Link Prediction Using Node Influence Based on Graph Neural Networks

BASWANI, Madhusudhana Rao <http://orcid.org/0009-0009-5999-8438>, YARRA, Khyathisree, BOYAPATI, Prasanthi <http://orcid.org/0000-0002-1051-5695>, RAYACHOTI, Eswaraiah <http://orcid.org/0000-0001-8605-0794> and TANGIRALA, Jaya Lakshmi <http://orcid.org/0000-0003-0183-4093>

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 March 2026, accepted 20 March 2026, date of publication 25 March 2026, date of current version 7 April 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3677526

RESEARCH ARTICLE

Link Prediction Using Node Influence Based on Graph Neural Networks

MADHUSUDHANA RAO BASWANI[1], KHYATHISREE YARRA[1], PRASANTHI BOYAPATI[1], ESWARAIAH RAYACHOTI[2], AND T. JAYA LAKSHMI[3], (Member, IEEE)

[1]Department of Computer Science and Engineering, School of Engineering and Sciences, SRM University-AP, Guntur, Andhra Pradesh 522502, India
[2]School of Computer Science and Engineering, VIT-AP University, Guntur, Andhra Pradesh 522241, India
[3]Department of Computing, Sheffield Hallam University, S1 2NU Sheffield, U.K.

Corresponding author: Prasanthi Boyapati (prasanthi.b@srmap.edu.in)

This work was supported by SRM University-AP toward the article processing charges (APC).

ABSTRACT Link prediction in complex networks aims to infer missing or future connections between nodes, a task crucial for understanding network evolution in domains such as social systems, biology, and recommender platforms. Traditional similarity-based approaches, which rely primarily on local neighborhood overlap, often fail to capture global structural properties and node importance, resulting in limited predictive accuracy in sparse or large-scale networks. To address these limitations, we propose a Link Prediciton based Centrality using Graph Neural Network (LP-GNN) framework that integrates centrality measures with similarity indices in a unified, parameterized learning paradigm. The proposed LP-GNN operates in three sequential stages. 1) Feature Construction: Edge features are enriched by a weighted fusion of structural similarity and node centrality measures, providing a comprehensive representation of both local and global graph information. 2) Centrality-Aware Encoder: A modified message-passing mechanism is employed to learn node embeddings, where the influence of structurally important nodes is amplified, ensuring that critical topological information is effectively captured. 3) Multi-Branch Decoder: Multiple complementary scoring mechanisms are integrated including dot-product similarity, a multilayer perceptron (MLP) to model non-linear interactions, and distance-based measures to generate robust estimates of link probabilities between node pairs. Our approach was evaluated across multiple real-world benchmark datasets, demonstrating its superiority over both traditional similarity-based methods and recent GNN-based models. Notably, it achieves improvements of up to 49% in Area Under the Receiver Operating Characteristic Curve (AUROC) and Area Under the Precision-Recall Curve (AUPR) over state-of-the-art baselines. The highlight the effectiveness of combining structural interpretability from centrality with the representational power of GNNs, offering a robust and scalable solution to the link prediction problem in complex networks.

INDEX TERMS Link prediction (LP), centrality measures (CM), graph neural networks (GNNs), graph auto-encoder (GAE).

I. INTRODUCTION

Complex networks are powerful mathematical representations of real-world systems in which entities are represented as nodes and their interactions as edges [1]. These networks capture intricate relationships and structural patterns that emerge in various domains, including social, biological, technological, information systems and scientific collaborations [2]. Fig. 1 illustrates a typical social network, such as Facebook, where nodes represent individual users and edges denote interactions between them, such as friendships, comments, or shared content. Complex networks have found applications in a wide range of fields. In social network analysis, they enable understanding of user behavior, influence propagation, and community formation. In biological systems, complex networks model protein-protein interactions, aiding in the understanding of cellular processes. Additionally, technological networks, such as the Internet or power grids, can be analyzed using complex network theory to enhance reliability and

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

efficiency. Despite their wide applicability, several fundamental problems persist in the study of complex networks. These include Link Prediction (LP) [3], Centrality Measures [4], Community Detection [5], and others. Among these, the task of predicting missing or future links between nodes has attracted increasing attention due to its growing importance in many real-world applications. Link prediction refers to the process of inferring potential or missing connections between nodes in a network, based on observed structural properties or node attributes. Applications of link prediction are widespread, including friend recommendation in social networks, predicting interactions in biological networks, and detecting potential collaborations in academic networks. Another fundamental aspect of complex networks is the study of centrality measures, which quantify the importance or influence of individual nodes within the network. Common centrality measures include Degree Centrality, Betweenness Centrality, Closeness Centrality, and Eigenvector Centrality. These measures are widely applied in identifying key users in social networks, influential proteins in biological networks, and critical routers in communication networks. A key motivation for integrating centrality measures into the link prediction task lies in the observation that not all nodes contribute equally to the network structure. Traditional link prediction methods, which rely solely on similarity measures [6], often overlook the inherent importance of individual nodes. Centrality measures capture global and local node influence, providing additional information that can enhance the accuracy and robustness of link prediction models. However, directly applying similarity-based link prediction approaches has notable disadvantages, such as limited generalization to unseen data, poor performance in highly sparse or large-scale networks, and an inability to effectively capture non-linear and high-order structural patterns. To address these challenges, recent advances in deep learning have introduced Graph Neural Networks (GNNs) [7] as a powerful framework for learning on graph-structured data. A Graph Neural Network is a specialized type of neural network designed to directly operate on graphs by leveraging both node features and the graph topology. The core idea of GNNs is message passing or neighborhood aggregation, where each node iteratively updates its own representation by aggregating information from its neighbors. This process enables the network to learn rich node and graph-level embeddings that capture not only local connectivity patterns but also global structural dependencies across the graph.

Mathematically, in each layer of a GNN, the representation of a node is updated by combining its own feature vector with aggregated information from its neighbors, typically through functions such as sum, mean, or more complex learned transformations. This iterative process enables the model to capture high-order relationships, where information propagates across multiple hops in the network, allowing the learned embeddings to reflect the broader network structure rather than just immediate connections.

The advantages of using GNNs in the context of link prediction are multifold:

- **End-to-end Learning:** Unlike traditional heuristic-based link prediction methods that rely on predefined similarity scores, GNNs can learn task-specific embeddings directly from the data, optimizing predictive performance in an end-to-end manner.
- **Handling Sparsity:** GNNs are particularly effective in sparse networks where limited structural information makes conventional methods unreliable. Through message passing, GNNs can effectively propagate structural and feature information even when direct connections are sparse.
- **Incorporating Node Attributes:** Real-world networks often contain rich node attributes (e.g., user profiles, item properties, etc.). GNNs seamlessly integrate these features with graph topology, resulting in more informative embeddings for prediction tasks.
- **Scalability and Generalization:** With advances in architecture design and optimization techniques, GNNs scale to large graphs and generalize well to unseen data, making them highly suitable for dynamic or evolving networks.

In recent years, various GNN architectures have been proposed, including Graph Convolutional Networks (GCNs) [8], Graph Attention Networks (GATs) [9], and GraphSAGE [10]. These models differ in how they aggregate neighbor information and how they handle graph sparsity, feature heterogeneity, and computational efficiency. Given these strengths, GNN-based link prediction models have demonstrated superior performance across numerous application domains, ranging from social network analysis and recommender systems to biological interaction prediction and infrastructure monitoring. This motivates the integration of centrality measures within the GNN framework, allowing the model to leverage both structural importance (captured by centrality) and learned topological embeddings, ultimately improving prediction accuracy and robustness.

A. PROBLEM SIGNIFICANCE

Accurate link prediction underpins applications such as recommendation, knowledge-base completion, and biological interaction discovery; errors propagate to downstream decisions. Many real-world networks are sparse, exhibit short average path lengths, and contain ego-centric regions where local structure provides limited signal. Classical similarity indices (CN, JC, RA) remain restricted to local topology and overlook system-level influence, whereas common deep baselines (e.g., GAE, GraphSAGE) primarily optimize node embeddings, provide limited explicit modeling of edge salience, and often underperform when structural variation is minimal. Moreover, existing GNN variants rarely combine global centrality information with local similarity in a parameterized, edge-aware manner.

Motivation and approach. We hypothesize that prospective links are governed jointly by (i) the global influence of their endpoints (centrality) and (ii) their relational proximity (similarity). To operationalize this, we construct a tunable, principled combination of node-centrality and similarity indices as an edge feature and inject it into a GAE-based architecture, harmonizing global and local structure, elevating edge importance to a first-class signal, and remaining lightweight for sparse graphs.

We propose *Link Prediction via Centrality in Graph Neural Networks (LP-GNN)*, which explicitly integrates local structural similarity and global node influence within a unified learning paradigm. The method (i) enriches edge attributes by combining classical similarity measures with centrality scores; (ii) employs a centrality-modulated encoder to emphasize influential nodes during message passing; and (iii) uses a multi-branch decoder (dot-product, MLP-based, and distance-based scoring) for robust link estimation.

Contributions. Specifically, this work advances the field in three key ways:

- It provides a parameterized fusion of centrality and similarity that captures both local and global network structures.
- It introduces a centrality-aware encoder that amplifies the influence of structurally critical nodes during message passing.
- It delivers a multi-branch decoding strategy that unifies dot-product, MLP-based, and distance-based link estimators for enhanced predictive robustness.

These contributions collectively address a fundamental shortcoming in current literature—the lack of edge-aware, centrality-driven models for link prediction—and thus represent a meaningful step forward in understanding and modeling network evolution.

In this work, We propose a novel approach of **Link Prediciton based Centrality using Graph Neural Network (LP-GNN)** framework for link prediction that explicitly integrates both local structural similarity and global node influence into a unified learning paradigm. The approach enriches edge features by fusing classical similarity indices with centrality measures, propagates them through a centrality-modulated encoder to highlight influential nodes, and employs a multibranch decoder combining dot-product, MLP, and distance-based scoring for robust link estimation. Extensive experiments on real-world networks demonstrate that LP-GNN consistently outperforms classical heuristics and recent GNN-based baselines in AUROC and AUPR, particularly in sparse, imbalanced, and long-range dependency settings, while maintaining interpretability by exposing how central nodes shape predicted connections.

The organization of this paper is as follows. Section II presents the fundamental concepts relevant to our study, emphasizing the node centrality metrics and similarity measures employed, along with a brief overview of Graph Neural Networks (GNNs). Section III reviews existing approaches to link prediction and centrality measures in both graphs and hyper-networks. In Section IV, we describe our parameterized framework for link prediction, which combines average node centrality and structural similarity measures within graph embeddings. Section V details the experimental setup, including the datasets, baselines, and evaluation metrics. Section VI provides an in-depth analysis of the outcomes, including a brief discussion on the robustness of LP-GNN under graph noise and structural perturbations. Finally, Section VII summarizes the findings and suggests directions for future research.

Table. 1 summarizes the key notations used throughout this work.

TABLE 1. Notations used in this research.

