A COMPREHENSIVE STUDY ON NODE CLASSIFICATION BY LATENT REPRESENTATION LEARNING METHODS IN NETWORK ANALYSIS

Abstract

Most Machine Learning models reply on learning the latent representation of vertices (nodes) in a large complex network. Recently, Network representation focuses on various tasks such as link prediction, community detection and node classification. This paper presents a comprehensive study on the various techniques applied in the latent representation of vertices in the network. Also, a new method is proposed to overcome the challenges faced while capturing the structural information in networks. The proposed method has gained its ability to assess the performance based on the robustness, scalability. DeepWalk, Node2vec, GraphSAGE are the existing algorithms used for the comparative analysis. We analyze the performance of these methods by leveraging the benefits, challenges and the application area using various parameters. Random Forest classifier is used for classifying the nodes in the network. The evaluation metric used for node classification includes F1 score. The experimental analysis using the face book network provides an improved result giving the valuable insights on the strengths and weaknesses of the algorithms.

Keywords: Community Detection, Latent Representation, Link Prediction and Random forest.

Authors

E. Chandra Blessie
Assistant Professor
Department of Computing (Artificial Intelligence and Machine Learning)
Coimbatore Institute of Technology
Coimbatore, India.

K. Anandan
Department of Computer Applications
Nehru College of Management
Coimbatore, India.

Sandhya Sharma
Research Scholar
Nehru College of Management
Coimbatore, India.

I. INTRODUCTION

Recently, most of the applications requires graph-based representation of complex network. Also, Machine Learning tasks such as link prediction, community detection, and node classification need latent representation of the nodes in large-scale network. Latent representation learning (Ozsel Kilinc, Ismail Uysal, 2018) has emerged as a vital task for extracting meaningful information from complex large-scale network. Link prediction model predicts the missing or potential edges which appear between the nodes. Latent representation learning estimates the likelihood of the existence of links between the pairs of nodes. The most prominent models are Graph Convolutional Networks (GCNs) and Graph Autoencoders (GAEs). These models capture the node similarities and network topology effectively. Community Detection aims at identifying the group of densely connected nodes. The main objective is to group the nodes into clusters by learning latent representation efficiently without sacrificing the structure of the network. Spectral Clustering, Node2vec, and GraphSAGE (Jonathan Crabbe et. al. 2021) are the algorithms for identifying node communities by capturing the network structure information. Node classification predicts the class to which the nodes are classified based on their attributes and connections. For this, the algorithm learns informative node embeddings in an effective way. DeepWalk, GraphSAGE, and Node2Vec have shown promising performance in node classification tasks by leveraging node embeddings to make accurate predictions.

In the era of big data and complex networks, graph-based data has become increasingly prevalent, encompassing a wide range of applications. In many of these domains, the task of node classification plays a crucial role, aiming to predict the class labels or categories of nodes within a graph based on their attributes and connections. To address this challenge, machine learning techniques have emerged as powerful tools for uncovering patterns and making accurate predictions.

One such versatile and effective method is the Random Forest Node Classifier. Random Forest is an ensemble learning algorithm that combines the strength of multiple decision trees to perform classification tasks. Random Forest algorithm has gained popularity due to its ability to handle high-dimensional and complex data, its robustness to over fitting, and its resistance to noise and outliers. In the context of node classification in graph-based data, the Random Forest Node Classifier operates by leveraging the features of nodes and their associated relationships within the graph. The nodes are represented as feature vectors, with each feature capturing relevant characteristics or attributes of the nodes. The Random Forest Node Classifier learns from labeled nodes in the graph, where class labels are available for a subset of nodes. Once trained, the model can be used to predict the class labels of unlabeled nodes, allowing for effective classification across the entire graph.

In this paper, the pros and cons of these node classification approaches are highlighted and a comparative analysis is made with the state-of-art traditional algorithms based on the F1 score, scalability and robustness. To overcome the challenges of these algorithms, we have proposed a novel method for handling the complex structure in large-scale networks. Our proposed approach is designed to assess the performance and to yield better result.

