

DEVELOPMENT OF LINK PREDICTION MODEL IN SOCIAL NETWORK

A Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in

Computer Science and Engineering

by

**FATIMA ZIYA
(2K21/PHDCO/502)**

Under the supervision of

Dr. Sanjay Kumar

Delhi Technological University



**Department of Computer Science and Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042**

February, 2026



©DELHI TECHNOLOGICAL UNIVERSITY-2026
ALL RIGHTS RESERVED

ACKNOWLEDGEMENT

कर्पूरगौरं करुणावतारं, संसारसारं भुजगेन्द्रहारम् ।
सदावसन्तं हृदयारविन्दे, भवं भवानीसहितं नमामि ॥

At the core of my academic odyssey, I express my heartfelt gratitude to the divine presence for being the unwavering source of strength, resilience, and guidance. In moments of challenge and triumph alike, the divine hand has provided the fortitude needed to navigate the complexities of this academic journey. The belief in a higher purpose and a guiding force has been a constant beacon, offering solace and inspiration during the trials and triumphs of my research endeavors. I am deeply grateful to the Almighty God for his boundless blessings, guidance, and strength throughout this journey. Without His grace, this achievement would not have been possible.

At the forefront, I express immense gratitude and profound appreciation to my supervisor, **Dr. Sanjay Kumar**. His steadfast support, faith in my capabilities, and comprehensive guidance have been pivotal throughout my doctoral studies. His genuine interest in the research topic, patience, and assistance have greatly enriched my academic experience. I am truly grateful for his mentorship and the invaluable insights they has provided.

Furthermore, I extend my heartfelt thanks to my beloved family, my father **S.R. Zaidi**, my mother **Shama Zaidi**, my siblings, and my best friend **Shivani Pachori** for their unwavering love, constant encouragement, and steadfast support through every stage of this Ph.D. journey. Your belief in me, even during moments of uncertainty, has been my greatest source of strength. Through all the ups and downs, your positive attitude, honest guidance, and shared faith in God have inspired me to persevere and remain determined.

I also wish to acknowledge the support of my colleagues, friends and the faculty/staff of Delhi Technological University. Their readiness to share their expertise, encouragement, and fellowship has been crucial in fostering my research and personal

development. The intellectual exchanges and support from this community have been invaluable.

This thesis is a testament to the collective efforts and encouragement of these remarkable entities and individuals. I am profoundly grateful for their enduring contributions, which have significantly impacted both my academic and personal growth.

Date: February 27, 2026

Fatima Ziya



DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahabad Daulatpur, Main Bawana Road, Delhi-110042, INDIA

CANDIDATE'S DECLARATION

I **Fatima Ziya, (2K21/PHDCO/502)** hereby declare that the work which is being presented in the thesis entitled "**Development of Link Prediction Model in Social Network** " in the partial fulfillment of the requirements for the award of the Degree of Doctor of Philosophy, submitted in the Department of Computer Science and Engineering, Delhi Technological University is an authentic record of my own work carried out during the period from 2022 to 2026 under the supervision of Dr. Sanjay Kumar.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Fatima Ziya
(2K21/PHDCO/502)

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis, and the statement made by the candidate is correct to the best of my knowledge.

Signature of Supervisor

Signature of External Examiner



DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahabad Daultapur, Main Bawana Road, Delhi-110042, INDIA

CERTIFICATE BY THE SUPERVISOR

Certified that **Fatima Ziya (2K21/PHDCO/502)** has carried out her research work presented in this thesis entitled "**DEVELOPMENT OF LINK PREDICTION MODEL IN SOCIAL NETWORK**" for the award of **Doctor of Philosophy** from the Department of Computer Science and Engineering, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by the student herself, and the contents of the thesis do not form the basis of the award of any other degree to the candidate or to anybody else from this or any other institution.

Dr. Sanjay Kumar

(Supervisor)

Department of Computer Science and Engineering
Delhi Technological University, Delhi

Date: February 27, 2026

Place: New Delhi

Abstract

Link prediction in social networks plays a crucial role in understanding network evolution, identifying potential interactions, and supporting applications such as recommendation systems, community analysis, and the discovery of biological networks. The fundamental problem of link prediction is to estimate the likelihood of future or missing connections between pairs of nodes based on existing network information, structural patterns, node attributes, and temporal evolution. However, real-world networks are highly complex, sparse, dynamic, and heterogeneous, making traditional similarity-based and shallow learning approaches insufficient to capture deep structural semantics and evolving behavioral patterns.

In this thesis, we introduce a robust and adaptive approach to link prediction in social networks. The present study integrates traditional similarity-based techniques with advanced deep music recommendations, among effective similarity scores existing methods for list structure- and attribute-aware information, a single similarity index, or paths from performance and reliability of the proposed methodology.

The first model, GSVAELP, introduces a hybrid GraphSAGE-VAE model that leverages local neighborhood aggregation with probabilistic latent-space embedding, successfully capturing both structural dependencies and latent relational patterns. This laid the foundation for robust structure-and-attribute-aware link prediction.

The second study, MetaLP-DGI, introduced centrality-aware Deep Graph Infomax with meta-learning, enhancing embedding quality by incorporating influential node characteristics while improving generalization across heterogeneous networks.

The third model, Hybrid Graph Embedding and Ensemble Learning, demonstrated that combining multiple embeddings with ensemble classifiers significantly improves predictive consistency and reduces model bias.

Further, the fourth model enhancement is achieved through MetaLP-DGI, which utilizes Deep Graph Infomax (DGI) embeddings integrated with a centrality-aware transition matrix to capture both global and local structural dependencies. The meta-learning component in MetaLP-DGI optimizes the learning process across heterogeneous datasets, improving robustness and adaptability. Complementing these in-depth approaches. The fifth study, Link Prediction in Social Networks: A Hybrid Approach with Graph Embedding and Ensemble Learning, combines structure- and attribute-based embeddings with ensemble classifiers, such as CatBoost and Random Forest, to deliver high-accuracy predictions in social network scenarios. Finally, the last study, UnifiedAttri2Vec-LSTM constructs a unified embedding by integrating multiple embedding algorithms through Attri2Vec and leverages LSTM to model temporal and structural dependencies simultaneously.

Overall, this thesis contributes a comprehensive exploration of hybrid, generative, and meta-learning-based frameworks for link prediction, establishing a strong foundation for adaptive and scalable graph analytics. The progressive integration of centrality, attention, temporal evolution, and ensemble learning provides a unified roadmap for advancing intelligent link prediction in complex and dynamic networked systems.

PUBLICATIONS

Published/ Accepted

SCI-INDEXED JOURNALS

1. Fatima Ziya, Sanjay Kumar, "GSVAELP: integrating graphSAGE and variational autoencoder for link prediction.,"**Multimedia Tools and Applications** Volume 84 (31 August 2024),doi:<https://doi.org/10.1007/s11042-024-20123-z>.
2. Fatima Ziya, Sanjay Kumar,"MetaLP-DGI: Meta-Learning-Based Link Prediction With Centrality-Aware Deep Graph Infomax Embeddings,"**Concurrency and Computation, Wiley** 37(21-22)(31 July 2025),doi:<https://doi.org/10.1002/cpe.70211>.(Impact Factor 1.5).
3. Fatima Ziya, Sanjay Kumar, "Enhancing Link Prediction through Graph Attention Network and Linear Discriminant Analysis", **Expert System,Wiley** doi:<https://doi.org/10.1111/exsy.70114>. (Impact Factor 2.3)

Communicated/Under Review

SCI-INDEXED JOURNALS

1. Fatima Ziya, Sanjay Kumar "Unified Link Prediction: Leveraging Multi-Algorithm Embeddings and Attri2Vec", **Journal of Chaos, Solitons and Fractals**, June,2025 (under revisions) (Minor Revisions Submitted)(Impact Factor 5.6)
2. Fatima Ziya, Sanjay Kumar "Integrating Structure and Similarity-Aware Graph Convolutional Network with Cross-Layer Representation learning in Multiplex

Networks”, **IEEE Transactions on Computational Social Systems**,(under review).

SCOPUS-INDEXED CONFERENCES

1. Fatima Ziya and Sanjay Kumar, "Link Prediction in Social Networks: A Hybrid Approach with Graph Embedding and Ensemble Learning," **8th International Conference on Innovative Computing and Communication (ICICC-2025)** Shaheed Sukhdev College of Business Studies, New Delhi, India, 2025.doi: https://doi.org/10.1007/978-981-96-6883-0_39. **(Published)**.
2. Fatima Ziya and Sanjay Kumar, "MetaLP-EC: A Meta-Learning Framework for Temporal Link Prediction with Evolving Centrality and GCN Embeddings," 6th International Conference on Data Analytics and Management (ICDAM-2025) Venue London Metropolitan University, United Kingdom June 13, 2025, https://doi.org/10.1007/978-3-032-03769-5_12. **(Published)**.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
CANDIDATE's DECLARATION	iv
CERTIFICATE BY THE SUPERVISOR(s)	v
ABSTRACT	vi
LIST OF PUBLICATIONS	viii
TABLE OF CONTENTS	xi
LIST OF FIGURES	xvi
LIST OF TABLES	xviii
LIST OF ABBREVIATIONS	xxi
1 INTRODUCTION	1
1.0.1 Link Prediction	2
1.0.2 Social Network Analysis (SNA)	3
1.0.3 Background	4
1.0.3.1 Similarity-Based Methods:	5
1.0.3.2 Path Based Approaches:	10
1.0.3.3 Probabilistic-based Approaches:	10
1.0.3.4 Classification-based Approaches:	10

1.0.3.5	Network Embedding-based Approaches:	11
1.0.4	Link Prediction Problem	11
1.0.5	Challenges in Link Prediction	11
1.0.6	Applications	13
1.0.7	Motivation & Scope	14
1.1	Research Gaps	15
1.2	Research Objectives	16
1.3	Problem Statement	17
1.4	Major Contributions of the Thesis	18
1.5	Organization of Thesis	19
2	LITERATURE REVIEW	26
2.1	Previous works in Link Prediction	26
2.1.1	Early Computational Era (2003–2010): Similarity-Based Methods	27
2.1.2	Probabilistic and Matrix-Factorization Methods (2010–2015)	30
2.1.3	Representation Learning and Network Embedding Era (2014–2019)	31
2.1.4	Multiplex, Multilayer, and Temporal Networks (2018–present)	34
3	GSVAELP: integrating GraphSAGE and Variational Autoencoder for Link Prediction	47
3.1	Background	48
3.2	Proposed Methodology	51
3.2.1	Feature extraction and generating network embedding using graphSAGE	52
3.2.2	Variational autoencoder for node embedding refinement	53
3.2.2.1	Encoder	54
3.2.2.2	Decoder	54

3.2.3	Training Classifier for Link Prediction	55
3.3	Experiments and analysis	56
3.3.1	Evaluation Metrics	58
3.4	Experiment and Result Analysis	60
3.4.1	Choice of Splitting Ratio	61
3.4.2	Parameter Sensitivity	63
3.4.3	Comparative Performance Study	64
3.4.4	Average Performance of each comparing method of link prediction across all datasets	68
3.4.5	Performance Measures	70
3.5	Summary	71
4	MetaLP-EC: A Meta-Learning Framework for Temporal Link Prediction with Evolving Centrality and GCN Embeddings	73
4.1	Background	74
4.2	Proposed Methodology	75
4.2.1	Temporal Graph Modelling	76
4.2.2	Evolving Centrality-Aware Transition Matrix	77
4.2.3	Temporal Graph Embedding using GCN	77
4.2.4	LSTM-based Classification for Link Prediction	78
4.3	Experimental Setup and Datasets	80
4.3.1	Evaluation Metrics	80
4.4	Performance of the proposed method against other methods	81
4.5	Conclusion	83
5	MetaLP-DGI: Meta-Learning-Based Link Prediction With Centrality-Aware Deep Graph Infomax Embeddings	85
5.1	Background	86
5.2	Proposed Methodology	88

5.2.1	Multi-step Connectivity Matrix Construction	89
5.2.2	Modified DGI with meta-learning connectivity matrix	91
5.2.3	Boosting Algorithm for Link Prediction	94
5.2.4	Algorithm Explanation	97
5.3	Experimental Result and Analysis	100
5.3.1	Hyperparamter Tunning	102
5.3.2	Comparative Performance Study	104
5.3.3	Relative Information Gain	112
5.3.4	Quantify Efficiency	113
5.3.5	Average performance of each comparing method of link prediction on all datasets	114
5.4	Conclusion	117
5.5	Summary	118
6	Link Prediction in Social Networks: A Hybrid Approach with Graph Embedding and Ensemble Learning	119
6.1	Background	120
6.2	Proposed Methodology	121
6.2.1	Creation of Katz Similarity Matrix	122
6.2.2	Representation Learning using node2vec Embedding	123
6.2.3	Creation and Balancing of Training and Testing Datasets	124
6.2.4	Training XGBoost Ensemble Classifier	124
6.3	Social Network Datasets	125
6.4	Experimental Result and Analysis	126
6.4.1	Comparative Performance Study	127
6.5	Conclusion	129
6.6	Summary	130

7	Enhancing Link Prediction through Graph Attention Network and Linear Discriminant Analysis	131
7.1	Background	132
7.2	Preliminary	134
7.2.1	Definition 1: Measuring Structural Similarity (MSS)	134
7.2.2	Definition 2: Measuring Topological Similarity (MTS)	135
7.2.3	Hubs and authorities centrality (HAC)	135
7.2.4	Definition 3: Weight Matrix	136
7.3	Proposed Work	136
7.3.1	Features Extraction	137
7.3.2	Generating Embedding using GAT	138
7.3.3	Model Training and Loss function	141
7.3.4	Algorithmic Explanation	144
7.4	Dataset and Evaluation Metrics	145
7.4.1	Evaluation Metrics	147
7.5	Results and Analysis	148
7.5.1	Hyperparameter Analysis	149
7.5.1.1	Other Parameter Analysis	150
7.5.2	Comparative Performance Study	153
7.5.3	Average Performance achieved by each LP model on all eight different datasets	162
7.5.4	Relative Information Gain	164
7.5.5	Limitations and Future Scope	164
7.6	Conclusion	165
7.7	Summary	166
8	UnifiedAttri2Vec-LSTM:Unified Link Prediction: Leveraging Multi-Algorithm Embeddings and Attri2Vec	168