Notation	Description
G	Network (also called a graph)
V	Set of nodes
E	Set of edges
u, v	Nodes within the network
d_u, d_v, d_z	Degree of a node u, v, and z
$d(u, v)$	Length of the shortest path between u and v
$C_D(u)$	Degree centrality of a node u
$C_C(u)$	Closeness centrality of a node u
$C_B(u)$	Betweenness centrality of a node u
$\sigma_{s,t}$	Number of shortest paths between s and t
$\sigma_{s,t}(u)$	Number of shortest paths between s and t passing through node u
$\Gamma(u), \Gamma(v), \Gamma(w)$	Immediate neighbor sets of u, v, and w
$C(u, v)$	Pairwise average centrality
$C_D(u, v)$	Pairwise average degree centrality
$C_C(u, v)$	Pairwise average closeness centrality
$C_B(u, v)$	Pairwise average betweenness centrality
$S_C^{CN}(v, u)$	Link prediction based on centrality using Common Neighbor
$S_C^{JC}(v, u)$	Link prediction based on centrality using Jaccard Coefficient
$S_C^{AA}(v, u)$	Link prediction based on centrality using Adamic–Adar
$S_C^{RA}(v, u)$	Link prediction based on centrality using Resource Allocation
$S_C^{PA}(v, u)$	Link prediction based on centrality using Preferential Attachment
$ADC(G)$	Average degree centrality of the graph
$ACC(G)$	Average closeness centrality of the graph
$ABC(G)$	Average betweenness centrality of the graph
AUROC	Area Under the Receiver Operating Characteristic curve
AUPR	Area Under the Precision-Recall Curve

II. PRELIMINARIES

This section reviews the core concepts for our study, highlighting the node centrality metrics and similarity measures we employ and providing a brief overview of Graph Neural Networks (GNNs).

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected, unweighted graph. The node set is $\mathcal{V} = \{v_1, \ldots, v_n\}$ and the edge set satisfies $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. For a vertex $v \in \mathcal{V}$ we write $\mathcal{N}(v) = \{u \mid (u, v) \in \mathcal{E}\}$ for its (one-hop) neighbourhood. Node features are stored in a matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ whose i-th row $\mathbf{x}_i \in \mathbb{R}^d$ is the d-dimensional feature vector of vertex v_i. Connectivity

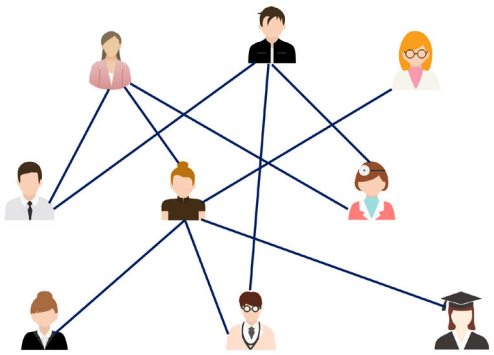

FIGURE 1. An example of a social network.

is encoded by the adjacency matrix $\mathbf{A} \in \{0,1\}^{n\times n}$, where $n = |\mathcal{V}|$ and the entry at position (u, v) is denoted A_{uv}. The (combinatorial) graph Laplacian is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, with $\mathbf{D} \in \mathbb{R}^{n\times n}$ the diagonal degree matrix.

Fig. 1 illustrates an example of the representation of a social network as a graph.

A. CENTRALITY MEASURES

Centrality scores indicate the significance of a node, and there are many variants, such as degree, semi-local, closeness, betweenness, eigenvector, Katz and PageRank, to address different analytical goals. Broadly, these measures divide into two groups. Local centralities (like degree or semi-local) depend only on a node's immediate neighbours and are computationally lightweight. Global centralities (such as closeness, betweenness, eigenvector, coreness, and PageRank) draw on the structure of the whole graph, making them more expensive to calculate. Selecting an appropriate centrality therefore hinges on both the specific definition of influence required and the computational resources available. These normalized centralities are combined into a pairwise term and used within the LP-GNN decoder.

1) DEGREE CENTRALITY

The degree centrality (DC) was first suggested by Shaw [11], later formalized by Nieminen [12], and brought to prominence by Freeman [4], they capture how ''visible'' a vertex is within its immediate neighbourhood.

The normalized degree centrality of a node u can be represented by Eq. 1 [13].

$$C_D(u) = \frac{d_u}{n-1} \tag{1}$$

where $C_D(u)$ denotes its degree centrality of a node u, d_u is the node's degree and $n-1$ is the maximum possible degree in a network of n nodes.

2) CLOSENESS CENTRALITY

Closeness centrality (CC) is grounded in the idea that a node positioned only a few hops away from the rest of the network can spread information most effectively [14]. To obtain $C_C(u)$, we first add up the shortest-path lengths from node u to every other node [15]. We then take the reciprocal of this total, so the CC value grows as those distances shrink, signaling tighter integration into the network. Freeman [4] formalized this measure as Eq. 2 [13]:

$$C_C(u) = \frac{n-1}{\sum_{u\neq v} d(u,v)} \tag{2}$$

where $C_C(u)$ denotes its closeness centrality of a node u, $d(u, v)$ is the length of shortest path between u to v.

3) BETWEENNESS CENTRALITY

In 1973, Granovetter showed that not all edges are equal. He defined strong ties as frequent contacts and weak ties as infrequent ones [16]. Weak ties often serve as bridges for information flow. The same idea extends to nodes through betweenness centrality. This metric, also called flow centrality, assumes information travels along shortest paths. It counts how many of those paths run through a node. More paths mean a higher betweenness score, denoted $C_B(u)$ as Eq. 3 [13].

$$C_B(u) = \frac{\sum_{s\neq u\neq t} \frac{\partial_{st}(u)}{\partial_{st}}}{(n-1)(n-2)/2} \tag{3}$$

where, $C_B(u)$ denotes its betweenness centrality of a node u, $\sigma_{s,t}$ is the number of shortest paths between s and t, $\sigma_{s,t}(u)$ is the number of shortest paths between s and t passing through node u.

B. LOCAL SIMILARITY MEASURES

Each unordered node pair (u, v) is given a similarity score $S(u, v)$. This value reflects how alike the two nodes are in structure or attributes. Unlinked pairs are scored in the same way. The highest scores signal the most probable future edges. Such metrics fall into several groups: local versus global, node-focused versus path-focused, and parameter-free versus parameter-driven [6]. [17] defined link prediction as estimating future links in complex networks using neighborhood-based similarity measures with hop distance of three for candidate node pairs yields optimal performance.

Local measures consider only each node's immediate neighbourhood. They blend counts of shared neighbours with degree information. Classic examples include Common Neighbour [18], Jaccard Coefficient [19], Preferential Attachment [20], Resource Allocation [21] and Adamic–Adar Index [22].

1) COMMON NEIGHBOR

Given two nodes u and v, the CN score is the number of vertices they both connect to [18]:

$$S_{\mathrm{CN}}(u, v) = |\Gamma(u) \cap \Gamma(v)| \tag{4}$$

where $\Gamma(u)$ and $\Gamma(v)$ denote the immediate neighbour sets of u and v, respectively. A higher count of shared neighbours

implies a greater likelihood that a link will appear between the pair.

2) JACCARD COEFFICIENT

This index extends Common Neighbours by dividing the number of shared neighbours by the size of the union of their neighbour sets [19]:

$$S_{\mathrm{JC}}(u, v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|} \tag{5}$$

where $\Gamma(u)$ and $\Gamma(v)$ are, respectively, the neighbors of the nodes u and v.

3) PREFERENTIAL ATTACHMENT

This index assumes that high-degree nodes are more likely to form new links. For a pair of vertices u and v, the score is simply the product of their degrees [20]:

$$S_{\mathrm{PA}}(u, v) = d_u \cdot d_v \tag{6}$$

where d_u and d_v denote the degrees of nodes u and v, respectively. A larger product indicates a higher likelihood that an edge will appear between the two nodes.

4) RESOURCE ALLOCATION INDEX

Derived from the resource-distribution model introduced by Ou et al. [23] in "Physical Review E", this measure imagines node u sending resources to node v through their shared neighbours. The amount of resource transmitted via each common neighbour z is inversely proportional to z's degree. The overall similarity between u and v is therefore Eq. 7

$$S_{\mathrm{RA}}(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{d_z}, \tag{7}$$

where d_z is the degree of neighbour z. Nodes linked by low-degree intermediaries thus receive higher RA scores.

5) ADAMIC–ADAR INDEX

Originally proposed by Adamic and Adar to measure the similarity of web pages, this score was later adapted for link prediction by Liben-Nowell et al. It weights each shared neighbour inversely by the logarithm of its degree [22]:

$$S_{\mathrm{AA}}(u, v) = \sum_{w \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log |\Gamma(w)|} \tag{8}$$

where $\Gamma(u)$ and $\Gamma(v)$ are, respectively, the neighbors of nodes u and v and $\Gamma(w)$ represents the set of neighbors of node w.

C. GRAPH NEURAL NETWORKS

Most Graph neural networks (GNNs) operate via message passing: each node refines its embedding by first collecting information from the embeddings of its immediate neighbours and then merging this aggregated message with its own prior state [24].

1) ENCODER (MESSAGE PASSING)

At layer l each node u updates its embedding by aggregating messages from its neighbours $\mathcal{N}(u)$ [25]:

$$m_u^{(l)} = \bigoplus_{v \in \mathcal{N}(u)} \mathrm{MSG}^{(l)}\big(h_u^{(l-1)}, h_v^{(l-1)}\big) \tag{9}$$

$$h_u^{(l)} = \mathrm{UPDATE}^{(l)}\big(h_u^{(l-1)}, m_u^{(l)}\big) \tag{10}$$

where $\bigoplus$ is a permutation invariant aggregator (e.g., *sum*, *mean*, *max* or attention), $\mathrm{MSG}^{(l)}$ and $\mathrm{UPDATE}^{(l)}$ are learnable functions [9], [10]).

After L layers we obtain final node embeddings $\{h_u^{(L)} \mid u \in \mathcal{V}\}$.

2) DECODER (EDGE SCORING)

For a candidate pair (u, v) we compute a link score via

$$\mathrm{similarity\_score}(u, v) = \sigma\big(h_u^{(L)} \cdot h_v^{(L)}\big), \tag{11}$$

where "$\cdot$" is the dot product and $\sigma(\cdot)$ is the sigmoid that converts the score to a probability. Alternative decoders concatenation + MLP, absolute difference, etc. can be plugged in without changing the encoder.

III. RELATED WORK

Link prediction has been extensively explored, with methods ranging from classical heuristics to advanced Graph Neural Networks (GNNs). Liben-Nowell and Kleinberg [3] formalized the problem and showed that proximity-based indices such as Common Neighbors and Jaccard Coefficient can effectively predict missing links. While these measures are simple and interpretable, they fail to capture higher-order dependencies. To overcome this, centrality-based similarity metrics have been proposed, where node influence is integrated into prediction. Reference [26] propose novel link prediction similarity scores based on average centrality measures (degree, betweenness, closeness, clustering coefficient), demonstrating significant improvements over existing and recent methods across multiple real-world datasets. and [27] developed the NCSM model, which embeds similarity and centrality into a GNN, outperforming traditional baselines.

Parallel to similarity-driven approaches, GNNs have become powerful alternatives by learning embeddings from graph topology. Position-aware GNNs (P-GNNs) [28] improved prediction through anchor-based aggregation, and DiffPool [29] introduced hierarchical pooling. However, flat architectures often struggle with long-range dependencies, motivating hierarchy-aware models such as HAGNN [30] and HAKE [31], which enhance performance in structured and semantic-rich networks. Recent advances also emphasize efficiency: ELPH and BUDDY [32] provide expressive yet scalable link predictors, while Wang et al. [33] balance node- and edge-wise architectures. Other innovations include capsule-enhanced GNNs (GCCL) [34], dual-channel GNNs [35], and generative models such as GraphLP [36].

