Shahid Bahonar University of Kerman
Journal of Mahani Mathematical Research
Print ISSN: 2251-7952 Online ISSN: 2645-4505

MULTI-LABEL NODE CLASSIFICATION IN HETEROGENEOUS NETWORKS USING GRAPH CONVOLUTIONAL NETWORKS

S. Bastami and MB. Dowlatshahi ✉

Article type: Research Article
(Received: 27 May 2024, Received in revised form 28 September 2024)
(Accepted: 01 November 2024, Published Online: 02 January 2025)

Abstract. This paper explores graph embedding techniques for effectively analyzing large, heterogeneous graphs with complex and noisy patterns. Graphs represent data through nodes (entities) and edges (relationships), and when dealing with large-scale data, effective search methods are crucial. Graph embedding helps evaluate node significance and transforms data into latent space representations. It also addresses challenges like handling multi-label data in heterogeneous networks, where nodes may have multiple labels describing complex concepts. Traditional methods struggle with such multi-label scenarios and fail to capture label dependencies. The paper introduces a Graph Neural Network (GCN)-based node embedding method, which extends traditional neural networks to graph data. GCNs allow the extraction of local features from nodes and their neighbors, making them useful for heterogeneous networks. By integrating label information into the embedding process, the method improves relationships between labels. The proposed approach transforms neighboring labels into continuous vectors, structured into a matrix for learning. This enhances the overall network embedding. The method outperforms previous techniques, demonstrating improved performance on real-world datasets, such as a 2.4% improvement on the IMDB dataset and 9.3% on the DBLP dataset. The paper discusses graph embedding techniques in the first section and explores the potential of multi-label embedding in non-uniform graphs, suggesting future research directions in the final section. The article's code link on GitHub can also be found at the following: https://github.com/sajadbastami/GCN.

Keywords: Graph embedding (GE), Multi-label (ML), Graph neural network (GNN), Heterogeneous network (HN), Weight learning (WL), Social Networks.
2020 MSC: 05C82.

1. Introduction

Machine learning predominantly relies on weight learning, representing instances through feature vectors and labels [1]. Traditional models focus on

✉ dowlatshahi.mb@lu.ac.ir, ORCID: 0000-0002-4862-0626
https://doi.org/10.22103/jmmr.2024.23506.1660
Publisher: Shahid Bahonar University of Kerman
How to cite: S. Bastami, MB. Dowlatshahi, *Multi-label node classification in heterogeneous networks using graph convolutional networks*, J. Mahani Math. Res. 2025; 14(2): 59-80.

observational learning, where each example signifies a distinct concept. In contrast, deep learning models frequently encounter datasets with multiple labels across various media, including YouTube videos, Instagram photos, newspaper articles, and genomic sequences. Multi-label node classification is crucial as it allows for the categorization of nodes with multiple labels in domains like machine learning, computer vision, and data mining. The primary problem is to develop effective techniques for enhancing the classification of nodes beyond a single label. The research questions guiding this study are:

(1) How can single-label classification algorithms be adapted for multi-label challenges?
(2) What are the limitations of current techniques, and how can they be overcome?
(3) How can deep learning facilitate multi-label classifications using structured data, social networks, and knowledge graphs?

Multi-label node classification finds its applications across various domains, including machine learning [2], computer vision [3], and data mining [4]. Multi-label classification involves predicting multiple labels for a single data point, unlike traditional classification, which predicts a single label. The main challenges include handling the exponential number of possible label combinations and capturing dependencies between labels. This discussion centers on the fundamental techniques for enhancing the classification of nodes beyond a single label. Consequently, algorithms designed for single-label classification are being adapted to address multi-label classification challenges, thereby supplanting older methods. These algorithms tackle multi-labels by transforming them into a single-label classification framework. However, certain learning techniques remain unsuitable for multi-label data.

In recent years, deep learning has facilitated the development of various classifications based on structured data, social networks, chemical molecules, and knowledge graphs [5]. Graphs depict interconnected objects, such as network graphs, chemical molecules, and knowledge graphs, all of which are linked [6]. Representational learning has expanded to include the multi-label learning model (MLL). Research in multi-label studies has revealed that MLL can be segmented into two distinct components: multi-label node classification (MLC) and label ranking (LR) [7].

Multi-label classification involves categorizing multiple independent classes, each representing different types. While these methods provided a baseline, they often failed to capture the interdependencies between labels, leading to suboptimal performance in complex datasets. Recent years have witnessed the integration of deep learning techniques in multi-label classification. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been employed to capture spatial and temporal dependencies in data [5]; [6]. Graph Convolutional Networks (GCNs) have further extended these capabilities by leveraging graph structures to enhance node classification [6]. Despite

these advancements, challenges remain in effectively integrating label information and graph structures.

To classify multi-labels, two phases are necessary: transforming the problem and ensuring algorithm compatibility [7]. Conversely, label ranking involves assigning a set of predetermined labels to a text and then creating a hierarchy based on those labels [8]. Graphs, commonly used data structures in computer science and related fields [9], present a significant challenge in graph-based machine learning—effectively integrating graph property information. The primary learning methods are categorized as supervised (or semi-supervised) and unsupervised. In unsupervised learning, graph nodes are positioned within a continuous vector space without relying on labels such as edges, subgraphs, or the entire graph. "DeepWalk" [10] emerged as an early attempt to adapt skip-gram models for graph embeddings, which are employed in training for multi-label classification.

To determine each node's embedding, DeepWalk employs NLP techniques and models probability distributions [11]. DeepWalk is an algorithm that learns latent representations of vertices in a graph by modeling a stream of short random walks, capturing neighborhood similarity and community membership. NLP techniques involve processing and analyzing natural language data to understand and generate human language, often using methods like word embeddings and neural networks. The algorithm uses unbiased random walks to enhance the probability of incorporating nodes into a graph. However, these methods often overlook the rich label information inherent in multi-label datasets. Future research should aim to develop models that can seamlessly integrate structural and label information, thereby enhancing the overall classification accuracy. Multi-label learning methods have received limited attention in graph-based systems [12].

For node embedding, auxiliary information is integrated to combine structural and non-structural data into node vectors [13]. In addition, knowledge graph information can assist in gauging the correlation between nodes and human perception. When regarded as supplementary information, this can improve the embedding of nodes within a meaningful space [14]; [15]. Some researchers have proposed the use of deep neural networks [16]; [17] to effectively learn representations of nodes in graph structures. A team [14] developed a deep natural language processing model for collaborative learning that incorporates node structure, content, and labels.

Analyzing the labels aids in understanding the representations of nodes within node embedding algorithms. Huang et al. [13] introduced a structural approach named Lane, which integrates labels into a featured network. To capture common meanings among nodes accurately, it is crucial for existing methods to consider the interplay of labels. Heterogeneous graphs, composed of various nodes and links, offer valuable insights when analyzed collectively. This paper addresses the challenge of extracting label information from nodes with multiple labels in networks. It emphasizes the multi-label classification of

nodes, where each node is labeled. The issue can be approached as graph-based observer learning by leveraging label information on the nodes.

