IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 February 2024, accepted 24 April 2024, date of publication 2 May 2024, date of current version 10 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3396209

RESEARCH ARTICLE

A Method for Predicting Links in Complex Networks by Integrating Enclosure Subgraphs With High-Frequency Graph Information

ZHIWEI ZHANG, GUANGLIANG ZHU, AND WENBO QIN
School of Informatics and Engineering, Suzhou University, Suzhou 234000, China

Corresponding author: Wenbo Qin (zzwloveai@gmail.com)

This work was supported in part by the Natural Science Foundation of Anhui Province under Grant 1908085QF283; in part by the Doctoral Scientific Research Funding under Grant 2019jb08, Grant 2021bsk016, and Grant 2023bsk024; in part by the University Synergy Innovation Program of Anhui Province under Grant GXXT-2022-047; in part by Anhui Provincial Universities' Excellent Young Teachers Training Program under Grant YQYB2023053; in part by the Natural Science Research Projects in Universities under Grant 2023AH040314; and in part by Suzhou University Scientific Research Funding under Grant 2021XJPT50.

ABSTRACT Link prediction in complex networks, crucial for uncovering hidden or upcoming links between nodes and widely applicable in fields such knowledge graphs, faces challenges with current techniques. Predominantly, graph neural networks (GNN) based methods focus on learning node representations and use predictive components to assess the similarity of these representations for achieving link prediction. However, these approaches tend to accumulate errors in the predictive model and complicates the training process. Additionally, existing GNNs often display a low-pass filtering effect during network data processing, prioritizing low-frequency information while overlooking high-frequency details in node representations. These bias make GNNs mainly used for link prediction in strongly assortative networks and limit their performance on highly disassortative networks. Addressing these issues, this article introduces a novel framework that redefines the link prediction problem. By extracting enclosure subgraphs of both 'observed' and 'unobserved' links, we represent these links by corresponding enclosure subgraphs and transform link prediction into a problem of subgraphs classification. We innovate by combining high- and low-frequency graph information from the subgraphs, using an attention mechanism for integration, and constructing a graph neural network tailored to learn these subgraph representations, thus accomplishing the task of link prediction indirectly and enhancing link subgraphs classification accuracy. Our extensive experiments on recognized benchmark datasets, evaluated using the *Hits@n* metric, demonstrate that our method not only shows remarkable performance but also possesses strong generalization capabilities, positioning it as a potent baseline for link prediction tasks.

INDEX TERMS Complex network, link prediction, graph neural network, enclosure subgraph, high-frequency graph information.

I. INTRODUCTION

Many systems in nature and society, from the World Wide Web to social and biological networks, can be aptly described as complex networks or graphs [1], [2], [3]. In these networks, nodes or vertices represent entities, while the interactions among these entities are depicted as edges or links [1]. Complex networks, serving as abstract models for understanding real-world systems, exhibit characteristics such as self-organization, self-similarity, small-world properties, and scale-free nature. Link prediction within these networks is a critical task that involves uncovering hidden or emerging

The associate editor coordinating the review of this manuscript and approving it for publication was Ghufran Ahmed.

links between nodes. This includes not only identifying unknown links that already exist but have not yet been detected in the network but also forecasting future links that, although not currently present, are likely or ought to exist in the foreseeable future [4].

In practical applications, link prediction plays a pivotal role in forecasting potential new connections within a network, thereby supporting and enhancing decision-making processes. For instance, in the field of biomedical research, particularly concerning protein interaction networks and metabolic networks, the determination of whether links (interaction relationships) exist between nodes often relies on extensive experimental inference. Taking protein interaction networks as an example, about 80% of interaction relationships in yeast proteins remain undiscovered, and for humans, only 0.3% of known protein interactions have been identified. Due to the high experimental costs associated with uncovering these hidden links in such networks, the development of effective link prediction algorithms based on existing network structures and characteristics is of paramount importance. Utilizing these predictive outcomes to guide experiments can significantly increase the likelihood of successful experimental results, thereby reducing costs. Furthermore, it can accelerate the pace of revealing the intrinsic mechanisms within these networks, offering substantial academic and research value [5], [6], [7].

In the realms of theoretical research and modeling, link prediction serves not only as a crucial tool for studying network structures and their evolutionary patterns but also as an effective method for simulating complex systems and constructing models. For example, in the field of knowledge representation, knowledge graphs, as a form of complex network, possess remarkable expressive power and modeling flexibility. In these graphs, nodes represent entities or concepts from the real world, and each link (edge) corresponds to a piece of knowledge in reality, thus embodying a wealth of rules and logical meanings. This allows for the deduction of unexpressed knowledge in the knowledge graph based on predefined rules. For instance, knowing that 'Tom is a cat', we can derive numerous new pieces of knowledge using rules such as 'cats have two ears, four legs, and come in various breeds', without the need to explicitly detail each one in the knowledge graph. Therefore, knowledge graphs not only effectively model the real world but are also readily processed by computers, leading to their wide application in fields like question-answering systems and criminal investigations in public security. However, issues like incomplete data and information loss during the construction of knowledge graphs can lead to missing entities, attributes, and relationships, causing inaccuracies in knowledge inference. To address these issues, knowledge graph completion techniques have emerged. These techniques aim to predict unobserved relationships between entities in the knowledge graph and to forecast tail entity attributes based on head entity attributes. Fundamentally, this technique parallels link prediction in complex networks. Thus, link prediction contributes significantly to enhancing the accuracy of knowledge graph completion, enriching knowledge inference, and its applications [8], [9].

With the advancement of deep neural networks, GNNs have been widely applied to link prediction in complex networks. On one hand, GNN-based prediction methods, being node-centric, update and learn node representations through repeated exchanges of neighborhood information, and then utilize prediction components to learn the similarity of these representations for link prediction. This approach, however, does not fully leverage the end-to-end learning advantages of GNNs and cumulatively increases the overall model error, making it challenging to dynamically adjust the node representations provided to the prediction components for optimization during training. Moreover, link-centric prediction models lack effective methods in areas such as the extraction and isomorphism testing of local pattern closure subgraphs of links. This project aims to focus on links by concentrating on the extraction and isomorphism testing of link closure subgraphs, transforming link prediction into a classification of these subgraphs, and establishing a single-task optimization model. On the other hand, previous studies have shown that current GNNs exhibit a low-pass filtering effect when processing network data [10], often learning only the low-frequency information representing commonalities of nodes and neglecting the high-frequency information that reflects node differences. This limitation confines the application of existing GNNs primarily to link prediction in strongly assortative networks, while their predictive performance is restricted in highly disassortative networks [8], [9], [10], [11]. However, highly disassortative networks are prevalent in the real world and play a crucial role, such as in biological, technological, and financial networks. Link prediction in these networks not only addresses practical application issues but also allows for exploration of network formation and evolution at a micro-level.

To address the aforementioned challenges, this article adopts a link-centric approach, proposing a method for predicting links in complex networks by **I**ntegrating **E**nclosure **S**ubgraphs with **H**igh-Frequency **G**raph **I**nformation (IESHGI). This approach indirectly facilitates link prediction through the classification of link enclosure subgraphs. The main contributions of this work are summarized as follows:

- We extract the closure subgraph, consisting of two nodes associated with a link and their neighboring nodes that do not exceed k hops, for each link, representing the link by its closure subgraph, and transform the link prediction problem into a classification task of these subgraphs. This leads to the establishment of a single-task optimization model for link prediction.
- We construct a GNN to learn the representations of link closure subgraphs, which extracts not only the low-frequency information from node representations but also the high-frequency information that reflects node differences. Furthermore, by applying an attention mechanism, we integrate high and low-frequency graph

information to realize a universal graph filter. This allows the model to effectively and adaptively aggregate the features of neighboring nodes.
- Extensive experiments have been conducted on widely recognized benchmark datasets to validate the feasibility and effectiveness of the proposed link prediction method.

The remainder of this article is organized as follows. Section I introduces the research background, main challenges faced, and the primary work of this article related to link prediction. Section II presents an overview of the literature involved to the topic under consideration. Section III outlines the link prediction framework that incorporate enclosure subgraphs, high- and low-frequency graph information. Section IV describes the experimental settings, datasets, along with the presentation of experimental consequences and their analysis. Finally, we concludes the key findings and drawbacks of the proposed method, and shed light on the future research directions in the final section V.

II. RELATED WORKS

As an emerging technique in complex network analysis, the concept of GNN was first introduced by Gori et al. [12] and further elucidated by Scarselli et al. [7]. However, in a comprehensive synthesis of existing research on link prediction, Philip S. Yu et al. pointed out that while methods based on similarity for link prediction have been extensively studied, the application of GNNs in link prediction has received comparatively less attention [5]. This paper will analyze and summarize the current state of research in representation learning within networks and the construction of GNNs for link prediction.