Reference [37] extends link prediction measures to heterogeneous and temporal networks, logically categorizes them. Reference [38] proposes a centrality-based link prediction measure (CLP) that integrates node importance with neighborhood similarity.

Hybrid approaches highlight the value of combining heuristics with representation learning. NLG-GNN [39] integrates Node2Vec and subgraph transformations into GCNs, while comparative studies [40] show limitations of similarity and GNN methods on homogeneous graphs. Overall, although GNNs have advanced the field, integrating local similarity with global influence remains underexplored. This motivates our work, which unifies centrality and similarity within a GNN framework to achieve robust and interpretable link prediction. Similarly in their study Le et al., survey demonstrates that infrastructure systems are increasingly modeled as interacting networks with cascading failures [41], which justifies exploring hyper-network methods for link prediction in such multi-dimensional settings. Similarly in their study He et al., employs high-way links derived from higher-order neighbors and adaptive feature perturbation to address missing information and enhance robustness in both node and graph classification tasks [42]. Recent studies have shown that graph neural networks are vulnerable to adversarial perturbations, particularly node injection attacks that exploit message passing and label propagation mechanisms. For example, [43] demonstrated that injecting malicious nodes can significantly distort neighborhood aggregation and degrade GNN performance, highlighting critical robustness concerns in graph-based learning models. Recent studies have emphasized the importance of robustness when graph information is incomplete or partially missing. For instance, [44] proposed GIN-SD, a GNN-based framework that leverages positional encodings and attentive fusion to distinguish incomplete nodes and focus on structurally influential users, demonstrating strong resilience under missing-node scenarios.

IV. PROPOSED METHODOLOGY

In this section, we introduce our parameterized framework for link prediction, which combines average node centrality and structural similarity measures within the context of graph embeddings.

A. MOTIVATION

Conventional similarity-driven link prediction methods (e.g., Common Neighbors, Jaccard, Resource Allocation) and deep learning approaches such as Graph Auto-Encoders (GAE) and GraphSAGE serve as strong benchmarks but exhibit notable shortcomings. Heuristic techniques depend solely on local topology, whereas embedding-based methods often struggle in networks with short paths or ego-centric structures where structural variation is minimal. Furthermore, many existing Graph Neural Network (GNN) models emphasize node-level representations while neglecting the structural significance of edges.

In real-world networks, node centrality plays a pivotal role: highly central nodes often act as hubs and are more likely to form new links, whereas nodes with low centrality contribute marginally to connectivity. Meanwhile, structural similarity continues to be an effective indicator of potential links. To overcome these challenges, we introduce a parameterized framework that jointly leverages node centrality and similarity indices, thus harmonizing global structural influence with local relational proximity. The resulting score is incorporated as an edge feature within a GAE-based model to enhance link prediction performance.

B. PARAMETERIZED SIMILARITY BASED ON CENTRALITY AND STRUCTURAL MEASURES

In this section, we propose a parameterized similarity function that integrates node centrality and structural similarity measures into a unified framework for link prediction. Let us consider a node centrality measure C, which can be either a local or a global centrality (e.g., Degree Centrality, Closeness Centrality, or Betweenness Centrality). For a candidate pair of nodes (u, v), we compute the average of their centrality scores as:

$$\overline{C}(u, v) = \frac{C(u) + C(v)}{2}, \tag{12}$$

which captures the structural importance of both nodes in the graph. On the other hand, structural similarity between two nodes, denoted as $Sim(u, v)$, can be estimated using classical similarity indices such as Common Neighbors (CN), Jaccard Coefficient (JC), Adamic–Adar (AA), Resource Allocation (RA), or Preferential Attachment (PA).

To balance these two aspects, we proposed the parameterized similarity score between nodes u and v as:

$$S(u, v) = \alpha \cdot \overline{C}(u, v) + (1 - \alpha) \cdot Sim(u, v), \tag{13}$$

where $\alpha \in [0, 1]$ is a user-defined parameter. The parameter α controls the relative influence of centrality and similarity: larger values of α emphasize the centrality-based structural prominence of nodes, while smaller values emphasize neighborhood-based similarity.

This formulation offers flexibility to adapt to different types of networks. In graphs where hub nodes or high-centrality vertices dominate link formation, larger values of α highlight the role of centrality. In contrast, in networks where connections are mainly driven by shared neighbors or local clustering, smaller values of α make similarity more influential. Thus, the proposed score effectively combines global structural prominence with local relational proximity in a single predictive framework.

We compute node degree (DC), closeness (CC), and betweenness (BC) on the training graph, form a pairwise average centrality for any candidate edge (u, v),

$$C(u, v) = \frac{C(u)+C(v)}{2}, \tag{14}$$

and blend it with a classical similarity score $\mathrm{Sim}(u, v) \in \{\mathrm{CN}, \mathrm{JC}, \mathrm{AA}, \mathrm{RA}, \mathrm{PA}\}$ as a parameterized edge prior:

$$S(u, v) = \alpha\, C(u, v) + (1 - \alpha)\, \mathrm{Sim}(u, v), \quad \alpha \in [0, 1]. \tag{15}$$

This edge prior is injected into the GAE decoder as an additive term:

$$p\big(A_{ij} = 1 \mid Z, S\big) = \sigma\big(z_i^\top z_j + \lambda\, S(u, v)\big), \tag{16}$$

so the decoder uses both the learned embedding similarity and the centrality-aware prior when scoring links.

We also define centrality-aware variants of the classical similarity indices by restricting contributing neighbors to those above the graph's average centrality (for a chosen centrality type). This yields Sim_D, Sim_C, and Sim_B (and their CN/JC/AA/RA/PA counterparts), making the handcrafted part of $S(u, v)$ itself centrality sensitive.

In the proposed LP-GNN framework, centrality is incorporated through both construction of the parameterized edge prior to the refinement of similarity measures. Specifically, we compute three widely used node centralities degree (DC), closeness (CC), and betweenness (BC) on the training graph and derive a pairwise average centrality for each candidate edge. This value is fused with classical similarity indices to form the parameterized score $S(u, v)$, which is injected as an additive prior to the GAE decoder, allowing the model to take advantage of both both learned embedding similarity and the influence-based structural cues. Furthermore, we define centrality-aware variants of CN, JC, AA, RA, and PA by allowing only neighbors with centrality values above the graph-wide mean to contribute, thereby emphasizing structurally significant nodes in the handcrafted similarity component.

C. PAIRWISE AVERAGE DEGREE CENTRALITY

For a candidate node pair (u, v), we compute their pairwise average degree centrality using Eq. 12:

$$\overline{C_D}(u, v) = \frac{C_D(u) + C_D(v)}{2}. \tag{17}$$

where $C_D(u)$ and $C_D(v)$ as shown in Eq.1 define the degree of the nodes u and v. This value represents the structural prominence of the nodes u and v jointly, averaging their degree centralities. Now we define the similarity of two vertices u and v using the average degree centrality of the graph as follows:

$$Sim_D(u, v) = \Big|\{w \mid w \in \Gamma(u) \cap \Gamma(v),\ C_D(w) \geq ADC(G)\}\Big|, \tag{18}$$

where $Sim_D(u, v)$ is the similarity score of the node pair (u, v). It is obtained by first collecting all common neighbors of u and v, and then counting only those neighbors whose degree centrality is greater than or equal to the average degree centrality of the graph. ADC is the average degree centrality of the graph which is defined in Eq.17. In this way, only structurally significant common neighbors contribute to the similarity score.

D. PAIRWISE AVERAGE CLOSENESS CENTRALITY

For a candidate node pair (u, v), we compute their pairwise average closeness centrality using Eq. 12:

$$\overline{C_C}(u, v) = \frac{C_C(u) + C_C(v)}{2}. \tag{19}$$

where $C_C(u)$ and $C_C(v)$ define the closeness centrality values of nodes u and v, respectively. This value represents the joint structural proximity of nodes u and v to all other nodes in the network. Now we define the similarity of two vertices u and v using the average closeness centrality of the graph as follows:

$$Sim_C(u, v) = \Big|\{w \mid w \in \Gamma(u) \cap \Gamma(v),\ C_C(w) \geq ACC(G)\}\Big|, \tag{20}$$

where $Sim_C(u, v)$ is the similarity score of the node pair (u, v). It is obtained by first collecting all common neighbors of u and v, and then counting only those neighbors whose closeness centrality is greater than or equal to the average closeness centrality of the graph. ACC is the average closeness centrality of the graph which is defined in Eq.19. In this way, only neighbors that are closer, on average, to the rest of the graph are considered structurally significant for the similarity measure.

E. PAIRWISE AVERAGE BETWEENNESS CENTRALITY

For a candidate node pair (u, v), we compute their pairwise average betweenness centrality using Eq. 12:

$$\overline{C_B}(u, v) = \frac{C_B(u) + C_B(v)}{2}. \tag{21}$$

where $C_B(u)$ and $C_B(v)$ define the betweenness centrality values of nodes u and v, respectively. This value reflects the joint importance of nodes u and v in controlling information flow through the network. Now we define the similarity of two vertices u and v using the average betweenness centrality of the graph as follows:

$$Sim_B(u, v) = \Big|\{w \mid w \in \Gamma(u) \cap \Gamma(v),\ C_B(w) \geq ABC(G)\}\Big|, \tag{22}$$

where $Sim_B(u, v)$ is the similarity score of the node pair (u, v). It is obtained by first collecting all common neighbors of u and v, and then counting only those neighbors whose betweenness centrality is greater than or equal to the average betweenness centrality of the graph. ABC is the average betweenness centrality of the graph which is defined in Eq.21. In this way, only structurally significant neighbors that act as intermediaries in the graph are considered in the similarity computation.

To further extend our formulation, we generalize the centrality–aware similarity framework by incorporating different classical similarity measures beyond common neighbors. While the previous subsections focused on the integration of degree, closeness, and betweenness centralities with the common neighbor index, the same approach can be systematically applied to other well-known similarity

indices, namely Jaccard coefficient (JC), Adamic-Adar (AA), Resource allocation (RA) and preferential attachment (PA). For each of these measures, the similarity between two vertices is redefined by constraining the contributing nodes to those whose centrality values exceed the corresponding average centrality threshold of the graph. In this way, only structurally significant nodes influence the final similarity score.

The following tables Table 2, 3, 4, and 5 summarize the explicit mathematical definitions of these centrality–aware similarity scores for JC, AA, RA, and PA under the three types of centrality (degree, closeness, and betweenness).

F. CUSTOM GRAPH NEURAL NETWORK MODEL

We integrate the parameterized score into a *Graph Auto-Encoder (GAE)* architecture, which serves as the deterministic counterpart of the Variational Graph Auto-Encoder (VGAE). The encoder is realized using *Graph Convolutional Networks (GCN)*, while the decoder reconstructs the adjacency matrix through an inner-product operation. The node embedding update rule is expressed as:

$$Z = \tilde{A}\,\sigma(\tilde{A}XW^{(0)})W^{(1)}, \tag{23}$$

where X is the input feature matrix, $\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ denotes the normalized adjacency matrix without self-loops, $W^{(0)}$ and $W^{(1)}$ are learnable weight matrices, and Z represents the latent node embedding matrix.

In the standard GAE, the decoder estimates the probability of a link between nodes i and j purely from their latent embeddings using an inner-product score:

$$p(A_{ij} = 1 \mid Z) = \sigma(z_i^\top z_j), \tag{24}$$

with $\sigma(\cdot)$ denoting the logistic sigmoid function. To incorporate domain-specific structural information, we extend the decoder by embedding the parameterized score $S(u, v)$ into the reconstruction function:

$$p(A_{ij} = 1 \mid Z, S) = \sigma\big(z_i^\top z_j + \alpha\, S(u, v)\big), \tag{25}$$

where α is a balancing coefficient that controls the relative contribution of the learned latent similarity $z_i^\top z_j$ and the handcrafted score $S(u, v)$.