Graph-based learning uses graph structures to represent and analyze data, capturing relationships and dependencies between entities. Graph Convolutional Networks (GCNs) extend traditional neural networks to graph data, enabling the extraction of local features from nodes and their neighbors. This paper utilizes Graph Neural Networks (GCNs) to transform neighboring labels into continuous vectors, which are then structured into a two-dimensional matrix for learning. The proposed method, based on a Graph Neural Network (GCN), effectively integrates label information into the embedding of heterogeneous networks, enhancing relationships between labels. Unlike traditional methods that often fail to handle multi-label data and capture label dependencies, this approach demonstrates superior performance with real-world data. However, it may face challenges in scalability and computational complexity compared to simpler, single-label transformation techniques. While significant progress has been made, there is a need for more comprehensive evaluations of existing methods.

Many studies focus on specific datasets or domains, limiting the generalizability of their findings. Additionally, the scalability of these methods to large-scale datasets remains an open question. Addressing these gaps will be crucial for advancing the field of multi-label node classification. The proposed multi-label node classification approach leverages deep learning models like CNNs, RNNs, and GCNs to enhance classification across various domains, including social networks and chemical molecules. Despite advancements in deep learning, challenges remain in integrating label information with graph structures effectively. Our approach utilizes Graph Convolutional Networks (GCNs) to transform neighboring labels into continuous vectors, structured into a two-dimensional matrix for learning. This method enhances label relationships in heterogeneous networks, showing superior performance with real-world data. Addressing these limitations can lead to more robust and effective multi-label classification.

Future research should focus on improving scalability and interpretability to address computational complexity and label dependencies. Practical implications include significant impacts on social networks, chemical analysis, and content categorization. This study focuses on enhancing multi-label node classification in heterogeneous graphs. Solving this problem enables evaluating connections between node labels and their neighbors, ensuring critical information is preserved for more precise graph analysis. Real-world applications include:

(1) Drug Side Effects and Discovery: Identifying potential drug interactions and side effects by analyzing chemical compound networks.

(2) Authorship Networks: Understanding relationships between authors and publications, which can assist in co-authorship predictions and expertise identification.
(3) Social Media Analysis: Categorizing user-generated content with multiple themes or topics, which aids in content recommendation and sentiment analysis.

This paper proposes three main contributions:

- Enhanced label representation: Our approach effectively captures the interdependencies between labels, leading to more accurate classifications.
- Improved scalability: We develop a scalable algorithm that can handle large-scale multi-label datasets efficiently.
- Enhanced interpretability: Our model provides insights into the decision-making process, enabling better understanding and analysis of the classification results.

1.1. **Scope of the article.** This article delves into the methods of embedding nodes with multi-labels within a heterogeneous graph. It elaborates on the concept of a multi-label graph, where each sample is linked to features and several labels. In such a network, the labels are interwoven with the topological structure, which complicates the content. The process of learning a multi-label graph representation poses challenges due to the distinctive traits that the nodes' labels exhibit.

1.2. **An overview of the article's structure.** The structure of this article is as follows: *Section 2* presents definitions and establishes the variables used. *Section 3* reviews related work, existing methods, and their classifications. Subsections *3-1*, *3-2*, and *3-3* detail the proposed approach. *Section 4* offers an analysis and comparison of the preemptive approach from a complexity standpoint. *Section 5* concludes with a discussion of challenges and future research opportunities.

1.3. **Definitions and concepts.** In this section, we will cover the fundamental concepts and definitions that are essential for understanding this article.
Definition 1. Information graph. A graph is $G = (V, E)$ where $v \in V$ is a multi-label node and $e \in E$ is an edge. Currently, we examine G using a multi-label node mapping function and an edge-type mapping function $f_v = V ß \tau^v$. Node $v_i \in V$ represents a specific type of node, which is represented by $f_v(v_i) \in \tau^v$. Similarly, for $e_{ij} \in E$, $f_e(e_{ij}) \in \tau^e$. τ^v and τ^e provide a list of types of nodes and edges with multi-labels. *Definition 2. A heterogeneous graph.* $G_h ete = (V, E)$ is a graph that $|\tau^e| > 1$ or $|\tau^v| > 1$.
Example 1. As shown in Fig 1, HG represents scientific data.
Definition 3. Multi-label classification. Multi-label classification (MLC) is an extended form of multi-class classification. MLC assigns multi-labels to samples simultaneously [18]. X represents the input vector, and $Y = y_i | i = 1, 2, \ldots, M$

represents the set of labels. When doing node training with multi-labels, you need to map a function f from the input vector x to the output space 2^Y. This function should determine a subset of the output space for each sample $x \in X$ in the input vector. Graph structures are maintained in a significant space through proximity criteria, revealing the correlation between nodes and edges, also known as the first order of proximity.

Definition 4. First-order proximity. There is an edge weight e_{ij} between nodes v_i and v_j, which is A_{ij}. The nodes connected with a greater weight are likely to be more similar. First-order proximity is the correlation between two connected nodes [19]. The vertices vi and vj, if $(v_j, v_i) \in E$, have a first-order proximity equal to equal to wi, j. which is the weight of the edge between nodes v_j and v_i. Otherwise, the first-order proximity between vi and vj will be zero. First-order proximity refers to the relationship between nodes that are close to each other. We are analyzing the similarity between neighboring nodes by comparing their proximity to second-order nodes. They have more similar neighbors if the second-order value between two nodes is higher.

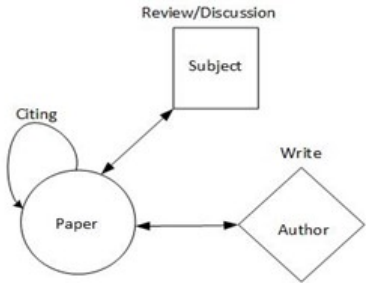

FIGURE 1. A graph illustrates heterogeneous scientific data.

Definition 5. Second-order proximity. The graph between nodes v_i and v_j shows the correlation between their neighbors. Vertices (v_j, v_j) in second-order proximity are counted if they have any familiar neighbors. It can also be calculated by considering the probability of a two-step transition between v_i and v_j. *Definition 6. Embedding heterogeneous graphs.* Using the heterogeneous input graph $G = (V, E)$ and the predetermined embedding dimensions d ($d \ll |v|$), in graph embedding, While preserving as many properties as possible, G is transformed into a d-dimensional vector. One can use proximity criteria such as first-order proximity and beyond to measure graph properties [19]. *Definition 7. An adjacency matrix.* The connections between the nodes are shown, and the adjacency matrix indicates if two nodes are adjacent through an edge [20].

Definition 8. Meta-path. The meta-path p is defined within a heterogeneous graph G. It is denoted as:

$$p = A_1 \square \xrightarrow{\perp R_1} A_2 \square \xrightarrow{\perp R_2} \ldots \xrightarrow{\perp R_i} A_{i+1},$$

where the types of nodes $A_1, A_2, \ldots, A_{i+1} \in A$ and the types of links $R_1, R_2, \ldots, R_i \in R$. The meta-path is used to illustrate relationships between entities. As illustrated in Table 1, a summary of the notations is presented.

