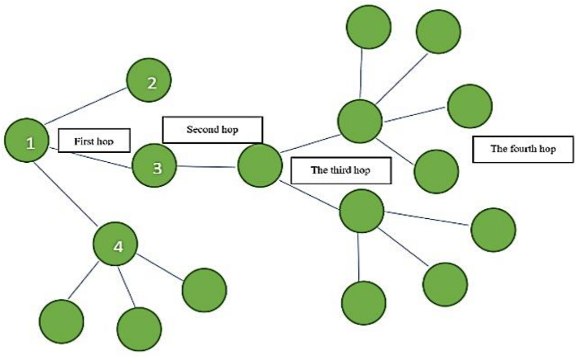


Fig. 1: The process of selecting nodes in the random walk process.

Therefore, a random walk is generated for all nodes in the network graph. Finally, the generated walk sequences are sent to the Skip-Gram model in the form of text strings to produce display vectors. Simultaneous with the random walk generation process, the textual content of the adjacent vertices is collected in a single document per node in the sequence of random walks. Then, the pattern in this document is extracted using a topic modeling approach in a word pairs fashion. Finally, the output of the context analysis constituent is linked with the display vector obtained from the Skip-Gram model.

The contributions and innovations of this paper are as follows:

- In this paper, a centrality and semantic-based random walk process entitled CSRW is proposed for the network embedding process with the aim of overcoming the problem of blind selection of vertices.
- The novelty of the proposed CSRW approach is that it focused on two dimensions, different criteria, and the semantic information and connection of vertices.

- The proposed approach is scalable. The increase and decrease in the size of the input network will not affect the effectiveness of the CSRW approach.
- Regarding the fact that networks from real-world scenarios have low degree vertices as a majority and high degree vertices as a minority, the greedy approaches for node selection cannot reflect the local and global properties of the network. To overcome this problem, the CSRW approach is proposed for the vertex selection process considering the effect of the evaluation-based criteria scores of neighboring nodes. So, another innovation of the algorithm is the better reflection of the local and global characteristics of the nodes in the network.

The method presented in this paper has been evaluated in several real-world networks such as reference networks.

The experimental results have proven that the proposed algorithm outperforms other community and text-based approaches in the fields of vertex classification, and link prediction.

The rest of this paper is organized as follows. Section 2 briefly presents previous research and algorithms related to the random walk strategies in network and topic modeling approaches. In section 3, the proposed algorithm of this paper is discussed with formal expressions along with mappings and explanations. Section 4 contains the results of the conducted experiments for verification of the proposed methods' effectiveness. Moreover, the parameter sensitivity of the proposed approach is analyzed. Finally, section 5 summarizes the discussion on the proposed method's framework, conclusions, and future work.

Related Work

Feature learning has been widely used in different filed as computer vision [10], [11] and natural language processing [12], [13]. With the development of the Internet, large amount of data are produced by complex networks. Unstructured data is a challenging issue in network embedding, and many methods are proposed to learn features of network.

In recent years, deep learning methods have been employed as an alternative to feature vector-based learning. These methods have used deep learning to learn representation vectors. They generate random walks with the help of different network search strategies and provide the input as contextual information to the Skip-Gram model [1].

Perozzi et al. (2014) confirmed the similarity between vertices in the random walk sequence in terms of the words in their contexts.

They proposed a DeepWalk model that uses Skip-Gram architecture to extract feature vectors from a sequence of random walks [1]. DeepWalk was the first method that used the Skip-Gram model to generate feature vectors. Although the DeepWalk method has shown good performance in vertex classification, because of not considering the neighborhood information of higher orders, it could not maintain the global structure of the network.

Tang et al. (2015) proposed the LINE algorithm in which they employ first and second-order neighborhoods together with preserving local information to learn node representations.

In the LINE method, two independent functions are defined for the first and second-order neighborhoods. The LINE method and the DeepWalk are unable to learn the representation vector for boundary nodes in the network [7].

Grover et al. (2016) proposed Node2Vec random walks based on strategies such as depth-first and breadth-first traversal approaches. This algorithm considers only the second-order neighborhoods and cannot reach the nodes whose distance from the starting node of the random walk is more than two. Therefore, like the DeepWalk method, it cannot maintain the global structure of the network [2].

Chen et al. (2019) presented the lateral information network embedding, which defined a semantic neighborhood to model the shape of each node, then applied random walks to explore this neighborhood [14]. Wang et al. (2016) proposed a deep model with a semi-supervised architecture entitled SDNE, which maps data to a nonlinear hidden space and can simultaneously optimize first-order and second-order neighborhoods [15].

Community detection in the network is one of the common methods in network embedding. Li et al. (2019) presented a network embedding method based on evolutionary algorithms that can maintain the neighborhood and communities of vertices in the network by optimizing a multi-objective function [16]. Chen et al. (2016) proposed a method with valuable group information for large-scale networks by considering the internal structures of groups and the information between them [17]. Kikha et al. (2018) presented an algorithm called CARE, which uses the Louvain community detection method to detect the communities of nodes in the network and construct a sequence of random walks. The CARE algorithm employs the Louvain method to discover communities [18]. Wang et al. (2020) proposed CANE algorithm, which describes the embedding of community-aware network through adversarial training. The CANE method minimizes the community assignment error with the aim of improving network embedding [19].

Criteria associated with vertices in the network have many usages in network embedding. Shi et al. (2019) proposed a network propagation embedding method with the aim of overcoming limitations such as the tendency to select high-degree nodes [20].

The drawback of their research is neglecting the global structure of very complex networks in the random walk. Chen et al. (2019) presented a generalizable model that uses both edge and node centrality information to learn low-dimensional vector representations that can maintain different vertex centrality information [21]. Zhao et al. (2019) proposed an integrated framework for social and behavioral recommendations with network embedding and introduced a joint network embedding approach as a pre-training step for hidden user representations [22].

Li et al. (2019) represented an unsupervised network embedding model for encoding edge relationship information, thus feature representation of vertices can be further captured [23].

The aim of topic models is to utilize observed text in order to infer hidden topic distribution. Some researchers have used topic models regarding authors' collaboration networks to infer the research community [24]. Mai et al. (2008) have presented a general solution of text mining with a network-based structure entitled NetPLSA for topic optimization in the network [25].

Wu et al. (2019) proposed a multi-task dual attention LSTM model to learn network representations for specific applications [26]. The model can capture structure, content and label information, then adjust vertex representations according to the downstream task. Yuan et al. (2019) proposed an algorithm called COANE, which uses the LDA topic modeling approach to detect the community of vertices in the network and construct a sequence of random walks [27].

In our other paper (2022), the DCB algorithm is proposed as a network attribute embedding framework that includes the contextual information of vertices in the network embedding process. In this research, the topic modeling based on word pairs has been utilized to investigate the relationship and semantic analysis of nodes [28]. Chen et al. (2022) proposed semantic feature-aware embedding via optimized random walk and paragraph2vec. By using textual semantics instead of contextual semantics, this method has been able to achieve higher accuracy in complex networks than the deep walking method [29].

In the recent works, random walk with the aim of embedding the network usually uses a single criterion, for example, random selection, centrality criterion, semantic analysis of texts, communities, etc. In the methods presented in this article to improve the quality of network embedding, two parameters of centrality criteria and semantic features are used.

In this paper, a random walk generation based on centrality and semantic information (CSRW) is proposed for network embedding. This method consists of two sections.

The first part includes the selection of nodes based on the criteria of centrality and the semantic influence of nodes in the network embedding. As mentioned before, the greedy selection of the highest degree or the largest centrality criterion for each node cannot reflect the properties of the network locally and globally. To solve this problem, a new approach is presented in this paper in the selection of vertices.

In the random walk generation process, a node is selected with the optimal value of the average evaluation criterion of itself and its neighboring nodes in the next order (in one to four hops).

In the next section, we will describe the proposed algorithm of this research:

Methodology

In this section, we will present a detailed description of the proposed CSRW algorithm and a brief overview of the existing node evaluation criteria.

In this paper, an algorithm called CSRW is proposed that generates a random walk based on the centrality and semantic information in network embedding, which is based on the average of the degree criteria or different centralities for the neighboring nodes of each vertex (in different orders) and semantic analysis of feature vectors of network nodes.

Among the important features of social networks that are able to maintain the local and global structure, we consider the average information of different degree or centrality criteria for the neighboring nodes to a vertex (in different hops) and the semantic analysis of textual features specific to each node as the dominant ones.

Suppose $\mathrm{G} = (\mathrm{V}, \mathrm{E}, \mathrm{T})$ is a feature graph where V is a set of vertices; $\mathrm{E} \subseteq \mathrm{V} \times \mathrm{V}$edges represent the relationships between the vertices and T represents the text content of the vertices; In particular, the textual information of each vertex $\mathrm{v} \in \mathrm{V}$ is related to the word sequence $\mathrm{T}_{\mathrm{v}} = (\mathrm{W}_1, \mathrm{W}_2, \dots \dots ., \mathrm{W}_{\mathrm{n}})$ where $\mathrm{n}_{\mathrm{v}} = |\mathrm{T}_{\mathrm{v}}|$.