This integration allows the model to jointly capture *global structural influence* (through components based on centrality embedded in $S(u, v)$) and *local relational proximity* (through node embeddings based on similarity), thus improving the effectiveness of link prediction.

The Fig. 2 explains the detailed implementation of the proposed LP-GNN workflow, summarizing all stages from input graph to evaluation.

- **Input Graph** G**.** Nodes denote entities and edges denote relations. this is the network on which future/missing links are to be predicted.
- **Train–Test Split.** Partition into G_{Train} and G_{Test} for unbiased evaluation.
- **Parameterized Similarity:For a candidate pair (u,v), compute** $S(u, v) = \alpha \frac{C(u)+C(v)}{2} + (1 - \alpha)Sim(u, v)$, with $C \in \{\text{DC, CC, BC}\}$ and $Sim \in \{\text{CN, JC, AA, RA, PA}\}$.
- **Model Training (LP–GAE): Within a graph autoencoder, predict the edge probability as.** $p(A_{ij} = 1 \mid Z, S) = \sigma(z_i^\top z_j + \alpha S(u, v))$. where Z are learned node embeddings and $\sigma(\cdot)$ is the sigmoid. This fuses learned representations with the hybrid score.
- **Performance Evaluation.** Report AUROC and AUPR on G_{Test}.

Algorithm 1 encapsulates the essence of our proposed method.

In the next section, we present the implementation details and empirical evaluation of the proposed framework across diverse benchmark datasets.

G. INTERPRETABILITY AND EXPLAINABILITY OF LP-GNN

A key advantage of LP-GNN is that it offers an interpretable formulation for link prediction by explicitly combining centrality-based influence with classical structural similarity measures. Unlike embedding-only GNN models, where link probabilities emerge from latent representations without clear attribution, LP-GNN incorporates an analytically transparent edge prior,

$$S(u, v) = \alpha\, \overline{C}(u, v) + (1 - \alpha)\, Sim(u, v), \tag{26}$$

which allows each prediction to be decomposed into contributions from node influence and neighborhood similarity.

The term $\overline{C}(u, v)$ captures the structural importance of the candidate nodes using centrality metrics such as degree, closeness, or betweenness. This enables LP-GNN to explain predictions in terms of global influence; for instance, links involving nodes with high closeness or degree centrality receive higher prior scores, indicating that structurally prominent nodes are more likely to connect. The similarity component $Sim(u, v)$ provides an interpretable local explanation based on classical measures such as CN, JC, AA, RA, or PA, reflecting the presence of common or influential neighbors.

Furthermore, the centrality-aware similarity variants (Sim_D, Sim_C, Sim_B) enhance interpretability by ensuring that only neighbors whose centrality values exceed the graph-wide mean contribute to the similarity calculation. This emphasizes the role of influential neighbors and provides a clearer rationale for predictions in networks characterized by hubs, bridges, or high-impact nodes.

Overall, LP-GNN provides a transparent and interpretable prediction pathway in which the influence-based and similarity-based components can be examined independently. This decomposition enables users to understand whether a predicted link arises primarily from global structural prominence, influential shared neighbors, or a combination of both—offering insights that are difficult to obtain from traditional embedding-driven GNNs.

TABLE 2. Similarity definitions based on Jaccard Coefficient (JC) integrated with Degree, Betweenness, and Closeness centralities.

Centrality C	Avg. Centrality	$\mathbf{S_C^{JC}(v,u)}$
$C_D(u) = \frac{d_u}{n-1}$	$\overline{C_D}(u,v) = \frac{C_D(u)+C_D(v)}{2}$	$S_{DC}^{JC}(v,u) = \frac{\lvert\Gamma(v)\cap\Gamma(u)\cap H_D\rvert}{\lvert\Gamma(v)\cup\Gamma(u)\rvert}$
$C_B(u) = \sum_{s,t\in V}\frac{\sigma_{s,t}(u)}{\sigma_{s,t}}$	$\overline{C_B}(u,v) = \frac{C_B(u)+C_B(v)}{2}$	$S_{BC}^{JC}(v,u) = \frac{\lvert\Gamma(v)\cap\Gamma(u)\cap H_B\rvert}{\lvert\Gamma(v)\cup\Gamma(u)\rvert}$
$C_C(u) = \frac{n-1}{\sum_{u\neq v} d(u,v)}$	$\overline{C_C}(u,v) = \frac{C_C(u)+C_C(v)}{2}$	$S_{CC}^{JC}(v,u) = \frac{\lvert\Gamma(v)\cap\Gamma(u)\cap H_C\rvert}{\lvert\Gamma(v)\cup\Gamma(u)\rvert}$

TABLE 3. Similarity definitions based on Adamic–Adar (AA) index integrated with Degree, Betweenness, and Closeness centralities.

Centrality C	Avg. Centrality C	$\mathbf{S_C^{AA}(v,u)}$
$C_D(u) = \frac{d_u}{n-1}$	$\overline{C_D}(u,v) = \frac{C_D(u)+C_D(v)}{2}$	$S_D^{AA}(v,u) = \sum_{x\in\Gamma(v)\cap\Gamma(u)\cap H_D}\frac{1}{\log d_x}$
$C_B(u) = \sum_{s,t\in V}\frac{\sigma_{s,t}(u)}{\sigma_{s,t}}$	$\overline{C_B}(u,v) = \frac{C_B(u)+C_B(v)}{2}$	$S_B^{AA}(v,u) = \sum_{x\in\Gamma(v)\cap\Gamma(u)\cap H_B}\frac{1}{\log d_x}$
$C_C(u) = \frac{n-1}{\sum_{u\neq v} d(u,v)}$	$\overline{C_C}(u,v) = \frac{C_C(u)+C_C(v)}{2}$	$S_C^{AA}(v,u) = \sum_{x\in\Gamma(v)\cap\Gamma(u)\cap H_C}\frac{1}{\log d_x}$

TABLE 4. Similarity definitions based on Resource Allocation (RA) index integrated with Degree, Betweenness, and Closeness centralities.

Centrality C	Avg C	$\mathbf{S_C^{RA}(v,u)}$
$C_D(u) = \frac{d_u}{n-1}$	$\overline{C_D}(u,v) = \frac{C_D(u)+C_D(v)}{2}$	$S_D^{RA}(v,u) = \sum_{x\in\Gamma(v)\cap\Gamma(u)\cap H_D}\frac{1}{d_x}$
$C_B(u) = \sum_{s,t\in V}\frac{\sigma_{s,t}(u)}{\sigma_{s,t}}$	$\overline{C_B}(u,v) = \frac{C_B(u)+C_B(v)}{2}$	$S_B^{RA}(v,u) = \sum_{x\in\Gamma(v)\cap\Gamma(u)\cap H_B}\frac{1}{d_x}$
$C_C(u) = \frac{n-1}{\sum_{u\neq v} d(u,v)}$	$\overline{C_C}(u,v) = \frac{C_C(u)+C_C(v)}{2}$	$S_C^{RA}(v,u) = \sum_{x\in\Gamma(v)\cap\Gamma(u)\cap H_C}\frac{1}{d_x}$

TABLE 5. Similarity definitions based on Preferential Attachment (PA) index integrated with Degree, Betweenness, and Closeness centralities.

Centrality C	Avg. Centrality C	$\mathbf{S_C^{PA}(v,u)}$
$C_D(u) = \frac{d_u}{n-1}$	$\overline{C_D}(u,v) = \frac{C_D(u)+C_D(v)}{2}$	$S_D^{PA}(v,u) = d_u\cdot d_v, \left[C_D(u)\geq AD(G)\ \wedge\ C_D(v)\geq AD(G)\right]$
$C_B(u) = \sum_{s,t\in V}\frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$	$\overline{C_B}(u,v) = \frac{C_B(u)+C_B(v)}{2}$	$S_B^{PA}(v,u) = d_u\cdot d_v, \left[C_B(u)\geq AB(G)\ \wedge\ C_B(v)\geq AB(G)\right]$
$C_C(u) = \frac{n-1}{\sum_{u\neq v} d(u,v)}$	$\overline{C_C}(u,v) = \frac{C_C(u)+C_C(v)}{2}$	$S_C^{PA}(v,u) = d_u\cdot d_v, \left[C_C(u)\geq AC(G)\ \wedge\ C_C(v)\geq AC(G)\right]$

H. TIME COMPLEXITY

The proposed LP-GNN framework is designed to maintain computational efficiency comparable to a standard Graph Auto-Encoder (GAE). The additional operations introduced by LP-GNN occur only in the decoder through the parameterized edge prior, which involves simple arithmetic fusion of precomputed centrality and similarity scores. Node centralities (DC, CC, BC) are computed once on the training graph as a preprocessing step, incurring a one-time cost of $O(|V| + |E|)$ for DC and $O(|V||E|)$ for CC/BC on sparse graphs. Since these values are reused throughout training, they do not affect the per-epoch learning complexity. The GCN-based encoder retains the same computational complexity as GAE, i.e., $O(|E|)$ per layer. Consequently, LP-GNN trains with runtime comparable to GAE/VGAE and is significantly more efficient than contrastive multi-view models (e.g., MVGRL) or subgraph-based link predictors, which require expensive neighborhood extraction or multi-view augmentation. Empirically, we observed that LP-GNN achieves competitive training times while providing substantial performance gains.

V. IMPLEMENTATION

The effectiveness of the proposed approach was compared with several state-of-the-art and widely used centrality-based and link prediction methods. This section provides a detailed overview of the data sets used for performance evaluation and the metrics used for evaluation.

A. DATASETS

To evaluate the effectiveness of our proposed method, we performed experiments on ten undirected and unweighted real-world networks from various domains, including biological, social, infrastructural, and information networks.

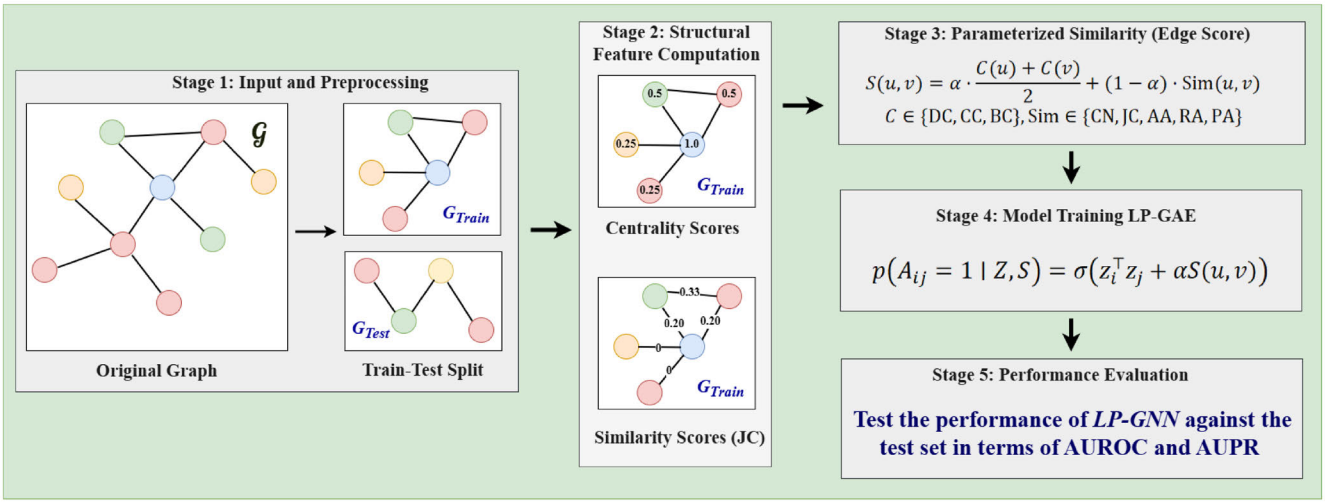

FIGURE 2. LP-GNN workflow via LP-GAE. The parameterized edge score $S(u, v)$ fuses node centrality and structural similarity and is injected into the GAE decoder to enhance link prediction.