Table 1 shows a Summary of Related Works.

TABLE 1. Summary of Related Works.

Notations	Explanations
V	The set of objects
E	The edges of the path
G_hete	A heterogeneous graph
x	Input space
Y	Set of labels
2^Y	Output space
e_{ij}	Edge weight
v	A node $v \in V$
Z	Latent space

2. Related Works

The real-world heterogeneous network revolves around individual instances and their interconnections. The multi-label classification process encompasses two stages: algorithm adaptation and problem transformation. Graphs containing semantic information are defined as follows. Numerous algorithms have been adapted from single-label to multi-label classifications. To address this, we will transform multi-label nodes into one or more nodes with a single label. Subsequently, we will employ a learning method appropriate for the existing single-label nodes to process the transformed data effectively. One such method is the binary relevance (BR) approach, analogous to the one-versus-all strategy used in multi-class scenarios. In BR, each label is modeled independently. However, this assumption may not hold in all real-world contexts. We also introduce classification chains (CC). An enhanced method, known as ensemble classifier chains (ECC), has been proposed. It acknowledges that the sequence of chains can influence their performance. CC links the classifiers in BR, where each link in the chain incorporates a feature space that encompasses the labels of all preceding links.

2.1. **Graph embedding.** The comparison encompasses graph embedding [21], the operation types for input and output, and the utilized attention mechanisms. The majority of embedding methods proposed are tailored for homogeneous networks. Depending on the scenario, the methods for monitoring and adapting inputs may differ. Homogeneous networks feature two embedding approaches: supervised and unsupervised. In the algorithms discussed, the input

graph is static and inductive. Often, the nodes and edges of a graph undergo changes through inductive adjustments. The recommended methods are adaptable, readily integrating new nodes, thereby rendering them scalable. While numerous embedding techniques exist for homogeneous graphs, these must be applicable to various network types, each with distinct characteristics. The challenges associated with such networks have been thoroughly investigated, as they are more pronounced in real-world applications. Embedding methods can be evaluated in the operation type settings, independent of the input data type [22]. These methods are categorized into four stages: node embedding, edge embedding, combined embedding, and entire graph embedding. The classification of embedding methods is based on the attention type they employ. Multi-label embedding techniques have been developed to manage extensive label spaces. These methods aim to accelerate computation and enhance the representation of conventional information by reducing the dimensions of the label space. This reduction is achieved by identifying models that are more distinctive than the label assignment matrix [23]. As a decision-maker in labeling, I have devised novel embedding methods that depend on statistical analysis, employing multivariate regressions and weak classifiers [23].

2.2. **GCN with a Structured Label Space.** A dependable method for addressing multi-class classification tasks is the integration of deep neural networks (DNNs) with graph convolutional networks (GCNs) [24]. This technique employs GCNs to encode the label structure and uses convolutional graph layers to derive features from latent variables, effectively interpreting label features. The entire network undergoes training as a deep learning model. The significance of neighboring nodes influences the probability value of each node in the graph. Every node or label possesses a vector representation. To ascertain the value of each label, this vector is transformed into a scalar. The input representation $z = f(x; _f)$ and each label y_i are embedded into a vector v_i. Node, in this case, the vector $h_i = [z; v_i]$, represents the label concatenation of the hidden input representation z and the label vector v_i. Between the output weights and the hidden representation of input z, the output feature vector u_i labeled y_i is encoded. The label score F_i is obtained by $F_i = u_i(z) + u_i(v_i)$, where u_i is the *i-th* row of U. To convey more detailed information, labels are encoded using vector representations [25].

2.3. **Label-Aware GCN.** When nodes and their neighbors share the same label, this is viewed positively; conversely, differing labels between them are viewed negatively. Edge classifiers refine graph structures by converting them into label-aware (LA) graph structures. This is achieved by eliminating negative neighbors and integrating unconnected yet positively labeled nodes as new neighbors. Enhanced GCN models can bolster their graph evaluation performance by training directly on the LA graph. In the analysis of LA graphs, the positive ratio serves as an indicator of the number of valuable neighbors, with a higher positive ratio correlating to improved GCN performance. This concept

posits that the primary categorization significantly influences graph classification effectiveness by establishing the fundamental criteria for a valuable edge categorization. Rigorous experiments on benchmark datasets have shown that LAGCN markedly improves the performance of existing GCN nodes, especially when the underlying graph exhibits a low positive ratio. Regarding node classification, LAGCN has been demonstrated to significantly boost the efficacy of GCN models, as detailed in the study referenced in [26]. These node classification tests were conducted using the benchmark dataset.

2.4. **Classification of multi-label graph nodes using semi-supervised graph embedding.** ML-GCN is a multi-label classification method based on GCN. It integrates multi-labels and nodes within the same space, enabling simultaneous label-label and label-node correlations. ML-GCN features two embedding sets: one for labels and another for both nodes and labels. GCN is utilized to merge node features with graph information, embedding labels and nodes into a cohesive vector space. Subsequently, a label matrix is generated at random, with each label vector signifying a distinct label type. Throughout the ML-GCN training phase, label and node vectors are amalgamated [27]. The skip-gram model is employed to discern correlations between node labels and label labels. Each neural network layer uses graph convolution operations to forge robust non-linear correlations among nodes. Label correlation is quantified by treating each label as a vector. The addition of closely associated labels reduces the distance between two nodes in the latent space. Training multi-label classifiers involves displaying all nodes on a graph and predicting labels for the unlabeled ones. The ML-GCN method applies a sigmoid layer to facilitate downstream learning [28].

3. Preliminaries

In this section, we define the notation that will be consistently employed throughout this study, examine the principles underlying heterogeneous graph classification, offer a concise overview of Heterogeneous graph neural network, and introduce the social networks case study chosen for our analysis.

3.1. **Notation.** A list of mathematical symbols for a heterogeneous graph and vector representations for its nodes and edges is provided. The heterogeneous graph $G = (v, \varepsilon)$ contains the set of vertices v and the set of links ε. There is a node mapping function $\phi(v) : v \to A$ and an edge mapping function $\phi(\varepsilon) : \varepsilon \to R$. Here, A and R refer to the type of node and edge, respectively. It is given that $A + R \geq 3$.

Each node contains various features and content, which may differ from one another. Therefore, each object $v \in V$ belongs to a specific class of object $\phi(v) \in A$, and each link $e \in \varepsilon$ is part of a special relationship $\phi(e) \in R$. Throughout this article, we will include additional symbols as needed.

3.2. **Heterogeneous graph neural network.** In heterogeneous neural networks, operations are conducted in four phases to extract meaningful information: (1) Extracting the heterogeneous neighbors. (2) Encoding the heterogeneous information of the node. (3) aggregating the heterogeneous neighbors. (4) Determining the objective function and training the network. To comprehend the characteristics of each node through their first-order neighborhood, it is necessary to ascertain the neighborhood information for each node. Unlike homogeneous graphs, it is not always feasible to extract feature information in a similar manner for heterogeneous graphs. Consequently, to access the content of each node, it is crucial to adjust the types and dimensions of features. This process entails the extraction of diverse content, including the embedded content within nodes. Neural networks are capable of amalgamating the characteristics of dissimilar neighbors.