A. NETWORK REPRESENTATION LEARNING

''Network representation learning is focused on embedding network nodes into a low-dimensional space, transforming high-dimensional sparse feature vectors into low-dimensional dense embedding vectors. Methods based on random walks in network representation learning generate contextual information for network nodes through these walks. Node sequences are then interpreted as sentences, and natural language processing techniques are applied for node embedding. Consequently, the more frequently two network nodes appear together in the same random walk, the more similar their embeddings become.

One of the most representative random walk-based network representation learning algorithms is DeepWalk, introduced by Perozzi et al. [13]. Its fundamental concept involves mapping the relationships and structural properties of nodes within a graph to a new vector space. In this space, nodes that are closer within the network are also closer in the vector space, thus transforming network data into vector space data through this optimization goal. Following this, feature vectors representing node structural information are concatenated with those representing node attribute information, and then used for downstream network data mining tasks, including link prediction [14]. However, LINE proposed by Tang et al. [15], seemingly does not utilize a random walk strategy. Nevertheless, literature [16] categorizes it under random walk approaches, primarily because LINE, similar to DeepWalk, employs a probabilistic loss function. This involves minimizing the empirical probability of nodes being connected and the distance in vectorized node similarity, considering both first and second-order similarities. This approach is inherently akin to the motivations of random walk strategies.

Furthermore, for strongly assortative network data, Xu et al. proposed the Graph Isomorphism Network (GIN) [17], which characterizes the discriminative ability of classical GNNs and their variants on assortative network data. GIN is proven not only to possess isomorphism testing capabilities as powerful as the Weisfeiler-Lehman test but also to perform exceptionally well on multiple network classification benchmark datasets. In terms of simultaneously learning network structure and embeddings, Chen et al. introduced an end-to-end learning framework, namely Deep Iterative and Adaptive Learning for Graph Neural Networks (DIAL-GNN) [18]. This framework converts the network structure learning problem into a similarity metric learning task, using an optimized regularization strategy to control the smoothness, connectivity, and sparsity of the generated network. Building on this foundation, they further proposed a new iterative method to search for hidden network structures, aiming to enhance the original network. This approach offers valuable guidance for constructing the link prediction Graph Neural Networks in this research topic.

In summary, network representation learning emphasizes the representation of network nodes and the preservation of network topology information in the embedding space, providing support for downstream link prediction tasks. However, current link prediction methods based on network representation learning rarely adopt a link-centric approach. They mainly focus on learning node representations in a node-centric manner and then utilize prediction components to calculate the similarity of these representations for link prediction. This approach not only accumulates the overall error of the model but also complicates the training process.

B. CONSTRUCTION OF GNNS FOR LINK PREDICTION

As a framework for deep learning on graphs, GNNs have been recognized for their potential in complex network link prediction. Yet, as noted by Philip Qiu et al., there is a relative scarcity of research in this area [11]. Baldassarre and Azizpour provided a general definition of Graph Networks (GNs) and focused their explanation on two main approaches: gradient-based and decomposition-based [20]. Their work, which concentrated on the interpretability of GNs, particularly for graph-based predictive tasks, offers valuable insights and inspiration for adapting link prediction in both assortative and disassortative complex networks, a focus of this study. To comprehensively learn node features in hierarchical

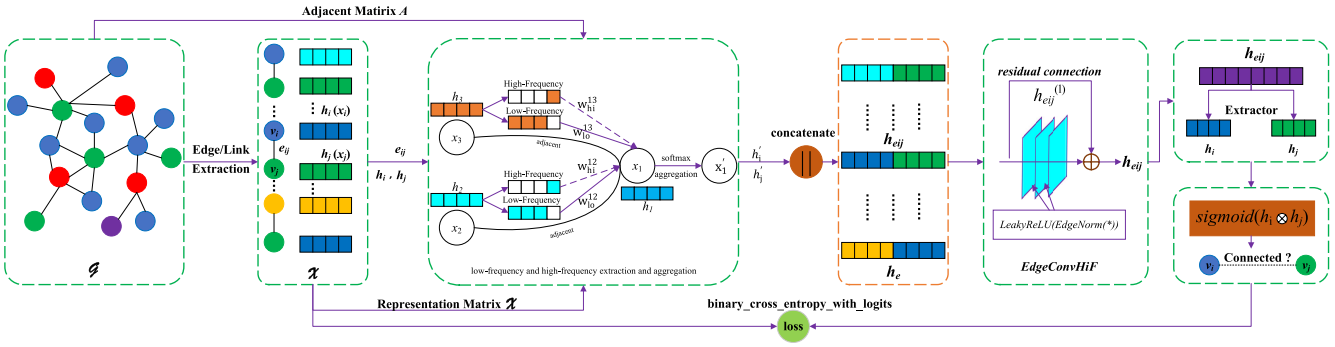

FIGURE 1. Link prediction graph neural network based on edge convolution [19].

graph-structured neural network models, information can be gleaned at various levels of the graph [21], [22]. The Capsule Graph Neural Network (CapsGNN) developed by Xinyi and Chen [22] stands out as one of the most representative models, employing the concept of capsules to address the limitations of existing graph embedding algorithms. By extracting node features in capsule form and utilizing a routing mechanism to gather vital information at different graph levels, CapsGNN generates multiple embeddings for each graph, capturing the macroscopic attributes of the entire graph in a data-driven manner from various perspectives. Our earlier work proposed a link-centric approach to link representation learning and GNN model training [19]. This model represents and learns link representations by 'merging' the representations of the two nodes associated with a link, defines an edge convolution layer, and constructs a link prediction GNN as shown in Figure 1 by stacking these layers. However, this model requires extracting node representations from the learned link representations and then computes link prediction based on the similarity of these node representations. Therefore, it accumulates errors in GNN training and link prediction, necessitating further refinement. However, combining the findings of Bo et al. published at AAAI 2021 [10], it is evident that current GNNs exhibit a low-pass filtering effect when processing network data, learning only low-frequency information and neglecting high-frequency information. This tendency limits GNNs to link prediction in strongly assortative networks and hinders their performance in highly disassortative networks. The smoothness of low-frequency information leads to GNN training retaining low-frequency features that reflect node commonalities, while high-frequency features that demonstrate node differences are overlooked. This paper will integrate the methods proposed by Bo et al. [10] and others to incorporate both high- and low-frequency graph information into the construction of graph link prediction neural networks.

III. LINK PREDICTION FRAMEWORK INTEGRATING ENCLOSURE SUBGRAPH AND HIGH-FREQUENCY GRAPH INFORMATION

This section will first introduce the symbols and task definitions related to link prediction. Subsequently, it will detail the scheme for extracting link enclosure subgraphs, the methods for high- and low-frequency graph information extraction, and the construction of Graph Neural Networks based on these elements. Finally, building on the aforementioned research, this section will outline a comprehensive framework for link prediction.

TABLE 1. Symbols employed in this article.

Symbol	Description and Explanation
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	$\mathcal{G}$ represents a network or graph, $\mathcal{V} = (v_1, v_2, \ldots, v_n)$ and $\mathcal{E} = (e_1, e_2, \ldots, e_n)$ correspondingly denote the set of nodes and the set of edges within $\mathcal{G}$, respectively.
$\mathcal{G}_i$	$\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_n)$, and $\mathcal{G}_i$ indicates the i-th enclosure subgraph of $\mathcal{G}$.
$\lvert\mathcal{V}\rvert$, $\lvert\mathcal{E}\rvert$	$\lvert\mathcal{V}\rvert$ and $\lvert\mathcal{E}\rvert$ respectively denote the number of nodes and the number of edges within $\mathcal{G}$.
$\mathcal{A}$	$\mathcal{A}$ represents the adjacent matrix of $\mathcal{G}$, and if there is a link between node v_i and v_j, $a_{ij} = 1$, otherwise $a_{ij} = 0$.
$\mathcal{N}(i)$	The neighbors of node v_i.
$\mathcal{X}^{(n\times d)}$	The representation of a link enclosure subgraph $\mathcal{G}_i$, where n indicates the nodes amount of $\mathcal{G}_i$, and d denotes the representation dimension of $\mathcal{G}_i$.
$\mathcal{H}^{(l)}$	The representation of a enclosure subgraph $\mathcal{G}_i$ in the l-th layer of a GNN.
h_i	The representation of node v_i.
h_i^{hi}, h_i^{lo}	h_i^{hi} and h_i^{lo} indicate the high- and low-frequency representation information of node v_i.
w_{hi}^{ij}, w_{lo}^{ij}	w_{hi}^{ij} and w_{lo}^{ij} respectively represent the weights corresponding to the high- and low-frequency information between nodes v_i and v_j.