8.1	Background	168
9	Conclusion, Future Scope and Social Impact	171
9.1	Conclusion	171
9.2	Future Scope	173
9.3	Social Impact	174
	References	177
	Appendices	201
.1	List of Prime Research Papers Selected in this SLR	202
	List of Publications	209
	Author Biography	218

List of Figures

1.1	Link Prediction in Social Network	3
1.2	Social Network Analysis.	4
1.3	Taxonomy for link prediction techniques.	5
1.4	Applications of Link Prediction in the real world	14
3.1	Architecture of proposed methodology	56
3.2	AUC Results obtained for different Dataset for various models as the splitting ratio varied from 50% to 95% in steps of 5%	62
3.3	Parameter Sensitivity on DBLP Dataset	64
3.4	Average Precision for each method on all eight datasets	67
3.5	Average Recall for each method on all eight datasets	68
3.6	Average F1-score for each method on all eight datasets	69
4.1	Workflow of the MetaLP-EC Framework for Temporal Link Prediction.	76
4.2	Performance Evaluation of Link Prediction Methods Based on Average Accuracy	83
5.1	Flowchart of the proposed meta-learning, deep graph infomax, and CatBoost-based link prediction model	95
5.2	Parameter Sensitivity on all Datasets	105
5.3	Average F1 Score for each model on all the datasets	115
5.4	Average AUC score for each model on all the datasets	115

6.1	Feature extraction of using node2vec model with Ensemble machine learning classifiers	126
7.1	Overview of the proposed method with feature generation and link prediction model.	139
7.2	a–d: Study of training and testing dataset ratio to obtain optimal allocation for training and testing. AUC outcomes for various datasets using different link prediction models, with the allocation of training data from 55% to 95% in 5% steps and the remaining for testing.	151
7.3	Various hyperparameter studies	152
7.4	Comparison of Average Accuracy across different link prediction models on all the datasets	163
7.5	Comparison of Average AUC Scores across different link prediction models on all the datasets	163
7.6	The relative improvement of the proposed model in precision, recall, F1 score, and AUC when compared to the chosen models for each dataset.	165

List of Tables

2.1	Comparison of Link Prediction Methods with Characteristics and Limitations	36
3.1	The Statistical Description of various Dataset used	58
3.2	Overall Performance Metrics of Caltech Dataset	65
3.3	Overall Performance Metrics of PolBlogs Dataset	66
3.4	Overall Performance Metrics of DBLP Dataset	67
3.5	Overall Performance Metrics of FB-TV show dataset	68
3.6	Overall Performance Metrics of Delaunay-N12 Dqataset	69
3.7	Overall Performance Metrics of Hamsterster Dataset	70
3.8	Overall Performance Metrics of Public Figure Dataset	71
3.9	Overall Performance Metrics of Wiki Vote Dataset	72
4.1	Basic Statistics of Selected Dynamic Networks	80
4.2	Comparative Analysis on Dynamic PPI and FB-Forum Datasets	81
4.3	Comparative Analysis on Wiki-Talk and Dynamic Email Datasets	82
5.1	Brief Statistical Description of Various Social Network Datasets used in this study	99
5.2	Evaluation Metrics for Link Prediction	100
5.3	Experimental results of different link prediction methods based on Prec, Recall, F1 Score, Acc, and AUC for the Facebook dataset.	104
5.4	Experimental results of different link prediction methods based on Prec, Recall, F1 Score, Acc, and AUC for the Email-University dataset.	106

5.5	Experimental results of different link prediction methods based on Prec, Recall, F1 score, Acc, and AUC for the FB-TV dataset.	107
5.6	Experimental results of different link prediction methods based on Prec, Recall, F1 score, Acc, and AUC for the Fb-Political dataset.	108
5.7	Experimental results of different link prediction models based on Prec, Recall, F1 score, Acc, and AUC for the Social-Wellesley dataset.	109
5.8	Experimental results of different link prediction methods based on Prec, Recall, F1 score, Acc, and AUC for the Epinions Dataset.	109
5.9	Experimental results of different link prediction methods based on Prec, Recall, F1 score, Acc, and AUC for the PolBlogs Dataset.	110
5.10	Experimental results of different link prediction methods based on Precision, Recall, F1 score, Accuracy, and AUC for the Amazon Dataset	111
5.11	Experimental results of different link prediction methods based on Prec, Recall, F1 Score, Acc, and AUC for the PPI Dataset.	111
5.12	Relative Gain in AUC of the proposed approach against the best baseline and recent findings	113
5.13	Model Efficiency Comparison in terms of Training Time, Inference Time, and Memory Usage	114
6.1	Details of Social Network Datasets	126
6.2	Outcomes of various link prediction techniques applied to the FB-food and Email-University datasets.	128
6.3	Outcomes of various link prediction techniques applied to the Hamsterster and FB-TV datasets.	128
7.1	Topological measures of real-world datasets	146
7.2	Accuracy of proposed SNGAT-LDA Model on different hidden layers of GAT over several networks	150

7.3	Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on Cora dataset	154
7.4	Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on Citeseer dataset	155
7.5	Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on Pubmed dataset	156
7.6	Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on Facebook dataset	157
7.7	Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on Wiki dataset	158
7.8	Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on the Hamsterster dataset	159
7.9	Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on the PolBlogs dataset	160
7.10	Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on the Email Dataset	161
7.11	Evaluation metrics for different link prediction models based on Precision, Recall, F1 Score, and AUC on the Twitter dataset	162

List of Abbreviations

LP Link Prediction

SNA Social network analysis

CN common neighbors

AA Adamic Adar

JC Jaccard Coefficient

PA preferential attachment

RA Resource Allocation Index

KI Katz Index

NSP Negated Shortest Path

RW Random Walk

LPI Local Path Index

LRW local Random Walks

SRW Superposed Random Walks

PRM Probabilistic Relational Model

PERM Probabilistic entity-relationship model

SRM Stochastic Relational Model

SVM Support Vector Machine

KNN K-Nearest Neighbors

DT Decision Tree

NB Naive Bayes

LR Logistic Regression

MLP Multi-Layer Perceptron

LINE Large-scale Information Network Embedding

DGI Deep Graph Infomax

VAE Variational autoencoders

GNN Graph Neural Network

LPI local path index

RAI Resource Allocation Index

GSVAELP graphSAGE and variational autoencoder for link prediction

MetaLP-EC Meta-Learning framework for Temporal Link Prediction

MetaLP-DGI Meta-Learning-Based Link Prediction model

LDA Linear Discriminant Analysis

GAT Graph Attention Network

ADB AdaBoost classifier

GNN Graph Neural Network

LID Local Information Dimensionality

GCN Graph Convolutional Network

MFI Matrix Factorization Index

LHN2 Leicht-Holme Newman

AUC Area under the Curve

LSTM Long Short-Term Memory

MLP Multilayer Perceptron

MCFMN-LP Multi-Correlation Features-based Link Prediction in Multiplex Networks

TP True Positive

TN True Negative

FP False Positive

FN False Negative

ML Machine Learning

DL Deep Learning

RF Random Forest

KNN K-Nearest Neighbour

Chapter 1

INTRODUCTION

In recent years, complex networks have attracted significant interest from researchers across various disciplines. Many real-world systems, including online social platforms, biological interaction networks, and transportation systems, can be modeled as complex networks consisting of numerous entities (nodes) connected through intricate relationships (edges). These networks present several important research challenges, such as community detection, influence maximization, and information diffusion. Among these, link prediction (LP) has emerged as one of the most prominent research areas. The primary objective of link prediction is to estimate whether a connection is likely to form between two currently unconnected nodes in the future [1].

Link prediction plays a vital role in many practical applications. It supports friend recommendation in social networks, helps identify potential future collaborations in co-authorship networks, and enhances recommendation systems by utilizing available user interaction data to make more informed predictions. Moreover, it provides insights into how networks evolve over time, as most real-world networks, such as social and communication networks, continuously grow and change. Over the years, numerous approaches for link prediction have been introduced [2], generally categorized into similarity-based, maximum-likelihood-based, and probabilistic models [3]. Additionally, social network analysis has increasingly contributed to addressing various modern challenges, including influential node detection, community discovery, opin-

ion leader identification, fake news mitigation, rumor prediction, and link prediction, among others [4, 5].

1.0.1 Link Prediction

Link prediction is a technique used to analyze a network and assess the likelihood of a future connection between any two nodes. In other words, it predicts whether a link can exist between two nodes that are not connected currently in the network. Figure 1.1 shows an example of the prediction of links between nodes **1-4**, **3-8**, and **7-9**.

The concept of predicting links has grown in popularity ever since it was first introduced in complicated network analysis [6]. In real-life scenarios, link prediction has many uses, including forecasting disease outbreaks, predicting link quality in wireless community networks, recommending new friends on social media, planning transportation, making product recommendations for e-commerce applications, and recommending hotels and music, among many others [7]. Link prediction facilitates the anticipation of network development as it evolves over time. Therefore, in today's world, which is entirely connected, link prediction can be very closely related to information and knowledge architecture. The basis for the link prediction process is the node pair's popularity and closeness. Nodes that are popular will connect with other nodes, and nodes that are similar to one another have a greater likelihood of connecting soon; thus, this relationship appears to be rational. Over the years, numerous link prediction techniques have been proposed in papers. These strategies can be generally categorized into similarity-based, maximum likelihood-based, matrix factorization-based, probabilistic, and random walk methods. Recent research has also investigated the potential of machine learning Machine Learning (ML) and deep learning Deep Learning (DL) methods. This can be useful for various applications, such as identifying potential friends or collaborators, recommending new connections to users, and detecting fraudulent activities.

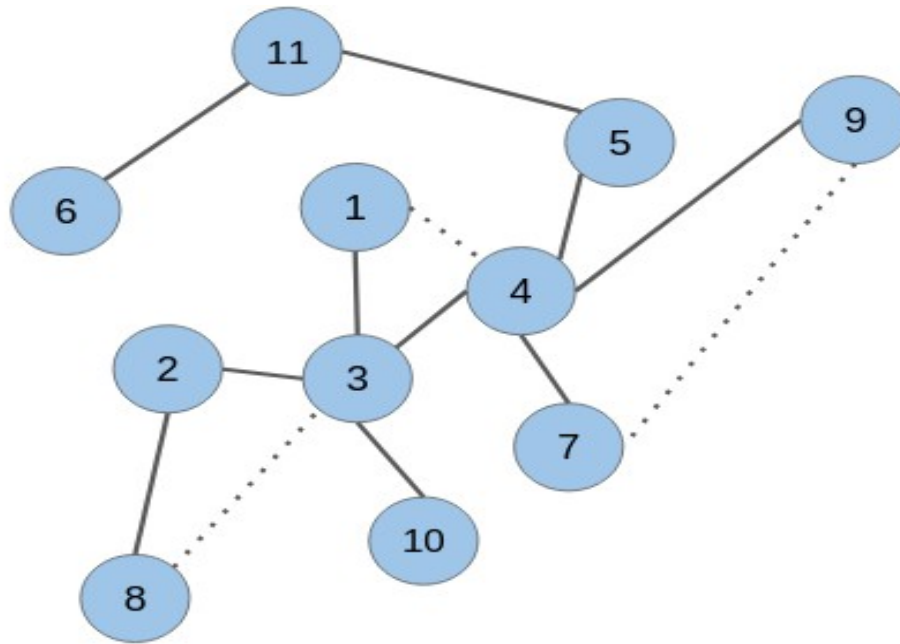


Fig. 1.1: Link Prediction in Social Network

1.0.2 Social Network Analysis (SNA)

Social network analysis (SNA) is an area of study that analyzes social networks and the different people or individuals who are related to each other. SNA determines how people are connected, their patterns of connection, and the structure and dynamics of social networks.

SNA focuses on the idea that social interactions and relationships can be represented as a network, where the nodes can be considered as individual people or groups and edges represent what are the connections between them[8]. By analyzing the patterns of these connections, SNA can reveal insights into how information, resources, and influence flow through social networks, and how these networks can impact behavior, attitudes, and outcomes. Figure 1.3 shows how different people are connected over social networks. SNA is applicable in several fields, including sociology, psychology, anthropology, communication studies, management, and marketing, to name a few[9]. SNA uses a range of analytical tools, including centrality measures, clustering algorithms, and network visualization techniques, to analyze social networks and identify



Fig. 1.2: Social Network Analysis.

key actors, groups, and patterns of interaction. Many methods have been devised for link prediction in social networks, including the Adamic-Adar, Jaccard coefficient, preferential attachment methods, etc.