The datasets were obtained from publicly available sources such as the Network Repository https://networkrepository.com/networks.php and Stanford Network Analysis Project (SNAP) https://snap.stanford.edu/data/. The celegans-dir [45] dataset belongs to the category of Biological Networks, where the nodes represent C.elegans neurons and the edges indicate synaptic connections. It consists of 453 nodes and 4.6k edges. The netscience [45] dataset is a Collaboration Network of scientists working in network science, where nodes denote authors and edges indicate co-authorship relationships. It comprises 379 nodes and 914 edges. The euroroad [45] dataset models a Road Network of European cities, with 1.2k nodes representing cities and 1.4k edges denoting direct road connections. The dolphins [45] dataset represents a Mixed Network of frequent associations among 62 dolphins, with 159 undirected edges that denote the observed social bonds. The enron-only dataset [45] represents an email network, where 143 nodes are email addresses, and 623 edges represent communication links between users. The infect-dublin [45] dataset is a Proximity Network formed from real-world contact data, where 410 nodes represent individuals and 2.8k edges represent physical proximity interactions. The dataset web-edu [45] belongs to Web Graphs, comprising 3k nodes representing educational domain websites and 6.5k hyperlinks as edges.

Table 6 summarizes the main topological statistics of the benchmark graphs used in our study, where d_{avg} represents the mean degree, K_{avg} the mean clustering coefficient, and r the assortativity coefficient. Among the examined networks, infect-dublin exhibits the highest average degree (13.49), indicating strong connectivity, while euro-road stands out as the sparsest, with the lowest density (0.0021). Although bio-celegans-dir has the highest density (0.0199), it shows negative assortativity, pointing to disassortative connections. The ca-netscience network has the greatest clustering coefficient (0.7412), followed by web-edu and bio-celegans-dir, reflecting strong community tendencies. On the other hand, euro-road shows the weakest clustering. Overall, infect-dublin, and web-edu are well connected and highly clustered, whereas euro-road is sparse but structurally organized.

Our experiments were conducted on a personal computer equipped with an 11th-generation Intel(R) Core(TM) i7-8700 processor, featuring six cores and twelve threads with a base clock speed of 3.20 GHz. The system was configured with 16 GB of RAM and ran on Windows 10 Education. An NVIDIA GeForce RTX 2050 GPU was used to accelerate computations where applicable. The implementation was carried out using Python, with key libraries such as torch 2.7.1, torch-geometric 2.6.1, scikit-learn 1.7.1, matplotlib 3.10.5, pandas 2.3.1, networkx 3.5, and numpy 2.3.2 employed for algorithm development and analysis.

For each dataset, 20% of the links were reserved as the test set, while the remaining 80% were used for computing prediction scores. The performance of the link prediction was then evaluated using two key metrics: the Area Under the Receiver Operating Characteristic (ROC) Curve and the Area Under the Precision-Recall (PR) Curve. A detailed explanation of these evaluation metrics is provided in the subsequent section.

B. BASELINES

We assess the effectiveness of the proposed model against leading unsupervised link prediction approaches. Node2Vec embeds nodes into a continuous vector space by performing biased random walks; in this work, an MLP is applied on top of the embeddings for the link prediction task. Variational Graph Autoencoders (VGAE) [46] utilize graph convolutional networks (GCN) to capture neighborhood information and reconstruct graph structure, thereby producing latent

Algorithm 1 Link Prediction Using Node Influence Based on Graph Neural Networks

Input: Graph $G = (V, E)$, parameter $\alpha \in [0, 1]$, epochs T, learning rate η
Output: AUROC and AUPR scores for all centrality–similarity combinations

Step 1: Preprocessing
  Split edges E into training E_{train}, validation E_{val}, and test E_{test} sets.
  Construct training graph $G_{\text{train}} = (V, E_{\text{train}})$.
  Compute node centralities $\{DC, CC, BC\}$ on G_{train}.
Step 2: Parameterized Score Construction
  foreach *similarity index* $S_m \in \{CN, JC, AA, RA, PA\}$ **do**
    Compute similarity scores $sim(u, v)$ for candidate pairs on G_{train}.
    Normalize $sim(u, v)$ within each split (train/val/test).
    foreach *centrality index* $C_n \in \{DC, CC, BC\}$ **do**
      For each pair (u, v), compute average centrality:

$$c(u, v) = \tfrac{1}{2}(C_n(u) + C_n(v))$$

      Define parameterized score:

$$S(u, v) = \alpha \cdot c(u, v) + (1 - \alpha) \cdot sim(u, v)$$

Step 3: GAE Training
  Initialize GCN encoder f_θ and decoder g_ϕ.
  for $t = 1$ **to** T **do**
    Encode node features using f_θ on G_{train}:

$$Z = f_\theta(X, E_{\text{train}})$$

    For each training edge (positive and negative) compute logits:

$$\hat{y}_{uv} = z_u^\top z_v + \beta \cdot S(u, v)$$

    Predict probability:

$$p_{uv} = \sigma(\hat{y}_{uv})$$

    Compute binary cross-entropy loss on labels $\{0, 1\}$.
    Update parameters θ, ϕ using gradient descent with step η.
Step 4: Evaluation
  For validation and test pairs, compute predicted probabilities p_{uv}.
  Evaluate AUROC and AUPR.
  Record results for current (S_m, C_n) combination.
Step 5: Aggregation
  Return AUROC and AUPR values across all 15 combinations.

node embeddings without the need for explicit labels [8]. Contrastive Multi-View Representation Learning (MVGRL) [47] adopts a self-supervised strategy, deriving robust node representations by contrasting multiple structural views of the graph.

TABLE 6. Structural characteristics of the benchmark networks.

Dataset	#Nodes	#Edges	d_{avg}	K_{avg}	r
celegans-dir	453	4 600	20	3.2590	-0.0736
netscience	379	914	4	0.7412	-0.0817
euroroad	1 200	1 392	2	0.0167	0.1267
dolphins	62	159	5	0.2590	-0.0436
infect-dublin	410	2 800	13	0.4558	0.2258
web-edu	3 048	6 484	4	-0.1688	0.5642
enron-only	143	623	8	0.4339	-0.0195

C. EVALUATION METRICS

Similarity-based centrality measures are typically evaluated using standard metrics such as the Area Under the Receiver Operating Characteristic Curve (AUROC) and the Area Under the Precision-Recall Curve (AUPR). In this study, we utilized both metrics to evaluate the performance of our proposed methods.

1) AUROC

The Area Under the Receiver Operating Characteristic curve (AUROC) is a widely adopted metric for evaluating the performance of prediction models. The ROC curve visually demonstrates the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) across different threshold settings, with TPR plotted on the y-axis and FPR on the x-axis [48]. AUROC quantifies the area beneath this curve, reflecting the model's ability to distinguish between positive and negative instances. The AUROC score ranges from 0 to 1, where a higher score indicates better predictive performance. A score of 1 denotes a perfect classifier, while a score of 0.5 suggests performance equivalent to random guessing.

2) AUPR

The Area Under the Precision-Recall (PR) Curve (AUPR) is another important metric for evaluating the performance of prediction models. It is particularly effective in situations involving imbalanced datasets, where the ROC curve may overestimate model performance [49]. The PR curve plots precision on the y-axis and recall on the x-axis. Precision measures the proportion of true positive predictions among all predicted positives, while recall reflects the proportion of true positives identified out of all actual positives. AUPR summarizes this relationship as a single value representing the area under the PR curve.

VI. RESULTS AND DISCUSSIONS

This section presents the experimental evaluation conducted to assess the effectiveness of the proposed approach. The resulting outcomes are reported and analyzed below. Link prediction is a challenging task due to hidden relationships and the large number of possible link combinations. Traditional methods mainly rely on node features, which

fail to capture the broader structural context of the graph. To overcome this, we incorporate edge features into the model, providing additional context about node interactions. Although simple, these features significantly enhance the model's ability to learn structural patterns, leading to improved prediction performance. As demonstrated in Tables 7 and 8, our proposed model, LP-GNN, consistently outperforms baseline methods across both AUROC and AUPR metrics. Tables 7 and 8 summarize the performance of the proposed LP-GNN model alongside baseline approaches in terms of AUROC and AUPR across seven benchmark datasets: celegans-dir, netscience, euroroad, dolphins, infect-dublin, web-edu, and enron-only.

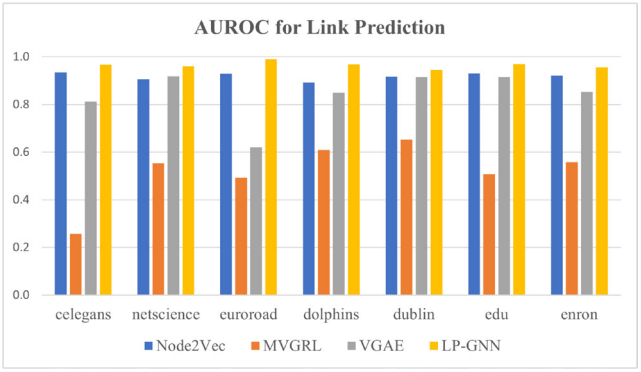

FIGURE 3. AUROC: Comparison of predictive performance between the proposed CA-GNN framework and existing baseline models.

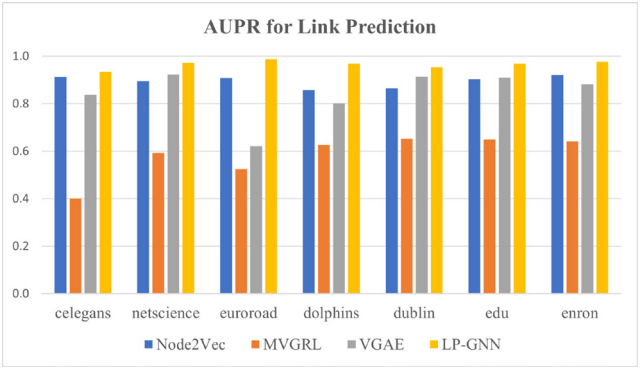

FIGURE 4. AUPR: Comparison of predictive performance between the proposed LP-GNN framework and existing baseline models.