A. PRELIMINARIES

1) SYMBOLS

In our endeavor to enhance the clarity and depth of descriptions and explanations pertaining to the domain of link prediction, we have diligently compiled an extensive and detailed list of symbols, as shown in Table 1, which provide a clearer understanding of the complex concepts and methodologies employed in our research. Throughout this article, these symbols are used consistently, serving as a fundamental cornerstone for elucidating our theoretical and experimental approaches.

2) LINK PREDICTION

Link prediction fundamentally aims to determine the existence or potential formation of links between two nodes.

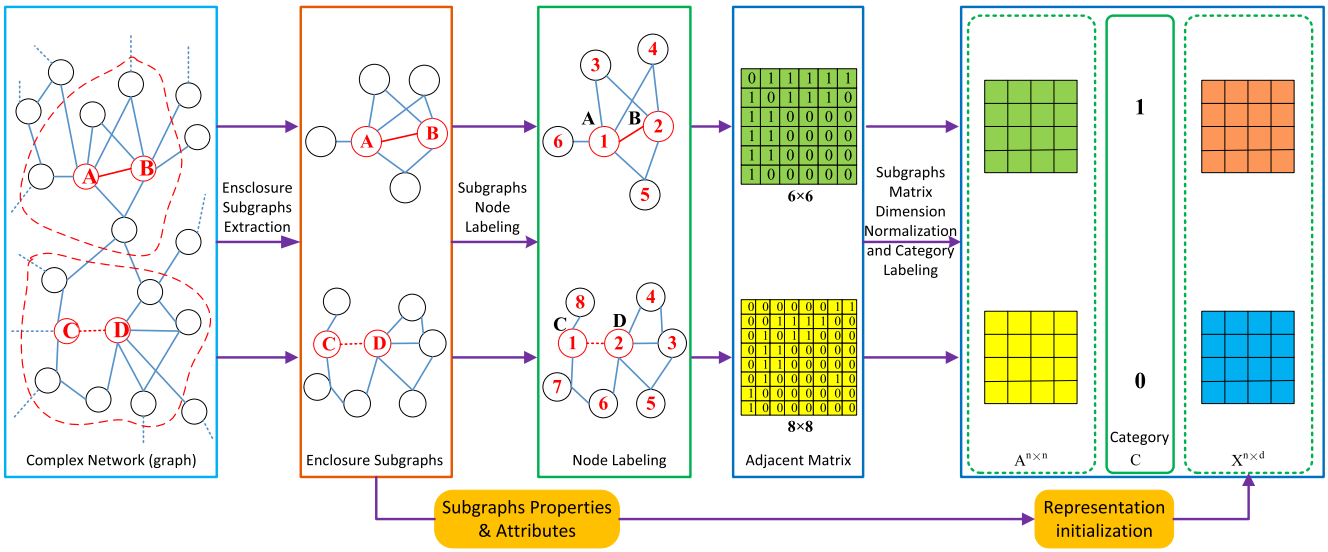

FIGURE 2. The process of extracting link closure subgraphs, labeling subgraph nodes, and generating initial subgraph representations constitutes a critical component of our methodology, where the edge 'A—B' is the 'observed' link and the 'C· · · D' is the 'unobserved' link, and their corresponding link enclosure subgraphs correspond to categories of '1' and '0', respectively.

Within the framework of a specified graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ the set of links, and considering the universal link set $\mathcal{U} = \mathcal{V} \times \mathcal{V}$, the goal of link prediction is to forecast links between nodes v_i and v_j ($v_i, v_j \in \mathcal{V}$). This forecast is based on the known topological features and properties of $\mathcal{G}$. To delineate, the link prediction process utilizing a GNN involves several key steps. Initially, $\mathcal{E}$ is partitioned into two subsets: $\mathcal{E}^T$ (training set) and $\mathcal{E}^P$ (validation set), while the complementary set $\mathcal{U} - \mathcal{E}$ is earmarked as the test set. Notably, $\mathcal{E}^T$ combined with $\mathcal{E}^P$ encompasses the entire set $\mathcal{E}$, with no overlap between them. Following this, a model based on graph attention neural network concepts is employed to derive node representations from $\mathcal{E}^T$. This model is then validated against $\mathcal{E}^P$ to refine its predictive accuracy. In the final phase, the model executes Hadamard product operations on the node representations for v_i and v_j, thereby effectively predicting the likelihood of a link between these nodes.

B. LINK ENCLOSURE SUBGRAPHS EXTRACTION

Guided by the SEAL proposed by Zhang and Chen [23], this article focuses on links within complex networks and extracts local pattern enclosure subgraphs of these links as the fundamental units for learning in a Graph Convolutional Neural Network (GCN) aimed at link prediction. We extract enclosure subgraphs not only for 'observed' links but also for 'unobserved' links, subsequently labeling the respective enclosure subgraphs' categories as '1' and '0'. This approach effectively translates the link prediction problem into a enclosure subgraph classification problem. To accommodate network types and node attributes, network embedding algorithms are utilized to generate initial representations of these link enclosure subgraphs. As illustrated in Figure 2, this process encompasses several key steps: the extraction of link enclosure subgraphs, the labeling of subgraph nodes, the dimension normalization of the subgraph adjacency matrix, and the generation of initial subgraph representations.

1) ENCLOSURE SUBGRAPHS EXTRACTION SCHEME

Motivated by the SEAL approach [23], this paper adopts a strategy of randomly sampling K-hop neighbors of link nodes to extract enclosure subgraphs. The process unfolds as follows:

Firstly, the method involves extracting enclosure subgraphs by randomly sampling K-hop (k=2 for simplicity in this article) neighbors of link-adjacent nodes. This process includes not only the extraction of closure subgraphs for 'observed' links (such as 'A—B' in Figure 2) but also for 'unobserved' links (like 'C· · · D' in Figure 2). This dual approach allows the model to learn patterns that both 'facilitate' and 'inhibit' the formation of links between nodes.

Secondly, the nodes within the extracted enclosure subgraphs are labeled to achieve 'sequential numbering' of nodes. In networks, there often exist links with similar or identical roles, leading to isomorphic properties in their corresponding enclosure subgraphs. Building upon the Weisfeiler-Lehman algorithm [24], [25] and incorporating the randomness of enclosure subgraph extraction, this paper develops a node labeling algorithm with equivalent node labeling and isomorphism testing capabilities. The steps of this algorithm are illustrated in Figure 3. This approach ensures that enclosure subgraphs with isomorphic properties have adjacency matrices with similar node indices.

Finally, the dimensions of the enclosure subgraph adjacency matrices are 'normalized'. Due to the sparsity, scale-free nature, and community structure characteristics of networks, there can be inconsistencies in the number of nodes in enclosure subgraphs formed by K-hop neighbors

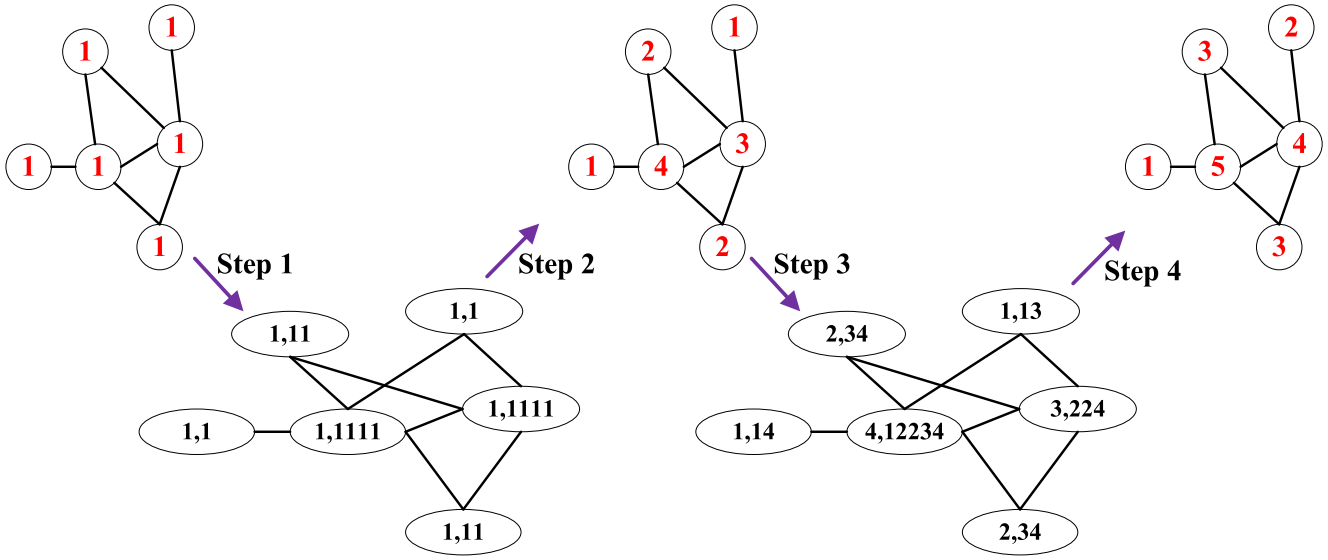

FIGURE 3. Node labeling process in enclosure subgraph of links, following the steps outlined in literatures [24] and [25]. Initially, all nodes are labeled with the same number, such as 1. Then, for each node, create a list of its neighboring nodes and represent it in the form of (node label, list of neighboring nodes label), such as (1, 11). Finally, based on the number of neighbors for each node, update its label to be the count of its neighbors; for example, the label of node (1, 11) is updated to 2. Repeat above process until each node within the enclosure subgraph has a unique label.

of link-adjacent nodes. This results in varying dimensions of the corresponding adjacency matrices for the enclosure subgraphs. In this paper, 'n' represents the final dimension of the subgraph adjacency matrix $\mathcal{A}^{(n\times n)}$. For matrices larger than n, a 'trimming' process is applied, while for those smaller, a 'padding' method is used. This approach ensures the consistency of node order in isomorphic subgraphs.