Network embedding tries to create a matrix of features with low dimensions called $\phi \in \mathrm{R}^{|\mathrm{v}| \times \mathrm{d}}$. $\mathrm{d} \ll |\mathrm{V}|$ defines the dimensions of the hidden representation space d is less than $|\mathrm{v}|$.

The Skip-Gram model has been used to obtain the best mapping performance of the Φ function.

A. Nodes Evaluation Criteria

Most of the classical methods for network embedding are focused on maintaining the network structure and generally do not pay proper attention to utilizing and collecting information related to the criteria and centrality nuggets of different vertices. Metrics with the ability to measure the importance of each node separately are practical in many applications. Of course, a greedy approach based on different criteria in selecting nodes is not able to reflect the characteristics of the network properly. In this paper, the main focus is on how to use criteria such as degree and different centrality-based metrics.

So in this domain, we employ various criteria such as degree criteria, degree centrality, closeness centrality [30], eigenvector centrality [31], and load centrality [32]. Our proposed CSRW model in this paper is able to maintain all mentioned centrality-based measures in its structure. In this paper, representation learning aims to preserve network structure, centrality information, and semantic connection between nodes. To this aim, we proposed a general model that employs various centrality criteria.

We briefly explain each criterion in continue:

I) Degree of node criterion

Simply, the number of connections of a node to other vertices in the graph is called the degree of that node. The centrality criterion means how important a node is in a social network. In the graph discussion, there are different types of centrality-based metrics, which can be used to identify impactful nodes in the network.

II) Degree centrality criterion

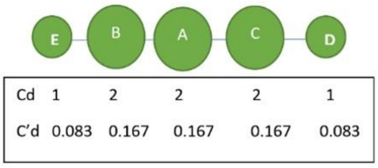
In a network graph, degree centrality is measured by the total number of direct edges with other nodes according to the basic formula in (1) [30]:

$$C_d(N_i) = \sum_{j=1}^{n} X_{ij} (i \neq j) \tag{1}$$

X_{ij} is a link between node i and j. with the increase in the size of the networks, in order to reduce the impact of the network size on this centrality criterion, formula (1) was standardized as formula (2) [30]:

$$C'_d(N_i) = \frac{\sum_{j=1}^{n} X_{ij}}{(n-1)(n-2)} (i \neq j) \tag{2}$$

$\sum_{j=1}^{n} X_{ij}$ represents the number of direct edges connected to node *N*, and *n* is equal to the total number of nodes in the network graph. Based on Fig. 2, the number of direct edges connected to node *A* is equal to 2, so, the value of C_d is equal to 2, and after standardization, the value of C'_d will be equal to 0.167 [30].



	E	B	A	C	D
Cd	1	2	2	2	1
C'd	0.083	0.167	0.167	0.167	0.083

Fig. 2: An example graph representing the degree centrality metric [30].

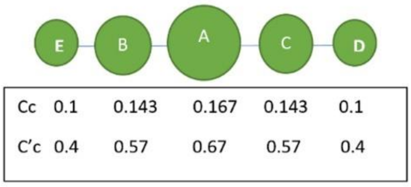
III) Closeness centrality criterion

In a graph, the closeness centrality of a vertex is the average length of the shortest path between the node and all other vertices in the network graph. Therefore, the more central a node is, the closer it is to other vertices.

Equation (3) is the basic equation of closeness centrality, which is equal to the total number of steps from node *N* to all nodes of the network [30].

$$C_c(N_i) = \frac{1}{[\sum_{j=a}^{n} d(N_i,N_j)]} (i \neq j) \tag{3}$$

According to Fig. 2, node A is located next to nodes B and C. Therefore, the distance of node A to these two nodes is equal to one order, and its distance to nodes E and D are equal to two orders. Hence, the closeness centrality value for node A is equal to $\mathrm{C_c(N_i)} = \frac{1}{1+1+2+2} \approx 0.167$, $\mathrm{C'_c} \approx 0.67$. The results for the rest of the nodes are shown in to Fig. 3 [30]:



	E	B	A	C	D
Cc	0.1	0.143	0.167	0.143	0.1
C'c	0.4	0.57	0.67	0.57	0.4

Fig. 3: An example graph representing the closeness centrality metric [30].

IV) Eigenvector centrality criterion

The eigenvector is the largest eigenvalue of an adjacency matrix, which can be a good measure of network centrality metric calculation. Unlike the degree-based metrics, which weight each edge equally, the eigenvector weights connections based on their centrality. The centrality of the eigenvector can be calculated as the weighted sum of not only direct edges but also indirect connections of any length.

Formula (4) illustrates the eigenvector centrality metric. It describes the centrality of the eigenvector *x* in two ways, as a matrix equation, and as a summation. The centrality of a node is proportional to the sum of the centralities of the nodes connected to it. λ is the largest eigenvalue of *A* and *n* is the number of vertices [31]:

$$A_x = \lambda_x, \lambda_{x_i} = \sum_{j=1}^{n} a_{ig} x_{ij}, i = 1, \dots \dots, n \tag{4}$$

V) Load centrality criterion

The measure of load centrality for each node is the fraction of the shortest paths that pass through that vertex. This criterion is obtained based on formula (5):

$$\mathrm{LC(v)} = \sum_{\mathrm{s,d \in V}} \theta_{\mathrm{s,d}}(\mathrm{v}) \tag{5}$$

The load centrality criterion employs an algorithm to calculate paths with the minimum weight between pairs of nodes (s, d). The variable $\theta_{s,d}$ is the value sent from node s to node d. It is assumed that this value is always transmitted to the next node with the lowest value. $\theta_{s,d}(v)$ is the total amount of value sent from node v. It is normally assumed that $s \neq d$ and $d \neq v$ [32].

B. Centrality and Semantic-Based Random Walk for Network Embedding

In this paper, the random walk algorithm based on the centrality and semantic information entitled CSRW is proposed for the network embedding process.

The diagram of the CSRW template is illustrated in Fig. 4. And the flowchart of the random walk is illustrated in Fig. 5 .

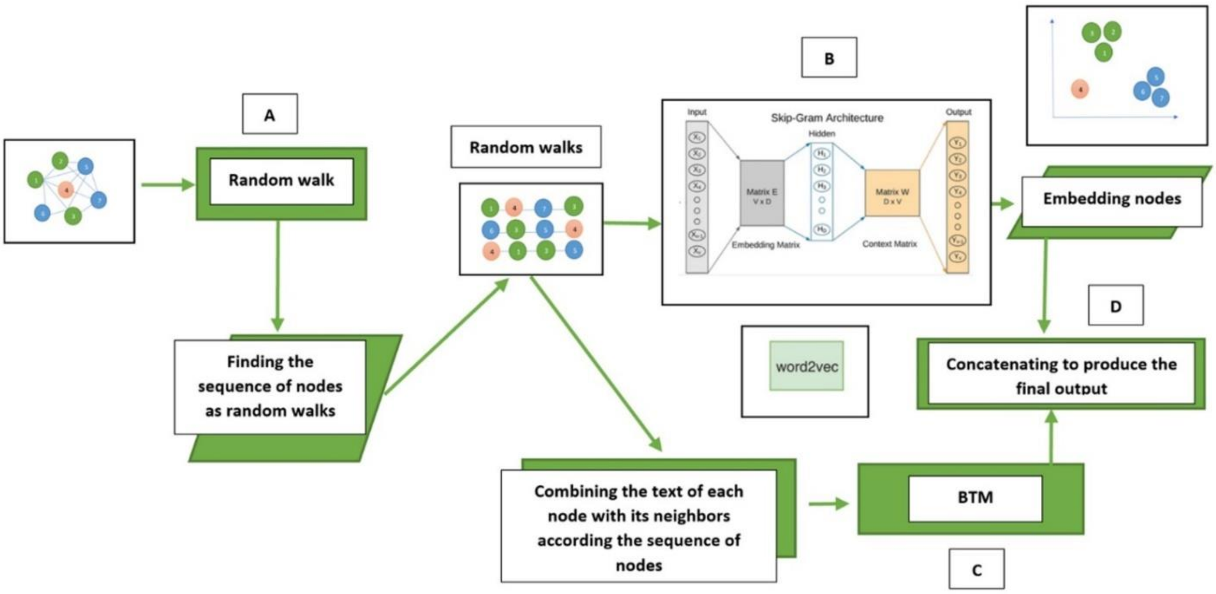

A
Random walk
Finding the sequence of nodes as random walks
Random walks
B
Skip-Gram Architecture
Input
Hidden
Output
Matrix E
V x D
Matrix W
D x V
Embedding Matrix
Context Matrix
word2vec
Embedding nodes
D
Concatenating to produce the final output
Combining the text of each node with its neighbors according the sequence of nodes
BTM
C

Fig. 4: Diagram of the proposed method of this paper (CSRW).

