International Journal of Mathematical, Engineering and Management Sciences
Vol. 11, No. 2, 634-653, 2026
https://doi.org/10.33889/IJMEMS.2026.11.2.026

Adaptive Structural Similarity Guided GCN–GAT Framework for Robust Link Prediction

Shambhu Kumar
Department of Computer Science & Engineering and Information Technology,
Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India.
E-mail: kumar.shambhu@gmail.com

Dinesh C. S. Bisht
Department of Mathematics,
Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India.
E-mail: drbisht.math@gmail.com

Arti Jain
Department of Computer Science & Engineering and Information Technology,
Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India.
Corresponding author: ajain.jiit@gmail.com

(Received on December 27, 2025; Revised on January 23, 2026; Accepted on February 2, 2026)

Abstract
Social Network Analysis (SNA) has experienced major advancements with the growth of technology, which has made link prediction an important aspect of SNA, particularly regarding social network applications, recommendation systems, and various biological networks. Traditional Graph Neural Networks (GNN), such as Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) which primarily focus on aggregating node features and often neglect structural similarity (SS) between node pairs. This paper proposes a Structural Similarity-Infused Multi-Head (MH) Graph Attention Network (SS-MH-GAT) for link prediction that integrates graph-topology-based features into a hybrid GCN-GAT model. SS-MH-GAT introduces a structural-similarity-weight-extraction module, which learns the relative importance of the adjacency, Jaccard Index, clustering coefficient, and path-distance similarity metrics at the graph level. These globally weighted structural features are fused to form a similarity-aware graph representation, which is further refined through a Graph Convolutional Network. This is followed by the use of a Multi-Head Graph Attention Network to impart a locality-focused adaptability to the network by assigning importance weights to neighboring nodes, culminating in a DenseNet Classifier to infer probabilities of links based on the enhanced feature representations. Experiments on five diverse real-world datasets demonstrate that SS-MH-GAT surpasses baseline models. There is a significant performance improvement in accuracy (+12%), precision (+14%), recall (+17%), and AUC (+13%), which validates the effectiveness of the proposed approach.

Keywords- Graph attention layer, Graph convolution layer, Graph neural network, Link prediction, Similarity metrics.

1. Introduction

Social network analysis (SNA) (Daud et al., 2020) has come to prominence in the last few years, mainly due to the rapid expansion of online social networks, the development of recommendation systems (Berkani, 2021), and advances in biological networks (Nasiri et al., 2021). One of the major problems in social network analysis is link prediction, which attempts to identify impending future relationships between nodes (people, organizations, etc.) based on the relationships that already exist within a particular network.

Traditional link prediction approaches leverage topology-based features (Kumar et al., 2025; Liben-Nowell & Kleinberg, 2007) that provide valuable insights into network structure, categorized into local and global structure metrics. Local features (e.g., Jaccard Index, Clustering Coefficient) take care of the immediate node connectivity and its connectivity patterns, which are very effective in dense subgraphs. Global topology-based features (e.g., Path Distance) measure long-range dependencies within and across a network and identify similarities among nodes that may not be directly connected to one another but are still important to the network structure (Liben-Nowell & Kleinberg, 2007). In networks with low density and few connections, global topology-based features play an important role.

Though the approaches based on topology have their own benefits, these approaches often lack adaptability and struggle with complex network structures. To overcome these limitations, GNN (Kumar et al., 2024; Liu et al., 2023) have emerged as powerful tools for learning node representations by propagating features across a graph. More specifically, Graph Convolutional Networks (GCN) (Kipf & Welling, 2017) and GAT (Veličković et al., 2017) are adaptive in nature and can be scaled effectively for link prediction by learning representations from both node attributes and the structural connectivity of all nodes in a given graph.

The state-of-the-art GNN models for link prediction are mainly based on feature-propagation methods, where a node's embedding is learned from its local neighborhood. In particular, existing methods may neglect global and local structure information explicitly. Specifically, they may overlook the following:

- ***Lack of structural awareness –*** It is observed that most existing GNN models only consider aggregated node features, overlooking graph-structure-based similarities such as the Jaccard Index, Clustering Coefficient, and Path Distance, which could offer more insights into the graph structure.
- ***Limited expressiveness in adjacency matrices –*** The conventional adjacency matrix in GCNs is either binary or weighted by edge strength, and so it does not capture structural similarities that can be useful in training a model.
- ***Suboptimal feature integration –*** Existing models do not effectively combine graph heuristics with deep learning techniques, which leads to less informative node embeddings.
- ***Inconsistent performance in sparse graphs –*** Most GNN-based link predictors struggle with sparse networks, where structural similarity metrics could provide more robust predictive power.

Although traditional topology-heuristics like Jaccard Index, Clustering Coefficient, or Path Distance (PD) are still used in link prediction tasks, they often lack adaptability and robustness across complex topological structures. With this intention, GNNs have received increasing attention as strong candidates for learning node representations. Moreover, GCN and GAT appear promising for improving link prediction performance by incorporating both node features and structural relationships.

Despite the strong performance of recent GNN-based link prediction models, most existing approaches primarily rely on feature propagation or attention-based neighborhood aggregation, while largely neglecting the explicit modeling of structural similarity patterns that govern link formation in real-world networks. In particular, structural heuristics such as neighborhood overlap, community cohesion, and long-range connectivity are either ignored or incorporated in a static, non-adaptive manner. To address this limitation, this work proposes a novel structural similarity–guided hybrid framework that uniquely integrates global adaptive weighting of multiple structural similarity metrics with local multi-head graph attention refinement. Unlike existing GCN–GAT models that focus solely on node-level feature aggregation, the proposed approach first learns dataset-level structural importance weights for adjacency, Jaccard Index, clustering coefficient, and path distance, and then combines them with edge-level attention mechanisms to capture fine-grained relational dependencies. This dual-level integration enables the approach to learn more expressive and robust node representations, particularly in sparse, structurally heterogeneous networks.

The paper is organized as follows: Section 2 outlines the literature review, identifies the research gap, and presents the contribution of the current research. Section 3 outlines a proposed methodology. Section 4 explains the experimental design, while Section 5 interprets the results. Finally, Section 6 concludes this research with future directions.

2. Literature Review

GNN are widely used for learning on graph-structured data, particularly for link prediction tasks. Graph Convolutional Networks (GCNs) (Kipf & Welling, 2017), which aggregate information from a node's local neighborhood to obtain new, expressive node representations. However, the downside of GCN is that it treats every node in its local neighborhood equally. Another downside of GCN is its inability to dynamically adjust node importance. The features of nodes from different neighbors of a node are aggregated uniformly within one hop. Therefore, it might not perform well in heterogeneous graphs where different nodes have different levels of importance.

GAT (Veličković et al., 2017) demonstrate the use of adaptive weights for neighboring nodes through self-attention. Unlike GCN, GAT applies a multi-head self-attention mechanism to learn dynamic weights for node features, thereby enhancing their ability to represent complex graph structures. GAT has a limitation when dealing with graphs with long dependencies, as it applies attention only to immediate neighbors. As a result, GAT may perform sub-optimally on tasks that require global structural information.

GraphSAGE, (Hamilton et al., 2017), provides an inductive learning framework in which node embeddings are created using sampling and aggregating the features of a node's neighbors. Its learnable aggregation functions, such as mean, LSTM, and pooling, improve accuracy by capturing diverse structural information. However, the sampling process may lead to information loss, reducing accuracy in highly connected graphs.