2) INITIAL REPRESENTATION GENERATION OF LINK ENCLOSURE SUBGRAPHS

To simultaneously capture the topological structure and attributes of enclosure subgraphs, and to learn and obtain the initial representation $\mathcal{X}^{(n\times d)}$ of the link enclosure subgraphs, where n represents the number of nodes in the subgraph, and d represents the dimensionality of the subgraph node representation. This paper employs the current mature network embedding algorithm, node2vec [26], to generate the initial representations $\mathcal{X}^{(n\times d)}$ for each link enclosure subgraph. This approach provides the necessary data support for link prediction through enclosure subgraph classification and for training the graph neural network constructed in this study.

C. CONSTRUCTION OF GNN INTEGRATING HIGH- AND LOW-FREQUENCY GRAPH INFORMATION

The adjacency matrix $\mathcal{A}^{(n\times n)}$ of the aforementioned link enclosure subgraphs reflects the structural information between nodes, whereas the information contained within the subgraph representation $\mathcal{X}^{(n\times d)}$ represents node features, with low-frequency information embodying common characteristics of nodes, and high-frequency information indicating node differences. To fully learn the inherent representations within link enclosure subgraphs for indirect link prediction through classification of these subgraphs, and to maintain robust performance in assortative and disassortative complex networks, it is first necessary to design corresponding high-pass and low-pass filters to extract high- and low-frequency graph information from $\mathcal{X}^{(n\times d)}$. Subsequently, by employing an attention mechanism, the proportion of high- and low-frequency graph information during the GNN information aggregation process is controlled. This achieves a universal graph filter capable of learning both high- and low-frequency graph information. Consequently, a Graph Convolutional Network (GCN) layer that integrates high- and low frequency graph information is constructed to learn and update the network node representations. The above process is shown in Figure 4.

1) EXTRACTION OF HIGH- AND LOW-FREQUENCY GRAPH INFORMATION

For each link enclosure subgraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and its adjacency matrix $\mathcal{A}(n \times n)$, the corresponding standard Laplacian matrix $\mathcal{L} = \mathcal{I}_n - \mathcal{D}^{-\frac{1}{2}}\mathcal{A}\mathcal{D}^{-\frac{1}{2}}$ can be derived. Here, $\mathcal{V}$ and $\mathcal{E}$ represent the node and link sets of $\mathcal{G}$ respectively, with $n = |\mathcal{V}|$, where $\mathcal{I}_n$ is the identity matrix and $\mathcal{D}_{ii} = \sum_j \mathcal{A}_{ij}$ is the diagonal degree matrix. Coupled with $\mathcal{G}$'s initial representation $\mathcal{X}^{n\times d} = [x_1, x_2, \cdots, x_n]$, where x_i denotes the signal of node v_i, that is, the node feature, and d represents the dimensionality of node features. This study delineates the two-step process for extracting high- and low-frequency graph information from the enclosure subgraph node features as follows, motivated by the idea proposed by Bo et al. [10].

Step One: Given that $\mathcal{G}$ is an undirected graph, $\mathcal{L}$ is a real symmetric matrix and can thus be decomposed into $\mathcal{L} = \mathcal{U}\Lambda\mathcal{U}^\top$, where $\mathcal{U} = [u_1, u_2, \cdots, u_n]$ comprises a complete set of orthonormal eigenvectors, and

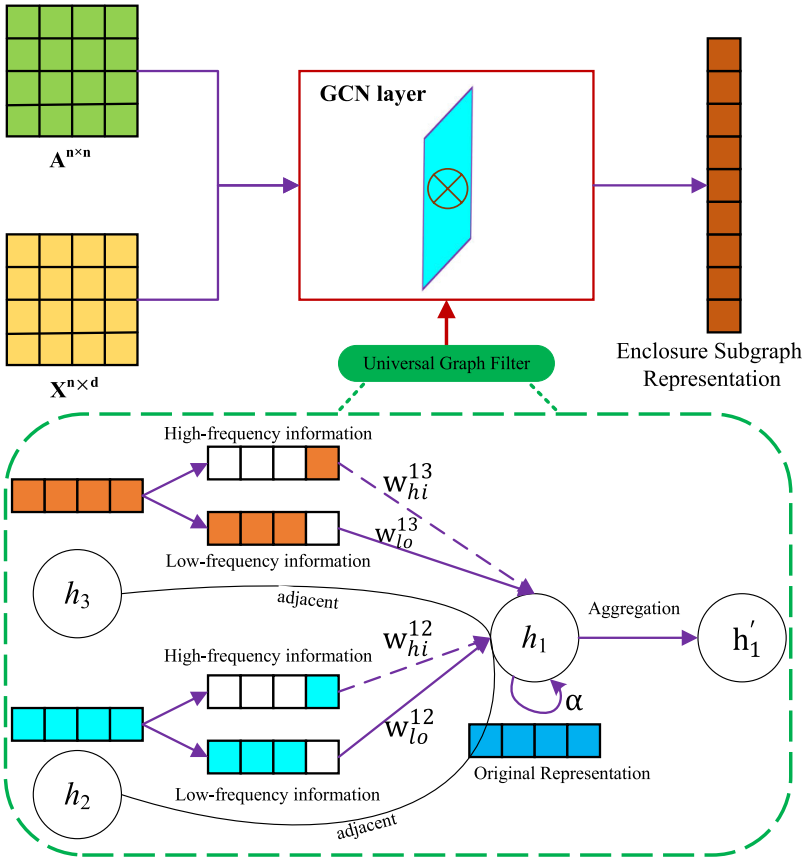

FIGURE 4. Construction process of GCN layer for learning representations of link enclosure subgraphs, where h_i denotes the representation of node v_i, w_{hi}^{ij} and w_{lo}^{ij} respectively represent the weights corresponding to the high- and low-frequency information between nodes v_i and v_j.

$\Lambda = diag([\lambda_1, \lambda_2, \cdots, \lambda_n])$, with λ_i being the eigenvalue corresponding to the eigenvector u_i ($i \in \mathbb{N}$, $1 \leq i \leq n$). In accordance with graph signal theory, the eigenvectors of $\mathcal{L}$ and $\mathcal{U}$ can serve as the basis for the graph Fourier transform, implying that for a graph signal $\mathcal{X}$, its graph Fourier transform is $\tilde{\mathcal{X}} = \mathcal{U}^\top \mathcal{X}$, and the inverse graph Fourier transform is $\mathcal{X} = \mathcal{U}\tilde{\mathcal{X}}$. Consequently, this study aims to define the convolution operation $\star$ between $\mathcal{G}$'s graph signal $\mathcal{X}$ and a convolution kernel $\mathcal{F}$ as shown in Equation (1).

$$(\mathcal{F} \star \mathcal{X})_{\mathcal{G}} = \mathcal{U}((\mathcal{U}^\top \mathcal{F}) \odot (\mathcal{U}^\top \mathcal{X})) = \mathcal{U} g_\theta \mathcal{U}^\top \mathcal{X} \quad (1)$$

Herein, $\odot$ signifies the Hadamard product, while $g_\theta = diag([\theta_1, \theta_2, \cdots, \theta_n])$ serves as the convolution kernel in the spectral domain. In this paper, g_θ is defined as $g_\theta = \sum_{k=0}^{k-1} \beta_k \Lambda^k$.