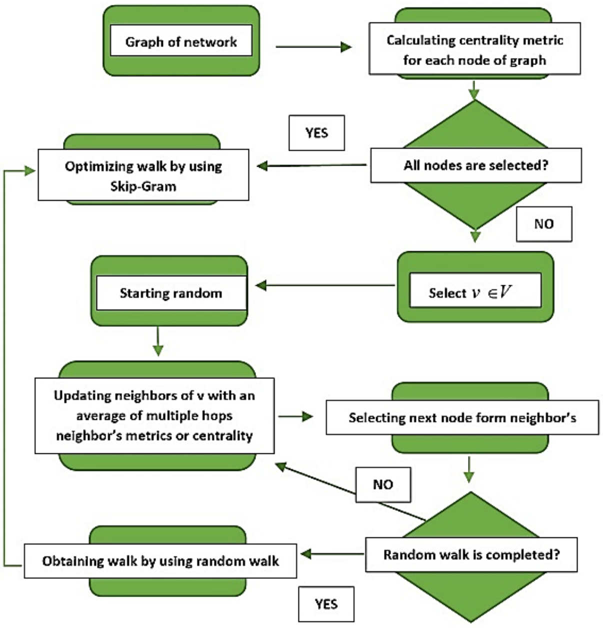

Graph of network
Calculating centrality metric for each node of graph
YES
All nodes are selected?
Optimizing walk by using Skip-Gram
NO
Select $v \in V$
Starting random
Updating neighbors of v with an average of multiple hops neighbor's metrics or centrality
Selecting next node form neighbor's
NO
Random walk is completed?
Obtaining walk by using random walk
YES

Fig. 5: Random walk flowchart process in CSRW method.

The diagram of CSRW method is illustrated in Fig. 4. This diagram composes four main parts:

- Part A: The aim of the random walk part is to find a sequence of nodes in such a way that the local and global characteristics of the nodes in the network are preserved. The random walk used in the CSRW method is shown in the flowchart in Fig. 5.
- Part B: A sequence of vertices $\mathrm{s} = (\mathrm{v}_1, \mathrm{v}_2, \dots \dots \dots \mathrm{v}_{|s|})$ obtained by a random walk through the network is considered a word sequence and each vertex in the sequence is considered a word. In the next step, the CSRW can obtain embedding nodes using the Skip-Gram model, which aims to maximize the average log of observing a vertex:

$$max_{\emptyset} \frac{1}{|s|} \sum_{i=1}^{|s|} log \ P_r(\{v_{i-w}, \dots, v_{i+1}, \dots, v_{i+w}\}|v_i) \tag{6}$$

- Part C: The BTM topic model is used for the contextual analysis of the collected documents from neighbors. In this part, the text content of the neighboring nodes is combined and analyzed. Latent semantic features are extracted in this section.
- Part D: In this part the embedding nodes and the semantic features are concatenating to produce final output.

In this section, the centrality and semantic-based random walk algorithm for network embedding (CSRW) are presented. The main purpose of this algorithm is to create representation vectors for network vertices. In

CSRW, the evaluation of nodes as the next selected vertex in the random walk sequence is based on the average of criteria such as degree, closeness, eigenvector and load centrality-based metrics. The evaluation is in such a way that each node is verified by the average value of its criterion and its neighboring nodes in the next order (one to four hops).

Among the evaluated neighboring nodes, the vertex that has a more optimal value with a higher priority for selection is chosen as the next vertex in the random walk process by employing the random selection of the Roulette Wheel genetic algorithm. The proposed method in this paper solves the problem of blind selection of nodes.

In this scenario, a node is selected that has a higher right to be chosen than other neighboring vertices in the future random walk sequence. Algorithm 1 depicts the steps of CSRW.

```
Algorithm 1: Framework of CSRW
Input: graph: $\mathbf{G}(\mathbf{V},\mathbf{E},\mathbf{T})$; Window size: w;
       representation dimension: d;
       walks per vertex: ɣ; walk length: l.
Output: matrix of network representations: $\phi \in \mathbf{R}^{|V|\times d}$
1: B= Benchmark measurement (G)
2: Sample $\phi$ from $u^{|v|\times d}$
3: for $i = 0$ to ɣ do
4:    $\vartheta = \mathbf{Shuffle}(V)$
5:    for each vertex $v \in \vartheta$ do
6:       $W_v = \mathbf{Random\ Walk}(G, v, B, l)$
7:       $D_v = \mathbf{ContextAggregation}(G,\ v,\ B)$
8:       $\mathbf{SkipGram}(\phi, S_v, w)$
9:    end for
10:   $Pr_t = \mathbf{BTM}_t(D)$
11: end for
12: $\phi = \phi \oplus Pr_t$
13: return $\phi$
```

Fig. 6: CSRW algorithm.

As shown in Fig. 6, in line 1, the desired criterion is measured for all network nodes. In line 2, Before learning the representation vectors for the network nodes, the *U* matrix is randomly generated to produce the nodes' representation vectors in the next section. The algorithm is now able to learn the final representation vectors in lines 3 to 11.

Before repeating the grid nodes, in line 4, at the start of each pass, the vertices are shuffled to prevent the node visiting order in the ϕ . The core task of the network embedding is done in line 6, where a random walk is generated for the selected vertex. Line 7 shows the semantic analysis section of the textual information of the nodes, which is explained in algorithm 3. In line 10, we aggregate text feature of a vertex to that's of its neighbors as D and input them into the text-based BTM model. Finally, the generated paths and the results of the node's semantic analysis are used to update the node's presentation in line 12.

Fig. 7 illustrates the *Random Walk* algorithm for the *CSRW* algorithm. A random walk starting at node *v* is denoted by W_v A random walk sequence for node *v* can be represented by random variables such as $W_v^1, W_v^2, \dots, W_v^k$.

To create a customized random walk starting from node *v*, first, all the neighbors of that node are extracted. Then a random variable *r* between 0 and 1 is created. α is random variable to select from neighbors.

If random variable is less than α then the criteria related to the neighboring nodes of the current node are checked.

This section in the random walk can have four types: In each type, the average criterion of the nodes of that order neighbors is examined for example in the second type, the average criterion of neighbor nodes up to the second order is examined.

Finally, a node is randomly selected from the final list obtained for the neighbors of the current vertex using the genetic algorithm of Roulette Wheel method.

Based on the algorithm illustrated in Fig. 7, random walks are generated independently. Hence, the current algorithm can be parallelized to speed up the embedding process.

In addition, if some new nodes are added or removed from the network, the random walk is calculated only for the new vertices.

The contextual analysis of the nodes in Fig. 6 are in performed in lines 7 and 10. In line 7, $D_v =$ $\mathrm{ContextAggregation}(G, v)$in which node v and network graph G are used in the text aggregation section to collect text documents related to node v and its first-order neighbors.

The text aggregation section is shown in Fig. 8 entitled Context Aggregation Algorithm. In Fig. 6, line 10, $P_{r_t} = \mathrm{BTM}(D)$ of the BTM topic model is used for the contextual analysis of the collected documents D_v.

The text type of nodes in social networks is short. In the Context Aggregation algorithm, the text content of the neighboring nodes is combined and analyzed. In another paper (2021), it has been shown that the BTM model has a better performance in relation to short texts than other topic models [33].

Finally, in Fig. 6, line 12, the $\emptyset = \emptyset \oplus P_{r_t}$ vectors obtained from the random walk analysis and the contextual analysis of the nodes are combined to produce the final $\emptyset$ vectors. Fig. 8 shows the process of summarizing the text. To reduce the deviation between the posterior population distribution and the actual population distribution, it should be considered that the length of a document should not be less than the total number of documents [34].

$$v.\log|v| \ll length(D_v) \qquad (7)$$

Line 5 of Fig. 8 shows how to assign probabilities to neighboring vertices.

The selection of the next vertices is based on the selection strategy of the Roulette Wheel genetic algorithm.

Algorithm 2: Random Walk

```
Input: graph:$\mathbf{G(V,E,T)}$; Source node of RW: $\boldsymbol{v_i}$;
       Benchmark measurement of graph: B;
       walk length: l; Random variable to select from
       neighbours: $\boldsymbol{\alpha}$.
Output: A path with max length l: Random Walk
initialize RW with $v_i$
While length(path)<l
  if the current node has neighbours
    if ((random (0,1) = r) < $\alpha$)
       For a node in current_ neighbours:
         For nodes in hop_1's node:
           list= Benchmark measurement list of
               nodes
           For nodes in hop_2's node:
             list= Benchmark measurement list of
                 nodes
             For nodes in hop_3's node:
               list= Benchmark measurement list
                   of nodes
               For nodes in hop_4's node:
                 list= Benchmark measurement
                     List of nodes
       Calculate the average of the list and add it
        into score list.
       next node for RW= Random from members
                         Sorted score list
    else:
       Select another $v_j$ at random from members
       $v_j$'s current_ neighbors
  else:
     backtrack in the path and select the last
     node which has neighbors that are in the
     path
end While.
```

Fig. 7: Random walk for CSRW algorithm.

Algorithm 3: Context Aggregation

```
Input: graph:$\mathbf{G(V,E,T)}$; Source node of RW $\mathbf{v_i}$;
       Benchmark measurement of graph: B;
Output: the contextual text information: $\mathbf{D_v}$
1: Initialize $D_v$ with $T_v$
2: While length ($D_v$ )< ɣ. log |v| do
3:   if current vertex has neighbors, then
4:     for each neighbor vertex u of v do
5          list= Benchmark measurement list of nodes
6:     end for
7:     select a vertex u list based on Roulette Wheel
8:     $D_v = D_v \oplus T_u$
8:   else
9:     $D_v = D_v \oplus T_v$
10:  end if
11: end while
13: return $D_v$
```

Fig. 8: The algorithm combines the semantic of each vertex with its neighbors.