Despite these improvements, the majority of GNN based systems do not explicitly model topological similarities, which are fundamental to link creation mechanisms. Classical topology-based heuristics (Liben-Nowell & Kleinberg, 2007) have widely been used for link prediction, which rely on similarity measures such as Common Neighbors (CN), Jaccard Index (JI), and Adamic-Adar (AA) to estimate connection probabilities.

Community structure has been explored to predict connections between peers, whereas traditional models rely solely on similarity (Muscoloni et al., 2018). By using the strength of connections between peers through their common neighbors, it assumes that the stronger the connection amongst these peers, the higher the probability of the two connecting. However, this method lacks accuracy within networks with weak clustering because of its heavy reliance upon strong local connectivity, which may lead to inaccurate predictions in sparse networks and in the very dynamic environment of most social networks.

The Jaccard Index (Jaccard, 1901) and Katz Index (Foster et al., 2001) have been developed to support link predictions by utilizing local and global topology-based approaches, respectively. The Jaccard index calculates importance using the number of common neighbours two nodes share, which allows for efficient usage but has low performance rates when used with sparse graphs. On the other hand, the Katz index calculates the importance using an aggregated calculation of the number of paths between the nodes, capturing both short-term and long-term cycles but is costly due to the complexity of the calculations required to compute all possible paths between the nodes. Because of this, while Jaccard provides good predictions on highly clustered graphs, the Katz index has difficulties providing accurate results on highly connected networks.

A similarity-based parameterized model has been illustrated for link prediction, leveraging three topological features to get a similarity score between nodes and predict the potential links (Rai et al., 2023). These topological features are equally weighted in this approach, leading to low accuracy in sparse networks.

Random walk-based approaches have recently proven to be useful techniques for learning lower-dimensional node representations for graphs. Node sequences generated via truncated random walks are modeled using the Skip-Gram framework to obtain node representations (Perozzi et al., 2014). This model is capable of learning local relationships but fails to capture various linkages of nodes.

A biased random walk strategy, known as Node2Vec, is introduced to improve DeepWalk by balancing the trade-offs between DFS and BFS (Grover & Leskovec, 2016). This strategy allows it to capture homophily and structural equivalence more effectively. However, Node2Vec still relies on manually tuned walk-strategy parameters, which can affect accuracy.

Structural awareness is further strengthened by removing direct dependence on node features and constructing a hierarchical similarity graph based on node structural identities prior to performing random walks (Ribeiro et al., 2017). With that, Struc2Vec is able to learn structural roles from distant nodes, making it more powerful in tasks that demand role-based embeddings. In contrast, this approach has higher computational overheads compared with DeepWalk and Node2Vec, which restricts scalability when dealing with large graphs.

Although GNN reinforcements such as GCN, GAT, and GraphSAGE perform well in terms of prediction accuracy, these algorithms neglect structural-similarity-based heuristics, which are necessary for link-creation simulation. On the other hand, traditional similarity-based methods such as the Jaccard Index, Katz, and Cannistraci-Hebb provide valuable relational insights but cannot learn adaptive node representations.

The random-walk-based embedding methods (DeepWalk, Node2Vec, and Struc2Vec) represent an advancement over traditional heuristics by capturing structural role. However, the downside of this technique is that it requires manual parameter tuning, and its computational overhead limits its effectiveness on large-scale graphs.

To address these challenges, this study proposes SS-MH-GAT, a Structural Similarity-Infused Multi-Head (MH) Graph Attention Network model (SS-MH-GAT) that combines global structural similarity fusion with local attention-based embedding refinement. The key contributions of the proposed approach are summarized below:

- Develop a structural similarity weight extraction module that learn global weights $(\alpha, \beta, \gamma, \delta)$, for similarity metrics such as adjacency, Jaccard Index, Clustering Coefficient, and Path Distance. These weights reflect the overall structural importance of each feature in the graph and are normalized so that $\alpha + \beta + \gamma + \delta = 1$.
- Developing enhanced representations of node embeddings through the use of structural similarity-based features in GCN.

- Multi-Head Graph Attention is employed to assign edge-level importance scores to neighboring nodes, allowing the model to capture fine-grained relational patterns and complex structural dependencies that are not captured by global weighting alone.
- Evaluating the effectiveness of SS-MH-GAT as an improvement over other baseline GNN models (GCN, GAT, SAGE) as well as traditional topological methods (Jaccard, Katz, CND) by testing for improvements in accuracy, robustness on sparse graphs, and generalizability across multiple datasets.

These four structural features (adjacency, Jaccard, clustering coefficient, and distance) provide immediate connectivity, neighborhood overlap, community cohesion, and long-range relational patterns, respectively. Whereas the combination of these four features offers a more robust and nuanced understanding of the relationships between nodes than any single feature would.

3. Proposed Methodology

This section presents SS-MH-GAT, a hybrid graph learning framework that integrates different structural similarities of a graph with local attention-based embedding for link prediction. The model consists of four components: (i) structural feature computation, (ii) structural similarity weight extraction module, (iii) GCN-based embedding refinement, and (iv) Multi-Head Graph Attention with DenseNet-based link prediction. The overall workflow is shown in **Figure 1**.

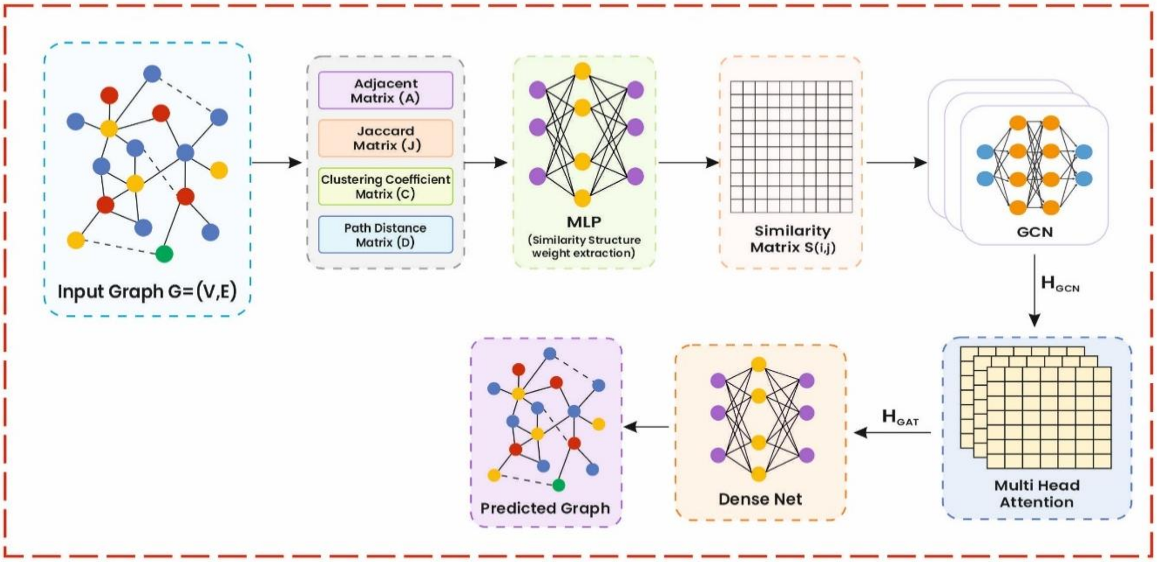

Figure 1. Illustration of the overall framework of the proposed framework SS-MH-GAT.