The paper is well organized with each section being well defined. Section 1 gives the introduction about learning latent representation with its techniques and the outline of the research towards the end. Section 2 provides study on the recent researches relevant to this paper followed by section 3 with the methodologies. Proposed approach with the algorithm is given in section 4. Section 5 covers the experimental analysis. Finally, we conclude with the summary of the study in section 6.

II. RELATED WORK

Tang et al. (2015) proposed a novel Large-scale Information Network Embedding (LINE) method, which optimizes proximity measurement and embedding between nodes in the network. This method is applicable for directed, undirected and weighted network. The authors introduced Edge sampling algorithm to improve efficiency and effectiveness of the inference. This algorithm can be applied on Citation network, Social Network and Language Network. Perozzi et al. (2014) proposed unsupervised algorithm named DeepWalk algorithm for learning latent presentation. Random walks and Skip-Gram model are used for capturing local network structure and to generate continuous representations for nodes respectively. DeepWalk learns structural regularities present within short random walks. They evaluated the representation on multilabel classification tasks on several social networks.

Grover and Leskovec (2016), builds node2vec upon DeepWalk by incorporating biased random walks that balance exploration and exploitation. This technique allows for the flexibility to control the trade-off between breadth-first and depth-first search strategies, enabling better capture of both local and global network information. node2vec has demonstrated superior performance in capturing community structures and identifying homophilic patterns in social networks.Hamilton et al. (2017) proposed GraphSAGE, whichintroduced the approach of node neighborhood aggregation in their framework for latent representation learning. GraphSAGE efficiently learns from the node's local neighborhood by aggregation and sampling features. This algorithm is suitable for large-scale applications even with the variation in network size and structure.

Kipf and Welling (2017) introduced Deep Learning based model Graph Convolutional Networks (GCNs) for graph-structured data. GCNs employ graph convolutional layers for learning node embedding by aggregating the local and high-order neighborhood node features. GCNs propagate information through the graph and shows the performance improvement in various network analysis tasks and are known for their scalability and interpretability.Kipf and Welling (2016) proposed Variational Graph Autoencoders (VGAE),a probabilistic model that uses variational inference to learn graph embeddings. VGAE can generate uncertain representations by distributingthe node embeddings. VGAE has been successfully applied in anomaly detection, recommendation systems, and network reconstruction tasks. Ozsel Kilinc and Ismail Uysal (2018) proposed a novel unsupervised clustering approach which uses k-means latent representation. This representation is useful for revealing the unknown clusters. The process randomly assigns a pseudo parent-class label to each observation.Generated pseudo-observation-label pairs are subsequently used to train a neural network with Auto-clustering Output Layer (ACOL) that introduces multiple softmax nodes for each pseudo parent-class. Jonathan Crabbe et. al. (2011) proposed SimplEx which uses the corpus, a freely selected examples to improve the user's understanding of the latent space. The authors proposed a novel approach, the Integrated Jacobian, that allows SimplEx to make explicit the contribution of each corpus feature in the mixture. The method explains the model representation by highlighting the relevant patterns. It shows accuracy improvement and robustness.

III. METHODOLOGIES

1. **Learning Latent Representation:** Most machine learning and Deep Learning algorithms learns the pattern, structure from the complex large networks by transforming the data into a low dimensional space.

   - **Definition:** Latent representation learning, also known as representation learning or feature learning, is a subfield of machine learning and deep learning that focuses on transforming raw data into a more informative, compact, and meaningful representation, often in a lower-dimensional space.

     The main objective of latent representation learning (William L et. al. 2017) is to extract relevant and useful features from the input and can be used for node classification, link prediction etc. This is an unsupervised approach which automates the process of extracting the features and enables the process of learning from unlabeled data.