1.0.3 Background

A link represents a connection between two nodes in a network, and this seemingly simple concept is capable of modeling highly complex systems involving numerous interacting entities. Link Prediction (LP) is based on the empirical observation that entities with higher similarity are more likely to establish connections. However, similarity in networks is often an abstract notion and may differ significantly across domains, making it essential to thoroughly understand the underlying context before defining what constitutes similarity between nodes. In many real-world networks, nodes naturally tend to form tightly connected [10], leading to a common interpretation of similarity as the presence of meaningful direct or indirect paths between nodes. One of the major challenges in link prediction lies in achieving an effective trade-off between uti-

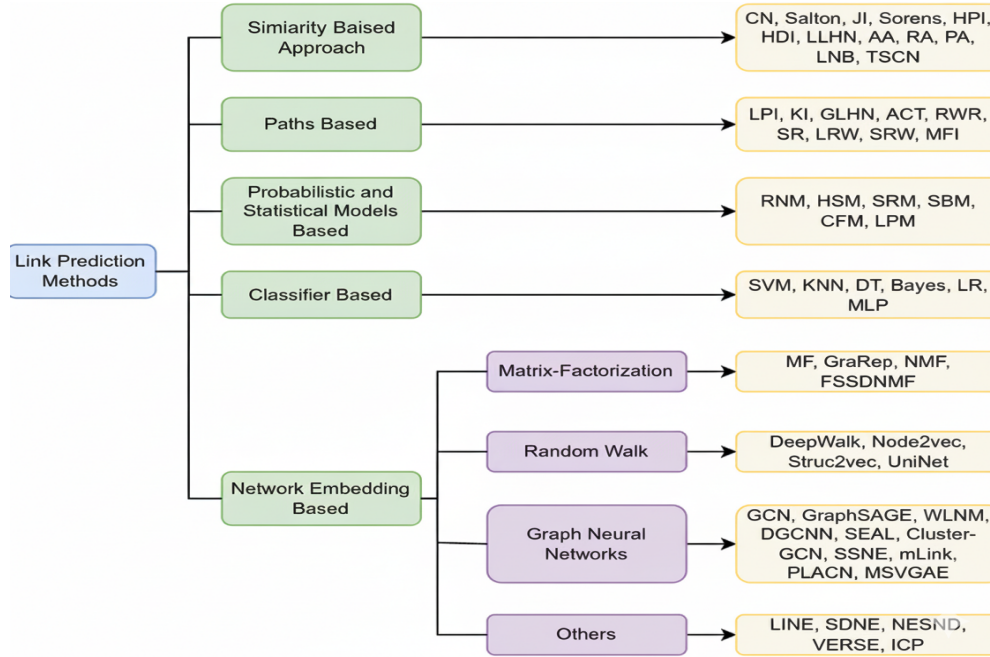


Fig. 1.3: Taxonomy for link prediction techniques.

lizing sufficient information for accurate prediction and managing the computational complexity associated with extracting and processing that information.

1.0.3.1 Similarity-Based Methods:

Similarity-based techniques assume that nodes typically establish connections with other nodes that are similar to them. These techniques are based on the idea that two nodes are similar if they are close together in the network based on a specific distance function or if they are connected to other similar nodes. These methods define a function $s(x, y)$ that assigns a similarity score to each pair of nodes x and y .

1.0.3.1.1 Local Approaches: Local similarity-based methods measure the similarity between each node and other nodes in the network using structural information related to its neighborhood. These methods are highly parallelizable and faster than nonlocal methods. Local similarity methods compute the similarity matrix using the local information of an edge. Numerous local feature-based algorithms have been presented in the literature [11, 3], including common neighbors (CN)[12], Adamic

Adar (AA)[13], Jaccard Coefficient (JC) [14], preferential attachment (PA) [15], and the Salton index [16].

- **Adamic Adar:** AA similarity measure is commonly used in network analysis for both link prediction and community detection. It was introduced by Lada Adamic and Eytan Adar in their 2003 paper titled "Friends and Neighbors on the Web [13]." AA is based on the concept that nodes in a network are more likely to be connected if they share many common neighbors. The measure calculates the similarity between two nodes by summing the inverse logarithm of the degree of each common neighbor they share. Essentially, AA quantifies the similarity of the "neighborhoods" surrounding two nodes.

$$AA(v_i, v_j) = \sum_{v_k \in (v_i) \cap \Gamma(v_j)} \frac{1}{\log \Gamma(v_k)} \quad (1.1)$$

The Eq.1.1 represents the Adamic-Adar similarity index for two nodes (v_i, v_j) . It measures their similarity by summing over their common neighbor v_k . where each neighbor's contribution is weighted inversely by the logarithm of its degree $\Gamma(v_k)$. This gives higher importance to rare (low-degree) common neighbors.

- **Jaccard Coefficient JC:** Data mining and machine learning use the Jaccard coefficient, also known as Jaccard similarity, as a statistic to determine how similar two sets are. The size of the intersection between the two sets is calculated by dividing the size of their union [14]. The Jaccard coefficient, which quantifies overlap between two sets in more straightforward terms, is frequently used to gauge how similar two sets are to one another. JC is defined as:

$$JC(v_i, v_j) = \frac{\Gamma(v_i) \cap \Gamma(v_j)}{\Gamma(v_i) \cup \Gamma(v_j)} \quad (1.2)$$

The equation 1.2 defines the Jaccard Coefficient (JC) for two nodes v_i and v_j calculated as the ratio of the number of their common neighbors to the number

of neighbors in their union. It reflects the proportion of shared neighbors relative to all unique neighbors of the two nodes.

- **Common Neighbor CN:** This similarity metric is a local measurement that depends on the number of friends that a particular pair of nodes has in common. In other words, it evaluates the similarity of the nodes by considering the number of connections they share [12].

$$CN(v_i, v_j) = \Gamma(v_i) \cap \Gamma(v_j) \quad (1.3)$$

This equation represents the Common Neighbors (CN) measure, which counts the number of nodes that are directly connected to both v_i and v_j . A higher CN value indicates a greater likelihood of a link existing between the two nodes.

- **Preferential Attachment PA:** A mechanism known as preferential attachment occurs in networks when nodes have a propensity to link with other nodes who already have a large number of connections. This process, sometimes known as the "rich get richer" effect, causes the establishment of densely connected hubs. Simply said, nodes with more links have a higher chance of attracting new linkages. The Barabási-Albert model can be used to mathematically characterize this process [15]. According to this model, new nodes are sequentially added to the network and connect to existing nodes with a probability proportional to the number of linkages each existing node currently has.

$$PA(v_i, v_j) = \Gamma(v_i) \times \Gamma(v_j) \quad (1.4)$$

This equation 1.4 defines the Preferential Attachment (PA) score for two nodes v_i and v_j computed as the product of their degrees $\Gamma(v_i)$ and $\Gamma(v_j)$. It is based on the idea that nodes with higher degrees are more likely to form new links.

-
- **Resource Allocation Index (RAI):** In order to forecast links in different networks, Tao Zhou, Liyuan Lü, and Yi-Cheng Zhang developed the Resource Allocation Index (RA) [17]. The method accounts for the similarity of nodes and their ability to share resources to efficiently allocate resources. This strategy is used by the algorithm to find potential connection formations depending on how resources are distributed throughout the network.

$$RA(v_i, v_j) = \sum_{v_k \in \Gamma(v_i) \cap \Gamma(v_j)} \frac{1}{\Gamma(v_k)} \quad (1.5)$$

This equation 1.5 represents the Resource Allocation (RA) index for two nodes (v_i, v_j) . It sums the contributions of their common neighbors v_k . where each contribution is weighted inversely by the degree $\Gamma(v_k)$ of that neighbor. In this way, low-degree common neighbors contribute more to the similarity score than high-degree ones.

1.3.1.1.2 Global Approaches: Global similarity-based indices assign a score to each link based on the topological information of the entire network. These techniques are not limited to comparing the similarities between two nodes. Global similarity-based methods, which are not restricted to measuring similarity between nodes up to two distances [18], utilize the entire topological information of the network to rank the similarity between node pairs. While examining the network's entire topology allows for greater flexibility in link prediction analysis, these approaches have high time complexity, making them impractical for large networks, and their parallelization can be extremely challenging. Global approaches include Katz Index (KI) [19], Negated Shortest Path (NSP), and Random Walk (RW).

- **Katz Index (KI):** The KI measures the closeness between two nodes by counting all possible paths connecting them, while giving shorter paths higher weight through the damping factor β . Direct links contribute most to the score, but indi-

rect connections through multiple steps are also considered, making KI effective at capturing both local and global structural relationships in a network.

$$KI(v_i, v_j) = \sum_{(l=1)}^{\infty} \beta^l \cdot |paths_{v_i, v_j}^l| \quad (1.6)$$

Where $paths_{v_i, v_j}^l$ indicates the number of distinct paths of length l between nodes v_i and v_j . The β is the damping factor ($0 < \beta < 1$) to penalize longer paths.

1.3.1.1.3 Quasi-Local Approaches: Compared to local methods, quasi-local approaches require more data and do not require global topological information. As a result, these methods have been developed to strike a balance between local and global measures, yielding better results and requiring less time complexity. Some popular quasi-local approaches are local path index (LPI), local Random Walks (LRW), and Superposed Random Walks (SRW).

- **Local path index (LPI):** The LPI [20] is a link prediction measure that extends the Common Neighbors idea by also considering paths of length 3 between two nodes, in addition to paths of length 2. This allows it to incorporate slightly more global structural information without the high computational cost of fully global methods like the Katz Index.

$$LPI(v_i, v_j) = (A^2)_{ij} + \alpha \cdot (A)_{ij}^3 \quad (1.7)$$

The Eq1.7 computes the omputes link similarity between nodes v_i and v_j by summing the number of paths of length 2 $(A^2)_{ij}$ and a weighted count of paths of length 3 $(A)_{ij}^3$.

1.0.3.2 Path Based Approaches:

These approaches analyze the paths of various lengths that connect two nodes to infer a potential link, often involving random walk or path counting techniques. Examples include the random walk with restart (RWR) and the link score calculation. The fundamental principle is expressed as: Two nodes are likely to connect if many short paths already exist between them.

1.0.3.3 Probabilistic-based Approaches:

The goal of the probabilistic-based approach under feature extraction techniques is to use the learned model to predict the missing links after extracting the underlying structure from the observed network. Three widely used probabilistic-based link prediction techniques are Probabilistic Relational Model (PRM), Probabilistic entity-relationship model (PERM), and Stochastic Relational Model (SRM) [21]. These models typically perform poorly on sparse, undirected networks but better on dense, directed networks. Graph embedding-based link prediction studies can be considered generative models [15].

1.0.3.4 Classification-based Approaches:

Classification-based link prediction treats the problem of predicting missing or future links in a network as a supervised binary classification task. In this framework, each pair of nodes (u, v) is represented by a feature vector describing their relationship, and a classifier is trained to distinguish between positive instances (node pairs with an existing link) and negative instances (node pairs without a link). Examples of some classification-based approaches for LP, including Support Vector Machine (SVM) [22], K-Nearest Neighbour (KNN) [23], Decision Tree (DT) [24], Naive Bayes (NB), Logistic Regression (LR) [25], and Multilayer Perceptron (MLP) [26].

1.0.3.5 Network Embedding-based Approaches:

Network embedding-based link prediction models are a class of methods that learn low-dimensional, continuous vector representations of nodes in a network, such that the learned embeddings preserve the structural and semantic properties of the original graph. These embeddings serve as features for predicting the likelihood of a future or missing link between pairs of nodes. A good network embedding method should be able to capture the internal structure of the network well to predict the possible future links. For example, Deepwalk [27], node2vec [28], Large-scale Information Network Embedding (LINE)[29], and graphSage [30] are network embedding-based approaches.

1.0.4 Link Prediction Problem

The link prediction problem in undirected networks is defined as follows. Given a snapshot of an undirected network in time t where each node represents an entity or actor and each link represents an interaction or relationship between the pair of entities connected through the link, the link prediction problem can be formally defined as inferring the subset of missing links (existing but non-observed links) in the current snapshot or that will be formed in time $t + 1$ [18].

1.0.5 Challenges in Link Prediction

Link prediction in social networks faces several significant challenges that affect the accuracy, scalability, and applicability of predictive models. One major issue is data sparsity, as real-world social networks are often highly sparse with far fewer existing links compared to the total number of possible connections. The dynamic nature of networks further complicates the task, since relationships evolve over time, requiring models to adapt to changes such as the addition or removal of nodes and links.

Additionally, noisy and incomplete data can mislead algorithms, while the cold-start problem makes predicting links for newly added nodes particularly difficult due to the lack of historical information. Scalability is another concern, as large-scale social networks can contain millions of nodes and links, creating high computational and memory demands. Algorithms also face bias and popularity effects, where highly connected nodes are over-predicted at the expense of less connected but potentially important nodes. The heterogeneity of relationships—where multiple types of connections exist—adds complexity to modeling, and privacy and ethical concerns arise when personal interaction data is used. Moreover, evaluation difficulties make it challenging to design realistic benchmarks that reflect true network evolution, and the lack of interpretability in many machine learning or deep learning models further hinders trust and understanding of predictions.

- **Data Sparsity:** Social networks are often highly sparse, with very few actual links compared to the number of possible links, making learning patterns difficult [24].

Dynamic Nature of Networks:: Social networks evolve over time — new nodes/links appear, and old ones disappear, requiring models to adapt to temporal changes[31].

Noisy and Incomplete Data: Missing, erroneous, or outdated relationship information can mislead prediction algorithms [32].

Cold-Start Problem: Predicting links for newly added nodes (with no or few connections) is difficult due to the absence of historical interaction data [33].

Scalability: Large-scale social networks with millions of nodes and links pose significant computational and memory challenges for link prediction algorithms [34].

Bias and Popularity Effect: Algorithms may over-predict links for highly

connected nodes due to degree bias, ignoring less connected but important nodes [35].

1.0.6 Applications

Link prediction has emerged as a vital research area in network science, offering significant value across various real-world domains. In the context of social networks, it enables the discovery of potential future relationships, the recovery of missing connections, and a deeper understanding of network evolution. By leveraging structural patterns, node attributes, and temporal dynamics, link prediction not only enhances user engagement in online platforms but also supports critical tasks such as community building, information dissemination, and security monitoring. The following are some prominent applications of link prediction in social networks.

- **Friend Recommendation:** Suggesting potential friends to users based on shared connections, mutual interests, or common communities (e.g., Facebook’s “People You May Know”)[9].
- **E-commerce Personalization:** Predicting links between customers and products to recommend items based on purchasing patterns, browsing behavior, and social influence.
- **Biological and Health Networks (Social Perspective):** Understanding disease spread in human contact networks or predicting patient–doctor connections in health communities.
- **Fraud and Spam Detection:** Spotting unusual or suspicious link formations that may indicate spam accounts or fraudulent activities.
- **Stock Exchange and Financial Networks:** Forecasting relationships between traders, companies, or stocks based on transaction patterns, market movements, and investor networks to assist in decision-making.