3.1 Structural Feature Computation

Let $G = (V, E)$ be an undirected graph with a node set V and edge set E. This study begins by constructing four structural similarity metrics that encode different topological properties:

a) Adjacency Matrix (A)

To ensure that node features contribute effectively to the embedding, self-loops are added to the adjacency matrix (A) as in Equation (1). Furthermore, to balance feature aggregation and prevent high-degree nodes from dominating, this study normalizes the $A^{\wedge}$ using Equation (2).

$$A^{\wedge} = A + I \tag{1}$$

$$A^{\sim} = D^{\frac{-1}{2}}(A^{\wedge})D^{\frac{-1}{2}} \tag{2}$$

Here,

- A is the adjacency matrix of the graph.
- I is the identity matrix ensuring self-loops.
- $A^{\wedge}$ is the modified adjacency matrix with self-loops.
- D is the degree matrix, where D_{ii} represents the degree of the node i.
- This normalization balances feature aggregation and prevent high-degree dominance.

b) Jaccard Index (*J*)

It measures the proportion of common neighbors between node pairs (i, j) as in Equation (3). It enhances local similarity for link prediction.

$$J(i,j) = \frac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i) \cup \Gamma(j)|} \tag{3}$$

Here,

- $\Gamma(i)$ is the set of neighbors of node i.
- $J(i,j)$ represents the similarity between nodes *i* and *j* based on shared neighbors.

c) Clustering Coefficient (*C*)

It captures community structure by quantifying how well a node's neighbors are interconnected, as written in Equation (4).

$$C(i) = \frac{2T_i}{d(i)(d(i)-1)} \tag{4}$$

Here,

- T_i: Number of triangles that include the node i.
- $d(i)$: Degree of node i (i.e., the number of direct connections).
- Averaged for node pairs (i, j):

$$C_{ij} = \begin{cases} \frac{c(i)+c(j)}{2}, & for\ each\ (i,j) \in V \\ 0, & Otherwise \end{cases} \tag{5}$$

d) Path Distance (*D*)

It represents the shortest path distance between node pairs (i, j) using Dijkstra's algorithm, normalized for consistency.

$$D(i,j) = \frac{1}{1+P(i,j)} \tag{6}$$

Here,

- $P(i,j)$ is the shortest path length between nodes *i* and *j* (computed using Dijkstra's algorithm).
- If $P(i,j) > 3, then\ P(i,j) = \infty$, ensuring distant nodes do not contribute to similarity calculations.
- The shortest path distance is capped at three hops to restrict the influence of distant node pairs that typically provide weak structural signals in link prediction tasks. This choice reflects common observations in social networks, where most meaningful interactions occur within a small number of

hops. Preliminary experiments with higher thresholds showed marginal performance gains at increased computational cost, motivating the use of a threshold of 3.

The selection of adjacency, Jaccard Index, clustering coefficient, and path distance is motivated by their complementary ability to capture distinct structural characteristics of a graph. Specifically, these metrics jointly represent direct node connectivity, neighborhood overlap, local community cohesiveness, and long-range relational dependencies, respectively. This combination enables a balanced and computationally efficient encoding of both local and global structural information for link prediction.

3.2 Structural Similarity Weight Extraction Module

This study introduces a global adaptive weighting module that learns and assigns heuristic weights to these four structural similarity metrics as in Equation (7). An MLP learns four graph-level coefficients.

$$W = [\alpha, \beta, \gamma, \delta] = Softmax(MLP([A^{\sim}, J, C, D])) \qquad (7)$$

This Softmax ensures the normalization of the coefficient as in Equation (8):

$$\alpha + \beta + \gamma + \delta = 1 \qquad (8)$$

These weights represent the importance of global structural similarities, optimized from data. It does not vary across edges or nodes, but provides dataset-level adaptiveness. The final fused similarity matrix is in Equation (9):

$$S = \alpha A^{\sim} + \beta J + \gamma C + \delta D \qquad (9)$$

Here, the learnable coefficients $\alpha, \beta, \gamma, \delta$ represent the global importance weights assigned to adjacency, Jaccard Index, clustering coefficient, and path distance, respectively. The Softmax normalization ensures that all weights are non-negative and sum to one, enabling a stable and interpretable contribution of each structural metric. The resulting fused similarity matrix S therefore captures dataset-level structural preferences and serves as a structure-aware input for subsequent GCN propagation.

This similarity matrix serves as a structure-aware input to the GCN.

It is important to note that the proposed structural similarity weight extraction module learns global (dataset-level) importance weights rather than node- or edge-specific weights. This design choice is intentional, as the objective is to capture a graph's overall structural tendencies while avoiding excessive model complexity and overfitting. Local and edge-level adaptiveness is subsequently handled by the multi-head graph attention mechanism, which dynamically assigns importance to neighboring nodes during message passing.

The learned structural similarity weights are dataset-specific and remain fixed during inference once training is complete. During training, the MLP-based weight extraction module is optimized jointly with the GCN and multi-head GAT components in an end-to-end manner using the same binary cross-entropy loss, enabling the weights to adapt to the structural characteristics of each dataset.

3.3 GCN-Based Structural Embedding Refinement

The fused similarity matrix S (Equation (9)) guides GCN propagation, enabling the encoder to incorporate richer topological information beyond raw adjacency. The GCN operation is defined as in Equations (10) and (11).

$$H^{(l+1)} = \sigma\left(SH^{(l)}W^{l}\right) \qquad (10)$$

$$H^0 \ = X \tag{11}$$

Here, $H^{(l)} \in \ R^{|V|\times d_l}$ is the node-embedding matrix at layer l. The final GCN embedding output H_{GCN} as in Equation (12) after L layers of convolution provides globally structure-aware node embeddings for subsequent attention refinement.

$$H_{GCN} = H^l \tag{12}$$

3.4 Multi-Head Graph Attention for Local Adaptiveness

While the structural similarity extraction module captures overall structural preferences, node relationships still vary significantly at the local level. To model these fine-grained dependencies, this study applies a Multi-Head Graph Attention Network (MH-GAT). The unnormalized attention score for an edge (i, j) under attention head k is computed as in Equation (13):

$$e_{ij}^k = LeakyreLU\left(a^{k^T}\left[W^k h_i || W^k h_j\right]\right) \tag{13}$$

The normalized attention coefficients are in Equation (14):

$$\alpha_{ij}^k = \frac{\exp\,(e_{ij}^k)}{\sum_{u\in N(i)} exp(e_{iu}^k)} \tag{14}$$

The node-level embedding update (Message Aggregation) for head *k* is given by Equation (15):

$$h_i^k = \sigma\left(\sum_{j\in N(i)} \alpha_{ij}^k \, W^k h_j\right) \tag{15}$$

In this formulation, the attention mechanism computes edge-level importance scores by jointly considering the transformed features of connected node pairs. The normalized attention coefficients quantify the relative contribution of each neighbor during message aggregation, allowing the model to emphasize structurally and contextually relevant neighbors while suppressing less informative ones.