     One of the techniques is Autoencoders, a popular class of unsupervised neural networks used for learning latent representations. The encoders map the input data to a latent space and a decoder that reconstructs the input data from the learned representation. Variational Autoencoders (VAEs) are a type of autoencoder which includes the probability reasoning concept in the model. It learns to encode data into a probability distribution in the latent space, allowing for generation of new data points.Generative Adversarial Networks (GANs) models consists of a generator and a discriminator, which are trained to generate data efficiently Word Embeddings model represents the words in a semantic space as continuous vector. These algorithms enableto capture the semantic meaning and contextual relationships between words.Transformer-based Models is used in NLP for capturing contextual dependencies and generate word embeddings efficiently.One of the benefits of latent learning representation is dimensionality reduction which converts and transforms the high-dimensional data into lower dimensional data. These models are more effect and robust since it focuses on the important data. latent representation learning techniques are used for various applications such as link prediction, community detection, and node classification tasks. These tasks are importance to identify missing connections, to predict the attributes of nodes and to understand the network structures and dynamics.

   - **Link Prediction:** Link prediction aims to predict missing or potential edges between nodes in a network. By learning meaningful latent representations of nodes, we can estimate the likelihood of the existence of links between pairs of nodes. Latent representation learning models, such as Graph Convolutional Networks (GCNs) and Graph Autoencoders (GAEs), have shown impressive results in link prediction by effectively capturing node similarities and network topology.

- **Community Detection:** Community detection involves identifying groups of densely interconnected nodes with higher intra-community connections than inter-community connections. By learning latent representations that preserve community structure, we can efficiently cluster nodes into different communities. Algorithms like Spectral Clustering, Node2vec, and GraphSAGE have been utilized for community detection, as they excel in capturing structural information and identifying node communities.

- **Node Classification:** Node classification focuses on predicting the class or label of nodes in a network based on their attributes or connections. By learning informative node embeddings, we can effectively classify nodes into predefined categories. DeepWalk, GraphSAGE, and Graph Convolutional Networks (GCNs) have shown promising performance in node classification tasks by leveraging node embeddings to make accurate predictions.

  In this study, we conduct comprehensive experiments on various real-world datasets to evaluate the performance of different latent representation learning methods for node classification. We compare the results of traditional approaches with state-of-the-art deep learning-based models.

- **Advantages of Latent Representation Learning:** Reduced Dimensionality: For learning, the data is represented in low dimensions which makes the visual representation much easier and effective analyze in various machine learning applications.

  Handling Missing Data: Latent representation learning can handle missing data by leveraging the relationships between available and most probable features.

  Compact and Informative Representation: Since the data is represented in low dimensions, the representation is very compact and informative.

- **Data Abstraction:** Data which are essential for the learning are only represented in the reduced vector space and other details are abstracted from its users.
- **Generalization:** Latent representations generalize well to unseen data, as they capture the underlying patterns and semantics of the data rather than memorizing specific instances.
- **Transfer Learning:** Latent representations facilitate transfer learning, where a model pre-trained on one task or dataset can be fine-tuned on another task or dataset with limited labeled data, saving time and computational resources.
- **Improved Performance:** Models using latent representations often achieve better performance on various tasks, as they can focus on the most relevant features and patterns in the data.

- **Disadvantages:** Loss of Interpretability: In some cases, latent representations may lack interpretability, making it challenging to understand the learned features and the reasoning behind specific predictions.

- **Domain Dependence:** The quality of the latent representations can depend on the domain and the characteristics of the data. Techniques that work well in one domain may not generalize as effectively to others.
- **Overfitting:** Overfitting may occur if the model does not learn in a regularized pattern or completely.
- **Computational Complexity:** Latent representation learning techniques can be computationally expensive, especially for large datasets and complex models.
- **Hyperparameter Tuning:** Latent representation learning methods often involve several hyperparameters, and finding the optimal settings can be challenging and time-consuming.
- **Biases and Noise:** While training the dataset, Latent representations may leads to biased or inaccurate prediction in various machine learning task.