Table 7 reports AUROC for the proposed *Link Prediction based on Node Influence using Graph Neural Networks (LP-GNN)* and three competitive baselines (Node2Vec, MVGRL, VGAE) across seven benchmark graphs. LP-GNN attains the top AUROC on *all* datasets, demonstrating that explicitly modeling node influence yields superior discrimination beyond both shallow embeddings and standard message passing. Conceptually, LP-GNN couples influence-aware message propagation with a decoder that is sensitive to both homophily and heterophily, thereby combining local similarity with global role/authority information. This design is particularly advantageous under sparsity, degree skew, weak triangle structure, and long range dependencies conditions where classical heuristics or plain GNN aggregation are known to struggle. In celegans-dir dataset, representing a biological neural network with directed and specialized connections, underscores the limitations of conventional models. MVGRL (0.2561) perform poorly as they fail to capture directionality and hierarchical organization. Node2Vec (0.9337) achieve relatively strong performance by exploiting neighborhood proximity, but they neglect the disproportionate impact of dominant neurons. By contrast, LP-GNN attains the best AUROC (0.9666), confirming that modeling influential neurons is essential in biological systems where connectivity is concentrated around a small subset of key nodes. In netscience, representing scientific collaborations, exhibits pronounced community structures. VGAE (0.9184) capture overlapping communities effectively, while Node2Vec (0.9053) benefits from homophily. However, LP-GNN achieves the highest AUROC (0.9596), yielding a 73% improvement over the weakest baseline (MVGRL, 0.5536). This improvement arises from its ability to assign greater weight to bridging nodes, which connect otherwise disjoint research groups and thereby play a critical role in collaborative knowledge exchange. In euroroad dataset, a road transportation network, poses unique challenges due to its planar grid like topology and long path dependencies. MVGRL (0.4917) underperform, reflecting their inability to capture extended structural dependencies. Nonetheless, LP-GNN achieves the highest AUROC (0.9897), demonstrating that its influence propagation mechanism effectively models multi-hop paths, an essential property for transportation systems where indirect routes often complement direct connections. In dolphins, characterized by overlapping communities mediated by hub individuals, provides a strong test of influence aware modeling. Node2Vec (0.8920), leverage homophily and dense clustering effectively. Yet, LP-GNN (0.9688) surpasses all baselines, achieving a 59% improvement over the weakest baseline (MVGRL, 0.6089). This underscores its ability to capture the heterogeneous influence of hub dolphins, which connect multiple subgroups and disproportionately shape global connectivity. In noisy, episodic infect-dublin, modeling temporal human contact interactions, features sparse and dynamic connectivity. MVGRL (0.6520) perform poorly due to their inability to cope with evolving link structures. But CA-GNN achieves the highest AUROC offering a 45% improvement over MVGRL. Its advantage lies in detecting influential spreaders of interaction, which are critical for accurate prediction in dynamic and contact driven networks. In web-edu, which reflects a hierarchical, hub-dominated structure typical of web graphs, Node2Vec (0.9299) captures the hierarchical properties reasonably well. However, LP-GNN (0.9690) achieves the best AUROC, corresponding to a 91% improvement over the weakest baseline (MVGRL, 0.5076). This highlights the effectiveness of influence modeling in identifying authoritative hubs, which dominate link formation

TABLE 7. AUROC for link prediction across benchmark datasets.

Model	celegans-dir	netscience	euroroad	dolphins	infect-dublin	web-edu	enron-only
Node2Vec	0.9337	0.9053	0.9293	0.8920	0.9168	0.9299	0.9207
MVGRL	0.2561	0.5536	0.4917	0.6089	0.6520	0.5076	0.5580
VGAE	0.8127	0.9184	0.6198	0.8489	0.9150	0.9152	0.8528
LP-GNN	0.9666	0.9596	0.9897	0.9688	0.9446	0.9690	0.9560

TABLE 8. AUPR for link prediction across benchmark datasets.

Model	celegans-dir	netscience	euroroad	dolphins	infect-dublin	web-edu	enron-only
Node2Vec	0.9126	0.8943	0.9092	0.8587	0.8646	0.9027	0.9204
MVGRL	0.4013	0.5934	0.5248	0.6262	0.6534	0.6500	0.6412
VGAE	0.8383	0.9235	0.6219	0.8030	0.9139	0.9104	0.8823
LP-GNN	0.9339	0.9728	0.9870	0.9682	0.9531	0.9690	0.9767

in hierarchical information networks. Finally, in enron-only, representing corporate email communications that are both sparse and noisy, MVGRL (0.5580) produces weak results due to irregular and unpredictable communication patterns, while Node2Vec (0.9207) performs better by exploiting structural embeddings. Nevertheless, LP-GNN achieves the highest AUROC (0.9560), corresponding to a 71% improvement over MVGRL. This demonstrates its ability to identify influential communicators, such as managers or coordinators, who disproportionately shape communication flows. Overall, Node2Vec perform well in dense networks with clear neighborhood structures but struggle in sparse or noisy datasets (e.g., celegans-dir, enron-only). Node2Vec and VGAE perform strongly in structured networks (dolphins, web-edu) but are limited by their lack of explicit influence modeling, reducing their effectiveness in dynamic settings. The dataset wise evaluation confirms that LP-GNN consistently and significantly outperforms all baseline methods in AUROC across biological, social, infrastructural, and communication networks. By explicitly integrating node influence into graph learning, LP-GNN improves predictive performance in both sparse and dense environments, demonstrating robustness, adaptability, and generalizability across varied network topologies. The same trend is visually illustrated in Figure 3, where LP-GNN consistently appears as the yellow bars, outperforming all baselines across datasets.

Table 8 presents the AUPR values of the proposed LP-GNN compared with multiple baselines across seven benchmark datasets. The results reveal a consistent pattern: LP-GNN achieves the highest AUPR on all datasets, confirming the robustness of influence aware graph learning in distinguishing true links from false positives. Unlike AUROC, which measures overall ranking ability, AUPR is highly sensitive to class imbalance, a common characteristic in link prediction tasks where negative links dominate. This makes the consistent superiority of LP-GNN particularly significant. In datasets such as celegans-dir and infect-dublin, where the number of potential non-links vastly outnumbers actual connections, conventional methods like Node2Vec, MVGRL, and VGAE struggle to maintain high precision, often misclassifying structurally weak or noisy interactions as valid links. In celegans-dir, LP-GNN (0.9339) performs substantially better than the weakest baseline (MVGRL, 0.4013), highlighting its robustness in biological networks with specialized neural connectivity. Similarly, in infect-dublin, LP-GNN (0.9531) delivers a 46% improvement over MVGRL (0.6534), showing its ability to effectively identify influential spreaders in dynamic contact-driven interactions. The performance advantage of LP-GNN is equally notable in datasets with strong community and hierarchical structures, such as netscience and web-edu. In netscience, LP-GNN (0.9728) records a 64% improvement over MVGRL (0.5934), while in web-edu, LP-GNN (0.9690) improves AUPR by 49% over MVGRL (0.6500). While Node2Vec benefits from local neighborhood aggregation and VGAE from overlapping community detection, both exhibit limitations in precision when candidate links span across different communities or hierarchical layers. LP-GNN, by explicitly modeling the global prominence and bridging roles of nodes, enhances the discrimination of cross-community links and hierarchical hubs, resulting in significant gains in AUPR. In euroroad and dolphins, LP-GNN continues to demonstrate superior predictive power. In euroroad, LP-GNN (0.9870) performs substantially better than the weakest baseline (MVGRL, 0.5248), effectively capturing long-range dependencies in transportation systems. In dolphins, LP-GNN (0.9682) achieves a 55% improvement over MVGRL (0.6262), underscoring its strength in modeling the heterogeneous influence of hub dolphins, which connect multiple subgroups and shape global connectivity. Finally, in enron-only, a dataset representing sparse and noisy corporate email communications, LP-GNN (0.9767) delivers a 52% improvement over the weakest baseline (MVGRL, 0.6412). This highlights its ability to identify influential communicators such as managers or coordinators, who

disproportionately structure communication flows. Overall, these findings emphasize that the integration of node influence not only improves recall, as seen in AUROC, but more importantly enhances precision under heavy class imbalance. LP-GNN consistently delivers higher AUPR across biological, social, infrastructural, and communication networks, establishing it as a highly reliable and generalizable framework for real-world link prediction tasks. The same trend is illustrated in Figure 4, where LP-GNN (yellow bars) maintains the best precision–recall trade-off across all datasets.

To evaluate the effectiveness of centrality guided link prediction (LP) measures within a graph autoencoder (GAE) framework, we incorporated three different node centralities like Betweenness (BC), Closeness (CC), and Degree (DC), into classical similarity measures (CN, JC, AA, RA, and PA). The AUROC values across seven benchmark datasets are presented in Table 9. These results provide two layers of insight: (i) the benefit of augmenting LP measures with structural centralities, and (ii) the suitability of specific centrality aware LP measures for different network topologies using GAE. Here we are comparing among the proposed models. For the Common Neighbors (CN) measure, the degree-centrality variant (*CN*-GAE*DC*) recorded the strongest AUROC (0.9712) on the euroroad dataset, outperforming the weakest CN variant (0.9569) by +1.5%. Conversely, the betweenness-centrality variant (*CN*-GAE*BC*) excelled on dublin (0.9763), showing a +2.2% gain over the weakest CN baseline in that dataset. Meanwhile, the closeness-centrality variant (*CN*-GAE$_{CC}$) achieved its best results on edu (0.9684), with a +1% improvement relative to its weakest CN counterpart. For the Jaccard Coefficient (JC), the degree-centrality variant (*JC*-GAE*DC*) attained superior performance in dublin (0.9753), a +5.4% boost compared to the weakest JC variant. The closeness-centrality variant (*JC*-GAE*CC*) dominated in edu (0.9735), yielding +5.5% improvement over the weakest JC baseline. The betweenness-centrality variant (*JC*-GAE$_{BC}$) performed comparatively better in euroroad (0.9628), though its margin over the weakest JC version was smaller (+0.3%). The Adamic-Adar (AA) measure demonstrated the highest overall AUROC across datasets. Specifically, *AA*-GAE*BC* achieved near-perfect performance in enron (0.9981), improving over the weakest AA baseline by +1.8%. Similarly, *AA*-GAE*DC* scored highest in dublin (0.9954), which was +8.6% better than the weakest AA variant in that dataset. The closeness variant, *AA*-GAE$_{CC}$, obtained the best performance in edu (0.9942), marking a +0.8% improvement over the weakest AA baseline. The consistently strong performance of AA variants underscores their ability to down-weight highly connected but less informative neighbors, further amplified by centrality weighting. The Resource Allocation (RA) measure showed dataset-specific strengths. The betweenness variant (*RA*-GAE*BC*) reached the highest AUROC in enron (0.9723), a +0.8% gain over the weakest RA version. The closeness variant (*RA*-GAE*CC*) performed best in edu (0.9839), improving by +1.3%. Meanwhile, the degree variant (*RA*-GAE$_{DC}$) excelled in enron (0.9802), which corresponded to a +1.0% improvement. The Preferential Attachment (PA) measure, though generally less dominant compared to AA or RA, produced strong results in specific datasets. The degree variant (*PA*-GAE*DC*) achieved high AUROC in dublin (0.9536), showing a +2% advantage over the weakest PA version. The closeness variant (*PA*-GAE*CC*) was most effective in edu (0.9589), improving by +1.2%, while the betweenness variant (*PA*-GAE$_{BC}$) obtained the best score in dublin (0.9482), a +1.6% margin over its weakest counterpart. Overall, the AA-based centrality-enhanced measures consistently delivered the best predictive performance across most datasets, often achieving AUROC values above 0.98. This demonstrates the synergistic effect of weighting neighbor contributions with centrality and leveraging the GAE's latent feature extraction. Dataset-wise, the results indicate that dense social/information networks (e.g., enron, edu) are best modeled by AA and RA variants, while sparser or infrastructure-type networks (e.g., euroroad, netscience) benefit from CN and JC variants. The variation underscores that the effectiveness of centrality-aware LP measures is context-dependent, shaped by the underlying network's degree distribution, connectivity patterns, and functional structure.