Let $H_k \in R^{|V|\times d_k}$ denote the matrix for head k, formed by stacking the embeddings of all nodes for the head k as in Equation (16).

$$H^k = \begin{bmatrix} \left(h_1^k\right)^T \\ \left(h_2^k\right)^T \\ \left(h_3^k\right)^T \\ \vdots \\ \left(h_{|V|}^k\right)^T \end{bmatrix} \tag{16}$$

The outputs of all heads (K) are concatenated as in Equation (17) to yield the final attention-refined embedding:

$$H_{GAT} = \|_{k=1}^{K} \ H^k \tag{17}$$

MH-GAT enables edge-level adaptability, complementing the structural-similarity weighting learned earlier. The output H_{GAT}, a refined embedding that captures both global structural information and localized neighborhood importance.

3.5 Dense Net-Based Link Prediction

DenseNet is employed instead of a simple MLP to enhance feature reuse and improve gradient flow during training. The dense connectivity enables the classifier to effectively exploit complementary information from both nodes in a pair, leading to more stable optimization and improved link prediction performance compared to shallow fully connected networks.

For a candidate node pair (i, j), this study constructs the input vector as in Equation (18)

$$z_{ij} = \ [h_i' \,||\, h_j'] \tag{18}$$

The concatenated vector is passed through multiple dense layers with non-linear activation functions as expressed in Equation (19).

$$H_{ij}^{(l+1)} = \sigma\left(W^{(l)} h_{ij}^{(l)} + b^{(l)} \right) \tag{19}$$

The output layer applies a sigmoid function to generate a probability score (0 to 1) as mentioned in Equation (20).

$$\hat{y}_{ij} = \sigma\left(W_{out} h_{ij}^{(L)} + b_{out} \right) \tag{20}$$

3.6 Training Using Binary Cross-Entropy Loss

The entire SS-MH-GAT architecture is optimized using a *single* binary cross-entropy loss (Equation (21)).

$$\mathcal{L} = \ -\sum_{(i,j)\in E} y_{ij} \log\left(\hat{y}_{ij}\ \right) + \left(1 - y_{ij}\right) \log(1 - \hat{y}_{ij}\ ) \tag{21}$$

The trained dense network produces a link-probability score, thereby enhancing link-prediction accuracy in SS-MH-GAT. **Table 1** lists the variables used in sections [3.3-3.6].

Table 1. List of Variables and description used in sections [3.3 to 3.6].

Variables	Description
X	Initial node feature matrix (identity features if unavailable)
$H^{(l)}$	Node embeddings at GCN layer l
$W^{(l)}$	Trainable weight matrix at GCN layer l
H_{GCN}	Output node embeddings from the last GCN layer
$W^{(k)}$	Trainable weight matrix for attention head k
$a^{(k)}$	Learnable attention vector for head k
$e_{ij}^{(k)}$	Raw (unnormalized) attention score between nodes i and j for head k
$\alpha_{ij}^{(k)}$	Normalized attention coefficient (edge-level importance)
h_i'	MH-GAT updated embedding for node i
H_{GAT}	Final embedding matrix from MH-GAT module
z_{ij}	Concatenated node-pair embedding for prediction
$W^{(l)}, b^{(l)}$	Weight and bias of DenseNet layer l
$z^{(l)}$	Hidden representation at DenseNet layer l
$\hat{y}_{ij}$	Predicted probability of a link between nodes i and j
y_{ij}	Ground-truth label (1 = link exists, 0 = no link)
$\mathcal{L}$	Binary cross-entropy loss used for end-to-end training
σ	Non-linear Activation function

4. Experimental Setup

This research work is implemented in the PyTorch environment. The system's configuration consists of an NVIDIA GeForce GTX 1050 4GB GPU, with an Intel® Core(TM) i9-9900 CPU@3.10 GHz.

4.1 Datasets

In this research, five different social network datasets of varying sizes were used. A brief description of these datasets is listed in **Table 2**, and dataset characteristics are summarized in **Table 3**:

Table 2. Brief description of datasets.

Datasets	Description	Web links
Anybeat (Rossi & Ahmed, 2015)	Anybeat is a digital community that serves as a public space for connecting with individuals in the local area or around the world.	https://networkrepository.com/soc-anybeat.php
Email (Rossi & Ahmed, 2015)	This dataset is the email exchange between students of a university.	https://networkrepository.com/email-univ.php
FB-Page-politician (Rossi & Ahmed, 2015)	The data was collected from Facebook pages in November 2017. The set of data exhibits blue verified Facebook page networks classified in various categories. The nodes in the data represent the Facebook pages, and the edges show the like relationship of the Facebook pages to one another.	https://networkrepository.com/fb-pages-politician.php
Fb-pages-ports (Rossi & Ahmed, 2015)	Data was gathered from Facebook pages in November 2017. This dataset encompasses networks of blue-verified pages categorized into various groups. The nodes symbolize the pages, while the edges depict the likes they share.	https://networkrepository.com/fb-pages-sport.php
Hamsterster (Rossi & Ahmed, 2015)	The network is undirected and depicts family connections between users of the website.	https://networkrepository.com/soc-hamsterster.php

Table 3. Dataset characteristics.

Datasets↓	Nodes	Edges	Clustering coefficient	Average degree	Density
Anybeat	12600	67100	0.227	10	0.0008
Email	1100	5500	0.22	9	0.0085
Fb-page-politician	5900	41700	0.385	14	0.002
Fb-page-sports	13900	86800	0.27	12	0.0009
Hamsterster	2400	16600	0.53	13	0.0056

4.2 Baseline Models

Table 4. Baseline model description model.

Sr. No.	Approach	Model	Description
1.	Local Topology (Similarity Based)	Common Neighbors (Lü et al., 2009)	Counts how many common neighbors exist between two nodes, assuming that a more significant overlap indicates a more substantial likelihood of forming a link.
2.		Cannistraci-Hebb Index (Muscoloni et al., 2018)	The common neighbor's approach is enhanced by incorporating topological regularities and Hebbian learning principles for improved link prediction.
3.	Global Topology (Similarity Based)	Katz Index (Foster et al., 2001)	Measures node similarity by considering all possible paths, giving higher weights to shorter paths.
4.	Local And Global Topology (Similarity Based)	CND (Ahmad et al., 2020)	Integration of common neighbor and node distance as similarity metrics.
5.	Random Walk Based	DW (DeepWalk) (Perozzi et al., 2014)	A random walk-based method that generates node embeddings by treating graph nodes as words in a corpus, using Skip-Gram.
6.		Node2Vec (Grover & Leskovec, 2016)	A variation of DeepWalk employs a biased random-walk approach to combine breadth-first and depth-first search strategies.
7.	GNN	GCN (Kipf & Welling, 2017)	Convolution-based message passing with uniform neighborhood aggregation.
8.		GAT (Veličković et al., 2017)	Attention-based propagation with learned edge-level importance.
9.		SEAL (Li et al., 2021)	Extracts link representations from labelled subgraphs through a Deep Graph Convolutional Neural Network.

The proposed model is evaluated against nine competitive baseline models, categorized into:

a) Four similarity-based methods.
b) Two random walk-based embedding techniques.
c) Three GNN-based architectures.

A brief overview of these models is provided in **Table 4**.