- **Applications:** Latent representation learning has found widespread use in various domains, including computer vision, natural language processing, recommendation systems, anomaly detection, and many others, improving the performance and efficiency of machine learning models.

2. Existing Methodologies

- **DeepWalk Latent representation Learning:** DeepWalk (Bryan Perozziet. al. 2014) is an Unsupervised Learning technique of Network Representations. The main objective is to learn meaningful representation of nodes in complex large networks. This algorithm has two processes; the first process is to apply random walk generator which takes graph as input and randomly selects a node as the root node during random walk and captures the network structures. By training the model on the random walks, DeepWalk learns continuous representations of nodes in the graph.

  The second process is to apply the language modelling technique such as skip-gram model that learns the node representation based on the effect of hyperparameters such as length of the walk, dimensions of node embedding and number of walks per node. The Skip-Gram model aims to predict the context (neighboring nodes) of a given node. DeepWalk offers facilities to visualize the complex network in low-dimensional space which is the embedding space where the algorithm captures structural relationships and similarity between nodes. Nodes with similar roles or similar neighborhoods in the graph are expected to have similar embeddings in the learned representation space.

- **Node2Vec Latent Representation learning:** Node2Vec (Aditya Groveret. al. 2016) is another latent representation learning technique which extends DeepWalk by introducing biased random walks that balance between breadth-first and depth-first exploration strategies. This flexibility allows node2vec to capture both local and global network characteristics. In this approach, both in-depth and wide exploration of node representation are done by changing the return parameter (p) and in-out parameter (q).

After generating the random walk, Node2Vec also applies skip-gram language model to predict the context (neighborhood nodes) of a given node in low-dimensional embedding space. It captures topological association between the nodes. Node2Vec is a semi-supervised learning algorithm which optimizes graph based objective function using SGD. Node2Vec uses 2^{nd} order random walk approach to generate network neighborhoods for nodes. After selecting the appropriate neighborhood, the algorithm learns about the node's role and community to which it belongs. For this, biased random walks are used efficiently

- **GraphSAGE Latent Representation learning:** GraphSAGE is a well-known approach for latent representation learning in a large complex network. This algorithm provides a scalable framework to learn about the unseen nodes based on their local neighborhood information. The process used are sampling and aggregation of features. This leads to robustness making it suitable for real time applications. This approach is different from DeepWalk and Node2Vec as it uses sampling-based approach within a certain hop distance and collects the features information and then aggregate them into comprehensive representation for the node.

The sampling techniques used are uniform sampling and neighbor-based sampling. This controls computational complexity while exploring the local neighborhood of each node. Finally, GraphSAGE process the aggregated node features using Graph Convolution Layers. These layers perform neighborhood aggregation and transformation, allowing the model to capture both local and higher-order structural information. By stacking multiple graph convolutional layers, GraphSAGE can learn more complex representations that consider information from distant nodes.

Table 1 gives a study on the benefits of these algorithms. Table 2 gives the challenges faced by these algorithms and application area where these algorithms are applicable are given in table 3.

Table 1: List of Benefits of DeepWalk, Node2Vec and GraphSAGE

Benefits	DeepWalk	Node2Vec	GraphSAGE
1. Scalability	Scalable. Can handle millions of nodes and edges	Scalable to handle massive graphs.	Scalable to large graph.
2.Neighborhood Preservation	Captures local structure of the graph	Captures both local and global structural information	Captures topological relationships between nodes.