Step Two: Inspired by Bo et al. [10], this paper employs high-pass and low-pass filters to extract high and low-frequency graph information from the features of nodes in enclosure subgraphs. The high-pass filter $\mathcal{F}_{hi}$ and the low-pass filter $\mathcal{F}_{lo}$ are respectively illustrated in Equations (2) and (3).

$$\mathcal{F}_{hi} = \alpha \mathcal{I}_n - \mathcal{D}^{-\frac{1}{2}} \mathcal{A} \mathcal{D}^{-\frac{1}{2}} \quad (2)$$

$$\mathcal{F}_{lo} = \alpha \mathcal{I}_n + \mathcal{D}^{-\frac{1}{2}} \mathcal{A} \mathcal{D}^{-\frac{1}{2}} \quad (3)$$

where α represents a hyperparameter that the model needs to learn. By substituting the high-pass filter $\mathcal{F}_{hi}$ from Equation (2) and the low-pass filter $\mathcal{F}_{lo}$ from Equation (3) for the convolution kernel function $\mathcal{F}$ in Equation (1), the high- and low-frequency graph information in $\mathcal{G}$ can then be obtained respectively through Equations (4) and (5), which lays the foundation for a universal graph filter capable of learning both high- and low-frequency graph information.

$$\mathcal{X}_{hi} = (\mathcal{F}_{hi} \star \mathcal{X})_{\mathcal{G}} = \mathcal{U}[(\alpha - 1)\mathcal{I}_n + \Lambda]\mathcal{U}^\top \mathcal{X} \quad (4)$$

$$\mathcal{X}_{lo} = (\mathcal{F}_{lo} \star \mathcal{X})_{\mathcal{G}} = \mathcal{U}[(\alpha + 1)\mathcal{I}_n - \Lambda]\mathcal{U}^\top \mathcal{X} \quad (5)$$

2) UNIVERSAL GRAPH FILTER INTEGRATING HIGH- AND LOW-FREQUENCY GRAPH INFORMATION

Following the extraction of high- and low-frequency graph information from the link enclosure subgraph representation $\mathcal{X}$, investigating a universal filter that can integrate both types of information and is adaptable to both highly disassortative and strongly assortative network link enclosure subgraph representation learning emerges as one of the key challenges to be addressed in this paper. To this end, an attention mechanism is employed to calculate the weights

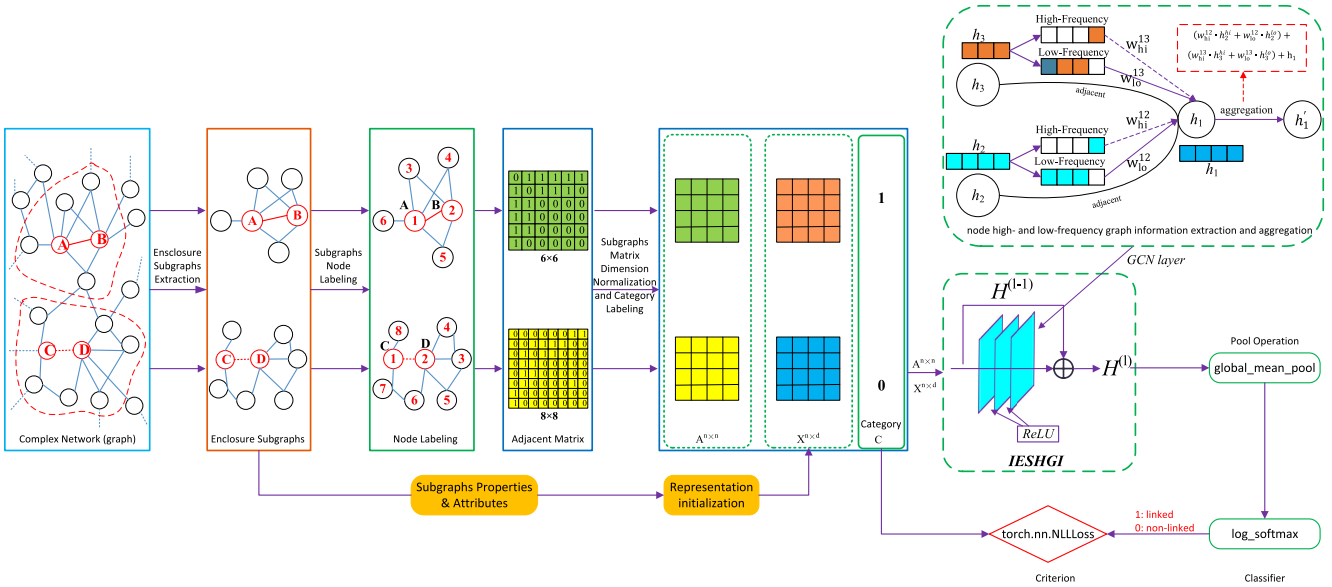

FIGURE 5. The whole GCN designed for link prediction in complex networks and named IESHGI,combines both high- and low-frequency graph information, where h_i is the representation of node v_i, h_i^{hi} and h_i^{lo} respectively denote the high- and low-frequency graph information for v_i, while w_{hi}^{ij} and w_{lo}^{ij} respectively represent the weights corresponding to the high- and low frequency information between nodes v_i and v_j.

of the extracted high and low-frequency graph information, as shown in Equations (6) and (7), thereby facilitating the provision of "fusion" coefficients for high- and low-frequency graph information during the representation learning of enclosure subgraph node v_i.

$$w_{hi}^{i} = softmax_i(\mathcal{X}_{hi}) = \frac{exp(\mathcal{X}_{hi}^{i})}{\sum_{k\in\mathcal{N}(i)} exp(\mathcal{X}_{hi}^{k})} \tag{6}$$

$$w_{lo}^{i} = softmax_i(\mathcal{X}_{lo}) = \frac{exp(\mathcal{X}_{lo}^{i})}{\sum_{k\in\mathcal{N}(i)} exp(\mathcal{X}_{lo}^{k})} \tag{7}$$

It should be noted that w_{hi}^{i} and w_{lo}^{i} represent the attention coefficients corresponding to the high and low-frequency graph information for node v_i ($v_i \in V$ in $\mathcal{G}$) during its representation learning. Incorporating the attention coefficients for high- and low-frequency graph information as specified in Equations (6) and (7), this paper designs a universal graph filter as shown in Equation 8 to accomplish the learning and updating of enclosure subgraph representation $\mathcal{H}^{(l)}$.

$$\mathcal{H}^{(l)} = w_{hi}((\mathcal{F}_{hi} \star \mathcal{X})_{\mathcal{G}}) + w_{lo}((\mathcal{F}_{lo} \star \mathcal{X})_{\mathcal{G}}) + \mathcal{H}^{(l-1)} \tag{8}$$

herein, $\mathcal{H}^{(l)}$ represents the new representation learned through the universal graph filter. With this, the paper completes the construction of the Graph Convolutional Network (GCN) layer for link enclosure subgraphs classification. Subsequent steps involve stacking GCN layers to build a comprehensive deep GCN for link prediction.

D. LINK PREDICTION FRAMEWORK IMPLEMENTED THROUGH LINK ENCLOSURE SUBGRAPH CLASSIFICATION

In this section, we delineate the construction of a comprehensive framework *IESHGI* for link prediction within complex networks. By stacking the aforementioned Graph Convolutional Network (GCN) layers and incorporating activation functions, pooling operations, and a classification function, we construct a Graph Convolutional Neural Network for the classification of link enclosure subgraphs, as depicted in Figure 5. This architecture enables the convolutional processing of the adjacency matrix and initial representations of link enclosure subgraphs, facilitating the acquisition of final subgraph representations. Consequently, this framework allows for the effective classification of link enclosure subgraphs, thereby enhancing our ability to predict links within complex networks.

This framework not only leverages the structural and feature information inherent in the subgraphs but also optimizes the information flow through the network by applying non-linear transformations and reducing dimensionality where necessary. The use of pooling operations, in particular, aids in abstracting higher-level features from the convolutionally processed subgraphs, while the classification function translates these features into probabilistic predictions for subgraph categories. This methodology underscores the potential of deep learning techniques in unraveling the intricate patterns of connectivity that characterize complex networks, offering a robust tool for the prediction of new or missing links based on observable network dynamics.

IV. EXPERIMENTS AND DISCUSSION

To rigorously evaluate the IESHGI model and conduct a comparison with established baseline methods, we follow the experimental framework from our prior work [4]. Our method

TABLE 2. The statistical information on the datasets utilized in this article.

Dataset	#Nodes	#Edges	Split Type
ogbl-ppa	576,289	30,326,273	Throughput
ogbl-collab	235,868	1,285,465	Time
ogbl-ddi	4,267	1,334,889	Protein target

encompasses an in-depth analysis across various benchmark datasets to fully gauge the model's efficacy. This section starts by detailing the benchmark datasets used, describes the baseline methods for comparison, and specifies the metrics for evaluating performance. Subsequently, we report our experimental findings and engage in a comparative discussion. Our aim is to showcase the *IESHGI* model's robustness and dependability through this comprehensive examination.