For fair comparison, all baseline models were evaluated using the same train–test split, negative sampling strategy, and evaluation protocol as the proposed SS-MH-GAT framework.

4.3 Evaluation Metrics

The performance of the proposed model is assessed using four key evaluation metrics, which are Accuracy, Precision, Recall, and AUC (Hanley & McNeil, 1982; Powers, 2020).

4.4 Experiment Strategy

The evaluation strategy for this research comprises multiple stages: determining similarity weights, GCN modeling, and MH-GAT integration for link prediction.

The dataset is then divided randomly into training and testing sets in the ratio of 80-20 percent. K-fold cross-validation with k = 5 is applied to the training set for better hyperparameter tuning and reduction of overfitting.

During K-fold cross-validation, the training set is divided into k subsets. In each iteration:

- (k-1) subsets are used for training.
- The remaining subset is used for validation.
- This is a process that repeats k times. This guarantees that every subset has been used for validation once.

Subsequently, the model is trained on the entire training set, applying the most effective hyperparameters derived from cross-validation. Finally, evaluate the model using the test set with respect to its ability to generalize to unseen data.

4.5 Parameter Settings

The proposed work will utilize the Grid Search Cross-Validation technique for hyperparameter optimization for all components within the SS-MH-GAT framework. The technique gives an exhaustive and deterministic exploration of the specified search space.

Although the entire architecture is trained in an end-to-end manner with a single binary cross-entropy loss function, hyperparameters of each module, such as MLP-based structural similarity weight extractor, GCN encoder, MH-GAT refinement, and DenseNet edge classifier, are distinct. Hyperparameters are optimized using a Grid Search CV method, as presented in **Tables 5** to **8** below.

Table 5 presents the optimized hyperparameter values for similarity-weight extraction.

Table 5. MLP hyperparameters of structural similarity-weight extractor.

Number of hidden layers	2
Number of neurons per layer	128
Activation function	ReLu
Optimizer	Adam
Learning rate	0.001
Dropout rate	0.2
Batch size	64
Number of epochs	100

Table 6 outlines the optimized hyperparameters used in the GCN implementation.

Table 6. GCN hyperparameters.

Number of GCN layers	3
Hidden dimension	128
Activation function	ReLu
Dropout rate	0.2
Learning rate	0.001
Weight decay	0.0001
Batch size	64

Table 7 details the hyperparameters applied in the MH-GAT model.

Table 7. GAT hyperparameters.

Number of GAT layers	3
Number of attention heads	8
Hidden dimension	128
Activation function	ELU
Dropout rate	0.2
Learning rate	0.001
Weight decay	0.0001
Batch size	64

Table 8 details the hyperparameters applied in the DenseNet classifier.

Table 7. DenseNet hyperparameters.

Number of dense layers	2
Number of neurons per layer	128
Activation function	ReLU
Dropout rate	0.3
Optimizer	Adam
Learning rate	0.001
Batch size	64
Epochs	100

5. Experimental Results

Figure 2 presents the experimental outcomes for the proposed study on five different datasets. These outcomes indicate that the model is robust for varying levels of network density. From the experimental results, it is evident that the proposed model is robust and flexible for both dense and sparse datasets. This approach performs well in most datasets, especially in denser datasets like Email and Hamsterster, where it reaches maximal values for Accuracy (0.90 & 0.87), Precision (0.89 & 0.86), Recall (0.91 & 0.89), and AUC (0.90 & 0.88). For sparse datasets like Anybeat, despite their complexity, this approach still performs well in maximizing scores for Accuracy (0.81) and Recall (0.82), implying that it can also efficiently handle less connected networks. The model's ability to strike a balance between precision and recall across diverse datasets indicates its effectiveness in both minimizing false positives and capturing meaningful connections. As per the experimentation result, the proposed link prediction model appears to be capable of producing reliable prediction results consistently across the various network density categories examined in this study.

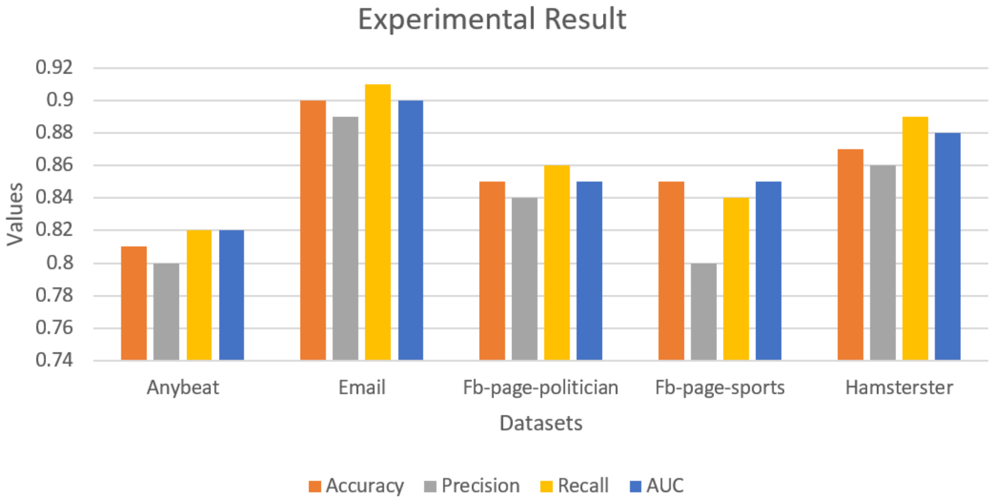

Figure 2. Performance comparison of SS-MH-GAT across five datasets in terms of accuracy, precision, recall, and AUC, demonstrating robustness over varying network densities.

5.1 Result Comparison Analysis

Table 9 compares the proposed model with other baseline models across five datasets.

- The results of the experiments indicate that the proposed SS-MH-GAT model outperforms all of the baseline methods (topological, random search-based, and other GNN-based embedding models) on all data sets. On average over the data sets, SS-MH-GAT achieves the highest levels of performance: accuracy 0.85, precision 0.83, recall 0.86, and AUC 0.86. This indicates that SS-MH-GAT captures complex graph structures better than all of the baselines.
- The baseline approaches do poorly on the sparse datasets (e.g., Anybeat and FB-page-sports), while SS-MH-GAT and the other GNN-based models have increased performance on these datasets compared to the other baselines. This shows that the use of Graph Attention Mechanisms and deep structural learning can yield significant improvement in link prediction on real-world networks.
- SS-MH-GAT continues to have strong performance on the sparse datasets such as Anybeat and outperforms the other models with an Accuracy of 0.81 and a Recall of 0.82. On the denser datasets, such as Email and Hamsterster, SS-MH-GAT continues to achieve the highest performance across all metrics, demonstrating that it can adapt to each data set's unique graph structure.
- Topology-based methods do reasonably well on denser datasets such as Email and Hamsterster, achieving accuracy values in the range of 0.71-0.79 and AUC scores between 0.66-0.80. However,

the baselines are struggling with the sparse networks such as Anybeat and FB-page-sports, whereas SS-MH-GAT has consistent improvements over all of the other topological approaches.

- Although the deep learning algorithms like GCN, GAT, and SEAL have reported a notable enhancement in comparison to heuristic and random search algorithms, the SS-MH-GAT outperforms in all evaluation criteria and proves to be a very efficient approach for the link prediction problem.