4. Method proposed

The foundational step in graph embedding techniques, such as DeepWalk [15] and node2vec [8], is the employment of a random walk process to capture the relationships between nodes. Random walks facilitate the embedding of the entire graph into node paths that reflect neighborhood relations. This method retains structural information during the embedding of all training instances, from context nodes to target nodes [29]. Node labels provide insights into the nodes themselves. To gauge the proximity of nodes, one may rely solely on individual and concealed labels. For an enhanced node embedding, it is imperative that the representations of nodes and their labels coincide within the representation space. Ignoring a node's multi-labels can lead to the omission of some features, as current methods typically adopt the simple paradigm of graph label embedding (SLNE), which does not consider all labels [13]; [15]. This section introduces a learning model that accounts for label correlation, aiming to resolve the challenges of embedding graph nodes in multi-label classification (MLC) [23]. Our proposed algorithm encompasses feature extraction, node encoding, and GCN training. In heterogeneous graphs, nodes with multiple labels form distinct connections based on their types. Classifying nodes with multiple labels necessitates the analysis of meta-paths and their weighted correlations [30] to accurately predict the node's labels. We extract meta-paths to illustrate the relationships among different nodes in the network and then determine path weights through a learning process. Our approach identifies each node's meta-path and assigns weights to these meta-paths according to the neighborhood matrix's outline. Subsequently, we assess the correlation between node labels by applying these weights. Nodes are classified based on the assigned weights to their meta-paths, enabling us to gauge the connectivity level between nodes and elucidate these paths. To tackle the complexity of multi-label nodes, we compute the correlations between neighbors via meta-paths.

4.1. **Analyzing heterogeneous neighbors.** To comprehend the correlation between nodes and their labels, it is essential to analyze the semantic connections among the nodes within their neighborhoods. Meta-path information is employed to assess the neighbors of nodes $p(1-2)$, beginning with node p_1. As depicted in Fig 2, nodes p_1 and p_2 possess neighbors with multiple labels [31], indicating that a path from p_1 to p_2 is denoted as $p(1.2)$. This approach facilitates improved communication between nodes and yields extensive information from the paths. We establish a link between the node type that shares a label and the adjacent node class to maximize the utilization of path information.

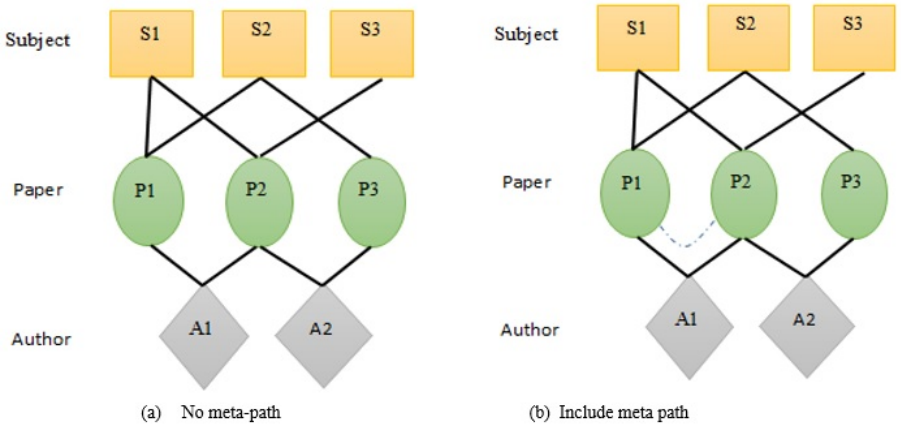

FIGURE 2. Analyzing heterogeneous neighbors for path extraction.

4.2. **Training with Meta-path weights.** By examining the diverse meta-paths within the heterogeneous graph, we can enhance the accuracy of multi-label node classification by training the weights for each meta-path. Each meta-path offers a distinct assessment of node classification, enabling the precise calculation of weights based on learned information. Thus, understanding the importance of a meta-path's weight in the heterogeneous graph is beneficial, as it provides significant insights. The heterogeneous network structure establishes a correlation between node labels by learning meta-path weights, effectively addressing the multi-label problem.

4.3. **Learning the weight of neighborhood matrices and meta-path.** The weight for learning is determined by the meta-path obtained from the neighborhood [32] of each node. For meta-path $P_L = VS, \ldots, E$, Source node V_s and destination node V_t are connected by path P_L representing different types of nodes. The correlation between node V_s and V_t under path P_L is

computed in Eq.(1).

$$\text{Cor}(V_s, V_t \mid \overline{P_L}) = \frac{z(V_s, V_t)\left(\frac{1}{\deg(V_s)} + \frac{1}{\deg(V_t)}\right)}{\frac{1}{\deg(V_s)}\sum_i z(V_i, V_s) + \frac{1}{\deg(V_t)}\sum_j z(V_t, V_j)} \tag{1}$$

Here, $z(V_s, V_t)$ represents the eigenvalues of the Neighborhood Matrix $Z(P_L)$ and, $deg(V_s)$ represents the number of edges connected to a vertex V_s. It is possible to obtain a correlation matrix between different vertices, shown in Table 2. This table represents a node proximity matrix for a heterogeneous graph, where the entries indicate the proximity between nodes based on meta-paths. The loss function of the weight parameter is shown in Eq.(2).

$$L(\boldsymbol{\theta}) = \frac{1}{2}\sum_{(V_i,V_j)\in V_d}\left\|1 + S(V_i, V_j) - \sum_{k=1}^{n}\theta_k \cdot \text{Cor}(V_s, V_t)\right\| + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2 \tag{2}$$

$$S(V_i, V_j) = \begin{cases} 1 & \text{if } V_i \neq V_j, \\ -1 & \text{otherwise.} \end{cases} \tag{3}$$

In the data set V_d, the nodes V_i and V_j have the classification label, and a regularization parameter is called [λ]. Learning the weight of each meta-path is shown in Eq.(4).

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \theta_k} = 0 \tag{4}$$

(5)

$$\theta_k = \frac{\sum_{(V_i,V_j)\in V_d} S(V_i, V_j) \cdot \text{Cor}(V_s, V_t) \cdot (1 + S(V_i, V_j)) - \sum_{k=1}^{n}\theta_k \cdot \text{Cor}(V_s, V_t)}{\lambda + \sum_{(V_i,V_j)\in V_d} S(V_i, V_j)^2 \cdot \text{Cor}(V_s, V_t)^2}$$

$$w_i = \frac{\theta_k}{\sum_{r=1}^{n}\theta_r} \tag{6}$$

$$\text{Label} = \frac{\sum_{V_j\in\{V_s,V_t\}} r(i,j)}{\sum_{V_j\in\{V_s,V_t\}} r(i,j) + \sum_{V_k\in V_d} r(k,d)} \tag{7}$$

TABLE 2. Heterogeneous graph node proximity matrix based on meta-paths $\overline{P_L}$.