A. EXPERIMENTAL SETTINGS

The experimental framework for this study was meticulously crafted to guarantee the precision and reliability of our research work. We conducted our experiments on a Dell T640 workstation, a high-performance deep learning platform running the CentOS-7 operating system. This setup provides a stable and efficient platform for computational tasks. The workstation was equipped with a Tesla V100s GPU. For this research, CUDA 10.2 was utilized to leverage optimized support for graph deep learning frameworks and enhance GPU acceleration. The programming environment was unified under Python 3.7 to ensure seamless compatibility and facilitate the development of *IESHGI*. PyTorch version 1.11, known for its dynamic computational graph capability and comprehensive library support, was selected for *IESHGI* implementation. Additionally, we incorporated torch_geometric 2.1, a PyTorch-based library dedicated to graph deep learning. This library offers vital tools for the implementation and evaluation of the *IESHGI* model, bolstering our experimental setup with the necessary resources for cutting-edge research.

B. DATASET

Without loss of generality, while maintaining continuity with our previous research work [4], the dataset employed in this article remains consistent with the dataset utlized in our previous work [4], all sourced from the Open Graph Benchmark (OGB),[1] as shown in Table 2, including ogbl-ppa [27], ogbl-collab [28] and ogbl-ddi [29]. We strictly followed the default partition settings provided by OGB for these datasets. This approach not only preserves the distinct size and characteristics of each dataset but also ensures consistency with the experimental methodologies outlined in [4], thereby maintaining the integrity and continuity of our research.

- The **ogbl-ppa** is a type of undirected and unweighted graph. In this graph, nodes correspond to proteins originating from 58 distinct species. The edges within this graph signify biologically significant relationships between proteins, which can include physical interactions, co-expression patterns, homology, or genomic proximity [27].
- The **ogbl-collab** is an undirected graph that captures a portion of the collaboration network among authors indexed by Microsoft Academic Graph. Each node in the graph represents an author, and the edges signify collaborations between these authors. All nodes in this dataset are associated with 128-dimensional features, which are derived by averaging the word embeddings of papers authored by these individuals. Additionally, each edge in the graph is accompanied by two pieces of meta-information: the year of collaboration and the edge weight, which reflects the number of co-authored papers published in that particular year. This graph can be understood as a dynamic multi-graph, allowing for the existence of multiple edges between two nodes if authors collaborate across multiple years [28].
- The **ogbl-ddi** is a homogeneous, unweighted, undirected graph that depicts the drug-drug interaction network. In this graph, each node represents either an FDA-approved drug or an experimental drug. The edges in the graph signify interactions between these drugs, indicating scenarios where the combined effect of taking two drugs together significantly deviates from the expected effect of each drug acting independently. This network captures and visualizes the complex relationships and interactions between different drugs based on their observed joint effects [29].

C. BASELINE METHODS

In this study, we extend our previous research by further validating the effectiveness of our enhancements and comparing them with established baseline methods. To ensure a robust comparison, we employed the same baseline techniques as mentioned in literature [4], which include the classic Graph Convolutional Network(GCN) [30], Graph Attention Network(GAT) [31], GraphSAGE [32], EdgeConvNorm [19], and our own previously developed LVGANN [4]. It's important to emphasize that our focus in using these GNNs is primarily on enclosure subgraph representation learning. For the link prediction task, which is indirectly approached through link enclosure subgraphs classification, we apply the *log_softmax* classifier to the pooled representations of these subgraphs, ensuring a consistent methodology for evaluating performance. The concise description of the baseline methods employed in this article are as follows.

- **GCN** advances the application of convolutional operations beyond their traditional realm of regular grids, as seen in image processing, to accommodate the

[1] https://ogb.stanford.edu/docs/linkprop/

complexities of irregular graphs. Within this framework, graphs are processed by applying a feature transformation to each node, which incorporates the attributes of its neighbors. This pivotal process preserves the network's local structure by aggregating neighboring features to forge a new representation for each node.

- **GAT** introduces attention mechanisms to GNNs, revolutionizing how significance is allocated among nodes. By incorporating an attention mechanism, GAT independently assesses the importance of each neighboring node, acknowledging that different neighbors exert varying levels of influence. This innovation enables nodes to focus more on significant neighbors and less on those with minimal impact. As a result, GAT improves the model's capacity to identify and learn from the most pertinent connections within the graph, facilitating more refined and effective node representations. This advancement underscores GAT's role in enhancing the nuanced understanding and processing of graph data.
- **GraphSAGE** an innovative adaptation of the GCN, addresses the critical issue of scalability. It creates node representations by selectively sampling and aggregating features, a departure from the GCN approach that requires processing all graph nodes in each forward pass. GraphSAGE stands out for its ability to efficiently train on large-scale graphs and produce embeddings for nodes not seen during training. This method greatly improves the model's flexibility and scalability, establishing GraphSAGE as a formidable solution for navigating and analyzing vast and intricate graph networks.
- **EdgeConvNorm** revolutionizes link representation learning by incorporating a specialized edge convolution operation, uniquely suited for distilling the essence of connections within a graph. This model further enhances the quality of link representations through a strategic normalization process, effectively addressing the prevalent challenge of over-smoothing often encountered in edge convolution-based link prediction models. A key feature of EdgeConvNorm is its deployment within a link prediction framework, employing multiple stacked edge convolutional layers. This structured approach allows the model to adeptly unravel and analyze intricate link characteristics, significantly boosting the precision and resilience of link prediction outcomes.
- **LVGANN** introduces a nuanced analysis of link value grounded in network structure and presents a novel methodology for its estimation. This approach integrates link value into the design and training phases of a link prediction graph attention network, enhancing the precision of link predictions. Moreover, this integration offers a theoretical framework for interpreting the prediction outcomes, thereby enriching our understanding of link dynamics within networks. This advancement not only elevates the accuracy of link predictions but also contributes significantly to the theoretical underpinnings of network analysis.

D. EVALUATION METRIC

Evaluating the efficacy of link prediction models, *Hits@n* emerges as a pivotal metric, quantifying prediction accuracy by counting how many correct links are identified among the top n predictions. This method involves ranking all predicted link enclosure subgraphs by their likelihood, in descending order, for each test scenario. A thorough investigation assesses whether the correctly predicted links fall within the top n positions. A *hit* is recorded when a correct link prediction (i.e. link enclosure subgraph classification) is found within this specified range. The collective *hits* across all scenarios are then compiled and normalized by the total number of accurately predicted subgraphs. In mathematical terms, for k correct link predictions, the link enclosure subgraph corresponding to each link $e_i \in \mathcal{E}$ is assigned a label of 1 if it ranks among the top n predictions, and 0 otherwise. The computation of *Hits@n*, as detailed in Equation (9), offers a precise metric for assessing the model's predictive precision.

$$Hits@n = \frac{1}{k} \sum p(e_i) \tag{9}$$

The *Hits@n* metric is pivotal for assessing the precision of link prediction models in identifying the top n links, with a higher *Hits@n* indicating superior model performance. It should be noted that in the performance evaluation of the model *IESHGI*, $p(e_i)$ represents the classification result of the enclosure subgraph corresponding to the link e_i. This research evaluates model efficacy using selected n values—10, 50, and 100—as key parameters. Notably, each model undergoes 10 iterations to produce a range of outcomes. The ultimate results are derived by averaging the *Hits@n* values across these iterations and calculating the corresponding standard error, providing a comprehensive measure of model accuracy and consistency.

E. RESULTS ANALYSIS AND DISCUSSION

For this study, the experimental settings of the baseline methods were carefully calibrated using the published source code from OGB. The model's parameters were not further optimized, prioritizing the comparison of relative performances across different models. However, it is crucial to emphasize that this paper serves as an enhancement and refinement of our previous research [4]. To maintain continuity, comparability, and validate the effectiveness of the methodology introduced here, all baseline methods experimental consequences, apart from the *IESHGI* model's experimental data, are sourced from our earlier studies [4]. This approach ensures a coherent and comparative framework for assessing the advancements presented in this work.

TABLE 3. The experimental results from various baseline methods and the model *IESHGI*, as applied to the ogbl-collab dataset, are documented.

Metric / Models	*Hits*@10		*Hits*@50		*Hits*@100	
	Val.	Test	Val.	Test	Val.	Test
GCN	0.3273±0.0138	0.2657±0.0128	0.5036±0.0130	0.4264±0.0118	0.5830±0.0054	0.5024±0.0056
GraphSAGE	0.3225±0.0111	0.2498±0.0115	0.5193±0.0065	0.4371±0.0061	0.5998±0.0037	0.5200±0.0035
GAT	0.3729±0.0123	0.3054±0.0112	0.5671±0.0114	0.4735±0.0183	0.6372±0.0026	0.5510±0.0141
EdgeConvNorm	0.3361±0.1307	0.3125±0.8014	0.5536±0.3012	0.4921±0.0513	0.5217±0.4033	0.4895±0.6104
LVGANN	0.3813±0.0046	0.3341±0.0019	0.5841±0.0027	0.5106±0.0134	0.6402±0.0130	0.5723±0.0048
IESHGI	**0.4107±0.0102**	**0.4001±0.0162**	**0.5985±0.0037**	**0.5481±0.0074**	**0.6637±0.0129**	**0.5927±0.0053**

TABLE 4. The experimental outcomes from a range of baseline methods and the model *IESHGI*, as implemented on the ogbl-ddi dataset, are presented.