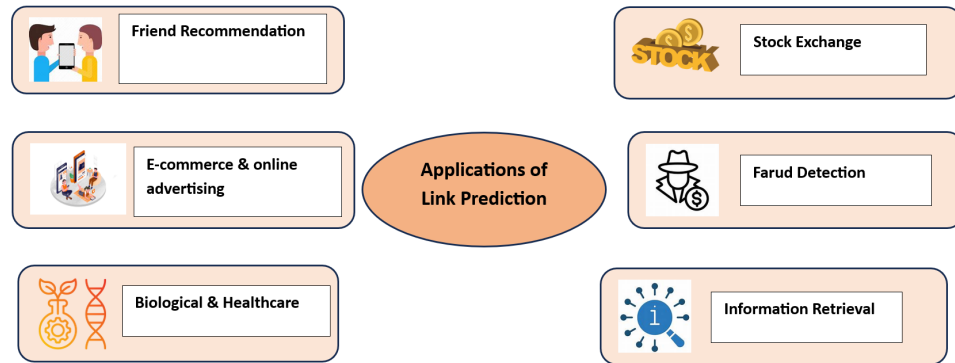


Fig. 1.4: Applications of Link Prediction in the real world

- **Information Retrieval:** Improving search results by predicting relevant connections between queries and documents or between topics and authors.

1.0.7 Motivation & Scope

There are many great potential applications of Link Prediction in social networks. The majority of previous link prediction applications were implemented for prediction purposes, for example, to predict future link connections, predict missing and spurious links in social networks (Liben-Nowell and Kleinberg, 2003). However, there are several new applications that serve more significant purposes than only predicting link tasks. The recent three categories of proposed link prediction applications are detection, recommendation, and influence analysis.

- The key motivation behind link prediction in social networks is its increasing applications in the real world, including social media marketing, online advertising, and recommendation systems[36].
- Link Prediction has numerous real-world applications in various domains, including healthcare, surveillance, sports analysis, human-computer interaction, and smart environments. By conducting research in link prediction in social networks, we can contribute to developing advanced systems/models that can

improve social links and enhance efficiency in different domains cite alman-soori2012link,ozkaynak2024common.

- Link Prediction technology has the potential to make a significant impact on society by enabling innovative applications such as Friend recommendation. In online social networks, link prediction is a popular application that deals with predicting new edges or connections between unconnected users based on their features, hobbies, and interests [37]. Some other notable applications of link predictions in real-life scenarios are predicting disease outbreaks, planning transportation, product recommendation in e-commerce applications, hotel recommendation, music recommendation, and many others.

1.1 Research Gaps

Link Prediction in social networks is a well-studied area in the field of network analysis and graph mining. However, there are still several research gaps that can be explored to enhance the accuracy and effectiveness of link prediction models. Some of these research gaps include. From the Literature Review, few research gaps which have been identified are as follows:

- **Data sparsity:** Existing link prediction methods often rely on complete or dense network data to achieve accurate predictions. However, real-world network data often suffers from data sparsity, where only a small fraction of possible connections are observed. This sparsity can limit the effectiveness of link prediction algorithms.[32, 38].
- **Scalability:** Some link prediction methods may not scale well to large-scale networks due to computational constraints. As networks grow, the computational complexity of link prediction algorithms can increase significantly, making them impractical for analyzing and predicting links in large networks [34, 39]].

- **Overfitting and generalization:** Link prediction methods can suffer from overfitting, where they perform well on the training data but fail to generalize to unseen data. This limitation arises when the algorithms rely heavily on specific features or patterns in the training network that may not be applicable in other networks[28, 40].
- **Incorporating temporal dynamics:** Most existing link prediction methods in social networks do not explicitly consider the temporal aspect of the network. Developing link prediction models that can capture temporal dynamics can lead to more accurate predictions[41, 42].

These notable gaps in existing literature highlight the limitations of current link prediction techniques in addressing the multi-directional requirements of high predictive accuracy, effective integration of structural and attribute information, adaptability to dynamic network changes, interpretability, and robust performance across diverse network types.

1.2 Research Objectives

The research gaps identified in the domain of link prediction in social networks highlight the need for comprehensive solutions that address the complex challenges of predictive accuracy, scalability to large-scale networks, robustness to dynamic and evolving structures, effective handling of sparsity, and the integration of both structural and attribute information. To address these concerns, a thorough study and experimental evaluation of the available link prediction methods across multiple paradigms, such as similarity-based, probabilistic, and machine learning/deep learning-based approaches, are crucial. Such an analysis, conducted on consistent and diverse benchmark datasets, can provide valuable insights into the strengths and limitations of existing approaches,

paving the way for the development of more reliable and effective predictive models. This necessitates the exploration of innovative, lightweight architectures that can achieve a balanced trade-off between predictive capability and computational efficiency without compromising robustness. In response to these identified gaps, research objectives are formulated to drive progress in the field of link prediction. These objectives aim to foster the creation of models that are not only accurate and efficient but also scalable, interpretable, and adaptable to real-world social network conditions. The research objectives are defined as follows:

- **Objective 1:** To study and compare the performance of network embedding-based link prediction models.
- **Objective 2:** To develop a model for Link Prediction using network embedding and ensemble machine learning classifiers.
- **Objective 3:** To propose a network embedding and deep learning-based link prediction model.

1.3 Problem Statement

There have been numerous studies on predicting the existence of links between nodes in social networks using traditional machine learning and deep learning approaches. These methods primarily focus on achieving high accuracy in classifying whether a potential link exists or not.

“To develop robust, scalable, and efficient Link Prediction methodologies that address the limitations of existing systems by enhancing prediction accuracy through advanced embedding generation and meta-learning-based structural and semantic modeling in social networks.”

1.4 Major Contributions of the Thesis

The research work presented in this thesis makes significant contributions to the field of Link Prediction (LP) in Social Networks, addressing several critical challenges, including data sparsity, cold-start problems, and the integration of structural and attribute-based information. Through extensive experimental evaluation on both benchmark and synthesized network datasets, this thesis provides a detailed analysis of existing LP techniques, identifying their strengths and limitations while highlighting potential areas for enhancement. This analysis serves as a valuable reference for future research in LP, guiding the development of more robust, generalized, and interpretable models for social network analysis.

- A key contribution of this thesis is the development of multiple novel frameworks tailored to real-world LP scenarios. These include meta-learning-based models that integrate centrality-aware transition matrices with deep graph embedding techniques such as Deep Graph Infomax (DGI), as well as ensemble-based methods combining Graph Neural Network (GNN) and machine learning classifiers like CatBoost and LSTM. Furthermore, a unified embedding framework is introduced that combines embeddings generated by Node2Vec, DeepWalk, GCN, GraphSAGE, and GAT into a single, optimized representation using the Attri2Vec model, thereby enhancing link prediction accuracy and robustness across heterogeneous networks.
- The thesis successfully addresses major research gaps, including the challenge of predicting links in sparse or evolving networks, handling cold-start nodes with limited information, and enhancing the quality of learned embeddings through structural and attribute fusion. By proposing innovative, centrality-guided, and meta-learning-based approaches, this work advances the design of LP systems that are more accurate, scalable, and adaptive to complex real-world social net-

works.

- This thesis includes the creation and utilization of synthetic and real-world network datasets to simulate diverse conditions such as sparsity, temporal evolution, and attribute imbalance. These datasets are used to benchmark existing methods and evaluate the proposed models, providing deep insights into their performance, stability, and generalization capabilities.
- The practical implications of this research are broad, particularly in enhancing social network analysis, recommender systems, and community detection. By improving link prediction accuracy, embedding quality, and scalability, the proposed methods contribute to the development of intelligent network analysis systems capable of uncovering hidden relationships, detecting anomalies, and supporting decision-making in large-scale social networks.
- Finally, this thesis outlines future research directions, encouraging exploration into multilayer and temporal link prediction, graph transformer architectures, and self-supervised embedding techniques to further enhance LP performance. Collectively, these contributions lay a strong foundation for advancing link prediction research and its applications in dynamic and heterogeneous social environments.

1.5 Organization of Thesis

- **Chapter 1: Introduction to the LP Field**

The introductory chapter lays the foundation for the research work presented in this thesis. It begins by highlighting the growing importance of intelligent network analysis systems capable of understanding and predicting relationships among entities, particularly in domains such as social media, recommender sys-

tems, biological networks, and communication platforms. The chapter then introduces the domain of LP, emphasizing its significance in uncovering hidden or potential connections within networks and enhancing the understanding of their underlying structure and dynamics. It provides an overview of the key challenges in LP, including data sparsity, cold-start problems, the temporal evolution of links, and the need for accurate and scalable prediction models. The chapter further explores the evolution of LP techniques, from traditional similarity-based and probabilistic methods to modern machine learning and deep learning-based approaches, underscoring the shift toward more automated, adaptive, and robust solutions. By presenting this comprehensive introduction, the chapter establishes the context, motivation, and relevance of the research, setting the stage for the detailed methodologies, analyses, and contributions discussed in the subsequent chapters.

- Chapter 2: **Literature Review**

The literature review chapter provides a detailed examination of existing research in the field of LP in Social Networks, with a particular focus on embedding-based and learning-driven approaches. It explores a wide range of techniques and frameworks, including traditional similarity-based methods, matrix factorization models, and modern graph embedding and graph neural network (GNN) architectures, analyzing their effectiveness in capturing structural and semantic relationships among nodes. Special attention is given to studies addressing critical challenges such as data sparsity, cold-start problems, and dynamic link evolution, and how various models manage these complexities. Furthermore, the chapter reviews advancements in meta-learning and ensemble-based LP techniques, highlighting strategies that enable robust and generalizable prediction across heterogeneous and multilayer networks. By critically analyzing these areas, the chapter identifies existing limitations and research gaps, establishing a

strong foundation for the proposed work presented in this thesis.

- Chapters 3: **GSVAELP: integrating graphSAGE and variational autoencoder for link prediction**

This chapter introduces graphSAGE and variational autoencoder for link prediction (GSVAELP), a hybrid framework that integrates GraphSAGE and Variational Autoencoder Variational autoencoders (VAE) to effectively capture both local neighborhood structures and latent probabilistic features. GraphSAGE is first employed as the encoder to learn node embeddings by sampling and aggregating neighbor information through nonlinear transformations. These embeddings are then passed to a Variational Autoencoder, where the latent variables represent the probabilistic distribution of node interactions. The decoder reconstructs potential links by estimating the likelihood of connections between node pairs. The model is trained by minimizing a reconstruction loss and Kullback–Leibler (KL) divergence, ensuring accurate link inference while maintaining latent space regularization. Experimental results demonstrate that GSVAELP achieves superior performance compared to traditional GNN-based link prediction methods, effectively handling uncertainty, sparsity, and complex topological patterns in social and biological networks.

- Chapter 4: **MetaLP-EC: A Meta-Learning Framework for Temporal Link Prediction with Evolving Centrality and GCN Embeddings**

Temporal networks, unlike static networks, evolve over time as nodes and edges are added, removed, or modified. This dynamic nature is observed in a wide range of real-world systems, including social media platforms, citation graphs, transportation systems, and communication networks. Predicting future links in such networks—known as temporal link prediction—is a challenging problem, as it requires capturing both the structural patterns and the temporal dependencies that influence network evolution. Existing approaches to link pre-

diction have shown promising results in static settings. However, they often fail to adapt to evolving topologies and changing node importance. Many dynamic embedding methods capture temporal patterns but ignore evolving centrality measures, which play a key role in understanding shifts in node influence over time. Moreover, most models lack the ability to generalize across different temporal snapshots without expensive retraining. To address these challenges, this chapter introduces Meta-Learning framework for Temporal Link Prediction (MetaLP-EC)—a Meta-Learning framework for Temporal Link Prediction that integrates Evolving Centrality measures with Graph Convolutional Network (GCN). The proposed model learns from historical snapshots, adapts quickly to new temporal patterns, and incorporates dynamic node influence into the prediction process. By combining structural learning, temporal dynamics, and meta-learning optimization, MetaLP-EC aims to provide accurate and efficient link prediction in evolving network environments.

- Chapter 5: **MetaLP-DGI: Meta-Learning-Based Link Prediction With Centrality-Aware Deep Graph Infomax Embeddings**

This chapter presents Meta-Learning-Based Link Prediction model (MetaLP-DGI), which integrates centrality-aware Deep Graph Infomax DGI embeddings with a meta-learning framework for enhanced structural representation in complex networks. The proposed model begins by constructing a centrality-guided multi-step connectivity matrix to preserve both local and global structural dependencies among nodes. Unlike traditional DGI, which treats all nodes uniformly, the modified DGI in MetaLP-DGI incorporates node centrality measures to adaptively weight node importance during embedding generation. A meta-learning-based feature extraction module further refines these embeddings, enabling the model to generalize efficiently across different network topologies. Finally, an ensemble classifier (CatBoost) is trained on the learned representations to predict

potential links with high precision. Experimental results on benchmark datasets demonstrate that MetaLP-DGI outperforms conventional graph embedding-based models in terms of accuracy, AUC, and F1-score, proving its robustness and adaptability. The integration of centrality awareness and meta-learning not only enhances the embedding quality but also contributes to a deeper understanding of node influence and the dynamics of connectivity in social networks.

- **Chapter 6: Link Prediction in Social Networks: A Hybrid Approach with Graph Embedding and Ensemble Learning**

This chapter presents a hybrid link prediction framework that integrates graph embedding algorithms with ensemble learning classifiers. The proposed approach leverages multiple embedding methods to capture diverse structural and semantic patterns in the network, which are then fused and passed to an ensemble of machine learning models for final prediction. By uniting representation learning with ensemble-based decision-making, the framework aims to enhance accuracy, robustness, and adaptability across different social network datasets. While traditional link prediction methods often rely on hand-crafted similarity metrics, recent advances in graph embedding techniques have enabled automated representation learning, capturing both structural and attribute information of nodes. However, embeddings derived from a single model may fail to capture all relevant patterns, leading to suboptimal predictions. On the other hand, ensemble learning offers a robust framework to combine the strengths of multiple models, reducing variance and improving generalization.