Result and Discussion

In this section, methods, experimental data sets, and parameter settings will be described. Then, the proposed algorithm of this paper is evaluated in two supervised learning tasks such as vertex classification and link prediction, and will be compared with other existing approaches.

As described in the literature review section, DeepWalk is an advanced network embedding algorithm that uses natural language processing for network embedding. CARE is an algorithm for community-aware network embedding and obtains community-related information using the Louvain method, and finally, random walks are converted into low-dimensional representation vectors using the Skip-Gram model. Also, CONE and COANE are algorithms related to network embedding that obtain community information with topic models and convert the generated random walk into low-dimensional representation vectors by using the Skip-Gram method.

DCB algorithm is an embedding of network attributes that can include the information and content of vertices' text in the network embedding process. This paper used the BTM topic model to examine the relationship and semantic analysis of nodes. In this section, we will compare our new method of CSRW with the classic methods of DeepWalk, CARE, CONE, COANE, and DCB on the following datasets of Cora and DBLP.

I) Data set

The *Cora* dataset contains 2708 machine learning articles from 7 classes and 5429 edges among the articles.

Each vertex represents an article and the citation relationships between documents form a typical complex network [35].

DBLP V12 contains 4 million articles and 45 million edges between them, and the date of this dataset is 09/04/2020. In this paper, two subgraphs with the number of 2000 and 5000 nodes are used for implementation from DBLP dataset [36].

Also, the content of the title of each article is used as the feature information. In *Cora* data set, the titles extracted from the main dataset had missing values, and in this regard, the link of the articles was used as a replacement. The characteristics of the dataset are shown in Table 1.

Table 1: The dataset used for the experiments

	Data set	Nodes	Edges	Labels
(1)	Cora	2708	5429	7
(2)	DBLP_2000	2000	4013	4
(3)	DBLP_5000	5000	11587	4

The display vector dimension is set to d = 128 for all datasets above. For DeepWalk, the number of walks is set to 20, the walk length is set to 20, and the window size w is set to 10.

In order to provide a fair comparison, the parameter settings used for DCB, CARE, CONE, COANE, and CSRW correspond to the values used for the DeepWalk. In all the above cases, the value of the variable k, the number of topics, is considered equal to 14.

II) Comparison based on link prediction metric

Link prediction is a task to estimate the probability of links between nodes in a graph. It is a supervised and semi-supervised learning task. The model is trained using a subset of link that have truth labels. For predicting link existence, the truth may just be whether the edge exists in the original data, rather than a separate label. Link prediction can also be done as a downstream task from node representation learning, by combining node embedding vectors for the source and target nodes of the edge and training a supervised or semi-supervised classifier against the result.

A standard evaluation criterion, the area under the curve (AUC), is adopted here, which indicates the probability that potentially connected vertices are more similar than unrelated ones.

The CSRW algorithm has been implemented in four different versions from one to four hops and each of them is implemented based on the criteria of degree, degree centrality, closeness centrality, eigenvector centrality and load centrality. In the implementation of the first type, the next node is randomly selected among the neighboring valued nodes. In the second type, the neighboring nodes are valued in such a way that the average value of each node is calculated with its second-order neighboring nodes.

In the third type, the averaging process continues until the neighbors of the third order, and in the fourth type, it continues until the fourth order. So, the CSRW algorithm has four execution types for each measurement criterion.

In this section, regarding Fig. 9 to Fig. 14, it should be noted that the bar graphs related to closeness centrality, degree centrality, degree, eigenvector centrality, and load centrality are calculated based on average values obtained from the implementation of four types of CSRW. For example, the degree bar graph is the average of the values for the execution of CSRW_Degree_Hop1, to CSRW_Degree_Hop4.

Regarding the datasets of DBLP_2000, DBLP_5000, and Cora, two different implementations have been performed with the aim of showing the high importance of context analysis in network embedding with no semantic analysis and impact of the text and by considering the semantic analysis of the nodes' context.

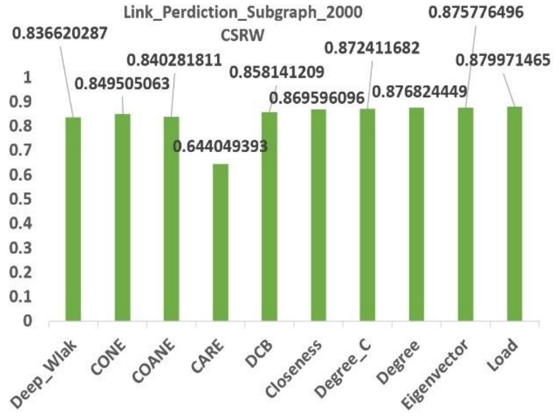


Fig. 9: ACU scores on link prediction criterion for DBLP_2000 for CSRW regardless of context.

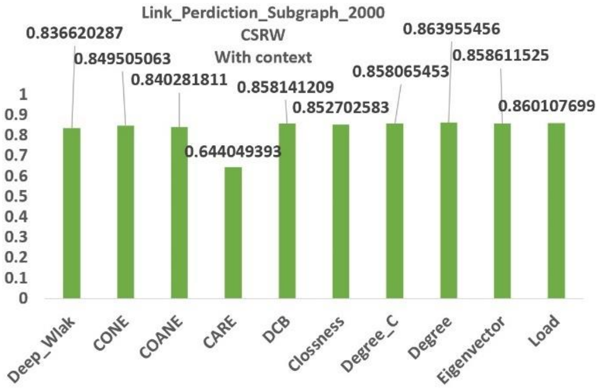


Fig. 10: ACU scores on link prediction criterion for DBLP_2000 for CSRW with semantic impact consideration.

Fig. 9 and Fig. 10 show the values obtained from the implementation of the CSRW algorithm without and with considering the semantic information in the DBLP_2000 dataset regarding the link prediction criterion. It can be seen from Fig. 9 and Fig. 10 that the last five bars which belong to our proposed method achieve the highest scores in the link prediction metric.

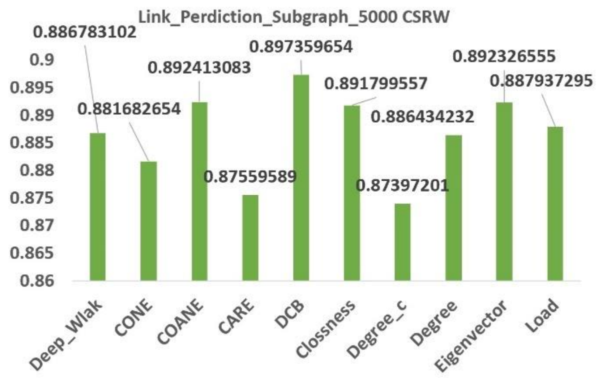


Fig. 11: ACU scores on link prediction criterion for DBLP_5000 for CSRW regardless of semantic.

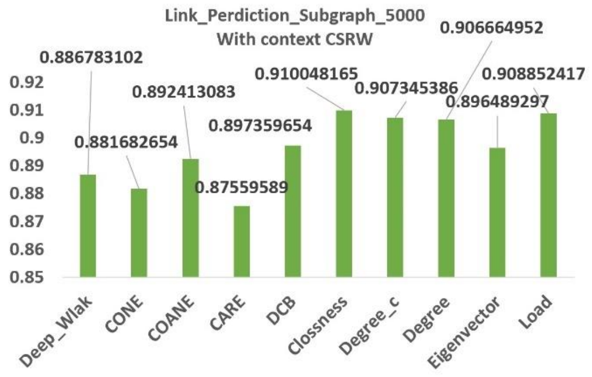


Fig. 12: ACU scores on link prediction criterion for DBLP_5000 for CSRW considering the semantic impact.

Based on Fig. 11 and Fig. 12 of CSRW implementation without and with context consideration in the DBLP_5000 dataset, it is clear that the semantic impact of the nodes in the accuracy obtained from the network embedding is very important.

As shown in Fig. 12, regarding the implementation of the CSRW algorithm, considering the context analysis in the DBLP_5000 dataset, the highest accuracy level of 0.91 has been obtained with CSRW_Clossness centrality. Also, the methods of degree, degree centrality, and load centrality have obtained accuracies above 0.9. That is, the average performance of CSRW_Clossness centrality in the first to fourth order is equal to 0.91 and has performed better than other classical methods. In

Fig. 16 , the accuracies of different CSRW runs for different metrics have been illustrated.

Based on Fig. 9 to Fig. 12 of the CSRW implementation without and with context consideration in the DBLP_2000 dataset and DBLP_5000 dataset, it is clear that the semantic impact of the nodes and Scalability of the dataset in the accuracy obtained from the network embedding is very important. the accuracy obtained has improved with the increase in the size of the data set.

Based on Fig. 12, considering the context analysis in the DBLP_5000 dataset, the highest accuracy has been obtained with different centrality criteria. so, the obtained accuracy will be better, when both centrality criteria and context analysis are used.

By examining these results, it can be seen that the CSRW algorithm is scalable considering the semantic analysis and has achieved the highest accuracy compared to other classical methods.