Metric / Models	*Hits*@10		*Hits*@50		*Hits*@100	
	Val.	Test	Val.	Test	Val.	Test
GCN	0.4483±0.3710	0.2386±0.1065	0.6890±0.1990	0.6800±0.3710	0.7847±0.1530	0.7929±0.2640
GAT	0.4716±0.0417	0.0994±0.0407	0.6573±0.0071	0.5407±0.0913	0.7110±0.0231	0.7642±0.0471
GraphSAGE	**0.6124±0.1421**	0.1069±0.1362	**0.7034±0.3510**	**0.8572±0.0199**	0.7201±0.0021	**0.9157±0.0069**
EdgeConvNorm	0.2307±0.4101	**0.3013±0.0027**	0.41573±0.6037	0.5283±0.4051	0.4947±0.6129	0.5247±0.2104
LVGANN	0.5347±0.0184	0.1004±0.0601	0.6917±0.1094	0.6017±0.0134	0.7238±0.1063	0.8035±0.1705
IESHGI	**0.6021±0.0371**	**0.2485±0.1036**	**0.7185±0.1062**	**0.8371±0.1058**	**0.7429±0.3107**	**0.9186±0.0127**

Besides, the final evaluation of performance is conducted by calculating the mean and standard deviation of the peak results from 10 iterations. Detailed findings, showcasing the comparative superiority of all baseline methods across diverse datasets, are systematically presented in Tables 3 to 5. This methodical approach underscores the robustness and consistency of our evaluation process, providing a clear demonstration of model efficacy.

In our analysis of the ogbl-collab dataset, we have adhered to the experimental protocol as described by OGB and broadened our methodology to include temporal aspects, utilizing the data splitting technique outlined in OGB. Our main objective is to forecast future author collaborations using historical data, placing a particular focus on prioritizing true collaborations.

Table 3 clearly demonstrates that the *IESHGI* method introduced in this paper outperforms other baseline methods, including our previously proposed LVGANN approach based on link value assessment. It achieves an average accuracy of 0.5356 and shows an average performance improvement of 6.3% compared to the LVGANN method. Given the dynamic and complex nature of the multi-graph dataset ogbl-collab, link enclosure subgraphs are more adept at capturing the local characteristics of links. They also delve into the high- and low-frequency information within the link enclosure subgraphs. This not only captures the similarity in subgraph node representations but also highlights the differences, effectively reflecting the intrinsic mechanisms through which nodes form links. These features enable *IESHGI* to more accurately predict the existence of a link between two nodes.

For the ogbl-ddi dataset, we employed a protein-target splitting method. This strategy aims to forecast drug-drug interactions based on data from established interactions previously. The experimental data in Table 4 clearly indicate that the *IESHGI* method introduced in this paper has achieved an average performance improvement of 17.7% over our previously proposed LVGANN method, attaining an average performance of 0.6780. Fortunately, compared to GraphSAGE, which also derives network representations by learning from the network's local structure, our method has shown a notable performance enhancement. The ogbl-ddi dataset is notable for its dense network structure, featuring 4,267 nodes and an impressive 1,334,889 edges. However, GraphSAGE's innovative node sampling strategy demonstrates significant effectiveness in addressing these challenges. By judiciously choosing nodes for the training process, GraphSAGE effectively navigates the complexities of the dataset's large-scale graph. It addresses the computational intensity of updating gradients over the entire graph and enhances training efficiency, illustrating a strategic approach to handling high-density networks. Fortunately, the *IESHGI* method proposed in this paper also predicts links by indirectly learning from the local structural enclosure subgraphs of a network. Overall, for dense network, local structures are more prevalent and better reflect the network's characteristics, making *IESHGI* particularly suited for these environments.

For the ogbl-ppa dataset, we adhere to the partitioning strategy outlined in the established OGB framework. The primary goal of this article is primarily aimed at predicting specific types of protein relationships, emphasizing the prediction of physical protein-protein interactions. These auxiliary connections have demonstrated a significant correlation with the targeted interactions, thereby boosting the reliability and accuracy of predictions within the realm of protein relationships. Experimental data from Table 5 reveal that, in comparison to the ogbl-collab and ogbl-ddi datasets, all methods, including IESHGI, did not achieve optimal results on ogbl-ppa, despite *IESHGI* showing a slight improvement over our previous method LVGANN. The

TABLE 5. The experimental consequences from various baseline methods alongside the model*IESHGI*, applied to the ogbl-ppa dataset, are detailed.

Metric / Models	*Hits*@10		*Hits*@50		*Hits*@100	
	Val.	Test	Val.	Test	Val.	Test
GCN	0.1307±0.0081	0.1293±0.1034	0.1637±0.0171	0.1501±0.0308	0.1736±0.0170	0.1637±0.0043
GAT	**0.1675±0.0602**	0.1435±0.0032	0.1796±0.0062	0.1464±0.0107	0.1582±0.0143	0.1601±0.0712
GraphSAGE	0.1401±0.0318	0.1386±0.0318	0.1709±0.0532	**0.1682±0.2041**	0.1803±0.0014	**0.1780±0.0510**
EdgeConvNorm	0.1183±0.0019	0.1037±0.0417	0.1206±0.3058	0.1282±0.0304	0.1402±0.0131	0.1513±0.5108
LVGANN	0.1594±0.0140	**0.1639±0.0216**	0.1801±0.1131	0.1407±0.0025	0.1812±0.0163	0.1697±0.0109
IESHGI	**0.1642±0.1065**	**0.1663±0.1174**	**0.1925±0.1073**	**0.1693±0.1805**	**0.1904±0.2031**	**0.1796±0.4133**

primary reason lies in the high sparsity structure of ogbl-ppa, which owns 576,289 nodes and 30,326,273 edges, making local structures like enclosure subgraphs less prevalent than in ogbl-collab and ogbl-ddi. This suggests that existing methods may be more suited to dense networks and those with high assortative.

Synthesizing the experimental results and analyses, it is evident that the *IESHGI* method proposed in this paper demonstrates strong overall performance, outperforming our previously introduced LVGANN method as well as conventional baseline methods, thereby advancing and refining our earlier work [4]. Data from Tables 3 and 4 indicate that both the existing baseline methods and the newly introduced *IESHGI* are particularly effective in dense networks, where local structures like enclosure subgraphs are more prevalent and exhibit high clustering. Additionally, *IESHGI* excels in learning and updating network representations by not only capturing low-frequency information that reflects commonalities among nodes but also high-frequency information that highlights node differences. This comprehensive approach to learning network representations is one of the key reasons *IESHGI* surpasses our previous link prediction methodologies.

Additionally, based on the dataset statistics shown in Table 2, the datasets employed in this study are of moderate size and exhibits high sparsity. Firstly, a sparse network implies fewer connections between nodes, leading to longer and more dispersed paths for information propagation. In such complex networks, conventional GNNs may face challenges in effectively propagating and aggregating information, resulting in information loss or decay. Secondly, the node degree distribution in sparse networks may be more uneven, with a few highly connected central nodes and a majority of nodes with lower degrees, which can impact GNNs' ability to recognize and learn the overall network structure and important nodes. Fortunately, the proposed *IESHGI* model in this article effectively integrates high-frequency and low-frequency information, improving the model's generalization ability and applicability.

V. CONCLUSION

This study introduces an inovative framework for predicting links within complex networks, a vital endeavor for uncovering latent or forthcoming node relationships applicable across diverse fields. Our research proposes a unique strategy that conceptualizes link prediction as a classification task of enclosure subgraphs, encapsulating both observed and unobserved links. This strategy leverages both high- and low-frequency information within these subgraphs, integrated via an attention mechanism, to construct a GNN specifically designed for acquiring intricate subgraph representations. This innovative approach not only sidesteps direct link prediction but also substantially boosts the accuracy of classifying link enclosure subgraphs.

Our method stands out by rectifying the biases present in conventional GNNs, showcasing impressive performance enhancements and robust generalization across well-established benchmark datasets. The strategy of using enclosure subgraphs to represent links simplifies link prediction into a straightforward optimization challenge. It captures a broad range of frequency information, effectively amalgamates features of adjacent nodes with a versatile graph filter, and sets a strong foundation for future link prediction endeavors.