The findings suggest that incorporating deep structural learning and graph attention mechanisms is critical for achieving robust link prediction performance in real-world networks.

Table 9. Comparative experimental result.

Performance matrix	Approach	Model	Anybeat	Email	Fb-page-politician	Fb-page-sports	Hamsterster
Accuracy	Traditional approach	CN	0.70	0.79	0.72	0.71	0.74
		Katz	0.66	0.76	0.72	0.70	0.74
		CND	0.78	0.76	0.73	0.72	0.75
		CH	0.67	0.75	0.7	0.68	0.71
	Random-based approach	DW	0.63	0.78	0.75	0.74	0.75
		Node2Vec	0.73	0.8	0.78	0.76	0.77
	GNN	GCN	0.75	0.83	0.81	0.8	0.83
		GAT	0.79	0.85	0.84	0.81	0.84
		SEAL	**0.82**	0.87	**0.85**	0.84	**0.87**
	Proposed	**SS-MH-GAT**	**0.81**	**0.90**	**0.85**	**0.85**	**0.87**
Precision	Traditional approach	CN	0.55	0.63	0.60	0.58	0.60
		Katz	0.64	0.68	0.64	0.63	0.64
		CND	0.62	0.7	0.66	0.65	0.68
		CH	0.60	0.65	0.61	0.6	0.63
	Random-based approach	DW	0.65	0.73	0.71	0.69	0.71
		Node2Vec	0.68	0.76	0.74	0.72	0.73
	GNN	GCN	0.72	0.8	0.75	0.74	0.78
		GAT	0.74	0.82	0.8	0.78	0.81
		SEAL	0.75	0.83	0.81	**0.80**	0.83
	Proposed	**SS-MH-GAT**	**0.80**	**0.89**	**0.84**	**0.80**	**0.86**
Recall	Traditional approach	CN	0.58	0.66	0.63	0.61	0.63
		Katz	0.63	0.74	0.67	0.67	0.68
		CND	0.70	0.73	0.71	0.69	0.7
		CH	0.60	0.68	0.65	0.63	0.64
	Random-based approach	DW	0.69	0.76	0.74	0.72	0.73
		Node2Vec	0.72	0.84	0.80	0.80	0.82
	GNN	GCN	0.74	0.85	0.8	0.78	0.85
		GAT	0.76	0.84	0.79	0.8	0.82
		SEAL	0.77	0.89	0.83	0.81	0.84
	Proposed	**SS-MH-GAT**	**0.82**	**0.91**	**0.86**	**0.84**	**0.89**
AUC	Traditional approach	CN	0.6	0.68	0.65	0.64	0.66
		Katz	0.71	0.80	0.76	0.79	0.76
		CND	0.67	0.75	0.72	0.71	0.74
		CH	0.69	0.72	0.72	0.70	0.72
	Random-based approach	DW	0.69	0.79	0.75	0.74	0.76
		Node2Vec	0.73	0.85	0.79	0.77	0.82
	GNN	GCN	0.76	0.84	0.82	0.8	0.79
		GAT	0.78	0.86	0.84	0.82	0.84
		SEAL	0.76	0.87	0.82	0.83	0.83
	Proposed	**SS-MH-GAT**	**0.82**	**0.9**	**0.85**	**0.85**	**0.88**

5.2 Ablation Study

In an ablation study, the SS-MH-GAT framework components are tested to see the effect of component removal on link prediction accuracy. For testing, the SS-MH-GAT performance has been measured using Accuracy, Precision, Recall, and AUC over five datasets.

The following ablation settings were tested:

- Without Structural Similarity Features: Evaluate the impact of the Jaccard Index, clustering coefficient, and path distance by relying solely on GCN and GAT embeddings based on the adjacency matrix.

- Without Structural Similarity Weight Extraction Module: This module replaces the Feed-Forward MLP with the direct input of raw graph features (adjacency matrix, Jaccard index, clustering coefficient, and node distance) to GCN and MH-GAT, assuming equal weight for all metrics.

- Without Graph Attention: This assessment assesses the effectiveness of attention-based learning by removing the GAT module and relying on standard neighborhood aggregation.
  For achieving reliable and universally applicable results, the ablative experiment has been performed on all datasets (Anybeat, Email, FB-page-politician, FB-page-sports, and Hamsterster), with the average result obtained to prevent any single dataset from affecting the outcome.

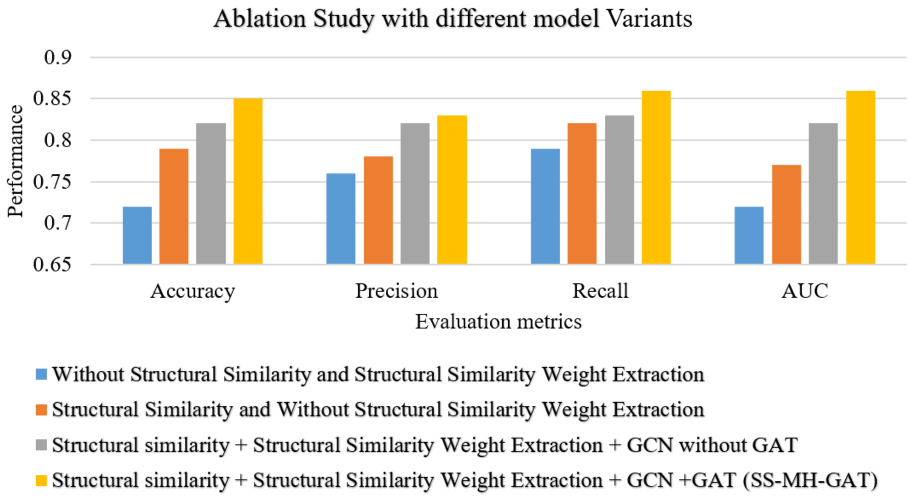

Figure 3. Ablation study illustrating the impact of removing individual components of SS-MH-GAT on link prediction performance across datasets.

Table 10. Ablation study with different model variants and corresponding result.

Model variant	Accuracy	Precision	Recall	AUC
Without structural similarity and feature weight extraction	0.83	0.8	0.81	0.85
Structural similarity but without feature weight extraction	0.85	0.82	0.84	0.87
Structural similarity + feature weight extraction + GCN without GAT	0.87	0.85	0.86	0.89
Structural similarity + feature weight extraction + GCN +GAT (SS-MH-GAT)	0.9	0.88	0.9	0.92

The results of the ablation study in **Figure 3** & **Table 10** show which part of our system is most important:

- ***Structural similarity features*:** Adding structural similarity features (Jaccard Index, Clustering Coefficient, Node Distance) improves accuracy from 0.72 to 0.79 (+9%) and AUC from 0.72 to 0.77 (+7%). This demonstrates that handcrafted topological features effectively enhance link prediction.

- ***Structural similarity weight extraction module*:** Introducing the Structural Similarity Weight Extraction Module boosts accuracy to 0.82 (+4%) and AUC to 0.82 (+6%), confirming the value of optimal feature weighting.
- ***Graph attention network* (GAT):** The final SS-MH-GAT model has the highest performance with an accuracy of 0.85 and an AUC of 0.86. It indicates the importance of the attention mechanism in modeling the dependencies between the nodes.