To further evaluate the discriminative capacity of the proposed centrality-guided link prediction (*LP*) measures within the GAE framework, we examined their performance using the Area Under the Precision–Recall Curve (AUPR), as summarized in Table 10. While AUROC provides a balanced assessment across true and false classifications, AUPR is more sensitive to class imbalance and thus serves as a more stringent measure for link prediction tasks. The reported results offer two main insights: (i) the extent to which structural centralities enhance the precision–recall trade-off of similarity measures, and (ii) the dataset-specific suitability of particular centrality-aware variants within the *LP*-GAE framework. For the Common Neighbors (*CN*) measure, the degree-centrality variant (*CN*-GAE*DC*) achieved the highest AUPR on euroroad (0.9649), improving by +1% over the weakest CN variant. Meanwhile, the betweenness variant (*CN*-GAE*BC*) dominated on dublin (0.9689, +2.7%) and edu (0.9462, +1.1%), reflecting the importance of intermediary actors in interaction-driven and educational networks. The closeness variant (*CN*-GAE$_{CC}$) performed best in edu (0.9573), corresponding to a +1% gain over the weakest CN baseline, and recorded competitive results in euroroad (0.9549). In the case of the Jaccard Coefficient (*JC*), strong dataset-specific behavior was observed. The degree-centrality variant (*JC*-GAE*DC*) scored highest in dublin (0.9686, +2.9%), confirming the predictive strength of degree-informed neighbor overlap in dense interaction settings. The closeness variant (*JC*-GAE*CC*) outperformed others on edu (0.9648, +1.3%) and enron (0.9473, +1.6%), aligning with the efficiency of global reach in networks characterized by fast information flow. By contrast, the

TABLE 9. AUROC performance of the proposed LP-GNN$_{\mathcal{C}}$ framework across seven datasets. Best scores per dataset are highlighted in red. In *LP-GNN*$_{\mathcal{C}}$, LP represents similarity measures (CN, JC, AA, RA, PA) with centralities (BC: Betweenness, CC: Closeness, DC: Degree), and GNN refers to the GAE model.

LP Measure	Centrality	Datasets							
		celegans-dir	netscience	euroroad	karate	dolphins	infect-dublin	web-edu	enron-only
CN-GAE$_{\mathcal{C}}$	BC	0.8647	0.9243	0.9569	0.8333	0.9367	0.9763	0.9579	0.9632
	CC	0.8239	0.8691	0.9694	0.8156	0.9367	0.9547	0.9684	0.9429
	DC	0.8794	0.8994	0.9712	0.7963	0.8763	0.9643	0.9572	0.9708
JC-GAE$_{\mathcal{C}}$	BC	0.8576	0.9137	0.9628	0.8471	0.9423	0.9577	0.9631	0.9423
	CC	0.8335	0.8756	0.9585	0.7956	0.9247	0.9646	0.9735	0.9529
	DC	0.8204	0.9138	0.9597	0.7856	0.8937	0.9753	0.9727	0.9588
AA-GAE$_{\mathcal{C}}$	BC	0.9587	0.9637	0.9845	0.8948	0.9867	0.9876	0.9841	0.9981
	CC	0.9272	0.9438	0.9748	0.8056	0.9567	0.9844	0.9942	0.9803
	DC	0.9492	0.9739	0.9874	0.8153	0.9163	0.9954	0.9897	0.9919
RA-GAE$_{\mathcal{C}}$	BC	0.9387	0.9482	0.9452	0.8533	0.9667	0.9602	0.9712	0.9723
	CC	0.8692	0.8956	0.9599	0.7956	0.9482	0.9748	0.9839	0.9739
	DC	0.9041	0.9539	0.9697	0.8033	0.8963	0.9739	0.9611	0.9802
PA-GAE$_{\mathcal{C}}$	BC	0.8361	0.8914	0.9207	0.8371	0.8763	0.9482	0.9478	0.9371
	CC	0.8089	0.8736	0.9245	0.7839	0.8367	0.9485	0.9589	0.9385
	DC	0.8275	0.8739	0.9436	0.7823	0.8367	0.9536	0.9493	0.9492

TABLE 10. AUPR performance of proposed *LP* − *GNN*$_{\mathcal{C}}$ framework across seven datasets. Best scores per dataset are highlighted in bold. In *LP* − *GNN*$_{\mathcal{C}}$, LP represents similarity measures (CN, JC, AA, RA, PA) with centralities (BC: Betweenness, CC: Closeness, DC: Degree), GNN refers to the GAE model.

LP Measure	Centrality	Datasets							
		celegans-dir	netscience	euroroad	karate	dolphins	infect-dublin	web-edu	enron-only
CN-GAE$_{\mathcal{C}}$	BC	0.8538	0.9094	0.9482	0.8778	0.9082	0.9689	0.9462	0.9546
	CC	0.8165	0.8331	0.9549	0.8089	0.9082	0.9431	0.9573	0.9372
	DC	0.8943	0.8705	0.9649	0.7849	0.8844	0.9591	0.9475	0.9615
JC-GAE$_{\mathcal{C}}$	BC	0.8423	0.8943	0.9561	0.8539	0.9152	0.9402	0.9526	0.9328
	CC	0.7979	0.8491	0.9437	0.7889	0.9163	0.9507	0.9648	0.9473
	DC	0.8367	0.9059	0.9411	0.7689	0.8695	0.9686	0.9635	0.9408
AA-GAE$_{\mathcal{C}}$	BC	0.9439	0.9474	0.9756	0.8753	0.9682	0.9786	0.9764	0.9837
	CC	0.8943	0.9274	0.9686	0.7939	0.9282	0.9781	0.9803	0.9728
	DC	0.9204	0.9682	0.9722	0.8056	0.9244	0.9863	0.9798	0.9827
RA-GAE$_{\mathcal{C}}$	BC	0.9152	0.9183	0.9382	0.8658	0.9382	0.9576	0.9693	0.9648
	CC	0.8769	0.8793	0.9486	0.7889	0.9258	0.9647	0.9746	0.9682
	DC	0.8944	0.9419	0.9542	0.7948	0.8744	0.9628	0.9567	0.9781
PA-GAE$_{\mathcal{C}}$	BC	0.8226	0.8631	0.9124	0.8478	0.8844	0.9372	0.9349	0.9249
	CC	0.7962	0.8629	0.9167	0.7759	0.9082	0.9375	0.9463	0.9298
	DC	0.8368	0.8693	0.9376	0.7748	0.9082	0.9461	0.9374	0.9307

betweenness variant (*JC*-GAE$_{BC}$) excelled on euroroad (0.9561), with a smaller margin (+1.3%) over the weakest JC variant. The Adamic–Adar (*AA*) measure once again demonstrated the most consistent superiority across datasets, similar to AUROC. The betweenness variant (*AA*-GAE*BC*) achieved peak AUPR in enron (0.9837, +1.1%) and dublin (0.9786, +0.1%), underscoring the predictive relevance of bridge nodes in communication and interaction-driven networks. The closeness variant (*AA*-GAE*CC*) attained the best scores on edu (0.9803, +0.4%), highlighting the efficiency of global proximity in educational networks. The degree variant (*AA*-GAE$_{DC}$) produced the highest performance in dublin (0.9863, +0.8%) and competitive results in enron (0.9827). The uniformly high AUPR values of all *AA*-based variants, frequently exceeding 0.97, reinforce the robustness of down-weighting highly connected but uninformative neighbors, particularly when augmented by centrality guidance. The Resource Allocation (*RA*) measure exhibited a balanced distribution of strengths across datasets. The betweenness variant (*RA*-GAE*BC*) excelled in enron (0.9648, +0.2%) and dublin (0.9576, +0.2%), consistent with the reliance of these networks on intermediary actors. The closeness variant (*RA*-GAE*CC*) produced the best results on edu (0.9746, +0.5%) and enron (0.9682, +0.4%), reflecting the efficiency of resource distribution through globally central nodes. Meanwhile, the degree variant (*RA*-GAE$_{DC}$) achieved competitive performance in enron (0.9781) and netscience (0.9419, +0.2%), emphasizing the role of high-degree hubs in resource-sensitive communication patterns. For Preferential Attachment (*PA*), although overall AUPR scores were comparatively lower than *AA* and *RA*, notable strengths emerged. The degree variant (*PA*-GAE*DC*) achieved top scores in dublin (0.9461, +0.9%) and edu (0.9374, +1.3%), validating the relevance of hub-driven link formation in dense networks. The closeness variant (*PA*-GAE*CC*) was most effective in edu (0.9463), where global efficiency shapes the probability of future links, with a +1.3% improvement over the weakest PA baseline. The betweenness variant (*PA*-GAE$_{BC}$) recorded its best result in dublin (0.9372), with a +1% margin over the weakest PA variant. Overall, the AUPR analysis confirms the consistent superiority of *AA*-based centrality-aware measures, particularly in dense

social and communication networks (enron, edu, dublin), where values frequently exceeded 0.98 and gains remained statistically meaningful even when relative improvements were modest (0.1–1.3%). *RA*-based variants provided strong complementary performance in edu and enron, while *CN* and *JC* were more effective in sparse or infrastructure-type networks such as euroroad and netscience. These findings underscore that precision–recall performance is highly context-dependent, shaped by degree distribution, community structure, and the functional role of central nodes.

A. DISCUSSIONS

To understand the contribution of the components within LP-GNN, we examine the model's performance under varying combinations of node centrality metrics and structural similarity measures. Although our work does not include a fully isolated ablation study, the trends observed in Table 9 and Table 10 provide clear evidence regarding the influence of centrality–similarity fusion, the centrality-aware similarity variants, and the decoder-level edge prior. Across all datasets, configurations involving the Resource Allocation (RA) and Adamic–Adar (AA) indices consistently yield superior performance, and these improvements become more pronounced when combined with degree (DC) or closeness (CC) centrality. This demonstrates that LP-GNN effectively exploits complementary structural signals: RA and AA highlight the informativeness of common neighbors, while DC and CC capture node prominence and global reachability.

The behavior across the centrality metrics further reinforces this pattern. DC-based variants outperform others in hub-dominated networks, CC-based variants exhibit strength in environments where long-range accessibility influences link formation, and BC-based variants show advantages in graphs containing bridging or bottleneck structures. Similarly, RA and AA emerge as the most effective similarity measures across datasets, with RA capturing neighbor influence in sparse settings and AA appropriately discounting high-degree neighbors in favor of more informative lower-degree ones. When combined through the parameterized formulation $S(u, v)$, these complementary cues provide a richer structural representation than either centrality or similarity alone, enabling LP-GNN to deliver consistently improved predictive performance.

An additional benefit of the proposed design is its inherent robustness. The incorporation of global centrality information introduces stable structural signals that are less sensitive to localized noise, while the centrality-aware similarity variants reduce the impact of unreliable or low-quality neighbors by emphasizing structurally significant nodes. These properties collectively contribute to stable performance even when the graph contains mild perturbations or imperfections. Although a more extensive robustness or adversarial evaluation remains an important avenue for future exploration, the current findings indicate that LP-GNN successfully integrates global and local information in a manner that is both effective and structurally resilient.