	$\mathbf{v_1}$	...	$\mathbf{v_i}$	...	$\mathbf{v_n}$
$\mathbf{v_1}$	0	...	1	...	0
...	...	...	...	...	...
$\mathbf{v_i}$	0	...	1	...	p
...	...	...	...	...	...
$\mathbf{v_n}$	0	...	1	...	p

Table 3 presents the correlation coefficients among various vertices in a network, where each value represents the strength and direction of the linear relationship between pairs of nodes.

Working with heterogeneous networks can be challenging when attempting to associate an object with a labeled object that possesses multiple labels. Algorithm 1 outlines the weight parameters of the meta-paths upon which the learning algorithm is predicated. The weight of each meta-path is ascertained through effective learning from training datasets, and this weight is instrumental in calculating the correlation between each object in the dataset. The algorithm learns weight parameters for a graph using GCN weight vectors. It initializes parameters $_k i = {}_{1,2}, \ldots, {}_i$ and iteratively updates them by calculating values using specific equations, such as $_k$ by Eq.(5). Labels are selected by Eq.(7) and the training set V_d. is updated in each iteration. Finally, the weight parameters are learned using Eq.(7). In addition to the weight learning process, this algorithm leverages the structure of the graph and the relationships between nodes to enhance the accuracy of multi-label classification. By iteratively refining the parameters and updating the training set, it ensures that the learned weights are well-suited to capture the complex patterns in the data. This approach is particularly effective in heterogeneous networks where different types of nodes and edges exist, making it a robust solution for various real-world applications. Here $r_{(i,j)}$ indicates the correlation between nodes V_i and V_j that if there is an edge, it will be adjusted to 1 a meta-path. Otherwise, it will be changed to 0. Thus, the object is added to the class set, and a label is assigned. As a result, multi-label nodes can be extended in heterogeneous networks.

TABLE 3. Heterogeneous graph node proximity matrix based on meta-paths $\overline{P_L}$.

Cor(V_s, V_t)	$\mathbf{V_1}$	$\mathbf{V_2}$	...	$\mathbf{V_i}$	...	$\mathbf{V_n}$
$\mathbf{V_1}$	1	0.4	...	0.4	...	0.5
$\mathbf{V_2}$	0.4	1	...	...	...	0.3
...	...	...	...	...	...	...
$\mathbf{V_i}$	0.4	...	...	1	...	0.2
$\mathbf{V_n}$	0.5	0.3	...	0.2	...	1

Algorithm 1: Structured graph learning using GCN weight vectors

Algorithm 1: Structured graph learning using GCN weight vectors

$G = (V, E)$; {λ: This parameter controls the regularization process; V_d: Set of data with nodes that have multi-labels} {W_d: $W_1, W_2, \ldots, W_i$}
Begin
Initialize $\theta_k = \{\theta_1, \theta_2, \ldots, \theta_i\}$, $k = 1, 2, \ldots, n$;
$t \leftarrow 0$;
While $t < N$ //The number N represents the number of iterations// **Compute** θ_k (using Eq. (5)), $k = 1, 2, \ldots, n$;
Select labels by Eq. (7) and update the training set V_d;
$t \leftarrow t + 1$;
By using Eq. (6), calculate the weight parameter learning;
End

5. The classification of multi-label nodes in heterogeneous networks using the proposed method and weight learning

Meta-paths represent connections between nodes that carry multiple labels within a heterogeneous graph, as detailed in reference [33]. We incorporate the neighborhood matrix into each meta-path and include them in our collection. Fig 3 depicts the node classification structures with multiple labels in heterogeneous graphs. A meta-path serves to ascertain the neighborhood matrix among nodes in a heterogeneous graph [26]. For the meta-path P_L, A neighboring matrix is constructed by examining the connections in a heterogeneous network, as shown in Table 3. We create a neighboring matrix for each meta-path type to analyze the graph structure. Then, we use information from the nodes' features matrix to classify the nodes with multi-labels. The limited research on multi-label node classification is largely due to the shortage of benchmark multi-label graph datasets. Although advancements have been primarily shown in multi-class classification, the more realistic scenario where each node can have multiple labels has been overlooked. The main obstacle in conducting detailed studies on multi-label node classification is the lack of publicly accessible multi-label graph datasets [34]. The complexity and high dimensionality of multi-label data make it challenging to obtain accurate label sets in real-world applications. Noisy label data can negatively impact the model's classification performance [35].

Fig 3 illustrates the process of classifying nodes in heterogeneous networks using multi-labels based on meta-paths. This process can be understood through the following five steps: 1. Identification of Node Categories: a) The procedure begins by classifying the nodes into three unique categories: "Subject," "Heterogeneous graphs," and "Author." b) Each category comprises nodes (e.g., S1, S2, S3 for topic) that are linked together, illustrating the relationships within that category. 2. Single-Label Classification: a) The initial kernel is the nodes

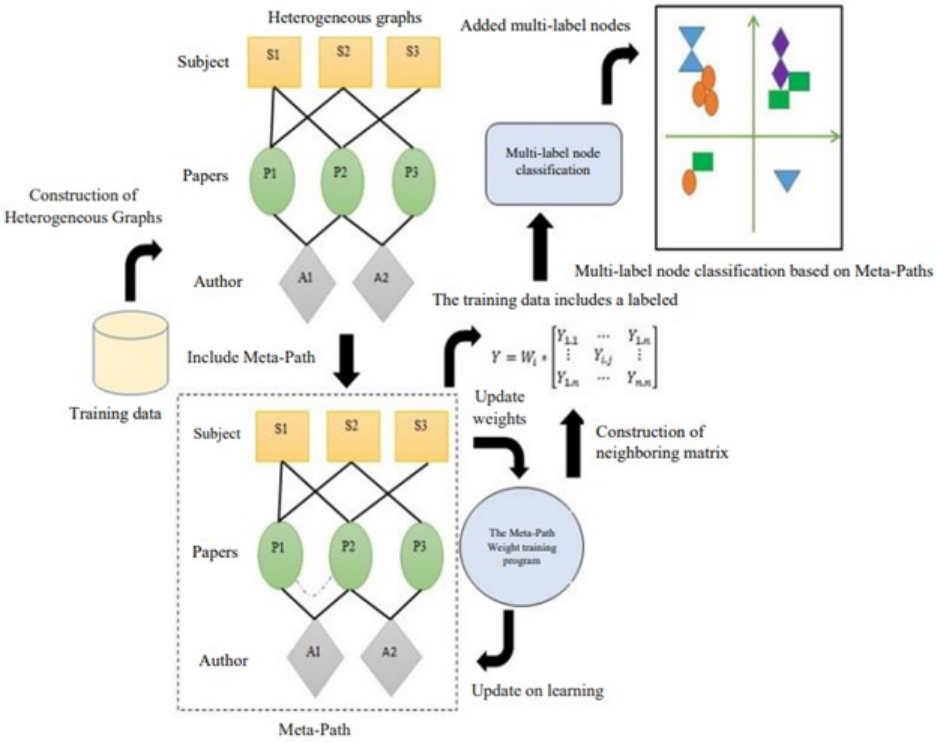

FIGURE 3. The stages of the projection function for mapping to a safe mode to reduce risk factors.

with a single label that represents an initial classification method. 3. Transition to Multi-Label Classification: a) The subsequent scheme shows nodes with multi-labels, illustrating the transition to a more complex classification system where nodes can possess multi-labels. 4. Analyzing Multi-Label Data: a) A phase starts with the creation of an adjacency matrix and the extraction of features from the training data using a GCN. b) This part describes the steps for data processing and weight updating during the classification procedure. 5. Meta-Path Algorithm: a) It generates a weight learning matrix from the training data and conducts multi-label classification within the heterogeneous network utilizing meta paths. b) meta-path plays a key role in establishing connections between various node types, which is essential for multi-label classification.