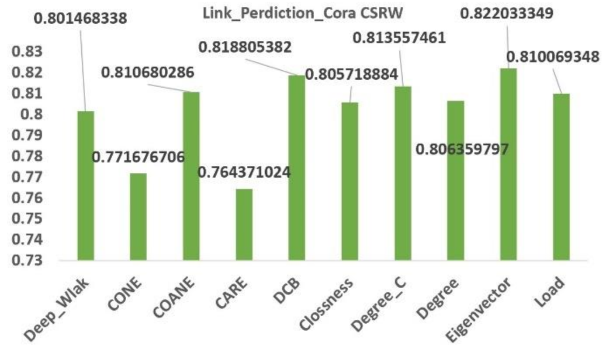


Fig. 13. ACU scores on the link prediction criterion for the Cora dataset for CSRW without considering the semantic.

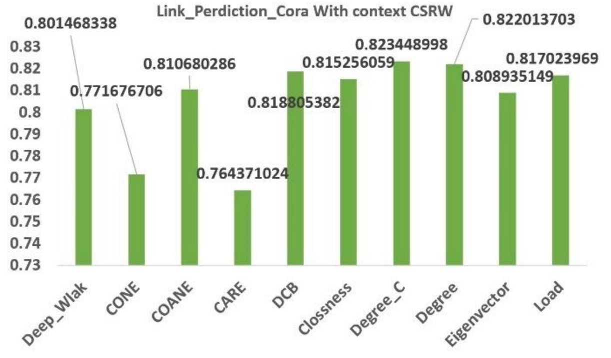


Fig. 14. ACU scores on the link prediction criterion for the Cora dataset for CSRW with semantic consideration.

In the Cora dataset based on Fig. 13, it can be observed that all the new methods, except the eigenvector centrality method, have not achieved high accuracy. But according to Fig. 14, it can be seen that under employing the semantic analysis of nodes in CSRW, all the new methods have reached high accuracy in comparison to the classical methods. Parts 1, 2و 3, 4 and 5 of Fig. 15 shows that the execution of the four types of CSRW algorithms considering the semantics for all five criteria. For example, part 1 shows the accuracy of the implementation of the CSRW_Clossness algorithm for one to four hops.

This comparison has been made with the aim of showing that increasing the hops to some extent increases the accuracy of the implementation. Increase the number of hops in the neighbor valuation process increases the number of loops in the CSRW algorithm and finally the time complexity increases. To increase the number of hops, you should pay attention to the following points:

- Memory and CPU power of the system to run the algorithm.
- The size of the network or dataset.

Finally, in order to achieve ideal accuracy, a trade-off must be made between two computational complexity and algorithm execution times. So, increasing the number of orders or hops in the CSRW algorithm should be continued until the improvement is achieved.

According to part 3 of Fig. 15, in the degree criterion, the accuracy of the algorithm has been improved by increasing the number of hops. And in parts 1 and 5 of Fig. 15 in closeness and load centrality the accuracy in lower hops has worked better. So, it is a better criterion to reach higher accuracy in lower hops with less complexity and time.

Fig. 16 illustrates the fact that with a high-dimensional network, i.e., a DBLP_5000, the CSRW embedding methods have achieved higher accuracy in lower orders in comparison to Fig. 15.

III) *Comparison based on Vertex Classification*

Vertex classification is used to evaluate the quality of the obtained representations, where L2-regularized logistic regression is employed as a supervised classifier. In the experiments, the training size of input datasets is in- creased from 10% to 90%. Precision, recall, Micro-F1, and Macro-F1 measures are applied to evaluate performance of different algorithms.

The experiments are repeated for 10 times and the average classification accuracy with different training ratio on the Cora dataset is shown in Fig. 17, Fig. 18, Fig. 19 and Fig. 20. The reason for choosing the Cora dataset at this stage is the number of seven tags that the articles are classified based on them. CSRW performs significantly better than other methods.

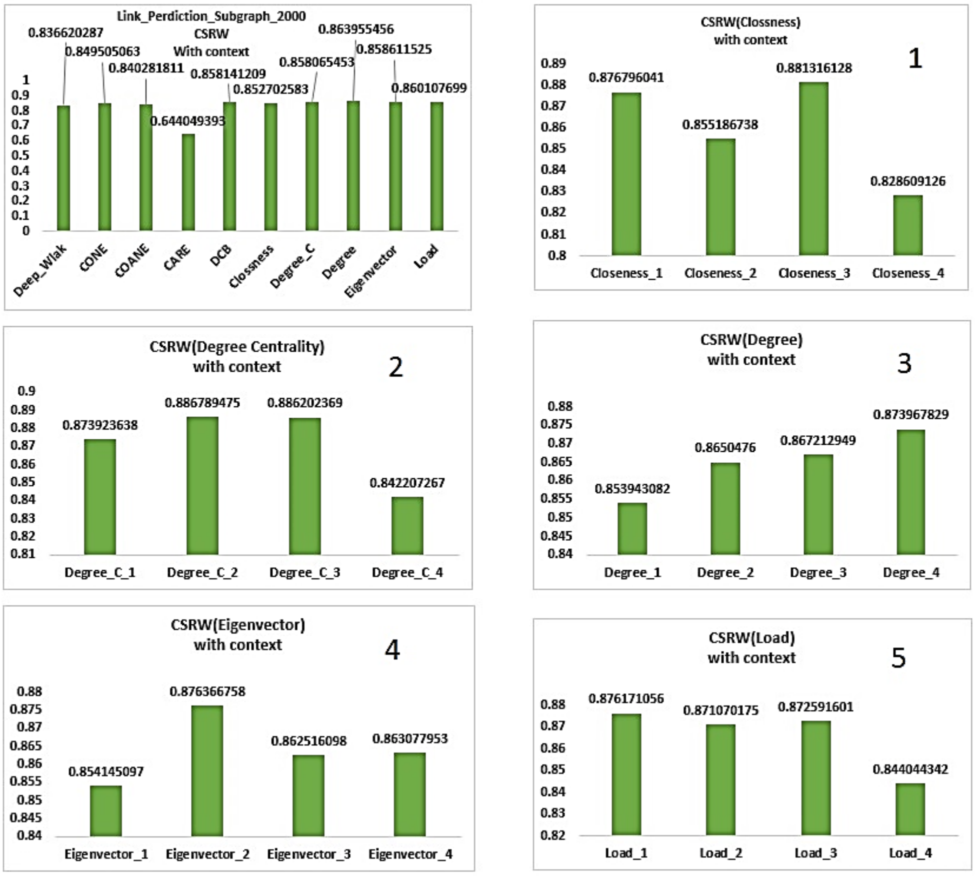


Fig. 15. Implementation of different variants of CSRW with semantic consideration.