- **Chapter 7: Enhancing Link Prediction through Graph Attention Network and Linear Discriminant Analysis**

This chapter introduces a hybrid model that integrates the Graph Attention Network (GAT) with Linear Discriminant Analysis (LDA) to enhance link prediction performance in complex social networks. The proposed framework lever-

ages the attention mechanism of GAT to capture the adaptive importance of neighborhoods, allowing the model to learn rich, structure-aware node representations. These embeddings are then refined using LDA, which performs dimensionality reduction and discriminative feature separation, thereby enhancing class separability between existing and non-existing links. By combining deep attention-based feature extraction with linear discriminant analysis, the model effectively balances nonlinear representation learning and interpretability. Experimental evaluations demonstrate that the GAT-LDA model achieves superior results compared to traditional graph embedding and neural link prediction methods, with improvements in Area under the Curve (AUC), precision, and recall across multiple benchmark datasets. This integration of attention-based learning and discriminant projection provides a robust and efficient framework for understanding and predicting link formation in social networks.

- **Chapter 8: Unified Link Prediction: Leveraging Multi-Algorithm Embeddings**

Attri2VecLink prediction (LP) plays a crucial role in uncovering potential relationships within graph-structured data. However, existing approaches often rely solely on either node attributes or structural topology, limiting their ability to capture comprehensive node relationships. To address this, we propose a UnifiedAttri2Vec-LSTM model that integrates both attribute-based and structure-based embeddings into a unified representation for improved link prediction in static graphs. First, node attribute embeddings are learned using Attri2Vec, while structural embeddings capture neighborhood information. A hierarchical fusion mechanism then combines these representations to generate enriched, unified embeddings. To model complex, position-aware dependencies among embedding dimensions, a Long Short-Term Memory (LSTM) network is employed as the classifier. Although the LSTM introduces additional computational cost,

it effectively captures higher-order correlations across features, enhancing prediction accuracy and generalization. Experimental evaluations on benchmark citation network datasets—Cora, Citeseer, and Pubmed—demonstrate that the proposed model consistently outperforms traditional classifiers such as MLP, Random Forest (RF), and Logistic Regression in terms of accuracy, AUC, and F1-score. The results confirm that the UnifiedAttri2Vec-LSTM framework effectively bridges attribute semantics and structural topology, leading to a more robust and generalizable link prediction model.

- **Chapter 9: Conclusion, Future work and Social impact**

This chapter summarizes the research conducted on link prediction in social networks, focusing on the development of advanced frameworks, such as GSVAELP, MetaLP-DGI, and MetaLP-EC, as well as hybrid approaches that integrate graph embedding and ensemble learning. This reflects the effectiveness of the proposed methods in capturing complex network structures, incorporating node centrality, and improving predictive accuracy through extensive experimental validation on benchmark datasets. The chapter also discusses future work, including the exploration of transformer-based graph models, temporal attention mechanisms, and the integration of heterogeneous and multimodal network data for a more comprehensive link prediction. Finally, it highlights the societal impact of this research in enhancing applications such as friend recommendation systems, knowledge graph completion, and e-commerce personalization, and biological network analysis, ultimately contributing to better connectivity, knowledge discovery, and decision-making in various domains.

Chapter 2

LITERATURE REVIEW

This chapter provides a brief literature overview, describing the most relevant and prominent state-of-the-art link prediction approaches and methods using a unified notation. We also discuss the critical points at which existing methods succeed and fail in predicting future links. A large number of link prediction techniques have been proposed in the specialized literature. These techniques differ in various aspects, including the evolution rules they model, their computational complexity, or the type and amount of information they consider.

Table 1 presents the various methods in their respective categories of machine and deep learning applications, a summary of different approaches; pros and cons are discussed in the table below. For comparative analysis only, a few approaches performing well in their categories are discussed.

2.1 Previous works in Link Prediction

Research on link prediction in social networks has undergone significant evolution over the past two decades. The earliest foundations were laid by sociologists, long before the term “link prediction” became popular in computer science. Early social network studies in the 1970s and 1980s—such as works on social proximity, triadic closure, and homophily established the idea that individuals with shared characteristics or mutual

friends are more likely to interact with one another. These sociological principles later became the backbone of computational link prediction.

2.1.1 Early Computational Era (2003–2010): Similarity-Based Methods

The formal introduction of link prediction as a computational problem is often attributed to Liben-Nowell and Kleinberg (2003), who analyzed evolving social networks and proposed that future links can be estimated using the structural patterns of the network. Their work popularized a variety of similarity-based indices, including:

1. **Common Neighbors CN:** This similarity metric is a local measurement that depends on the number of friends that a particular pair of nodes has in common. In other words, it evaluates the similarity of the nodes by considering the number of connections they share[12].
2. **Jaccard Coefficient JC:** Data mining and machine learning use the Jaccard coefficient, also known as Jaccard similarity, as a statistic to determine how similar two sets are. The size of the intersection between the two sets is calculated by dividing the size of their union[14]. The Jaccard coefficient, which quantifies overlap between two sets in more straightforward terms, is frequently used to gauge how similar two sets are to one another.
3. **Adamic–Adar:AA** similarity measure is commonly used in network analysis for both link prediction and community detection. It was introduced by Lada Adamic and Eytan Adar in their 2003 paper titled "Friends and Neighbors on the Web[13]." AA is based on the concept that nodes in a network are more likely to be connected if they share many common neighbors. The measure calculates the similarity between two nodes by summing the inverse logarithm of the degree of each common neighbor they share. Essentially, AA quantifies the similarity of

the "neighborhoods" surrounding two nodes.

4. **Preferential Attachment PA:** A mechanism known as preferential attachment occurs in networks when nodes have a propensity to link with other nodes who already have a large number of connections. This process, sometimes referred to as the "rich get richer" effect, leads to the formation of densely connected hubs. Simply said, nodes with more links have a higher chance of attracting new linkages. The Barabási-Albert model can be used to mathematically characterize this process [15].
5. **Resource Allocation RA:** In order to forecast links in different networks, Tao Zhou, Liyuan Lü, and Yi-Cheng Zhang developed the RA[17]. The method considers the similarity of nodes and their ability to share resources with one another, enabling efficient resource allocation. This strategy is employed by the algorithm to identify potential connection formations based on the distribution of resources throughout the network.

Through a personalized survey and field study, [43] introduced a common-neighbor-based connection prediction problem in social networks, known as Friend of a Friend (FOAF). Their research revealed that Likelihood-based approaches are essential for learning new connections. However, a significant limitation of these methods is their exclusive dependence on local information, which restricts the potential links to nodes two or more neighbors apart [44]. Kumar et. al [45] proposed a supervised learning framework that includes various node centralities like Betweenness centrality, Katz centrality, and Betweenness centrality paired with Light Gradient Boosting Machine to predict the missing links. It is a form of ensemble learning technique that amalgamates multiple weak models—usually decision trees—to produce a powerful predictive model.

Li et al.[46] proposed a deep-dynamic embedding method for link prediction in evolving networks. The paper addresses the problem of network embedding in dy-

dynamic networks. The approach utilized historical data from network snapshots at past timestamps to acquire the latent representation of the future network. Anand et al. [18] proposed a method that integrates node centralities and similarity measures with machine learning classifiers, such as the Random Forest RF, AdaBoost classifier AdaBoost classifier (ADB), and ANN-based classifier. The proposed method incorporates four distinct node centrality measures, five similarity measures and three machine learning classifiers like external coreness centrality, page rank, gateway local rank (GLR), and Local Information Dimensionality (LID), the similarity measures used common neighbors, cosine similarity, Leicht-Holme Newman (LHN2) and Matrix Factorization Index (MFI) are fed into the random forest, Adaboost classifier and Artificial neural network. The network's local, semi-local, and global structures were better captured using the combination of node centralities and similarity measures. Elkabani et al.[46] proposed a method based on homophily, using rough set theory, on the Facebook social network dataset. The method is used for dealing with uncertain, imprecise, or incomplete information in classification tasks. The rough set theory facilitates the rapid selection of the optimal subset of attributes, thereby improving the efficiency of the predictor. In many real-world networks, particularly social networks, numerous links exist between two hops.

Global Likelihood-based approaches, such as SimRank and Random Walk with Restart (RWR), extend beyond evaluating node similarity within two hops by utilizing broader graph structures to determine node relationships. SimRank measures similarity based on common third nodes. On the other hand, Quasi-local methods require less global information but consume more data for similarity assessments. As a result, these methods have been introduced to balance local and global metrics, consume less time and complexity, and produce superior outcomes. The most common examples of quasi-local approaches are LPI and Random walk-based methods [47]. Li et al. [48] proposed a link prediction model that integrates local structural information and node degree information along 3-hop paths to enhance the prediction accuracy of existing

local likelihood-based methods. [49] proposed a generalized assortativity measure for complex networks, which analyzes the local assortativity in weighted and directed networks. The researcher also highlighted the importance of individual vertices and edges in contributing to the network's overall assortativity. The latest research [31] investigates an extended self-representation model (ENSR) that calculates the likelihood of a link using an adjacency matrix. The model ENSR adaptively discovers the underlying link predictions of the formation mechanism of complex networks.

2.1.2 Probabilistic and Matrix-Factorization Methods (2010–2015)

As social networks became larger and more complex, researchers introduced more sophisticated statistical models. Methods such as:

1. **Probabilistic graph models (Stochastic Block Models, Hierarchical Random Graphs):** SBMs assume that nodes belong to latent communities (blocks), and the probability of a link depends only on the communities of the two nodes. For each node i , has a latent community label $z_i \in \{1, \dots, k\}$. The probability of an edge between nodes i and j depends on a block affinity matrix θ . Where $A_{ij} = 1$, if an edge exists, 0 otherwise

$$P(A_{ij} = 1 | z_i = r, z_j = s) = \theta_{rs} \quad (2.1)$$

2. **Matrix factorization and latent space models:** Matrix factorization assumes that the adjacency matrix can be approximated by a low-rank combination of latent factors. Each node i is embedded into a latent vector $u_i \in \mathbb{R}^d$. The probability of a link depends on the similarity of these latent vectors.
3. **Bayesian relational models:** These models treat the existence of an edge as a random variable governed by Bayesian priors and posterior inference. The

Edge probabilities depend on latent features of nodes, which are given prior distributions.

These techniques attempted to uncover hidden patterns and latent social dimensions that drive link formation. Although more powerful than basic similarity measures, they often suffer from scalability issues when applied to large online social networks.

2.1.3 Representation Learning and Network Embedding Era (2014–2019)

A major breakthrough occurred with the introduction of network representation learning, where nodes were embedded into low-dimensional vector spaces. Techniques such as:

1. DeepWalk (2014): DeepWalk is one of the earliest and most influential methods for network embedding. It treats random walks on a graph the same way natural language processes treat sentences. Each node becomes a “word,” and a random walk produces a “sentence.” These sequences are then fed into the Skip-Gram model to learn embeddings.
2. LINE (2015): Large-scale Information Network Embedding is designed for very large graphs (millions of nodes). It focuses on preserving two key types of proximity.

- First-Order Proximity: direct adjacency/edge weights

$$\min = \sum_{(u,v \in E)} w_{uv} \log \sigma(u, v) \quad (2.2)$$

Where u, v are the embedding vectors of nodes u and v . The Equation represents the two directly connected nodes in the graph.

- Second-order Proximity: Similarity of neighborhood distributions

$$\min = \sum_{(u,v \in E)} w_{uv} \log \sigma(\mathbf{u}', v) \quad (2.3)$$

Where $\mathbf{u}'$ is the context embedding of u . Second-order uses context embeddings to capture neighborhood similarity.

3. Node2vec (2016): It extends DeepWalk by introducing a biased random walk mechanism. This enables node2vec to capture both homophily (nodes with similar properties tend to be close to each other) and structural equivalence (nodes with similar roles tend to be close to each other).

$$P(c|t, v) = \begin{cases} \frac{1}{p}, & \text{if } c = t, \\ 1, & \text{if } c \text{ is connected to } t \\ \frac{1}{q}, & \text{otherwise.} \end{cases} \quad (2.4)$$

Where $P(c|t, v)$ is the probability of moving from the current node v , to a candidate node c , given that the previous node in the walk was t . The parameters p and q control the likelihood of returning to t and the degree of exploration, allowing the walk to balance between BFS and DFS behavior.

4. SDNE (Structural Deep Network Embedding): It uses deep autoencoders to capture both first-order and second-order proximities. It is one of the earliest deep learning approaches for graph embedding. The core idea of this approach is to utilize the autoencoder to reconstruct adjacency information.
5. HOPE (High-Order Proximity preserved Embedding): HOPE is designed to preserve asymmetric transitivity, which is important in directed graphs such as citation networks or follower–follower networks. It uses matrix factorization of similarity matrices that capture high-order proximities (e.g., Katz index, Rooted

PageRank).

$$S = \sum_{k=1}^{\infty} \beta^k A^k \quad (2.5)$$

Where S is the similarity matrix, which captures all possible multi-step relationships between nodes in the graph. β is the decay vector (between 0 and 1) that reduces the influence of longer paths, and A is the adjacency matrix. The term A^k represents the number of paths of length k between two nodes.

Chen et al. [50] proposed a Nonnegative Matrix Factorization via Asymmetric link clustering and PageRank model (NMF-AP), which utilizes PageRank to capture the global network structure and asymmetric link clustering to preserve the local structure, and jointly optimizes these in the matrix factorization model. Whereas Mo et al. [51] integrated temporal and topological information for effective link prediction in dynamic networks. The approach utilized a Graph-Wavelet Neural Network (TT-GWNN), which integrates temporal and topological information through a second-order balanced sampling algorithm based on a random walk. This technique highlighted the importance of considering temporal and topological factors for precise link prediction in dynamic networks.

Rozemberczki et al. [52] proposed multi-scale embedding algorithms that capture the information about the node from the local distribution of node attributes around it. These algorithms aim to efficiently map units from different modalities into a common latent space.

J. Chen et al. [53] proposed an ensemble learning model based on graph embedding for link prediction. The model addresses the challenge of representing the latent structure. They designed a two-stage link prediction model based on the three graph embedding models and multi-classifiers. This method also introduced a self-supervised learning strategy to utilize the graph attribute information for graph-structure learning. In recent years, graph neural network (GNN) techniques have revolutionized net-

work analysis tasks, including link prediction, by effectively capturing the complex relationships between nodes and graph structures. They learn rich node embeddings through iterative neighborhood aggregation, which enhances the accuracy of predicting future links. Graph Neural Networks (GNNs) are particularly effective at handling vast, dynamic, and heterogeneous networks, making them well-suited for various network analysis tasks. Examples of GNNs are Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs) [44].