6. Experiments

IMDB: In an online database, you can find information about movies and TV shows, such as actors, production teams, and storylines. After undergoing data pre-processing, the IMDB dataset now includes 4278 movies, 2081 directors, and 5257 actors. Based on their genre information, movies are categorized into three classes - action, comedy, and drama. Additionally, a set of fictional keywords is provided for each film. Within the nodes, 9.35% (400) are associated with film nodes on semiconductor learning models, and the remaining 81.30% (3478) are related to semiconductor learning model nodes. DBLP: This is a website that provides computer science bibliographies. Once pre-processing is complete, the DBLP dataset will have 4057 authors, 14328 articles, 7723 terms, and 20 sites. The authors represent four research areas: databases, data mining, artificial intelligence, and information retrieval. A set of related keywords is used to describe authors and their papers. For their study on semiconductor learning models, the authors divided their nodes into three sets: educational, validation, and experimental. These sets contained 400 nodes each (9.86% of the total nodes) and 3257 nodes (80.28%) in the experimental set. Statistics are summarized from the dataset in Table 4.

Table 4. Heterogeneous graph node proximity matrix based on meta-paths ($\overline{P_L}$).

Meta-paths	**Edges**	**Nodes**	**Datasets**
FIA, FAF, IFI, IFAFI, AFA, AFIFA	#Film(F): 4,278, #Initiator(I): 2,081, #Assistant(A): 5,257, #F-I: 4,278, #F-A: 12,828	11,616	IMDb
IAI, IAIAI, IAVAI	#Inventor(I): 4,057, #Article(A): 14,328, #Item(I): 7,723, #Venue(V): 20, #I-A: 19,645, #A-I: 85,810, #A-V: 14,328	25,607	DBLP

Experiments were conducted on a system equipped with an Intel® Core™ i7-10750H CPU @ 2.60 GHz, 16.0 GB DDR4 memory, and a Windows 11 Home 64-bit operating system (x64 processor). Table 5 indicates that the accuracy of classification algorithms has improved by approximately 29.5% with the expansion of multi-label data in heterogeneous graphs. The most precise results are obtained by utilizing meta-paths for label classification in heterogeneous graphs. Fig 4 demonstrates the correlation of multi-label nodes through meta-paths. The application of three distinct classification algorithms has led to improved accuracy in two datasets, DBLP and IMDB. The graphs show precision for varying training data proportions (2% to 10%). GCN consistently achieves

the highest precision across both datasets, followed by SVM and Neural Network. KNN and Random Forest have lower precision but show improvements with more training data.

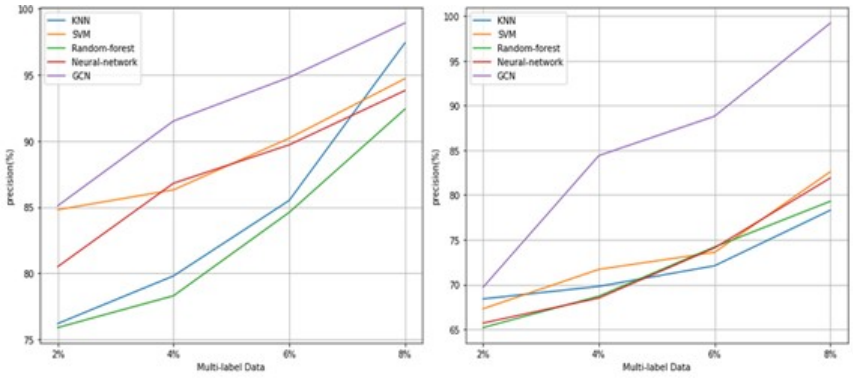

FIGURE 4. Evaluation of the efficiency of classification algorithms based on different meta-paths.

6.1. **Analyses of different classification methods.** Utilizing the neighborhood matrix of each path in heterogeneous graphs, we classify each node by employing three methods on the dataset. The meta-path symbolizes the relationship between its influence on the classification of multi-label nodes in the heterogeneous graph and the outcomes in the specified dataset, which are detailed with corresponding labels in Table 5. The accuracy of multi-label classification is enhanced as more significant label data is incorporated. This table compares the accuracy of different classifiers (KNN, SVM, Random Forest, Neural Network, and GCN) on IMDB and DBLP datasets with varying training data percentages. GCN consistently shows the highest accuracy across both datasets. SVM and Neural Network also perform well, especially with more training data. KNN and Random Forest improve with more data but generally have lower accuracy compared to GCN, SVM, and Neural Network.

As the dataset size increases, the performance accuracy improves. For example, when evaluating the IMDB dataset, the GCN method demonstrates an accuracy of approximately 98.9% within the 2% to 8% range. Furthermore, the DBLP dataset achieves the highest accuracy, reaching up to 99.8%. Fig 5 displays the ROC curves representing the favorable rates of various datasets, encompassing both accurate and inaccurate results. It has also been noted that the classification performance on the DBLP dataset is markedly different from that of predicting class labels in the IMDB dataset. The image shows ROC (Receiver Operating Characteristic) curves for DBLP and IMDB datasets. The x-axis represents the False Positive Rate, and the y-axis represents the True Positive Rate, both ranging from 0 to 1. The curves illustrate the trade-off

TABLE 5. An overview of the accuracy of different classifiers on IMDB and DBLP.

Methods	**IMDB (%)**				**DBLP (%)**			
	2%	**4%**	**6%**	**8%**	**2%**	**4%**	**6%**	**8%**
KNN	76.2	79.8	85.5	97.4	68.4	69.8	72.1	78.3
SVM	84.8	86.3	90.2	94.7	67.3	71.7	73.6	82.6
Random-forest	75.9	78.3	84.6	92.4	65.2	68.7	74.2	79.3
Neural-network	80.5	86.8	89.7	93.8	65.7	68.5	74.1	81.9
GCN	85.1	91.5	94.8	98.9	69.7	84.4	88.8	99.2

between true positive and false positive rates for different threshold settings. The graph helps evaluate the performance of binary classification systems on these datasets. Training typically involves using between 10% to 80% of the