Table 2: List of challenges of DeepWalk, Node2Vec and GraphSAGE

Challenges	DeepWalk	Node2Vec	GraphSAGE
Computational Complexity	Computationally expensive for large-scale graphs with millions of	Node2Vec's random walk and sampling processes can be computationally	Aggregation process can still be

	nodes and edges. Training DeepWalk on massive networks can require significant computational resources and time, limiting its applicability to extremely large graphs.	expensive for large-scale graphs. Performing multiple random walks and aggregating neighborhood information can require significant computational resources and time, limiting its scalability to massive networks.	computationally expensive for large graphs with millions of nodes and edges. The computational complexity increases with the number of layers and the size of the sampled neighborhood,
Bias	DeepWalk relies on random walks to generate training sequences, and as a consequence, it can introduce a homophily bias in the learned node embeddings	homophily bias, where nodes with similar attributes or connections are more likely to be placed close together in the embedding space.	Inductive Bias: GraphSAGE is an inductive learning method. When new nodes are added to the graph, they require retraining the model,
Sensitive to parameters	Sensitive to Hyperparameters: DeepWalk's performance can be sensitive to the choice of hyperparameters, such as the length of random walks, the number of walks per node, and the dimensionality of the embeddings.	Node2Vec relies on two hyperparameters, namely the return parameter (p) and the in-out parameter (q), which control the exploration and exploitation during random walk	Sensitivity to aggregation function

Table 3: List of application area of DeepWalk, Node2Vec and GraphSAGE

Applications	DeepWalk	Node2Vec	GraphSAGE
Node Classification	nodes are classified into predefined classes or categories based on their embeddings.	By learning embeddings that capture node properties and structural information,	Inductive learning capability allows it to efficiently classify unseen nodes based on their aggregated neighborhood information.
Link Prediction	to predict the likelihood of forming links between nodes	to estimate the likelihood of forming connections between nodes,	predicting missing or potential links between nodes
Community Detection	can be visualized to gain insights into the	capture community	applicable to graph-level detection and

	structure and relationships	structures in the graph makes it suitable for clustering nodes	divisions

As a summary, DeepWalk, Node2Vec and GraphSAGE are powerful and successful latent representation learning methods with some limitations on applying to specific network analysis tasks and in different applications.

IV. PROPOSED METHOD

After a comparative based study on the benefits, challenges and applications, we have proposed a new concept for latent representation learning. The main purpose behind this new approach is to give better idea which can give better solution for the challenges of the algorithms studied in this paper. The objective is to capture the structural information from the network in an efficient way to reduce the computational complexity and to increase the scalability with effective hyperparameter tuning.Node2Vec uses 2nd order Random Walk which uses two parameters p and q with the restricted distance of {0,1,2} between the nodes from where the walk starts and reaches. For node embedding, GraphSAGE focuses on using of inductive learning method and uses the Graph Convolution Network architecture for aggregation. So, the new approach introduces two main components

1. h-hop neighborhood approach where we can vary the value of "*h*" and arrange the random walk list on the basis of number of nodes in a walk instead of two parameters introduced in Node2Vec.

2. For node aggregation, Graph Attention Networks (GAN) is used to weigh the importance of neighboring nodes while aggregating information for each node instead of Graph Convolution Networks (GCN) as it is used in the GraphSAGE algorithm. This allows the model to focus on relevant neighbors during the learning process.

Algorithm : New Latent Representation Learning algorithm with h-hop and Graph Attention Network (GAN)

Algorithm(G, h) // G is the graph and 'h' is the hop level of a node with its neighborhood.
Input: graph G(V, E)
walks per vertex hop 'h' walk length
Output: matrix of vertex representations $\Phi \in R^{|V| \times d\ 1}$:
Initialization: vertex v_i
for i = 0 to h do
Wvi = BiasW alk(G, v_i,h^2) // walk made by using 'h' hops
end for
// for aggregation of nodes
for v in v_i do
    GAN(G,V_i,h)   // Graph Attention Network algorithm is used
end for