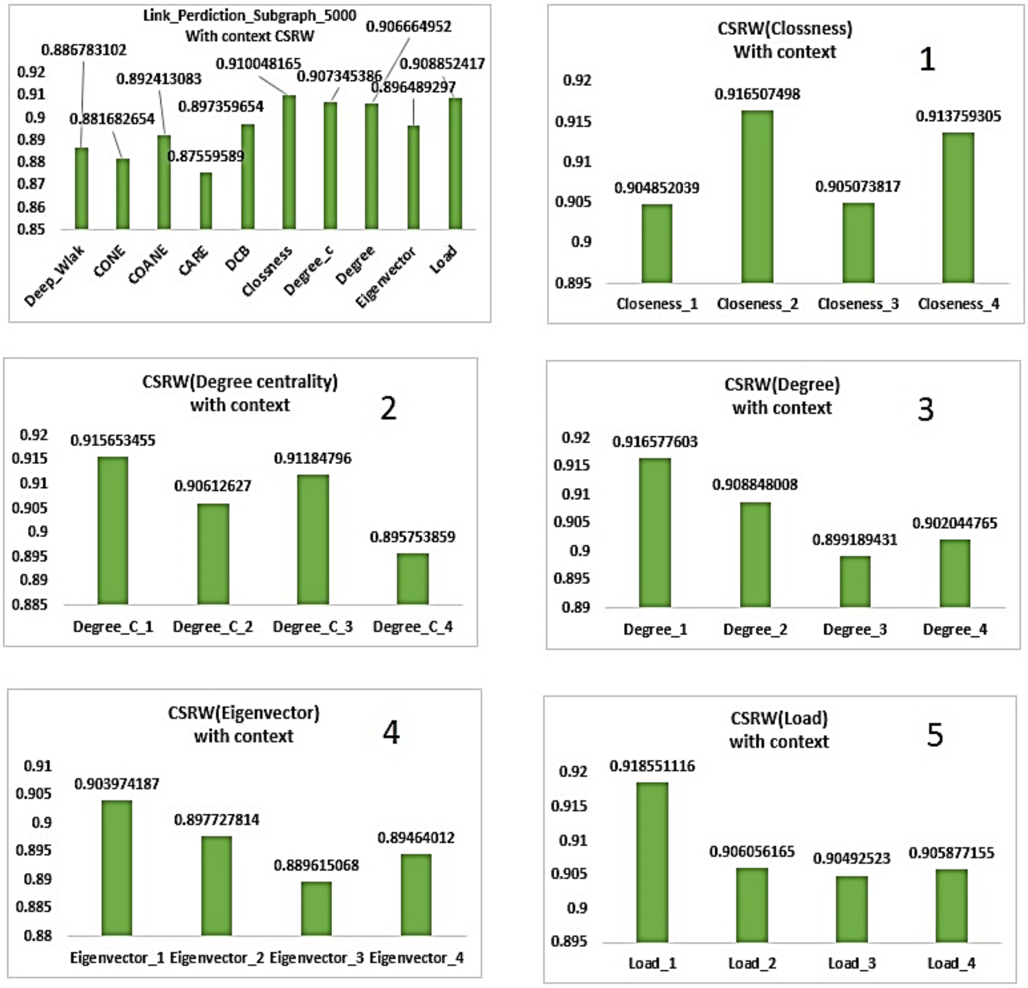
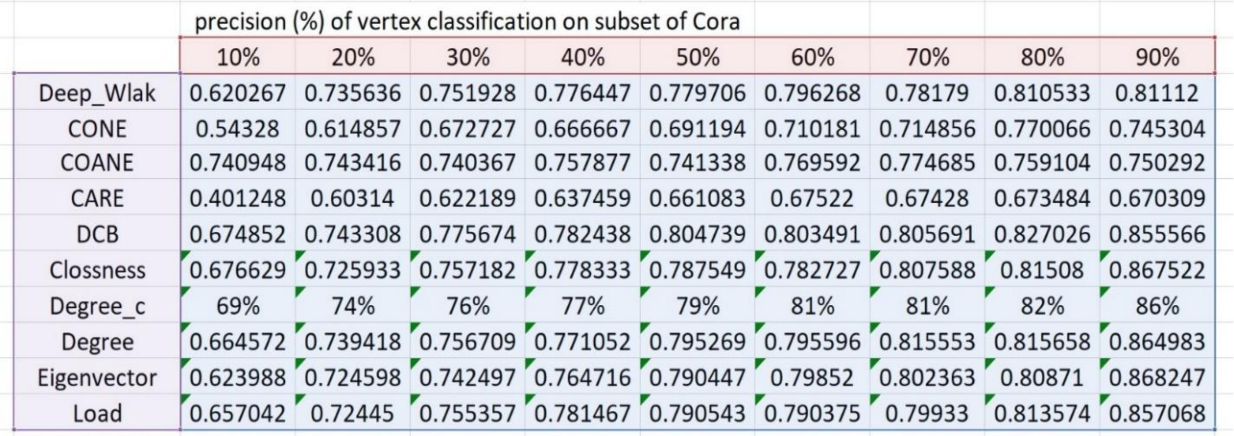
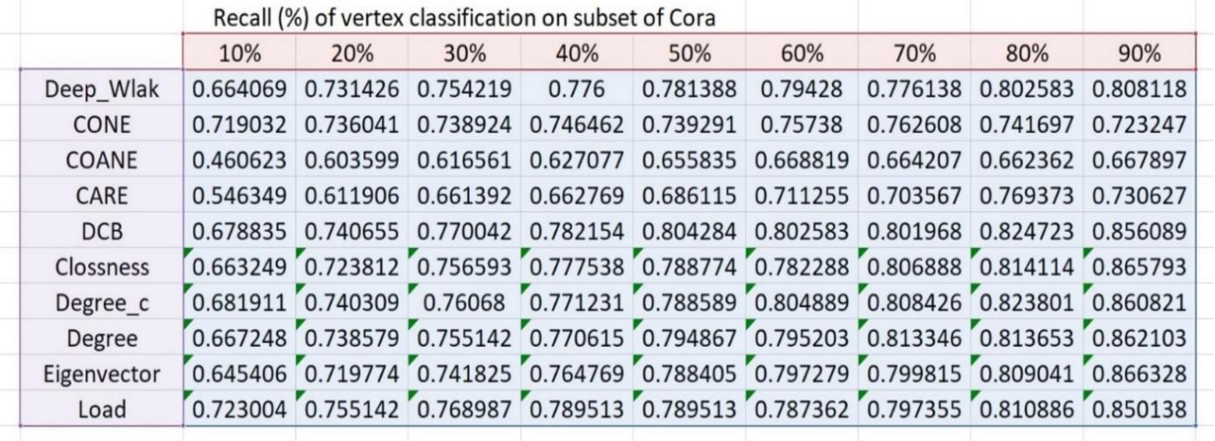


Fig. 16 Different runs of CSRW with semantic consideration in a DBLP_5000.



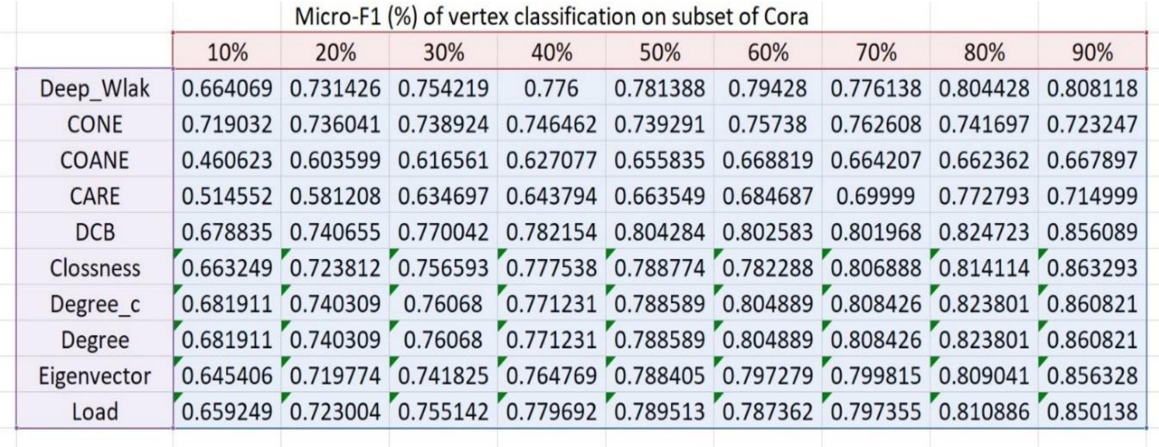
	precision (%) of vertex classification on subset of Cora								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
Deep_Wlak	0.620267	0.735636	0.751928	0.776447	0.779706	0.796268	0.78179	0.810533	0.81112
CONE	0.54328	0.614857	0.672727	0.666667	0.691194	0.710181	0.714856	0.770066	0.745304
COANE	0.740948	0.743416	0.740367	0.757877	0.741338	0.769592	0.774685	0.759104	0.750292
CARE	0.401248	0.60314	0.622189	0.637459	0.661083	0.67522	0.67428	0.673484	0.670309
DCB	0.674852	0.743308	0.775674	0.782438	0.804739	0.803491	0.805691	0.827026	0.855566
Clossness	0.676629	0.725933	0.757182	0.778333	0.787549	0.782727	0.807588	0.81508	0.867522
Degree_c	69%	74%	76%	77%	79%	81%	81%	82%	86%
Degree	0.664572	0.739418	0.756709	0.771052	0.795269	0.795596	0.815553	0.815658	0.864983
Eigenvector	0.623988	0.724598	0.742497	0.764716	0.790447	0.79852	0.802363	0.80871	0.868247
Load	0.657042	0.72445	0.755357	0.781467	0.790543	0.790375	0.79933	0.813574	0.857068

Fig. 17: Precision score on Cora dataset.



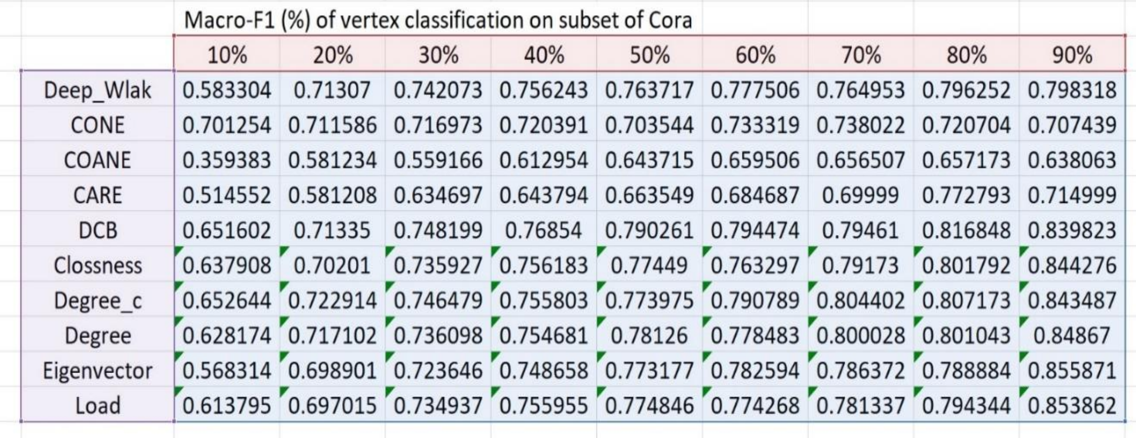
	Recall (%) of vertex classification on subset of Cora								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
Deep_Wlak	0.664069	0.731426	0.754219	0.776	0.781388	0.79428	0.776138	0.802583	0.808118
CONE	0.719032	0.736041	0.738924	0.746462	0.739291	0.75738	0.762608	0.741697	0.723247
COANE	0.460623	0.603599	0.616561	0.627077	0.655835	0.668819	0.664207	0.662362	0.667897
CARE	0.546349	0.611906	0.661392	0.662769	0.686115	0.711255	0.703567	0.769373	0.730627
DCB	0.678835	0.740655	0.770042	0.782154	0.804284	0.802583	0.801968	0.824723	0.856089
Clossness	0.663249	0.723812	0.756593	0.777538	0.788774	0.782288	0.806888	0.814114	0.865793
Degree_c	0.681911	0.740309	0.76068	0.771231	0.788589	0.804889	0.808426	0.823801	0.860821
Degree	0.667248	0.738579	0.755142	0.770615	0.794867	0.795203	0.813346	0.813653	0.862103
Eigenvector	0.645406	0.719774	0.741825	0.764769	0.788405	0.797279	0.799815	0.809041	0.866328
Load	0.723004	0.755142	0.768987	0.789513	0.789513	0.787362	0.797355	0.810886	0.850138

Fig. 18: Recall score on the Cora dataset.