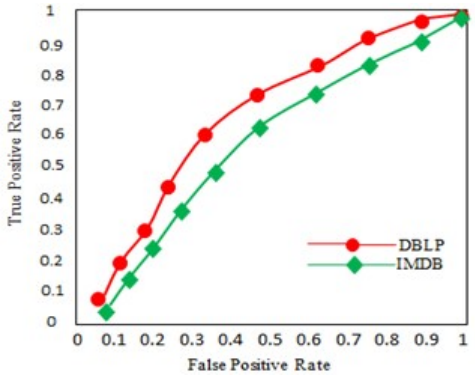

FIGURE 5. OC curves on DBLP, IMDB.

available data. Generally, having a larger number of training samples leads to enhanced performance in experiments. Algorithms are evaluated based on their micro-F1 score performance. Table 6 displays the experimental Micro-F1 results for the DBLP and IMDB datasets. Table 6 compares the micro-F1 scores of various methods on IMDB and DBLP datasets with different training data percentages. The proposed method consistently outperforms other methods, especially on the DBLP dataset, achieving the highest scores. Metapath2vec and Gat also perform well, but the proposed method shows superior accuracy. Node2Vec, DeepWalk, SLNE, and LINE have lower and relatively stable scores across different training percentages. GCNs effectively capture local graph structures, showing consistent performance improvements with increased labeled data. However, the proposed method outperforms GCNs, especially on the DBLP dataset, demonstrating superior robustness and effectiveness.

TABLE 6. Analyses of the proposed method in comparison with other micro-F1-based basic methods.

Methods	**IMDB (%)**				**DBLP (%)**			
	2%	**4%**	**6%**	**8%**	**2%**	**4%**	**6%**	**8%**
Node2Vec	31.2	31.2	31.2	31.2	31.2	31.2	31.2	31.2
DeepWalk	28.1	27.8	26.5	26.6	25.8	25.9	24.2	24.7
SLNE	30.8	30.8	30.8	30.8	30.8	29.9	29.9	29.9
LINE	31.6	31.6	31.6	30.9	30.8	29.4	28.7	28.7
Metapath2vec	49.4	50.1	52.3	52.1	93.4	93.9	94.2	94.7
GAT	56.2	57.1	58.4	58.6	77.2	80.3	81.8	83.2
GCN	52.7	53.8	54.5	55.0	78.6	79.9	84.7	88.1
Proposed Method	56.1	57.7	59.2	60.4	96.1	96.5	97.2	97.4

These results establish a definitive connection between the network structure and the labels. Utilizing labels allows for more straightforward access to hidden representations. A method that accounts for the learning weight of the meta-path has shown substantial progress and surpasses other techniques. The class labels in the training dataset are presented in various formats, enabling diverse analytical approaches. The superiority of our method is attributed not to disparate information sources but to the embedding of graph node labels within our meta-path framework, facilitating a more comprehensive understanding. By assigning effective weights to the meta-paths, the proposed method enhances the correlation among nodes, leading to a more precise latent representation and classification of multi-label nodes. As indicated, all methods have shown a relationship between the network structure and the labels. Given the network's diversity, expanding the dataset size could further improve the algorithms' accuracy.

6.2. **A study of the performance impact of heterogeneous graph neural networks.** A proficient latent space embedding technique should effectively encapsulate the most crucial and distinctive features of nodes. This section compares various methods of differentiating nodes by classifying multilayered networks. The proposed method surpasses DeepWalk, Node2vec, SLNE, and LINE in the dataset, as demonstrated by the superior Micro-F1 scores across various training ratios. An increase in the training ratio correlates with enhanced accuracy of the proposed method. In comparison to DeepWalk, Node2vec, SLNE, and LINE, the proposed method exhibits a higher accuracy level, boosting classification performance by 43% to 53%. We have established a correlation between node labels and meta-path weights using the proposed method, leading to a significantly improved feature representation. Employing nodes in a neighborhood model proves highly effective in accurately

depicting meta-paths between nodes, even in the absence of structural information during the modeling process. By leveraging the graph data structure in the proposed method, a more optimal model is achieved compared to vector-based approaches. The proposed method has increased the model's accuracy, proving to be more efficient in utilizing meta-path-based embeddings than the initial methods. In calculating the neighborhood matrix between nodes, the proximity of features within meta-paths is crucial. It is important to recognize that each method in the educational dataset accesses and manipulates class nodes distinctively. The recommended approach embeds the network structure and meta-paths into a latent space to discern the features of each node. Our proposed method is capable of representing all nodes in a network, both structurally and textually. This approach is more efficacious compared to other methods.

7. Conclusion and Future Directions

General methods for integrating graphs from different domains and incorporating other features have received considerable attention. However, the applications of graph embedding have received less attention. In this article, we will discuss a technique for embedding multi-label data in graphs while focusing on the concepts of data mining, machine learning, and natural language processing. We will focus on enhancing networking applications as we look to the future. This includes improving natural language processing, information recovery models, and social networks. However, there are still challenges to be addressed regarding multi-label data categories. Despite advancements in graphic learning methods for classifying nodes or predicting links, these challenges still need to be addressed. In heterogeneous graph information, node classification is performed using several semantic paths. Different paths are created based on the Meta-path Neighbor Matrix, and the weight of the meta-path is determined. According to the experimental results using the DBLP and IMDB data sets, better performance is achieved with more paths. Enriched connections between nodes are possible in a heterogeneous graph. The proposed method shows an improvement of up to 2.4% on the IMDB dataset and up to 9.3% on the DBLP dataset compared to other methods, demonstrating its superiority in handling multi-label data. Currently, researchers can explore and tackle various research paths and challenges, including the following topics: (1) Detection and classification of nodes using high-performance GPU computing or other parallelization techniques. (2) Non-Euclidean embedding. (3) Collections of high quality. (4) A large-scale application environment can be achieved using the proposed method. (5) The detection of hostile attacks on nodes requires further study. (6) Enhancing the classification system's accuracy, robustness, and consistency by integrating the label-based method with a feature graph. (7) Reliability issues in multi-label data embedding.

References

[1] Zhang, M.-L. and Z.-H. Zhou, A review on multi-label learning algorithms. IEEE transactions on knowledge and data engineering, 2013. 26(8): p. 1819-1837, doi://doi.org/10.1109/TKDE.2013.39.

[2] Read, J., et al., Classifier chains for multi-label classification. Machine learning, 2011. 85(3): p. 333-359, doi://doi.org/10.1007/s10994-011-5256-5.

[3] V. Chauhan and A. Tiwari, "Randomized neural networks for multilabel classification," Appl. Soft Comput., vol. 115, p. 108184, Jan. 2022, doi://doi.org/10.1016/j.asoc.2021.108184.

[4] M.A. Tahir, J. Kittler, and A. Bouridane, "Multilabel classification using heterogeneous ensemble of multi-label classifiers," Pattern Recognit. Lett., vol. 33, no. 5, pp. 513-523, Apr. 2012, doi://doi.org/10.1016/j.patrec.2011.10.019.

[5] Tsitsulin, A., et al., Graph clustering with graph neural networks. arXiv preprint arXiv:2006.16904, 2020, doi://doi.org/10.48550/arXiv.2006.16904.

[6] Kipf, T.N. and M. Welling, Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016, doi://doi.org/10.48550/arXiv.1609.02907.