V. EXPERIMENTAL ANALYSIS

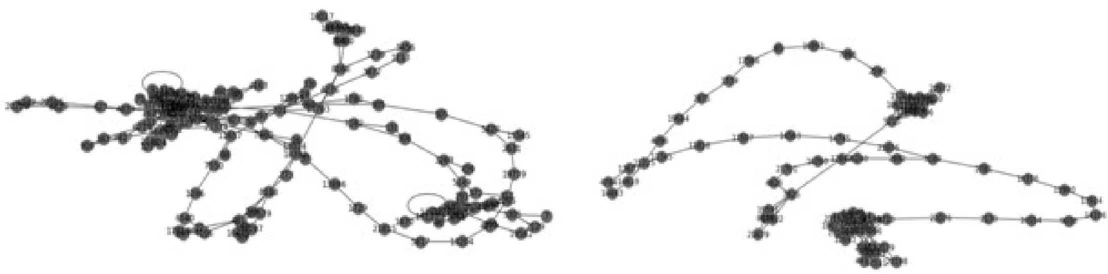
In this section, we provide an overview of the datasets and methods used in our experiment. The network used for this analysis is Facebook network.

1. **Network Used:** We have takenFacebook page dataset which is a well-known dataset for network analysis. This network represents Facebook page networks of different categories.It consists of 8 different types of pages each represented by nodes and edges. Nodes represent the pages and edges are mutual likes among them. For each dataset, we listed the number of nodes and edges. Table4shows the number of nodes, edges, density of the nodes and the transitivity used in the eight pages of the Facebook. The

Table 4 : Description of the Facebook network feature information

	Nodes	Edges	Density	Transitivity
Politicians	5,908	41,729	0.0024	0.3011
Companies	14,113	52,310	0.0005	0.1532
Athletes	13,866	86,858	0.0009	0.1292
News Sites	27,917	206,259	0.0005	0.1140
Public Figures	11,565	67,114	0.0010	0.1666
Artists	50,515	819,306	0.0006	0.1140
Government	7,057	89,455	0.0036	0.2238
TV Shows	3,892	17,262	0.0023	0.5906

To validate the performance of our approach, we compare it against a number of baseline algorithms studied in this paper. Figure 1 (a) – (d) shows the graph network representation by applying the existing and the proposed algorithms



(a) (b)

(c)

(d)

Figure 1: 1(a) graph latent representation using DeepWalk (b) graph representation using Node2Vec (c) graph latent representation using GraphSAGE (d) graph representation using proposed method

2. **Metrics Used:** For node classification using Random Forest classifier, we used F1 score metric to analyze the performance of the algorithms. F1 score is a single value which combines both recall and precision, making it suitable for imbalanced datasets. The value ranges from 0 to 1. The higher F1 score indicates better model performance in correctly classifying nodes across different classes.When different models are compared, the F1 score is one of the useful criterion for selecting the best-performing model or parameter configuration for node classification tasks. Confusion matrix shows the performance of Random Forest classifier by comparing the predicted values with the true values of the nodes in the network. It provides valuable information to assess the model's effectiveness in node classification.

3. **Result and Discussions:** F1 score and the confusion matrix after using DeepWalk mechanism to learn the embeddings and train them using a Random Forest Classifier is given below.

```
0.9374721851357365
[[0.94724943 0.03617182 0.00301432 0.01356443]
 [0.01568335 0.9626587  0.0201643  0.00149365]
 [0.00423729 0.04576271 0.94830508 0.00169492]
 [0.11419753 0.00925926 0.0308642  0.84567901]]
```

We see that the F1 score is 0.93 which means that the model has done a good job in its performance. The confusion matrix also assists in indicating the same with true positive, false positive, false negative and true negative rates.