Despite these advances, there remain areas ripe for further exploration and enhancement, particularly concerning the time complexity, attention mechanism refinement, and adaptability to highly sparse complex networks. (1) Time Complexity: A key challenge with sophisticated GNN models, including ours, is their substantial computational demand, which escalates with the network's size and intricacy. The incorporation of both high- and low-frequency data, along with an attention mechanism for detailed subgraph representation learning, significantly contributes to the model's increased time complexity. Future efforts could aim at optimizing these computational aspects, perhaps through refined algorithmic methods or by harnessing advancements in parallel computing. Such improvements would aim to lower the overall time complexity, enhancing the model's suitability for real-time analysis and application to extensive datasets. (2) Refinement of the Attention Mechanism: The utilization of an attention mechanism is central to our model's enhanced performance, enabling more precise integration of features from enclosure subgraphs. However, the potential for optimizing this mechanism further remains vast. Future research might investigate more advanced or dynamic attention frameworks that adjust responsively to the network's specific features or the task at hand. Enhancements in this

area could yield more accurate link predictions by more effectively discerning the network's structural nuances and node interrelations. (3) Adaptability to Sparse Complex Networks: Our model shows exceptional performance in dense networks, where enclosure subgraphs are common. Yet, its efficacy in highly sparse complex networks, which typify many real-world environments, could be improved. Future work should consider devising strategies or adaptations to the model that bolster its performance in sparse contexts. This may involve creative approaches to deduce or reconstruct local structures in such networks or introducing novel mechanisms to capture essential long-range dependencies more effectively.

In summary, this study propels the link prediction field and the use of GNNs for network analysis forward. Yet, the identified avenues for enhancement underscore the roadmap for future investigations. By tackling these identified challenges, forthcoming studies have the opportunity to expand GNNs' utility further, affirming their role as comprehensive tools for navigating and understanding the complexities of vast and diverse networks.

REFERENCES

[1] Z. Zhang and Z. Wang, "Mining overlapping and hierarchical communities in complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 421, pp. 25–33, Mar. 2015.

[2] Y. Cui, X. Wang, and J. Li, "Detecting overlapping communities in networks using the maximal sub-graph and the clustering coefficient," *Phys. A, Stat. Mech. Appl.*, vol. 405, pp. 85–91, Jul. 2014.

[3] C. Mu, Y. Liu, Y. Liu, J. Wu, and L. Jiao, "Two-stage algorithm using influence coefficient for detecting the hierarchical, non-overlapping and overlapping community structure," *Phys. A, Stat. Mech. Appl.*, vol. 408, pp. 47–61, Aug. 2014.

[4] Z. Zhang, X. Wu, G. Zhu, W. Qin, and N. Liang, "A graph attention network-based link prediction method using link value estimation," *IEEE Access*, vol. 12, pp. 34–45, 2024.

[5] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[6] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowl.-Based Syst.*, vol. 151, pp. 78–94, Jul. 2018.

[7] F. Scarselli, M. Gori, A. Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "Computational capabilities of graph neural networks," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 81–102, Jan. 2009.

[8] B. P. Chamberlain, S. Shirobokov, E. Rossi, F. Frasca, T. Markovich, N. Y. Hammerla, M. M. Bronstein, and M. Hansmire, "Graph neural networks for link prediction with subgraph sketching," in *Proc. 11th Int. Conf. Learn. Represent. (ICLR)*, 2023, pp. 1–27. [Online]. Available: https://openreview.net/forum?id=m1oqEOAozQU

[9] L. Cai, J. Li, J. Wang, and S. Ji, "Line graph neural networks for link prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5103–5113, Sep. 2022.

[10] D. Bo, X. Wang, C. Shi, and H. Shen, "Beyond low-frequency information in graph convolutional networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, 2021, pp. 3950–3957.

[11] Z. Qiu, J. Wu, W. Hu, B. Du, G. Yuan, and P. S. Yu, "Temporal link prediction with motifs for social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 3145–3158, Mar. 2023.

[12] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IEEE Int. Joint Conf. Neural Networks(IJCNN)*, vol. 2, 2005, pp. 729–734.

[13] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: Association for Computing Machinery, Aug. 2014, pp. 701–710.

[14] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Network representation learning: A survey," *IEEE Trans. Big Data*, vol. 6, no. 1, pp. 3–28, Mar. 2020.

[15] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 1067–1077.

[16] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Eng. Bull.*, vol. 40, no. 3, pp. 52–74, Sep. 2017. [Online]. Available: http://sites.computer.org/debull/A17sept/p52.pdf

[17] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–17. [Online]. Available: https://openreview.net/forum?id=ryGs6iA5Km

[18] Y. Chen, L. Wu, and M. J. Zaki, "Deep iterative and adaptive learning for graph neural networks," in *Proc. AAAI Workshop Deep Learn. Graphs: Methodologies Appl. (AAAI DLGMA)*.

[19] Z. Zhang, L. Cui, and J. Wu, "Exploring an edge convolution and normalization based approach for link prediction in complex networks," *J. Netw. Comput. Appl.*, vol. 189, Sep. 2021, Art. no. 103113.

[20] F. Baldassarre and H. Azizpour, "Explainability techniques for graph convolutional networks," 2019, *arXiv:1905.13686*.

[21] J. Li, Y. Rong, H. Cheng, H. Meng, W. Huang, and J. Huang, "Semi-supervised graph classification: A hierarchical graph perspective," 2019, *arXiv:1904.05003*.

[22] Z. Xinyi and L. Chen, "Capsule graph neural network," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–16. [Online]. Available: https://openreview.net/forum?id=Byl8BnRcYm

[23] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2018, pp. 5171–5181.

[24] B. Weisfeiler, *On Construction and Identification of Graphs*, vol. 558. Springer, 2006.

[25] M. Zhang and Y. Chen, "Weisfeiler-lehman neural machine for link prediction," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: Association for Computing Machinery, Aug. 2017, pp. 575–583.

[26] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016, pp. 855–864.

[27] D. Szklarczyk, A. L. Gable, D. Lyon, A. Junge, S. Wyder, J. Huerta-Cepas, M. Simonovic, N. T. Doncheva, J. H. Morris, P. Bork, L. J. Jensen, and C. V. Mering, "STRING v11: Protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D607–D613, Jan. 2019, doi: 10.1093/nar/gky1131.

[28] K. Wang, Z. Shen, C. Huang, C.-H. Wu, Y. Dong, and A. Kanakia, "Microsoft academic graph: When experts are not enough," *Quant. Sci. Stud.*, vol. 1, no. 1, pp. 396–413, Feb. 2020, doi: 10.1162/qss_a_00021.

[29] D. S. Wishart et al., "DrugBank 5.0: A major update to the DrugBank database for 2018," *Nucleic Acids Res.*, vol. 46, no. D1, pp. D1074–D1082, Jan. 2018, doi: 10.1093/nar/gkx1037.

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–14. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[31] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[32] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf

ZHIWEI ZHANG was born in Fuyang, Anhui, China, in 1988. He received the Bachelor of Science degree in computer science and technology from Suzhou University, Suzhou, China, in 2010, and the Master of Science degree in computer system architecture from Kunming University of Science and Technology, Kunming, in 2013, and the Ph.D. degree in computer science and technology from the South China University of Technology, Guangzhou, in 2016. Since 2019, he has been an Associate Professor with the Data Science and Big-Data Application Laboratory, School of Informatics and Engineering, Suzhou University. He has authored more than 20 scholarly articles. His research interests include social computing, social network analysis, the dynamics of complex networks, and big-data applications.

GUANGLIANG ZHU was born in Heze, Shandong, China, in 1974. He received the Bachelor of Science degree in electronic instruments and measurement technology from Changchun University of Science and Technology, Jilin, in 1997, the Master of Science degree in military equipment from the PLA University of Science and Technology, Jiangsu, in 2004, and the Ph.D. degree in military operations research from the Second Artillery Engineering University, Shanxi, in 2013. He has been an Associate Professor of computer science and technology, since 2019. He has contributed to more than 20 scholarly articles. His research interests include network security, the integration of artificial intelligence in educational settings, and the innovative applications of big data.

WENBO QIN was born in Xuchang, Henan, China, in 1994. He received the Bachelor of Science degree in measurement and control technology and instrumentation and the Master of Science degree in mechanical engineering from Henan Polytechnic University, in 2015 and 2018, respectively, and the Ph.D. degree in detection technology and automation devices from Northeastern University, in 2022. Since 2022, he has been a part of a Faculty Member with the School of Information Engineering, Suzhou University. From September 2022 to September 2023, he held the position of an Associate Researcher with the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center. An active contributor to the scientific community, he has published more than ten articles in SCI-indexed journals. His research interests include micro-nano detection, intelligent detection systems, and the application of artificial intelligence in these fields.