	Micro-F1 (%) of vertex classification on subset of Cora								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
Deep_Wlak	0.664069	0.731426	0.754219	0.776	0.781388	0.79428	0.776138	0.804428	0.808118
CONE	0.719032	0.736041	0.738924	0.746462	0.739291	0.75738	0.762608	0.741697	0.723247
COANE	0.460623	0.603599	0.616561	0.627077	0.655835	0.668819	0.664207	0.662362	0.667897
CARE	0.514552	0.581208	0.634697	0.643794	0.663549	0.684687	0.69999	0.772793	0.714999
DCB	0.678835	0.740655	0.770042	0.782154	0.804284	0.802583	0.801968	0.824723	0.856089
Clossness	0.663249	0.723812	0.756593	0.777538	0.788774	0.782288	0.806888	0.814114	0.863293
Degree_c	0.681911	0.740309	0.76068	0.771231	0.788589	0.804889	0.808426	0.823801	0.860821
Degree	0.681911	0.740309	0.76068	0.771231	0.788589	0.804889	0.808426	0.823801	0.860821
Eigenvector	0.645406	0.719774	0.741825	0.764769	0.788405	0.797279	0.799815	0.809041	0.856328
Load	0.659249	0.723004	0.755142	0.779692	0.789513	0.787362	0.797355	0.810886	0.850138

Fig. 19: Micro-F1 score on the Cora dataset.



	Macro-F1 (%) of vertex classification on subset of Cora								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
Deep_Wlak	0.583304	0.71307	0.742073	0.756243	0.763717	0.777506	0.764953	0.796252	0.798318
CONE	0.701254	0.711586	0.716973	0.720391	0.703544	0.733319	0.738022	0.720704	0.707439
COANE	0.359383	0.581234	0.559166	0.612954	0.643715	0.659506	0.656507	0.657173	0.638063
CARE	0.514552	0.581208	0.634697	0.643794	0.663549	0.684687	0.69999	0.772793	0.714999
DCB	0.651602	0.71335	0.748199	0.76854	0.790261	0.794474	0.79461	0.816848	0.839823
Clossness	0.637908	0.70201	0.735927	0.756183	0.77449	0.763297	0.79173	0.801792	0.844276
Degree_c	0.652644	0.722914	0.746479	0.755803	0.773975	0.790789	0.804402	0.807173	0.843487
Degree	0.628174	0.717102	0.736098	0.754681	0.78126	0.778483	0.800028	0.801043	0.84867
Eigenvector	0.568314	0.698901	0.723646	0.748658	0.773177	0.782594	0.786372	0.788884	0.855871
Load	0.613795	0.697015	0.734937	0.755955	0.774846	0.774268	0.781337	0.794344	0.853862

Fig. 20: Macro-F1 score on the Cora dataset.

IV) *Parameter Sensitivity*

The effect of the number of communities or topics (k) is shown in Fig. 21.

The k parameter varies from 9 to 24 and it shows the link prediction values for the CSRW method for different criteria of closeness centrality, degree centrality, degree centrality, eigenvector centrality, and load centrality for a DBLP_5000.

Here, load centrality has provided a better result than other criteria.

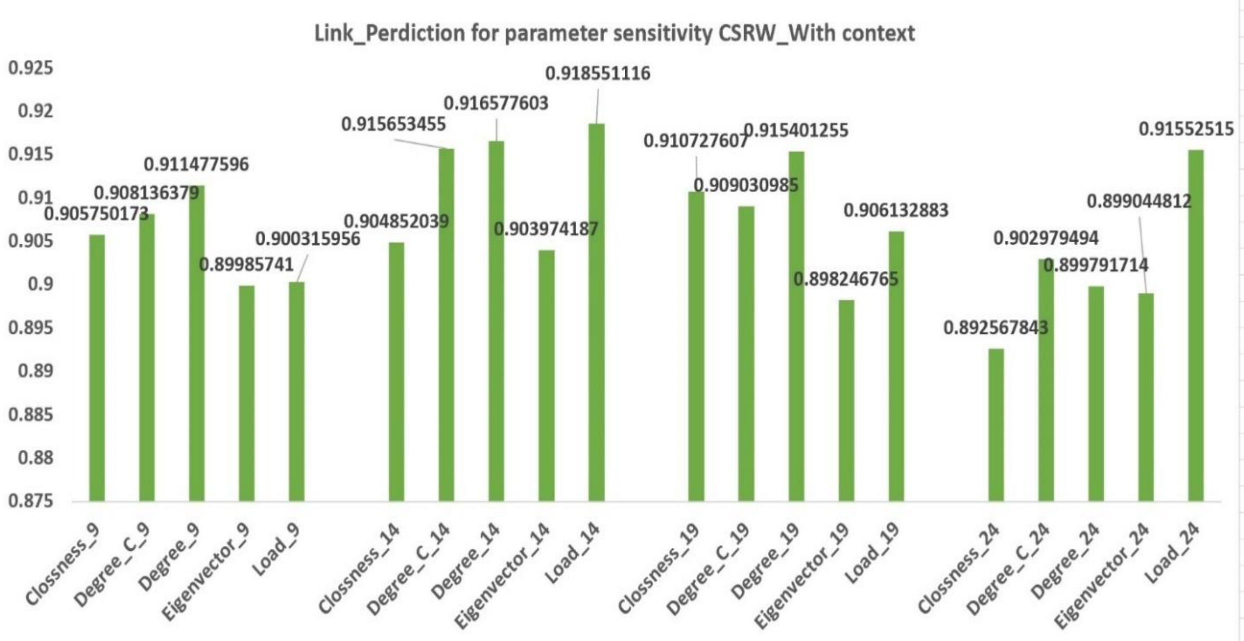


Fig. 21: The effect of the number of communities on the link prediction criterion for the CSRW algorithm.

To determine the best value of k, the averaging process has been performed for five methods of proximity centrality, degree, degree centrality, eigenvector centrality and load centrality in CSRW under setting different values for k.

According to Fig. 22, the best average value for k is 14. In this paper, the value of k is considered equal to 14 in all the implementations.

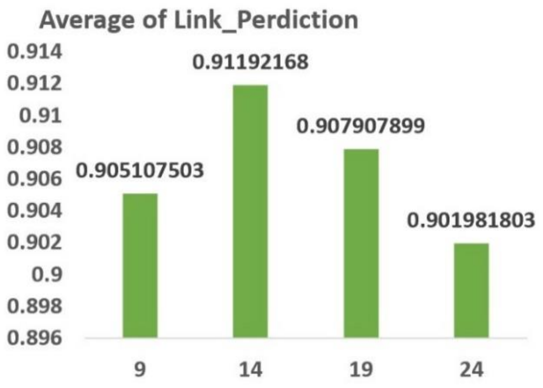


Fig. 22: The average values of the link prediction values of different criteria for the number of communities in the CSRW method.

Conclusion and Future Work

In this paper, the random walk algorithm based on the centrality and semantic consideration named CSRW is presented for the network embedding process. This method consists of two parts: random walk and semantic analysis.

In the random walk section, various criteria such as degree, degree centrality, closeness centrality, eigenvector centrality, and load centrality have been employed.

The non-greedy usage of these criteria is the specific innovation of this paper. Each node is measured based on the average of its criteria and the higher-order vertices. In the semantic analysis section, the textual information of each node is aggregated with the neighboring vertices of the next order and is analyzed by the BTM topic model. Finally, the final vector is obtained by combining the vectors obtained from the random walk and contextual analysis.

The proposed approach of this paper overcomes the shortcomings of classical research and has reached a high efficiency in the network embedding phase.

The experimental results on real-world concept-based networks show the effectiveness and robustness of CSRW compared to five basic methods DeepWalk, CARE, CONE, COANE, and DCB. The CSRW method is focused on two dimensions, different criteria, and the semantic connection of nodes.

In future works, we plan to expand this work to obtain the most influential vertices and maximize their influence throughout the network.

The methods presented in this article focus on one type of node. However, real-world networks are usually composed of different types of vertices, relationships, and explicit information.

Therefore, in the continuation of this research, the proposed method can be extended on heterogeneous networks. And also, in the following, it is possible to optimize the use of network embedding to obtain influential nodes and maximize the impact in social networks. In the future work, we plan to expand this work in terms of obtaining Influential nodes and optimizing the network.