F1 score and confusion matrix for Node2Vec algorithm in node classification is given below. The F1 score is 0.92 which also shows that the model is trained well.

```
0.9296840231419671
[[9.43410853e-01 3.33333333e-02 3.10077519e-03 2.01550388e-02]
 [1.46950771e-02 9.54445261e-01 3.01249082e-02 7.34753857e-04]
 [5.18134715e-03 6.04490501e-02 9.31778929e-01 2.59067358e-03]
 [9.92700730e-02 2.04379562e-02 2.91970803e-02 8.51094891e-01]]
```

F1 score and confusion matrix by using Node2Vec algorithm for embedding in node classification is given below. The F1 score is 0.88 which also shows that the model is trained better.

```
0.8802848242100578
[[0.92790698  0.03643411  0.01395349  0.02170543]
 [0.02680566  0.9091586   0.06105733  0.00297841]
 [0.02356902  0.07828283  0.8956229   0.00252525]
 [0.17384844  0.04309064  0.07875186  0.70430906]]
```

F1 score and confusion matrix for the proposed algorithm in node classification is given below. The F1 score is 0.95 which also shows that the model is trained well. From this, the proposed algorithm showed improved accuracy based on F1 score than the other algorithms

```
0.953215133753690
[[0.97739280  0.06439026  0.03019438  0.02546321]
 [0.03683542  0.98354782  0.03739235  0.02345336]
 [0.00532659  0.05368203  0.96329553  0.01859200]
 [0.18149174  0.00935623  0.03295432  0.91445331]]
```

VI. CONCLUSION

The proposed algorithm for latent representation learning is compared with the existing algorithms. A comprehensive study on the strength and weakness was made. Deepwalk captures structural information effectively using random walk and skip-gram model. Node2Vec introduces biased random walk which uses two parameters which controls the selection of node in each iteration of random walk. GraphSAGE uses inductive learning to generate embedding. It uses Graph Convolution Network for aggregating the information. Based on the experimental analysis using Face Book network, the proposed algorithm classifies the nodes more accurately than the existing algorithms.

REFERENCES

[1] Ozsel Kilinc, Ismail Uysal, "Learning Latent Representations in Neural Networks for Clustering through Pseudo Supervision and Graph-based Activity Regularization", International Conference on Learning Representations 2018.

[2] Jonathan Crabbe, Zhaozhi Qian, Fergus Imrie and Mihaela van der Schaar, "Explaining Latent Representations with a Corpus of Examples", 35th Conference on Neural Information Processing Systems (NeurIPS 2021).

[3] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "DeepWalk: Online Learning of Social Representations." KDD 2014.

[4] William L. Hamilton, Rex Ying, and Jure Leskovec. "Inductive Representation Learning on Large Graphs." NeurIPS 2017.

[5] Aditya Grover and Jure Leskovec. "node2vec: Scalable Feature Learning for Networks." KDD 2016.

[6] William L. Hamilton, Rex Ying, and Jure Leskovec. "Inductive Representation Learning on Large Graphs." NeurIPS 2017.

[7] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. In Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on, pages 475–486. IEEE, 2006.

[8] L. Tang and H. Liu. Leveraging social media networks for classification. Data Mining and Knowledge Discovery, 23(3):447–478, 2011.

[9] L. Tang and H. Liu. Scalable learning of collective behavior based on sparse social dimensions. In Proceedings of the 18th ACM conference on Information and knowledge management, pages 1107–1116. ACM, 2009.

[10] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, Qiaozhu Mei, "LINE: Large-scale Information Network Embedding", WWW, 2015.

[11] Bryan Perozzi, Rami Al-Rfou, Steven Skiena, "DeepWalk: Online Learning of Social Representations", KDD, 2014.

[12] Aditya Grover, Jure Leskovec, "node2vec: Scalable Feature Learning for Networks", In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[13] Kilinc, Ozsel, Uysal, Ismail, "Learning Latent Representations in Neural Networks for Clustering through Pseudo Supervision and Graph-based Activity Regularization", 2018.

[14] Thomas N. Kipf, Max Welling, "semi-supervised classification with Graph Convolutional Networks", ICLR, 2017.

[15] Seong Jin Ahn, MyoungHo Kim, "Variational Graph Normalized AutoEncoders", CIKM '21, November 1–5, 2021.