Chen [54] proposed the GC-LSTM, a deep-learning model that combines Graph Convolutional Network and long-short-term memory for dynamic network link prediction. It is designed to handle the challenges of predicting links in the network that change with time. GC-LSTM uses GCN to learn the node structure of the network and LSTM to learn the temporal features. Another deep learning model was proposed by Meng et al. [55], an innovative embedding learning model called hyper-motif of the network (LEHMN) to optimize the link prediction process. They introduce the depth and breadth motif of the random walk strategy and apply this model to an open and closed hyper-motif structure to emphasize its effectiveness. Using the concept of the network's topology to forecast weak links, Zhou et al. [56] presented a neural-network-based method called weak link prediction based on hyper-latent distance (HLD). The pair-wise long strength in the embedding space is computed.

2.1.4 Multiplex, Multilayer, and Temporal Networks (2018–present)

In a multilayer network, Wang et al.[57] proposed an attention layer shareable method for link prediction, which consists of a feature encoder, a knowledge learner, and a fusion predictor. By varying the significance of structural information for each link, the feature encoder introduces several attention mechanisms to encode the feature representations of links. Whereas Jiang et al. [58] proposed a temporal link prediction method with transfer learning, aiming to leverage knowledge learned from historical

time snapshots or related networks to improve prediction accuracy in evolving graphs. Nie et al. proposed a local optimization policy for link prediction using reinforcement learning, where the link formation process is modeled as a sequential decision-making problem. An agent explores the network by learning optimal transition actions that maximize long-term rewards associated with discovering missing or future links. This approach effectively captures local structural patterns and adapts dynamically to complex network environments. On the other hand, Qiu et al [59] proposed a temporal link prediction that leverages recurring higher-order subgraph patterns to model the evolution of social networks over time. Zhao et al. [60] proposed Multi-Correlation Features-based Link Prediction in Multiplex Networks Multi-Correlation Features-based Link Prediction in Multiplex Networks (MCFMN-LP), which exploits multiple relational features to improve prediction performance. By jointly integrating interlayer correlations, nodal attributes, and community-level relationships, the method captures the complex hierarchical dependencies and community structures that exist across layers. This comprehensive feature fusion enables a deeper understanding of multiplex network dynamics, leading to more accurate and robust link prediction results.

Table 2.1: Comparison of Link Prediction Methods with Characteristics and Limitations

Author (Year)	Method	Characteristics	Limitation
Li et al., 2013[61]	Graph-kernel based recommendation using associative interaction graphs (AIG)	AIG captures topologically similar users and items with similar usage behaviours and access characteristics. Novel graph-kernel designed to classify user-item pairs using their AIG.	May not be suitable for all types of recommendation systems or large-scale datasets.
Fire et al., 2014[62]	Friend measure and the same community feature-based method with ML classifiers	Framework for constructing efficient classifiers based on features that are easy and fast to calculate.	Experiments performed on limited datasets, unclear generalizability.
Perozzi et al., 2014[27]	DeepWalk (unsupervised ML)	Learns latent representations of vertices in a network, handles large and complex graphs.	Ignores attributed information, cannot handle new nodes/edges without historical data.
Ibrahim et al., 2015 [32]	Temporal, community structure and node centrality-based LP	Predicts social network behavior over time, useful for understanding dynamics.	May not apply to all social network datasets.

Author (Year)	Method	Characteristics	Limitation
Elkabani et al., 2015 [63]	Rough set theory classification-based LP	Demonstrates effectiveness of homophilic features like common groups/page likes.	Only considers a limited set of homophily; it ignores other features like location or age.
Grover et al., 2016 [28]	node2vec based LP	Learns continuous feature representations for nodes.	Static embedding; ignores temporal dynamics.
Dai et al., 2017[64]	Belief propagation + similarity + nonnegative matrix factorization	Calculates belief vectors, measures influence between relation types.	May not work in real-world scenarios; depends on network-specific characteristics.
Chen et al., 2019 [65]	E-LSTM for dynamic link prediction	End-to-end deep learning model: Uses an encoder–decoder architecture with LSTMs to model temporal dependencies in dynamic networks.	Performance depends on network size and complexity.

Author (Year)	Method	Characteristics	Limitation
Li et al., 2020 [66]	EMLP (Similarity + LR + XG-Boost)	Integration of Diverse Features. By combining multiple predictions, the ensemble reduces the effect of errors or biases that are specific to individual models.	Works on small datasets, Reduced Interpretability, High computational cost.
Zareie et al., 2020 [67]	DICN: Similarity-based LP with direct/indirect neighbors	Considers latent relationships via second-order neighbors.	Unclear performance in other contexts; complexity not discussed.
Kumar et al., 2021 [8]	NC-LGBM based LP	Combines node centralities with LGBM classifier for better LP performance.	Feature Dependency and Generalizability .
Berahmand et al., 2021 [5]	CSADW based LP	Modified DeepWalk improves LP in attributed networks.	Very high time complexity.

Author (Year)	Method	Characteristics	Limitation
Singh et al., 2021 [41]	PILHNB (popularity, interests, location + HNB)	Feature integration: Combines multiple social factors—popularity, user interests, and location—to improve link prediction. .	1. Feature dependence assumption. 2. Limited scalability. 3. Temporal granularity.
Anand et al., 2022 [68]	NSMLLP (Node centrality + similarity + ML classifiers)	1. It captures local, global, and topological features for LP 2. Improved accuracy: ML classifiers learn from combined features, outperforming single-measure heuristics.	1. High feature engineering effort. Requires careful selection and computation of multiple centralities and similarity indices. 2. Scalability issues. Centrality computation can be costly in large-scale networks.
Mishra et al., 2022[69]	ELP (Ego perspective LP)	Computes ego strength and likelihood score for target links.	Limited performance, small datasets, ignores node count in initial info.

Author (Year)	Method	Characteristics	Limitation
Tripathi et al., 2022[70]	Skip-gram edge embedding + LinkVec + LGBM	Temporal representation learning: Learns low-dimensional embeddings that preserve both structural information and temporal dynamics of evolving networks	1. Cold-start problem: Performance drops for new nodes with little interaction history. 2. Limited interpretability: Embeddings act as black boxes, making it hard to explain predictions.
Chen et al., 2022[38]	Graph embedding + ensemble ML classifiers	1. Transforms network into a low-dimensional space while preserving structure. 2. Ensemble learning: Combines multiple classifiers (e.g., Random Forest, SVM, Neural Networks) to improve prediction accuracy.	1. Embedding quality dependency: Performance heavily depends on the effectiveness of the chosen embedding techniques. 2. Limited interpretability: Ensemble + embeddings reduce transparency of prediction results.

Author (Year)	Method	Characteristics	Limitation
Barracchia et al., 2022[71]	LP-ROBIN (incremental node embedding)	Learns hidden patterns from latent representations to find new links over time.	1. Embedding quality degradation – Incremental updates may accumulate errors over time, leading to less accurate embeddings compared to full retraining. 2. Cold-start problem – Struggles with newly added nodes or edges that lack sufficient historical information.
Liu et al., 2022[11]	Hybridized community detection	1. Inductive Matrix Completion (IMC) framework – Utilizes side information (e.g., node attributes, metadata) along with network structure to predict missing links.	1. Computational overhead – Solving matrix completion with side information can be expensive for very large-scale networks. 2. Static assumption – Primarily designed for static networks, limiting applicability to dynamic or temporal link prediction tasks.

Author (Year)	Method	Characteristics	Limitation
Zhao et al., 2022 [72]	ICP (inductive matrix completion) + low-rank + bilinear	Predicts connection probability between nodes.	Feature construction is expensive; not ideal for highly dynamic networks; requires node features.
Liu et al., 2023 [73]	GNN + Capsule Networks (CapsNet)	Hybrid architecture Combines Graph Neural Networks (GNNs) for structural feature learning with Capsule Networks for capturing hierarchical relationships among features.	1. Overfitting risk: Increased model parameters may lead to overfitting, especially on small or sparse networks. 2. Limited temporal handling: Primarily designed for static graphs.
Wang et al, 2023[74]	MTTM (multi-type transferable method)	Uses a generative predictor and a discriminative classifier in a mini-max game.	Limited description of heterogeneous structure.

Author (Year)	Method	Characteristics	Limitation
Li et al.,2023[39]	Meta-HGT	Metapath-aware em-beddings capture heterogeneous re-lationships, while transformer attention enhances information aggregation across nodes and metapaths.	1. Limited temporal modeling: Focused on static HINs, with weak adaptability to dynamic or evolving heterogeneous networks. 2. Metapath dependence: Model effectiveness is influenced by the quality and selection of metapaths; poor metapath design may degrade performance.
Dong Shu et al, 2024[75].	KG-LLM for LP	Improves multi-hop LP by converting KG into prompt format.	1. High Computational Cost – Training and inference with LLMs alongside KGs is resource-intensive. 2. Hallucination Risk – LLMs may generate semantically plausible but incorrect links.

Author (Year)	Method	Characteristics	Limitation
Xiu et al.,2024 [31]	ENSR model	Uses matrix factorization to capture latent structures and similarities, with an extended formulation that better preserves global patterns.	1. Static Network Focus – Limited applicability to dynamic or evolving networks. 2. Parameter Sensitivity – Performance is highly dependent on regularization and optimization choices.
Si et al.,2024[34]	A link prediction method for multiplex social networks using information transmission approach, namely ITMLP	The proposed ITMLP model predicts links by measuring node similarity through the total information transmitted between them and aggregating these similarity scores across all layers via information transmission.	1. Scalability – High computational cost on large multiplex networks. 2. Dynamics – Limited ability to model evolving temporal links.

Author (Year)	Method	Characteristics	Limitation
Cai al.,2024[35]	A novel transition matrix based on a similarity metric based on reliable local random walk (BEC-RLRW).	BEC-RLRW introduces a reliable local random walk for multiplex similarity in multilayer social networks, followed by node feature extraction using node2vec embeddings.	1. Embedding Dependency – Performance heavily depends on the quality and parameter tuning of node2vec embeddings. 2. Limited Attribute Integration – Focuses mainly on structural features; node or edge attributes are not fully leveraged.
Nie et al 2025[76]	ASLAM (attribute + neighborhood attention)	Constructs semantic + structural representations with attention mechanisms.	Depends heavily on attribute quality.
Zhou et al.,2025[60]	MCFMN-LP (multilayer correlation features)	1. Uses intra-/inter-layer similarities for LP. 2. Captures complementary information from multiple layers to enhance prediction accuracy.	1. Generalizability and interpretability concerns. 2. High computational complexity when handling many layers and features. 3. Limited scalability on large-scale multilayer networks.

2.1. Previous works in Link Prediction

Author (Year)	Method	Characteristics	Limitation
Garompolo et al.,2025[40]	GCN-LSTM for SIoT	Predicts both link formation and dissolution using spatial-temporal features.	Risk of overfitting on sparse/limited data.
Karami et al., 2025 [42]	SEAL+ A subgraph-enhanced framework for link prediction with graph neural networks	1. Extends SEAL by incorporating enhanced subgraph extraction and node labeling strategies. 2. Utilizes graph neural networks (GNNs) to learn structural patterns within local enclosing subgraphs.	1. Scalability issues on very large-scale networks with millions of nodes/edges. 2. Prediction quality highly depends on the subgraph sampling strategy.

Chapter 3

GSVAELP: integrating GraphSAGE and Variational Autoencoder for Link Prediction

This chapter introduces a deep learning-based link prediction model in social networks by leveraging graphSAGE (graph sample and aggregation) and Variational Autoencoders (VAE). The proposed work starts by utilizing graphSAGE to generate node embeddings of the input network by sampling and aggregating information from neighborhood nodes. The generated embeddings are sufficiently expressive to capture the local and global network structure. Further, we adopt VAE to learn a latent space representation of the graphSAGE embeddings. The VAE helps in refining the node embeddings and learning a meaningful representation (latent space) of the input data, which can be useful for the downstream link prediction task. The encoder's output (latent space) can capture important features that aid in link prediction. Finally, train a logistic regression classifier using the latent representations from the VAE as features to predict the upcoming links in the network. The necessary hyperparameter studies are performed to obtain the optimal values of the various model parameters. The results from experiments and analyses carried out on eight distinct real-world network datasets demonstrate the effectiveness of the suggested link prediction model. The motivation behind integrating GraphSAGE with a Variational Autoencoder (VAE) is to jointly leverage inductive neighborhood aggregation and probabilistic latent represen-

tation learning for link prediction. GraphSAGE effectively captures local structural information by aggregating features from sampled neighborhoods, making it scalable to large social networks. However, deterministic embeddings alone may not adequately model the uncertainty and sparsity inherent in real-world networks. The VAE component addresses this limitation by learning a smooth latent space that captures uncertainty and enables better generalization. Their integration allows GSVAELP to learn robust, uncertainty-aware node representations, leading to improved link prediction performance.

3.1 Background

Link prediction (LP) entails the endeavor of foreseeing potential or new links among nodes within a network by leveraging the network’s present structure and characteristics [77]. Link Prediction (LP) is one of the most widely studied research problems in the field of network science and complex network analysis. Complex networks are characterized by their intricate topological structures, often seen in social networks, biological networks, citation networks, transportation networks, and more. Link prediction plays a crucial role in estimating the growth and evolution of complex networks by providing valuable insights into the potential future connections and interactions within the network. LP has been helpful for a variety of tasks including suggestions for friendship in social networks, predicting potential future author collaborations, and enhancing recommendation systems. It finds applications in diverse domains where predicting relationships or links between entities is crucial for decision-making and system efficiency. Link prediction in social networks helps anticipate future connections, aiding in better recommendations and understanding network dynamics for forming online communities and identifying influencers [1, 78]. It has several other applications in various domains of recommendation systems, suggesting relevant items to users that might be interested based on their existing preferences or connections

such as music recommendation, and hotel recommendation [79, 80]. LP can be used in analyzing biological networks like protein-protein interaction, disease-gene associations, and discovering drug targets [81]. Hence, the forecasting of links through link prediction significantly contributes to the improvement of decision-making, efficient resource allocation, and the overall enhancement of user experiences in a wide array of real-world applications [82].