Author Contributions

M. Taherparvar: Programmer, Validation, Conceptualization, Visualization, Investigation, collected the dataset, Writing-Reviewing and Editing, Writing - Original draft preparation. F. Ahmadi Abkenari: Supervision, Project administration, Conceptualization, Methodology, Visualization, Investigation, Writing-Reviewing and Editing, Programmer, Writing - Original draft preparation. P. Bayat: Writing-Reviewing and Editing. All authors discussed the results.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Abbreviations

CSRW	Centrality, and a Semantic-based Random Walk
BTM	Biterm Topic Models for Short Text

References

[1] B. Perozzi, R. Al-Rfou, S. Skiena, “Deepwalk: Online learning of social representations,” in Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 701–710, 2014.

[2] A. Grover, J. Leskovec, “Node2vec: Scalable feature learning for networks,” in Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 855–864, 2016.

[3] S. Bhagat, G. Cormode, S. Muthukrishnan, “Node classification in social networks,” Social Network Data Analytics (Springer) Ed. Charu Aggarwal, 2011.

[4] A. Faroughi, R. Javidan, “CANF: Clustering and anomaly detection method using nearest and farthest neighbor,” Future Gener. Comput. Syst., 89: 166–177, 2018.

[5] S. Aslan, M. Kaya, “Topic recommendation for authors as a link prediction problem,” Future Gener. Comput. Syst., 89: 249–264, 2018.

[6] T. Mikolov, K. Chen, G. Corrado, J. Dean, “Efficient estimation of word representations in vector space,” arXiv:1301.3781, 2013.

[7] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, “Line: Large-scale information network embedding,” in Proc. International Conference on World Wide Web: 1067–1077, 2015.

[8] H. Gao, H. Huang, “Deep attributed network embedding,” in Proc. International Joint Conference on Artificial Intelligence: 3364–3370, 2018.

[9] X. Huang, J. Li, X. Hu, “Label informed attributed network embedding,” in Proc. ACM International Conference on Web Search and Data Mining: 731–739, 2017.

[10] J. Butepage, M. J. Black, D. Kragic, H. Kjellstrom, “Deep representation learning for human motion prediction and classification,” in Proc. IEEE Conf. on Computer Vision and Pattern Recognition: 1591–1599, 2017.

[11] X. Du, J. J. Y. Wang, “Support image set machine: Jointly learning representation and classifier for image set classification,” Knowledge-Based Syst., 78: 51–58, 2015.

[12] J. Li, J. Li, X. Fu, M. A. Masud, J. Z. Huang, “Learning distributed word representation with multi-contextual mixed embedding,” Knowledge-Based Syst., 106: 220–230, 2016.

[13] M. Janner, K. Narasimhan, R. Barzilay, "Representation learning for grounded spatial reasoning," Trans. Assoc. Computer. Ling., 6: 49–61, 2018.

[14] Z. Chen, T. Cai, C. Chen, Z. Zheng, G. Ling, “SINE: Side information network embedding,” in Proc. International Conference on Database Systems for Advanced Applications: 692–708, 2019.

[15] D. Wang, P. Cui, W. Zhu, “Structural deep network embedding,” in Proc. International Conference on Knowledge Discovery and Data Mining: 1225–1234, 2016.

[16] M. Li, J. Liu, P. Wu, X. Teng, “Evolutionary network embedding preserving both local proximity and community structure,” IEEE Trans. Evol. Comput., 24(3): 523-535, 2019.

[17] J. Chen, Q. Zhang, X. Huang, “Incorporate group information to enhance network embedding,” in Proc. International Conference on Information and Knowledge Management: 1901–1904, 2016.

[18] M. M. Keikha, M. Rahgozar, M. Asadpour, “Community aware random walk for network embedding,” Knowl. Based Syst., 47–54, 2018.

[19] J. Wang, j. Cao, W. Li, S. Wang, “*CANE: community-aware network embedding via adversarial training,*” Knowl. Inf. Syst., 63: 411-438, 2020.

[20] Y. Shi, M. Lei, H. Yang, L. Niu, “Diffusion network embedding,” Pattern Recognit., 88: 518–531, 2019.

[21] H. Chen, H. Yin, T. Chen, Q.V.H. Nguyen, W.-C. Peng, X. Li, “Exploiting centrality information with graph convolutions for network representation learning,” in Proc. IEEE International Conference on Data Engineering: 590–601, 2019.

[22] W. Zhao, H. Ma, Z. Li, X. Ao, N. Li, “SBRNE: An improved unified framework for social and behavior recommendations with network embedding,” in Proc. International Conference on Database Systems for Advanced Applications: 555–571, 2019.

[23] Q. Li, J. Zhong, Q. Li, Z. Cao, C. Wang, “Enhancing network embedding with implicit clustering,” in Proc. International Conference on Database Systems for Advanced Applications: 452–467, 2019.

[24] M. RosenZvi, T. Griffiths, M. Steyvers, P. Smyth, “The author-topic model for authors and documents,” in Proc. Uncertainty in Artificial Intelligence: 487–494, 2004.

[25] Q. Mei, D. Cai, D. Zhang, C. Zhai, “Topic modeling with network regularization,” in Proc: International Conference on World Wide Web: 101–110, 2008.

[26] L. Wu, D. Wang, S. Feng, Y. Zhang, G. Yu, MDAL: “Multi-task Dual Attention LSTM Model for Semi-supervised Network Embedding,” in Proc: International Conference on Database Systems for Advanced Applications: 468–483, 2019.

[27] Y. Gao, M. Gong, Y. Xie, H. Zhong, “Community-oriented attributed network embedding,” Knowledge-Based Systems, 193: 105418, 2019.

[28] M. Taherparvar, F. Ahmadi Abkenari, P. Bayat, “Attribute network embedding based on maintaining the structure and semantic

features of the graphs," in Proc: International Conference on The New Horizons in The Electrical Engineering, Computer and Mechanical, 2022.

[29] L. Chen, Y. Li, X. Deng, Z. Liu, M. Lv, T. He, "Semantic-aware network embedding via optimized random walk and paragaraph2vec," J. Comput. Sci., 63: 101825, 2022.

[30] J. Zhang, Yu. Luo, "Degree centrality, betweenness centrality, and closeness centrality in social network," in proc. International Conference on Modelling, Simulation and Applied Mathematics, 2017.

[31] P. Bonacich, "Some unique properties of eigenvector centrality," Social Network, 29(4): 555-564, 2007.

[32] L. Maccari, L. Ghiro, A. Guerrieri, A. Montresor, R. Lo Cigno, "On the distributed computation of load centrality and its application to DV routing," in proc. IEEE Conference on Computer Communications, 2018.

[33] M. Taherparvar, F. Ahmadi Abkenari, P. Bayat, "Conformance evaluation of topic modeling approaches on web-based short text dynamic graph databases," in proc. International Conference on Web Research, 2021.

[34] J. Tang, Z. Meng, X. Nguyen, Q. Mei, M. Zhang, "Understanding the limiting factors of topic modeling via posterior contraction analysis," in Proc. International Conference on Machine Learning: 190–198, 2014.

[35] A. K. McCallum, K. Nigam, J. Rennie, K. Seymore, "Automating the construction of internet portals with machine learning," Information Retrieval: 127–163, 2000.

[36] https://www.aminer.org/citation, last access June 17, 2023.

Biographies

Mohadeseh Taherparvar received B.Sc. in software engineering from Azad University of Lahijan, Iran, in 2008, MSc software engineering from Azad University of Qazvin, Iran, in 2013 and she is PHD candidate in Department of Computer Engineering, Rasht Branch, Islamic Azad University, Rasht, Iran. Her research interests are Data Mining, Deep Learning, and Optimization Algorithms. she also has experience in Python.

- Email: mtaherparvar@phd.iaurasht.ac.ir
- ORCID: 0000-0002-7822-5088
- Web of Science Researcher ID:NA
- Scopus Author ID:NA
- Homepage: NA

Fatemeh Ahmadi-Abkenari received the M.Sc. degree in information technology from the Polytechnique (Amirkabir) University of Tehran, Iran, in 2007, and the Ph.D. degree in computer engineering from UTM, Malaysia, in 2012. She is currently an Assistant Professor with the Faculty of Computer Engineering and Information Technology, Payam-Noor University, Rasht Branch, Iran. Her main research interests include machine learning, data mining, text mining, sentiment and opinion mining, arti_cial neural networks, deep learning, and natural language processing.

- Email: Fateme.Abkenari@pnu.ac.ir
- ORCID: 0000-0001-5175-6826
- Web of Science Researcher ID:NA
- Scopus Author ID:NA
- Homepage: NA

Pyman Bayat received the M.Sc. degree from Islamic Azad University, Arak Branch, Iran, and the Ph.D. degree in computer engineering from UCSI University, Malaysia. He is currently an Assistant Professor with the Faculty of Computer Engineering, Islamic Azad University, Rasht Branch, Iran. His main research interests include distributed systems, image processing, and data mining.

- Email: bayat@iaurasht.ac.ir
- ORCID: 0000-0003-2291-1369
- Web of Science Researcher ID:NA
- Scopus Author ID:NA
- Homepage: NA

How to cite this paper:
M. Taherparvar, F. Ahmadi Abkenari, P. Bayat, "Centrality and Latent Semantic Feature Random Walk (CSRW) in large network embedding," J. Electr. Comput. Eng. Innovations, 11(2): 311-326, 2023.

DOI: 10.22061/jecei.2023.9279.600

URL: https://jecei.sru.ac.ir/article_1834.html