[7] You, J., Z. Ying, and J. Leskovec, Design space for graph neural networks. Advances in Neural Information Processing Systems, 2020. 33, doi://doi.org/10.48550/arXiv.2011.08843.

[8] Gibaja, E. and S. Ventura, A tutorial on multilabel learning. ACM Computing Surveys (CSUR), 2015. 47(3): p. 1-38, doi://doi.org/10.1145/2716262.

[9] Wu, Z., et al., A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems, 2020. 32(1): p. 4-24, doi://doi.org/10.1109/TNNLS.2020.2978386.

[10] Kumar, V., et al., Multi-label classification using hierarchical embedding. Expert Systems with Applications, 2018. 91: p. 263-269, doi://doi.org/10.1016/j.eswa.2017.09.020.

[11] Vembu, S. and T. Gärtner, Label ranking algorithms: A survey, in Preference learning. 2010, Springer. p. 45-64, doi://doi.org/10.1007/978-3-642-14125-6-3.

[12] Liu, S., et al., Query2Label: A Simple Transformer Way to Multi-Label Classification. arXiv preprint arXiv:2107.10834, 2021, doi://doi.org/10.48550/arXiv.2107.10834.

[13] Hamilton, W.L., R. Ying, and J. Leskovec, Representation learning on graphs: Methods and applications. arXiv preprint arXiv:1709.05584, 2017, doi://doi.org/10.48550/arXiv.1709.05584.

[14] Mikolov, T., et al. Distributed representations of words and phrases and their compositionality. in Advances in neural information processing systems. 2013, doi://doi.org/10.48550/arXiv.1310.4546.

[15] Perozzi, B., R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014, doi://doi.org/10.1145/2623330.2623732.

[16] Huang, X., J. Li, and X. Hu. Label informed attributed network embedding. in Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. 2017, doi://doi.org/10.1145/3018661.3018667.

[17] Shi, M., Y. Tang, and X. Zhu, MLNE: Multi-label network embedding. IEEE transactions on neural networks and learning systems, 2019. 31(9): p. 3682-3695, doi://doi.org/10.1109/tnnls.2019.2945869.

[18] Guo, Z., et al., A two-level topic model towards knowledge discovery from citation networks. IEEE Transactions on Knowledge and Data Engineering, 2013. 26(4): p. 780-794, doi://doi.org/10.1109/TKDE.2013.56.

[19] Zhang, L., et al., A modified stochastic neighbor embedding for multi-feature dimension reduction of remote sensing images. ISPRS journal of photogrammetry and remote sensing, 2013. 83: p. 30-39, doi://doi.org/10.5194/isprsannals-I-3-395-2012.

[20] Szymański, P., T. Kajdanowicz, and N. Chawla, LNEMLC: Label network embeddings for multi-label classification. arXiv preprint arXiv:1812.02956, 2018, doi://doi.org/10.48550/arXiv.1812.02956.
[21] Sun, X., et al., Label embedding network: Learning label representation for soft training of deep networks. arXiv preprint arXiv:1710.10393, 2017, doi://doi.org/10.48550/arXiv.1710.10393.
[22] Cai, H., V.W. Zheng, and K.C.-C. Chang, A comprehensive survey of graph embedding: Problems, techniques, and applications. IEEE Transactions on Knowledge and Data Engineering, 2018. 30(9): p. 1616-1637, doi://doi.org/10.48550/arXiv.1709.07604.
[23] R. Rastogi and S. Mortaza, "Multi-label classification with Missing Labels using Label Correlation and Robust Structural Learning," Knowl.-Based Syst., vol. 229, p. 107336, Oct. 2021, doi://doi.org/10.1016/j.knosys.2021.107336.
[24] I. Ullah, M. Manzo, M. Shah, and M. G. Madden, "Graph convolutional networks: analysis, improvements and results," Appl. Intell., vol. 52, no. 8, pp. 9033-9044, Jun. 2022, doi://doi.org/10.1007/s10489-021-02973-4.
[25] Meihao Chen, Zhuoru Lin, Kyunghyun Cho. Graph Convolutional Networks for Classification with a Structured Label Space. arXiv preprint arXiv: 1710.04908, 2017, doi://doi.org/10.48550/arXiv.1710.04908.
[26] Z. Li, Y. Liu, Z. Zhang, S. Pan, J. Gao, and J. Bu, "Semi-supervised classification with graph convolutional networks,". arXiv preprint arXiv:1609.02907, 2016, doi://doi.org/10.48550/arXiv.1609.02907.
[27] Y. Wu, Y. Song, H. Huang, F. Ye, X. Xie, and H. Jin, "Enhancing Graph Neural Networks via auxiliary training for semi-supervised node classification," Knowl.-Based Syst., vol. 220, p. 106884, May 2021, doi://doi.org/10.1016/j.knosys.2021.106884.
[28] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016, doi://doi.org/10.48550/arXiv.1609.02907.
[29] Shi, M., et al., Multi-Label Graph Convolutional Network Representation Learning. IEEE Transactions on Big Data, 2020, doi://doi.org/10.1109/TBDATA.2020.3019478.
[30] Y. Du, W. Guo, J. Liu, and C. Yao, "Classification by multi-semantic meta path and active weight learning in heterogeneous information networks," Expert Syst. Appl., vol. 123, pp. 227-236, Jun. 2019, doi://doi.org/10.1016/j.eswa.2019.01.044.
[31] Y. Xiao, P. Quan, M. Lei, and L. Niu, "Latent neighborhood-based heterogeneous graph representation," Neural Netw., vol. 154, pp. 413-424, Oct. 2022, doi://doi.org/10.1016/j.neunet.2022.07.028.
[32] L. Zong, X. Zhang, H. Yu, Q. Zhao, and F. Ding, "Local linear neighbor reconstruction for multi-view data," Pattern Recognit. Lett., vol. 84, pp. 56-62, Dec. 2016, doi://doi.org/10.1016/j.patrec.2016.08.002.
[33] N. Tong, Y. Tang, B. Chen, and L. Xiong, "Representation learning using Attention Network and CNN for Heterogeneous networks," Expert Syst. Appl., vol. 185, p. 115628, Dec. 2021, doi://doi.org/10.1016/j.eswa.2021.115628.
[34] T. Zhao, N. T. Dong, A. Hanjalic, and M. Khosla, "Multi-label Node Classification On Graph-Structured Data," 2023, doi: 10.48550/ARXIV.2304.10398.
[35] Z. Qin, H. Chen, Y. Mi, C. Luo, S.-J. Horng, and T. Li, "Multi-label Feature selection with adaptive graph learning and label information enhancement,"Knowledge-Based Systems, vol. 285, p. 111363, Feb. 2024, doi: 10.1016/j.knosys.2023.111363.

Sajad Bastami
Orcid number: 0009-0008-1229-2887
Department of Computer Engineering
Kordestan University
Sanandej, Iran
Email address: sajad.bastami@uok.ac.ir

Mohammad Bagher Dowlatshahi
Orcid number: 0000-0002-4862-0626
Department of Computer Engineering
Lorestan University
Khoramabad, Iran
Email address: dowlatshahi.mb@lu.ac.ir