A complex network intuitively can be presented as a graph $G = (V, E)$, where V represents the set of users, people, and entities. E represents the set of present edges or the links and $E \subseteq V \times V$. An edge connects two nodes such as friendship, follow-follower relationships, co-authorship, and other similar relations. Mathematically, the link prediction problem can be stated as finding the function $f(i, j)$ such that

$$f(i, j) = w_{ij}, \text{ for all } (i, j) \in (V \times V) \setminus E \quad (3.1)$$

where w_{ij} is the prediction score representing the likelihood of a link between nodes i and j . Hence, the link prediction model involves devising efficient algorithms and models to estimate w_{ij} accurately, aiding in predicting potential edges in the network. The task of link prediction is viewed as a binary classification problem where we aim to predict whether an edge exists or not. In this case, we can define a threshold θ and classify the pair (i, j) as follows.

$$f(i, j) = \begin{cases} 1, & \text{if } w_{ij} \geq \theta \\ 0, & \text{otherwise} \end{cases} \quad (3.2)$$

if the prediction score is above the threshold $\theta = 0.5$, then the link prediction between the pair of nodes (v_i, v_j) is said to be true, otherwise false.

Real-life networks contain a large number of nodes and edges with intricate relationships between nodes. The traditional representation of the networks as adjacency matrix and adjacency list lack in capturing crucial relationships between nodes and

can be bottlenecks for various network analysis tasks like link prediction. The emergence of various network embedding techniques has taken care of such limitations and become pivotal in diverse analysis tasks. The objective of network embedding is to present a network using low-dimensional vector representations. In this technique, each node is represented as a vector in a reduced dimensional space while retaining the structural information of the network [46]. If an edge between the two nodes is close to one another in a graph, they might have similar vector representation or the distance in a vector space is relatively close. Various graph embedding algorithms have been introduced, the only difference between these embedding techniques is how to define the proximity between the nodes [83]. Typically, first and second-order proximities are commonly utilized metrics for assessing the node similarity[84]. Over the course of time, numerous link prediction methods have been proposed in scholarly works [85, 86]. The methods of link predictions can be broadly classified into three main categories: maximum likelihood-based, similarity-based, and probabilistic-based methods, where recent research has also looked into the potential of deep learning and machine learning techniques. In recent years, machine learning techniques have seen widespread adoption of graph embedding algorithms for addressing the challenges of link prediction [87]. The Traditional autoencoder-based models rely on deterministic encoders and often suffer from overfitting, particularly in sparse networks. GSVAELP improves upon these approaches by employing a Variational Autoencoder, which enforces a probabilistic latent space through distribution learning and regularization. This enables better modeling of network uncertainty and reduces sensitivity to noise. Furthermore, using GraphSAGE as the encoder enables GSVAELP to generalize to unseen nodes, whereas conventional autoencoder-based link prediction models do not.

In this paper, we propose an improved link prediction model that combines graphSAGE for generating low-dimensional node representations and feature extraction, an VAE to refine the generated embeddings for meaningful representation, and a logistic regression classifier for link prediction. The network embedding generation in the

proposed model uses a neighborhood aggregation function that combines information from a node’s neighbors to update its representations. The technique captures both the local and global structure of the network, can generalize to unseen nodes, and does not require the entire network structure during learning. After obtaining the embeddings for every node, the generated embeddings are passed through the VAE for further refinement. The VAE takes the embedding as input and learns to encode it into a low-dimensional space, whereas the decoder aims to reconstruct the original embedding from the compressed representation. Here, the VAE is trained to obtain a latent space representation for the nodes, resulting in the development of a continuous and meaningful latent space that plays a crucial role in the task of link prediction. Finally, we perform a performance comparison of the proposed model based on various evaluation criteria against several baselines and recently introduced models of link prediction using eight different datasets.

3.2 Proposed Methodology

In this section, we discuss the proposed link prediction model, GSVaelP, in detail. The overall work of the proposed model is divided into four phases. In the first phase, feature extraction is done by using the graphSAGE, where embeddings of each node in the network are obtained. Here, the generated node embeddings represent network nodes or entities as low-dimensional vectors and aim to capture the network’s structural and relational information. The second phase involves the utilization of variational autoencoders (VAE) that take input as node embeddings generated by graphSAGE. This phase trains the VAE to learn a latent-space representation of nodes. The VAE helps refine node embeddings and learn a continuous, meaningful latent space. The third phase of the model uses a logistic regression classifier to predict upcoming links in the network. Finally, the necessary hyperparameter studies are conducted to determine the optimal values for the hyperparameters involved in the model’s different phases.

The combination of GSVAELP leverages neighborhood information and node features captured by graphSAGE embeddings, while VAE learns a compact latent space and captures patterns for link prediction. Fig. 3.1 depicts the general architecture of the model. The description of the various phases of our proposed work is provided below.

3.2.1 Feature extraction and generating network embedding using graphSAGE

The network embedding generation in the proposed model uses a neighborhood aggregation function that combines information from a node's neighbor to update its representations. The technique captures both the local and global structure of the network can generalize to unseen nodes and does not require the entire network structure during learning. Rather than considering the complete k-hop neighborhood of a target node, graphSAGE initially samples or prunes the computational graph of the k-hop neighborhood and then performs the feature aggregation procedure to produce the embeddings of the target node.

Consider the input network as $G = (V, E)$, where V is the set of nodes and E is the set of the links or edges in the graph. The node feature matrix $Y \in \mathbb{R}^{|V| \times D}$, where D is the dimension of node features. The adjacency matrix of graph $A \in \mathbb{R}^{|V| \times |V|}$ matrix, if the two nodes u, v are connected, the value of A is equal to 1, otherwise 0. The main purpose of using graphSAGE is to learn the node embedding H_v for each node $v \in V$. For each node v , create a set of neighbors $N(v)$ from its neighborhood, ensuring that each node has the same number of neighbors. For each node v , aggregate the embeddings of its neighbors $H_N(v)$ by using an *AGGREGATE* function represented as Eq 3.3.

$$H_{agg}(v) = AGGREGATE(\{H_u, \forall u, \in N(v)\}) \quad (3.3)$$

After calculating the aggregate embeddings $H_{agg}(v)$ at of the graphSAGE model, then

updating the embedding of node v by using an *UPDATE* function is shown in Eq 3.4.

$$H'_{agg}(v)^k = UPDATE(H_v, H_{agg}(v)) \quad (3.4)$$

$H'_{agg}(v)^k$ is the updated embedding at the $k - th$ layer of the graphSAGE model, and the updated embedding is then passed as input in the next phase of the model.

3.2.2 Variational autoencoder for node embedding refinement

In this step, variational autoencoders (VAEs) are utilized that take input as node embeddings generated by graphSAGE and help in refining the node embedding, leading to learning a continuous, meaningful latent space of nodes. Variational Autoencoders are an extension of traditional autoencoders that incorporate additional probabilistic concepts. The objective of VAEs is not only to capture the latent representation of the network but also to underlying the model data distribution. The architecture of a Variational autoencoder comprises an input layer, an encoding layer, a decoding layer, and an output layer. The encoding layer reduces the dimensionality of the incoming data, and the decoding layer reconstructs the original data from the encoded form. To generate the embedding graphSAGE use the AGGREGATOR function and then update the embedding $H'_{agg}(v)$ of the network and then transfer it into the encoder layer. The Encoder uses the graphSAGE embedding to learn the network structure. At the same time, the Decoder is a layer that is fully connected layer, it transforms the extracted embedding to the original space. GSVaelP will output the predicted social network link prediction. The Linear Autoencoder's encoding layer, sometimes referred to as the encoding space, learns a lower-dimensional representation of the input data. Usually, the latent space's dimensions are less complex than the dimensions of the incoming data. Prior to the model's in-depth description. We discuss some terms and notations that will be used in the paper.

3.2.2.1 Encoder

An encoder layer takes the graphSAGE embedding $H'_{agg}(v)^k$ as input. It is responsible for transforming the high-dimensional input layer into a lower-dimensional encoding that captures the most essential features of the data. An encoder framework is designed that takes the graphSAGE embedding as input and produces the mean μ and the log-variance $\log(\sigma)$ as outputs. The mathematical equation of the encoder is presented as Eq. 3.5 and Eq.3.6

$$\mu, \log(\sigma) = \text{Encoder}(H'_{agg}(v)^k) \quad (3.5)$$

$$Z = \mu + \epsilon \odot \sigma, \text{ where } \epsilon \sim N(0, I) \quad (3.6)$$

where Z represents the latent variable vector obtained from the latent variable distribution, ϵ is a random sample from a standard distribution ($N(0, I)$), where I is the identity matrix from a normal distribution with mean 0 and variance 1. Now we design a decoder framework that takes Z as input and generates the reconstructed representation of the graphSAGE embeddings.

3.2.2.2 Decoder

The decoder is responsible for transforming the encoded representation into the original data space. The decoder converts the latent representation Z into the original graphSAGE Embedding $H'_{agg}(v)^k$

$$H''_{agg}(v)^k = \text{Decoder}(Z) \quad (3.7)$$

$H''_{agg}(v)^k$ is the reconstructed embedding for each node v in the $k - th$ layer of the graphSAGE model. Decoder is a function that takes the sampled latent variable Z as input and attempts to reconstruct the original embeddings $H'_{agg}(v)$. Its purpose is to generate the reconstruction of the input based on the sampled latent variables.

3.2.3 Training Classifier for Link Prediction

After getting the latent representation from VAE, the proposed model exploits a binary classifier for link prediction. For the classification task, use logistic regression to improve the prediction ability of our proposed work. The input for logistic regression consists of a pair of nodes with their corresponding latent representation. For a pair of nodes v_i and v_j , the input features would be $Z_i - Z_j$, where Z_i, Z_j are the latent representation obtained from VAE model.

$$P(y_{ij}) = 1 = \frac{1}{1 + e^{-(w^T(Z_i - Z_j) + b)}} \quad (3.8)$$

The regression model predicts the probability of a link between the node v_i and v_j , where y_{ij} is the binary label indicating, whether there is a link between nodes v_i and v_j , w are the weights and b is the bias term. In the regression prediction problem, the L_2 regularization term is usually used to measure the discrepancies between the predicted probabilities and the actual labels.

$$L_2 = ||P(y_{i,j}) - A(y_{i,j})||^2 = \sum_{i=0}^V \sum_{j=0}^V (P(i,j) - A(i,j))^2 \quad (3.9)$$

where P represents the probability matrix, $A(y_{i,j})$ is the adjacency matrix, and V is the number of nodes in the network. However, if we utilize only L_2 distance as a loss function, it may cause biases towards zero, especially in cases where the network exhibits sparsity. To avoid overfitting, we incorporate an additional regularization term, L_{reg} . The overall loss during the training process is defined as:

$$L_{total} = L_2 + \beta L_{reg} \quad (3.10)$$

where β is the parameter that balances the importance of the two terms L_2 and L_{reg} . The regularization loss L_{reg} encourages the model to keep the weights of the parame-

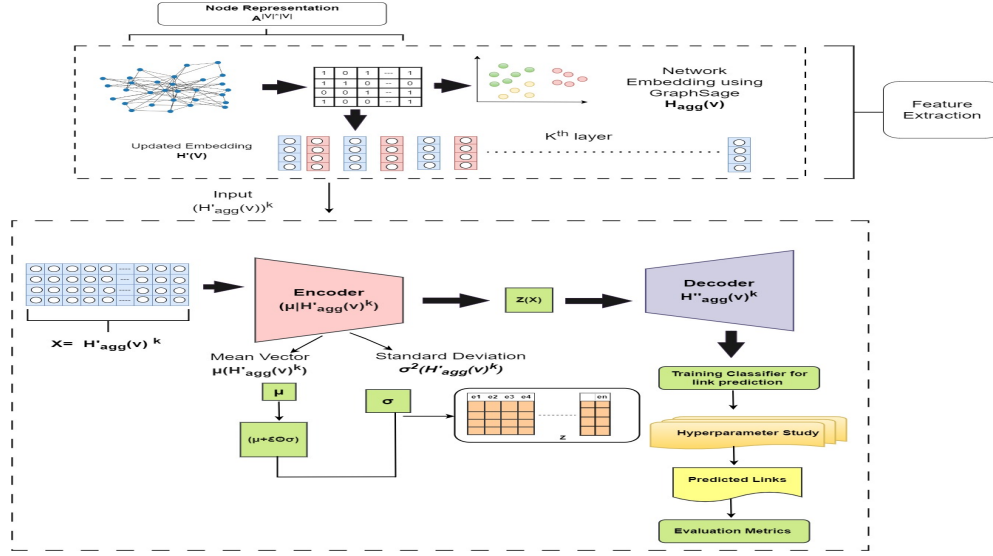


Fig. 3.1: Architecture of proposed methodology

ters small, which helps to prevent the overfitting problem, whereas, the L_2 represents the sum of the square of the GSVAELP model weights and with the help of Eq.3.2 we calculate the prediction score between the nodes $\theta(V_i, V_j)$, if the prediction score is above or equal the value of threshold, then link will be predicted. To measure the effectiveness of the GSVAELP model, utilize the appropriate evaluation metrics as discussed in the next section.

$$Prediction = \begin{cases} 1, & \text{if } w_{ij} \geq \theta \\ 0, & \text{otherwise} \end{cases} \quad (3.11)$$

The related discussion of the technicalities mentioned in the proposed methodology is presented as follows.

3.3 Experiments and analysis

In this study, we utilize five diverse network datasets, including the social network and citation network, to evaluate the performance of the proposed model. The details of the datasets are described as follows.