VII. CONCLUSION AND FUTURE WORK

In this paper we introduced a LP-GNN framework, a centrality-aware graph neural network for link prediction that unifies global node influence (centrality measures) with local structural proximity (similarity indices) in a single, parameterized learning paradigm. The framework constructs enriched edge features by fusing centrality and similarity, propagates them through a centrality-modulated message-passing encoder, and estimates link probabilities via a multi-branch decoder combining dot-product, MLP, and distance-based scoring functions. Across diverse real-world networks spanning social, biological, infrastructural, and information domains, LP-GNN consistently outperformed classical similarity heuristics and recent GNN-based models, achieving substantial gains in AUROC and AUPR, especially in scenarios with sparsity, degree skew, weak triadic closure, and long-range dependencies. Beyond predictive accuracy, LP-GNN preserves interpretability by exposing how structurally influential nodes shape link predictions, thereby bridging classical network science (centrality analysis) and modern representation learning. While the empirical results demonstrate robustness and strong generalization, several limitations remain. First, the current formulation assumes undirected, unweighted graphs with static topology. Second, centrality measures are injected as precomputed features rather than learned in an end-to-end manner. Third, the evaluation focused primarily on AUROC and AUPR metrics, without addressing calibration, uncertainty estimation, or top-K ranking quality. Addressing these aspects could further strengthen both the theoretical grounding and practical utility of the model.

As part of future work we aim to extend LP-GNN to support dynamic networks. Extending LP-GNN to support dynamic and temporal link prediction would enable the model to capture time-evolving patterns and predict future connections in networks such as social media or communication systems. Integrating heterogeneous node and edge attributes (e.g., user profiles, item metadata, interaction types) could improve the model's representational capacity, improving accuracy in complex multi-modal datasets. Developing automated hyperparameter optimization techniques would allow for adaptive and task-specific tuning of fusion and decoder parameters, improving generalization across diverse datasets. Finally, improving the interpretability and explainability of predictions by quantifying the contribution of centrality measures and embeddings could make the model more transparent and applicable in critical domains such as healthcare and finance. These future enhancements will advance the scalability, adaptability, and practical relevance of LP-GNN for link prediction in increasingly complex network scenarios.

REFERENCES

[1] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

[2] M. E. J. Newman, "The structure of scientific collaboration networks," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 2, pp. 404–409, Jan. 2001.
[3] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proc. 12th Int. Conf. Inf. Knowl. Manage.*, Nov. 2003, pp. 556–559.
[4] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Netw.*, vol. 1, no. 3, pp. 215–239, Jan. 1978.
[5] M. E. Newman, "Detecting community structure in networks," *Eur. Phys. J. B*, vol. 38, no. 2, pp. 321–330, 2004.
[6] A. Kumar, S. S. Singh, K. Singh, and B. Biswas, "Link prediction techniques, applications, and performance: A survey," *Phys. A, Stat. Mech. Appl.*, vol. 553, Sep. 2020, Art. no. 124289.
[7] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 2, Jul. 2005, pp. 729–734.
[8] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[9] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
[10] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 1024–1035.
[11] M. E. Shaw, "Group structure and the behavior of individuals in small groups," *J. Psychol.*, vol. 38, no. 1, pp. 139–149, Jul. 1954.
[12] J. Nieminen, "On the centrality in a graph," *Scandin. J. Psychol.*, vol. 15, no. 1, pp. 332–336, 1974.
[13] A. Saxena and S. Iyengar, "Centrality measures in complex networks: A survey," 2020, *arXiv:2011.07190*.
[14] M. A. Beauchamp, "An improved index of centrality," *Behav. Sci.*, vol. 10, no. 2, pp. 161–163, 1965.
[15] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, no. 4, pp. 581–603, Dec. 1966.
[16] M. S. Granovetter, "The strength of weak ties," *Amer. J. Sociol.*, vol. 78, no. 6, pp. 1360–1380, 1973.
[17] Y. Nandini, T. J. Lakshmi, and M. K. Enduri, "Link prediction in complex networks: An empirical review," in *Proc. Int. Conf. Frontiers Intell. Comput., Theory Appl.* Cham, Switzerland: Springer, 2023, pp. 57–67.
[18] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 64, no. 2, Jul. 2001, Art. no. 025102.
[19] P. Jaccard, "Distribution de la flore Alpine dans le Bassin des dranses et dans quelques régions voisines," *Bulletins Société Vaudoise des Sci. Naturelles*, vol. 37, pp. 241–272, Jan. 1901.
[20] A. L. Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations," *Phys. A, Stat. Mech. Appl.*, vol. 311, nos. 3–4, pp. 590–614, Aug. 2002.
[21] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *Eur. Phys. J. B*, vol. 71, no. 4, pp. 623–630, Oct. 2009.
[22] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Netw.*, vol. 25, no. 3, pp. 211–230, Jul. 2003.
[23] Q. Ou, Y.-D. Jin, T. Zhou, B.-H. Wang, and B.-Q. Yin, "Power-law strength-degree correlation from resource-allocation dynamics on weighted networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 75, no. 2, Feb. 2007, Art. no. 021102.
[24] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" 2018, *arXiv:1810.00826*.
[25] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.
[26] Y. V. Nandini, T. J. Lakshmi, M. K. Enduri, and H. Sharma, "Link prediction in complex networks using average centrality-based similarity score," *Entropy*, vol. 26, no. 6, p. 433, May 2024.
[27] H. Lu and S. Uddin, "A parameterised model for link prediction using node centrality and similarity measure based on graph embedding," *Neurocomputing*, vol. 593, Aug. 2024, Art. no. 127820.
[28] J. You, R. Ying, and J. Leskovec, "Position-aware graph neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7134–7143.
[29] Z. Su, Z. Hu, and Y. Li, "Hierarchical graph representation learning with local capsule pooling," in *Proc. ACM Multimedia Asia*, Dec. 2021, pp. 1–7.
[30] D. Wu, H. Wu, and J. Li, "Hierarchy-aware adaptive graph neural network," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 1, pp. 365–378, Jan. 2025.
[31] Z. Zhang, J. Cai, Y. Zhang, and J. Wang, "Learning hierarchy-aware knowledge graph embeddings for link prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 3, pp. 3065–3072.
[32] B. P. Chamberlain, S. Shirobokov, E. Rossi, F. Frasca, T. Markovich, N. Hammerla, M. M. Bronstein, and M. Hansmire, "Graph neural networks for link prediction with subgraph sketching," 2022, *arXiv:2209.15486*.
[33] Y. Wang, X. Hu, Q. Gan, X. Huang, X. Qiu, and D. Wipf, "Efficient link prediction via GNN layers induced by negative sampling," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 1, pp. 253–264, Jan. 2025.
[34] X. Liu, X. Li, G. Fiumara, and P. De Meo, "Link prediction approach combined graph neural network with capsule network," *Expert Syst. Appl.*, vol. 212, Feb. 2023, Art. no. 118737.
[35] S. Fang, L. Li, X. Chen, Q. Zhang, H. Chen, L. Yu, and J. Liu, "Link prediction algorithm based on dual-channel graph neural network," in *Proc. 10th Int. Conf. Big Data Inf. Anal. (BigDIA)*, Oct. 2024, pp. 887–894.
[36] X. Xian, T. Wu, X. Ma, S. Qiao, Y. Shao, C. Wang, L. Yuan, and Y. Wu, "Generative graph neural networks for link prediction," 2022, *arXiv:2301.00169*.
[37] T. J. Lakshmi and S. D. Bhavani, "Link prediction measures in various types of information networks: A review," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2018, pp. 1160–1167.
[38] Y. Nandini, T. Jaya Lakshmi, and M. K. Enduri, "Link prediction based on node centrality measure," in *Proc. Int. Conf. Frontiers Intell. Comput., Theory Appl.* Cham, Switzerland: Springer, 2024, pp. 63–74.
[39] Y. Tai, H. Yang, H. He, X. Wu, and W. Zhang, "A representation learning link prediction approach using line graph neural networks," in *Proc. Chin. Conf. Pattern Recognit. Comput. Vis. (PRCV)*. Cham, Switzerland: Springer, 2023, pp. 195–207.
[40] M. Kamrul Islam, S. Aridhi, and M. Smail-Tabbone, "A comparative study of similarity-based and GNN-based link prediction approaches," 2020, *arXiv:2008.08879*.
[41] T. Liu and F. Liu, "Graph neural networks for evaluating the reliability and resilience of infrastructure systems: A systematic review of models, applications, and future directions," *IEEE Access*, vol. 13, pp. 164883–164904, 2025.
[42] M. He, P. Zhu, Y. Liu, and K. Tang, "PLGNN: Graph neural networks via adaptive feature perturbation and high-way links," *Complex Intell. Syst.*, vol. 11, no. 7, p. 288, Jul. 2025.
[43] P. Zhu, Z. Pan, K. Tang, X. Cui, J. Wang, and Q. Xuan, "Node injection attack based on label propagation against graph neural network," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 5, pp. 5858–5870, Oct. 2024.
[44] L. Cheng, P. Zhu, K. Tang, C. Gao, and Z. Wang, "GIN-SD: Source detection in graphs with incomplete nodes via positional encoding and attentive fusion," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, no. 1, pp. 55–63.
[45] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proc. AAAI*, 2015, pp. 4292–4293. [Online]. Available: https://networkrepository.com
[46] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016, *arXiv:1611.07308*.
[47] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4116–4126.
[48] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 233–240.
[49] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data Mining and Knowledge Discovery Handbook*. Boston, MA, USA: Springer, 2010, pp. 853–867.

MADHUSUDHANA RAO BASWANI received the M.C.A. degree from Andhra University, Visakhapatnam, Andhra Pradesh, India, in 2010. He is currently pursuing the Ph.D. degree in computer science and engineering with SRM University-AP, Amaravati, India. His research interests encompass recommender systems, graph analytics, link prediction, and big data analytics.

KHYATHISREE YARRA is currently pursuing the bachelor's degree in computer science and engineering with a specialization in artificial intelligence and machine learning. She has a strong foundation in algorithms, data structures, and object-oriented programming concepts. A passionate coder and aspiring researcher, she is keen to expand her technical, analytical, and problem-solving skills, with a particular interest in algorithmic challenges, machine learning, and artificial intelligence. She is currently engaged in research exploring advanced computational models and their applications.

PRASANTHI BOYAPATI received the B.Tech. and M.Tech. degrees in computer science and engineering from Jawaharlal Nehru Technological University, India, and the Ph.D. degree in image processing from Acharya Nagarjuna University, India, in 2019. She is currently an Assistant Professor with the Department of Computer Science and Engineering, SRM University, AP, India. She has more than 17 years of teaching and research experience. Her research interests include medical image processing, machine learning, deep learning, and big data analytics.

ESWARAIAH RAYACHOTI received the Bachelor of Technology degree from Sri Venkateswara University, Andhra Pradesh, the Master of Technology degree from JNTUH, Hyderabad, and the Ph.D. degree in computer science and engineering from Acharya Nagarjuna University, Andhra Pradesh. He worked in various engineering colleges in Andhra Pradesh. He is currently a Professor with the School of Computer Science and Engineering, VIT-AP University. His research interests include digital image processing and computer vision.

T. JAYA LAKSHMI (Member, IEEE) received the Ph.D. degree from the School of Computer and Information Sciences, University of Hyderabad, India, in 2019, focusing on "LP in Heterogeneous Social Networks." She is currently a Lecturer with the Department of Computing, Sheffield Hallam University, Sheffield, U.K. Her research interests include graph mining, recommender systems, natural language processing, and security analytics. She is an active reviewer for prominent international journals and brings over 22 years of teaching experience.