Among all components, the structural similarity weight extraction module contributes the most to performance improvement, as it enables adaptive fusion of multiple topological features. The inclusion of the multi-head graph attention mechanism further refines local neighborhood representations, resulting in the highest overall performance when all components are combined.

5.3 Learned Structural Similarity Weights

Figure 4 shows the contributions of different structural similarity metrics to link prediction over various datasets and the weights α, β, γ, and δ of those contributions.

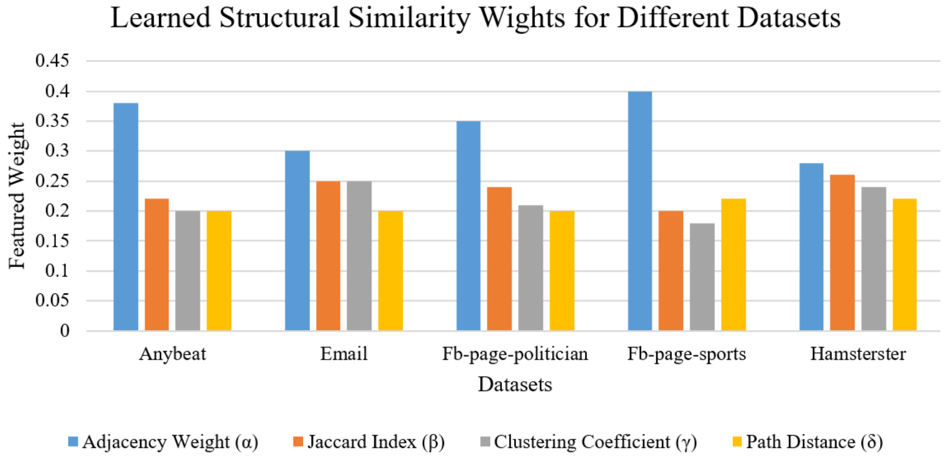

Figure 4. Learned global structural similarity weights (α, β, γ, δ) for adjacency, Jaccard Index, clustering coefficient, and path distance across different datasets.

The weights show how much each of the metrics contributed to link prediction. The dominance of the adjacency weight (α) shows a consistent and strong tendency for direct node connections to have a greater influence on determining link formation. The weights for the Jaccard Index (β) and Clustering Coefficient (γ) were similar, suggesting that neighborhood overlap and local community structure also played a significant role in Link Prediction for these datasets. Meanwhile, the path distance (δ) holds a moderate weight, suggesting that long-range dependencies provide valuable relational insights but are less significant than immediate and local connections. Further, dataset-specific variations highlight interesting trends that Email and Hamsterster networks show higher clustering coefficient (γ) values, suggesting that tightly knit communities are more indicative of link formation in these graphs. Conversely, Fb-page-sports possesses high value for adjacency weight (α) compared with others, implying that direct graph associations are most dominant compared with neighborhood associations in predicting graph associations. These variations demonstrate the need to use an adaptive approach to determine the relative importance of different structural similarities across context-dependent networks during link prediction.

5.4 Complexity Analysis

Table 11 presents the complexity of its key components to assess the computational efficiency of the proposed approach.

Table 11. Complexity analysis.

Sr. No.	Components	Complexity	Final complexity
1.	Structural similarity computation	Jaccard Index: $O(\lvert E\rvert)$ Clustering coefficient: $O(\lvert V \rvert + \lvert E \rvert)$ Node Distance: $O(\lvert V \rvert + \lvert E \rvert))$ Adjacency Information: $O(1)$	Total complexity (similarity computation): $O(\lvert E \rvert + \lvert V^2 \rvert))$
2.	Structural similarity weight extraction (feed-forward MLP)	For L layers, the total complexity for one node is : $O(L.d.h)$	Considering all nodes, the total complexity is as follows: $O(\lvert V\rvert.L.d.h)$
3.	Graph convolutional network (GCN)	GCN performs node aggregation in two ways: Feature propagation: $O(\lvert E \rvert.d)$ per layer Feature transformation (Matrix multiplication): $O(\lvert V \rvert.d^2)$	For L_{GCN} layers, Total complexity: $O(L_{GCN}.(\lvert E\rvert.d + \lvert V\rvert.d^2))$
4.	Graph attention network (GAT)	GAT computes attention scores for each edge, which increases its computational complexity compared to GCN. Attention coefficient computation: $O(H.\lvert E\rvert.d)$, where H is the number of attention heads Feature transformation: $O(\lvert V\rvert.d^2)$ Softmax and message passing: $O(H.\lvert E\rvert)$	For L_{GAT} layers, the total complexity: $O(L_{GAT}.(H.\lvert E\rvert.d + \lvert V\rvert.d^2)$
5.	Dense Net for final prediction	DenseNet processes embeddings using multiple densely connected layers. Each Dense layer: $O(d^2)$ For each L_D Layer: $O(L_D.d^2)$	It needs to process all node pairs so that the complexity would be: $O(\lvert V\rvert^2.L_D.d^2)$

Combining all components, the total worst-case complexity is written in Equation (22):

$$O(\lvert E \rvert + \lvert V^2 \rvert + \lvert V\rvert.L.d.h + L_{GCN}.(\lvert E\rvert.d + \lvert V\rvert.d^2) + L_{GAT}.(H.\lvert E\rvert.d + \lvert V\rvert.d^2) + \lvert V\rvert^2.L_D.d^2) \tag{22}$$

By looking at the complexity, it appears that the similarity computation $O(\lvert V\rvert^2)$ in a dense network is the bottleneck, but in a sparse network, it reduces to $O(\lvert E\rvert)$. The proposed work incurs higher computational overhead than traditional GNN models, but its improved accuracy (~12-15% higher AUC than the baselines) outweighs the cost.

From the complexity analysis, it can be observed that the computation of structural similarity metrics constitutes the dominant cost, particularly in dense graphs where the number of node pairs is large. However, in sparse real-world networks, this cost is significantly reduced due to limited neighborhood sizes and shorter path computations. In contrast, the GCN and multi-head GAT components scale efficiently with the number of edges and benefit from sparse matrix operations, making the overall framework computationally feasible for large-scale sparse graphs.

5.5 Training Time Analysis

Figure 5 highlights the training time of several link predictors in five datasets. Simple models like CN and CH have shorter training times because of their simple computations. GNN models like GCN, GAT, and SEAL have higher training times because of their complex network structure. The proposed SS-MH-GAT has relatively higher training times compared with GNN models, but is more effective based on all evaluation criteria. SS-MH-GAT scales efficiently with dataset size, maintaining computational feasibility for large networks. Such a trade-off in enhanced predictability and increased trainability demonstrates the efficacy of the proposed model in a practical scenario of link prediction.

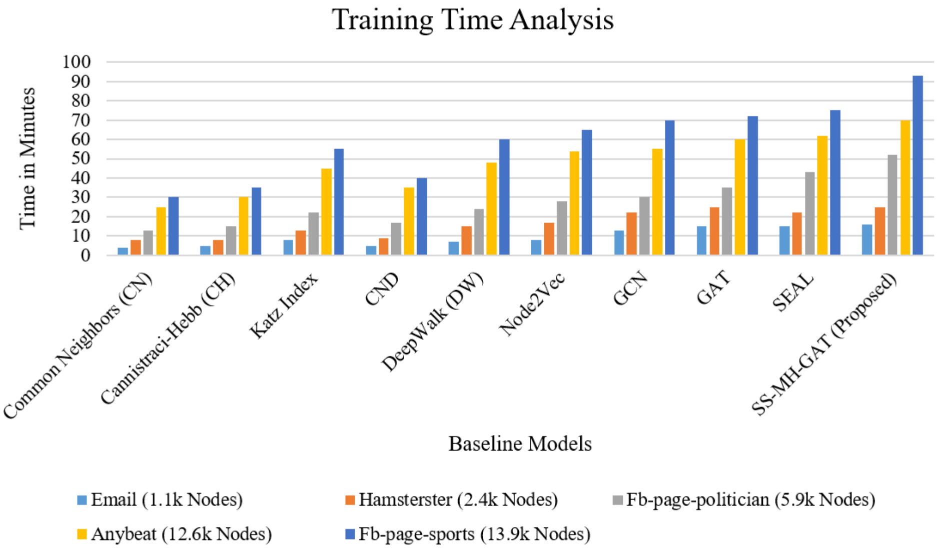

Figure 5. Training time comparison of SS-MH-GAT with baseline link prediction models across datasets, highlighting the trade-off between computational cost and predictive performance.

6. Conclusions

This study presented SS-MH-GAT, a structural-similarity enhanced hybrid GCN–GAT framework for link prediction. The proposed model combines global adaptive weighting of four key structural features, such as adjacency, Jaccard Index, clustering coefficient, and path distance, through a structural similarity weight extraction module. The resulting fused similarity matrix is then processed by a GCN to refine structure-aware embeddings, followed by a multi-head graph attention mechanism that captures local-neighbor adaptiveness. SS-MH-GAT produces expressive node representations and more reliable link predictions by integrating these complementary components within a single end-to-end learning pipeline.

Thorough experiments conducted on various real-world graphs have established the efficacy of the method in terms of improvement over state-of-the-art similarity measures as well as robust neural models like GraphSAGE and GAT. The ablation study has validated the performance contribution of the two important components of the proposed method.

Future work would be to extend the framework to handle heterogeneous, weighted, and temporal graphs, to incorporate more scalable methods for computing similarity, and to demonstrate the application of SS-MH-GAT on practical applications like recommendation, fraud detection, and biological networks.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

Acknowledgments

This research has not received any funding from any agencies in the public, commercial, or not-for-profit sectors.

AI Disclosure

The author(s) declare that no assistance is taken from generative AI to write this article.

References

Ahmad, I., Akhtar, M.U., Noor, S., & Shahnaz, A. (2020). Missing link prediction using common neighbor and centrality based parameterized algorithm. *Scientific Reports*, *10*(1), 364. https://doi.org/10.1038/s41598-019-57304-y.

Berkani, L. (2021). Recommendation of items using a social-based collaborative filtering approach and classification techniques. *International Journal of Data Mining, Modelling and Management*, *13*(1-2), 137-159. https://doi.org/10.1504/IJDMMM.2021.112919.

Daud, N.N., Ab Hamid, S.H., Saadoon, M., Sahran, F., & Anuar, N.B. (2020). Applications of link prediction in social networks: a review. *Journal of Network and Computer Applications*, *166*, 102716. https://doi.org/10.1016/j.jnca.2020.102716.

Foster, K.C., Muth, S.Q., Potterat, J.J., & Rothenberg, R.B. (2001). A faster Katz status score algorithm. *Computational & Mathematical Organization Theory*, *7*(4), 275-285. https://doi.org/10.1023/A:1013470632383.

Grover, A., & Leskovec, J. (2016). node2vec: scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 855-864). Association for Computing Machinery. New York, USA. https://doi.org/10.1145/2939672.2939754.

Hamilton, W.L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Social and Information Network.* https://doi.org/10.48550/arxiv.1706.02216.

Hanley, J.A., & McNeil, B.J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, *143*(1), 29-36. https://doi.org/10.1148/radiology.143.1.7063747.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et du jura. *Bulletin de la Société Vaudoise des Sciences Naturelles, 37*(142), 547-579. https://doi.org/10.5169/SEALS-266450.

Kipf, T.N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *Machine Learning.* http://arxiv.org/abs/1609.02907.

Kumar, S., Bisht, D., & Jain, A. (2024). An anchor-aware graph autoencoder fused with Gini index model for link prediction. *SN Computer Science*, *5*(6), 745. https://doi.org/10.1007/s42979-024-03081-z.

Kumar, S., Jain, A., & Bisht, D.C.S. (2025). Enhancing link prediction in dynamic social networks: a novel algorithm integrating global and local topological structures. *International Journal of Data Mining, Modelling and Management*, *17*(1), 26-53. https://doi.org/10.1504/IJDMMM.2025.144611.

Li, Y., Yin, J., & Chen, L. (2021). SEAL: semisupervised adversarial active learning on attributed graphs. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(7), 3136-3147. https://doi.org/10.1109/TNNLS.2020.3009682.

Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, *58*(7), 1019-1031. https://doi.org/10.1002/asi.20591.

Liu, Z., Fang, Y., Liu, Y., & Zheng, V.W. (2023). Neighbor-anchoring adversarial graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, *35*(1), 784-795. https://doi.org/10.1109/TKDE.2021.3087970.

Lü, L., Jin, C.-H., & Zhou, T. (2009). Similarity index based on local paths for link prediction of complex networks. *Physical Review E*, *80*(4), 046122. https://doi.org/10.1103/PhysRevE.80.046122.

Muscoloni, A., Abdelhamid, I., & Cannistraci, C.V. (2018). Local-community network automata modelling based on length-three-paths for prediction of complex network structures in protein interactomes, food webs and more. *bioRxiv* (pp. 1-23). Paris, France. https://doi.org/10.1101/346916.

Nasiri, E., Berahmand, K., Rostami, M., & Dabiri, M. (2021). A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding. *Computers in Biology and Medicine*, *137*, 104772. https://doi.org/10.1016/j.compbiomed.2021.104772.

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). DeepWalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 701-710). Association for Computing Machinery. New York, USA. https://doi.org/10.1145/2623330.2623732.

Powers, D.M.W. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *International Journal of Machine Learning Technology, 2*(1), 37-63. https://doi.org/10.48550/ARXIV.2010.16061.

Rai, A.K., Tripathi, S.P., & Yadav, R.K. (2023). A novel similarity-based parameterized method for link prediction. *Chaos, Solitons & Fractals*, *175*(Part 2), 114046. https://doi.org/10.1016/j.chaos.2023.114046.

Ribeiro, L.F.R., Saverese, P.H.P., & Figueiredo, D.R. (2017). Struc2vec: learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 385-394). Association for Computing Machinery. New York, USA. https://doi.org/10.1145/3097983.3098061.

Rossi, R., & Ahmed, N. (2015). The network data repository with interactive graph analytics and visualization. *Proceedings of the AAAI Conference on Artificial Intelligence*, *29*(1), 4292-4293. https://doi.org/10.1609/aaai.v29i1.9277.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017). Graph attention networks. *Machine Learning.* https://doi.org/10.48550/arxiv.1710.10903.

Publisher's Note- Ram Arti Publishers remains neutral regarding jurisdictional claims in published maps and institutional affiliations.