1. Caltech36 [88]: It is a Facebook friendship dataset, made up of Caltech, Pasadena students. Here, each node signifies an individual, while the edges denote the friendship or social ties within the network. The dataset encompasses 769 nodes and approximately 16,656 edges.
2. PolBlogs [89]: It is a Twitter network that shows users' retweet activity in the 2010 U.S. congressional elections. Retweets are represented by the edges, while Twitter users are represented by the nodes. There are 61157 edges and 18470 nodes in the network.
3. DBLP [90]: It is a citation network dataset. The citation data is sourced from a variety of databases including DBLP (Digital Bibliography and Library Project), ACM, and other relevant sources. Each node represents a written piece, and each edge symbolizes a reference connecting these written pieces. There are 49743 edges and 12591 nodes.
4. Fb-TV Show [91]: The Dataset represents a social network derived from Facebook, and shows the links among different TV program pages on the platform. In this dataset, nodes represent the TV program pages, whereas the edges show the instances of shared links between these pages. The dataset comprises 17260 edges and 3892 nodes.
5. Delaunay-N12[92]: This network dataset falls under the DIMACS10 category. These files were generated by triangulating random locations in the unit square using the Delaunay method. It consists of 4100 nodes and 12300 edges.
6. Public figures [53]: This dataset is derived from Facebook and serves as a social network representation. The nodes in this network correspond to public figures, and the edges depict their interactions. It encompasses a total of 11,565 nodes and 67,036 edges.

Table 3.1: The Statistical Description of various Dataset used

Dataset	$ V $	$ E $	$\langle k \rangle$	$\langle cc \rangle$
Caltech	769	16656	43	0.409
Polblogs	18470	61157	6	0.059
DBLP	12591	49743	7	0.0119
FB-TV Show	3892	17260	8	0.373
Delaunay-N12	4100	12300	5	0.434
Hamsterster	2426	16630	13	0.537
Public Figure	11565	67036	11	0.1793
Wiki-Vote	889	2914	7	0.1409

7. Hamster [93]: It is a popular social network dataset, a representation of the interaction between the users, the nodes denote the users whereas the edges correspond to the link between the users. It consists of 2426 nodes and 16630 edges.
8. Wiki [94]: This dataset comprises voting data sourced from Wikipedia. In this network, the nodes represent Wikipedia users, and the edges illustrate the voting interactions between these users. The dataset encompasses 889 nodes and 2914 edges.

A brief statistical description of the datasets is discussed in Tab. 3.1 where $|V|$ represents the total number of nodes, $|E|$ shows the total number of links, k represents the average degree of the nodes, and cc denotes the average clustering coefficient.

3.3.1 Evaluation Metrics

To evaluate the effectiveness of the proposed model, we use distinct classical evaluation metrics like precision, recall, F1-score, AUC (Area under the curve), and accuracy. Below is a detailed description of these metrics.

1. Precision: The precision measures the positive instances of the class. It is an indicator to measure the shared positive classifications that were true. It is also

known as positive predictive value. The mathematical formula of precision is:

$$Precision = \frac{TP}{TP + FP} \quad (3.12)$$

Where TP represents the correctly predicted links in the network whereas FP denotes the falsely predicted links.

2. Recall: Recall is used to measure the model's ability with the help of detecting the positive samples. It is a ratio between the number of positive samples correctly to the number of positive samples. The mathematical representation of recall is followed as:

$$Recall = \frac{TP}{TP + FN} \quad (3.13)$$

TP denotes the True positive, which classifies the positive samples as positive, whereas FN denotes the false negative means, which classifies the positive sample as negative.

3. F1-score: The F1-score measures the Harmonic mean of precision and the recall. The mathematical formula of the F1-score is expressed as :

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3.14)$$

4. AUC: The Area under the receiver operating characteristics curve (AUC). It is a graphical representation of the true positive rate versus the false positive rate. The range of AUC varies from 0.0 to 1.0. A higher AUC score indicates a higher model's accuracy. AUC can be defined as :

$$AUC = \frac{M' + 0.5M''}{M} \quad (3.15)$$

M denotes the independent comparisons, M' is the occurrence of the missing links, M'' is the occurrence of missing links and non-existent links in the network and the value 0.5 is the discrimination threshold.

5. Accuracy: Accuracy is calculated as the proportion of correct predictions divided by the total number of predictions made. It is an important evaluation metric to measure the effectiveness of the model. The mathematical expression of accuracy is:

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \quad (3.16)$$

3.4 Experiment and Result Analysis

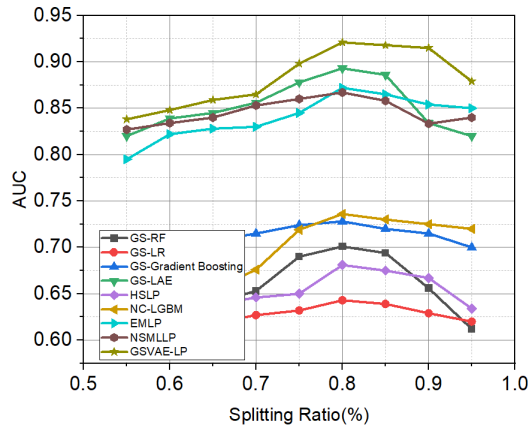
This section provides experimental results and analysis using the datasets and evaluation metrics. We compared the performance of the proposed model with some baseline methods of link prediction using classical network embedding techniques such as deepwalk [44], node2vec [43], and Line [95]. The link prediction methods using classical baseline machine learning classifiers such as Random Forest (RF), Logistic Regression (LR), Gradient Boosting (GB), and Linear Autoencoders (LAE) are also used as comparing methods in this analysis. Before performing binary classification tasks for link prediction (LP) using the above-mentioned machine learning classifiers, the graph-SAGE embedding is used for feature representation or latent representation of nodes of the network. We named these ML classifiers based LP method RFLP (random forest-based link prediction), LRLP (logistic regression based LP), GBLP (gradient boosting LP), and LAELP (Linear Autoencoders based LP). Apart from these comparing methods, we evaluated the performance of our proposed model against some recently popular methods of link prediction namely, HSLP [84] (H-index-based link prediction), NC-LGBM [96] (Link Prediction using node centrality and light gradient boosting

machine), EMLP [93] (Ensemble-model based link prediction) and NSMLLP [94] (integrating node centralities, similarity measures, and machine learning classifiers based LP). Further, we performed various hyperparameter studies such as the choice of splitting ratio for training and testing the models and various model parameters to find out their optimal values.

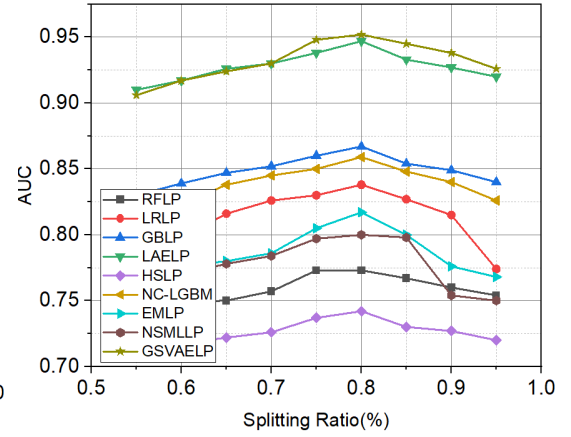
3.4.1 Choice of Splitting Ratio

In this section, we examine the optimal ratio for dividing the dataset into training and testing phases. To analyze this, we performed experiments on the splitting ratio to evaluate the performance of AUC metrics for each dataset, for various baseline methods, existing methods, and our proposed method. In the conducted experiments, the splitting ratio ranges from 50% to 95% in increments of 5%. The results we have acquired are shown in Fig.3.2. The x-axis value signifies the dataset's splitting ratio, while the y-axis denotes the corresponding AUC (Area Under the Curve) value. It is evident that the dataset splitting ratio of 80 : 20 is ideal for training and testing the various methods of LP considered in this study. Hence, 80% of the data is dedicated to training, leaving the remaining 20% for evaluating the model's performance. The decision to divide the dataset in this way ensures that there will be sufficient data for properly training the classifier while also avoiding the overfitting problem.

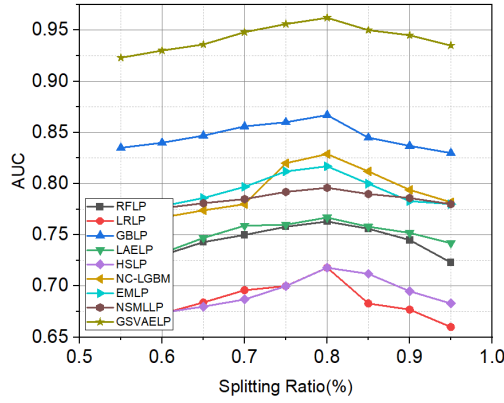
Figures show that with an increasing splitting ratio, the AUC value rises across all datasets, peaking at approximately 80%. This indicates that when the splitting ratio reaches 50%, the data is in the under-lifting situation. In other words, at lower splitting ratios, there is not enough data available to train the dataset efficiently, and the proposed technique struggles to fully capture the network's features. So, based on these findings, it is usually clear that AUC performance is better as the split ratio rises from 50% to 80%, with the optimum performance being at 80%. Additionally, it has been seen that the AUC value begins to decrease when the splitting ratio approaches



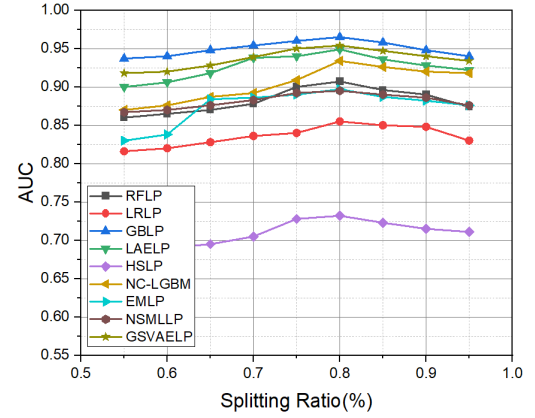
(a) Caltech Dataset



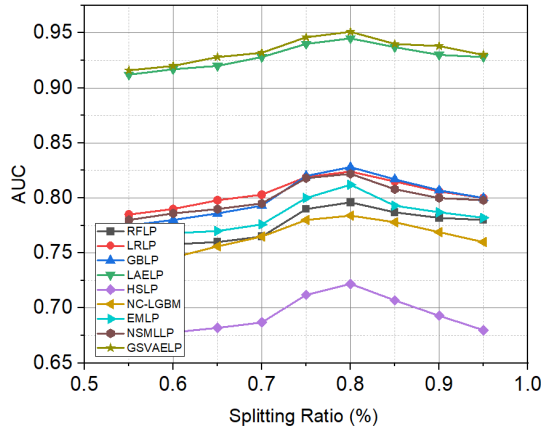
(b) PolBlogs Dataset



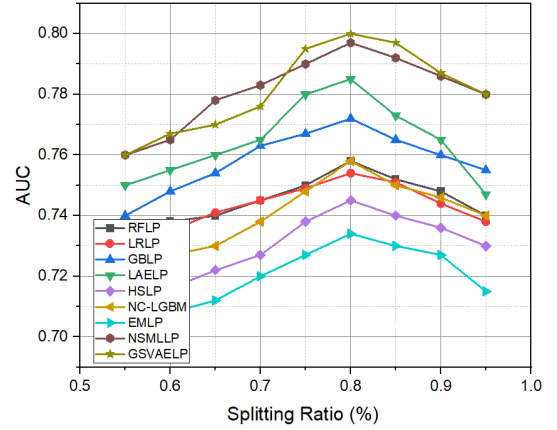
(c) DBLP Dataset



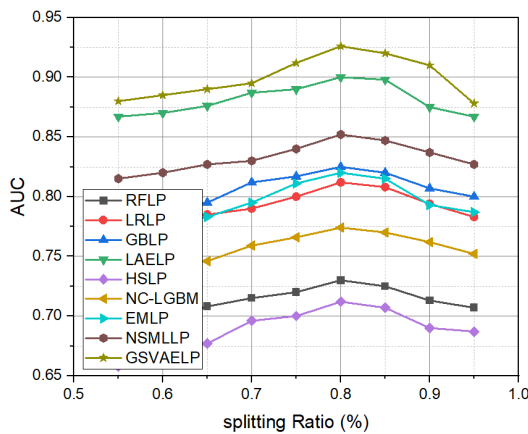
(d) FB-TV Show Dataset



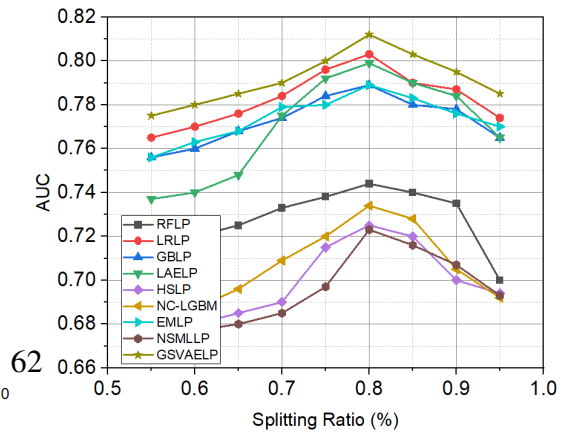
(e) Delaunay N12 Dataset



(f) Hamsterster Dataset



(g) Public Figure Dataset



(h) Wiki Vote Dataset

80%. This shows that the data tends to overfit as the amount of training data increases. Therefore, this study leads us to the conclusion that an 80 : 20% split of the data is the best one.

3.4.2 Parameter Sensitivity

In addition to the previous analysis, we have conducted some experiments to evaluate the effectiveness of the proposed GSVAELP model. The optimal experiments are the most effective means of determining the ideal parameter settings. To fine-tune our model, we utilized one dataset DBLP shown in Fig.3.3. We used the distinct hyperparameters including batch size 300, activation function *leaky_relu*, learning rate 0.0003, dropout 0.8, hidden feats 512, epochs 20, and loss function. The AUC for GSVAELP generally increases with the dimensions of the embedding vectors. However, there is a noticeable drop in AUC when the embedding dimension changes from 4 to 8. The optimal AUC is achieved when the dimension is set to 32 as shown in Fig. 3.3a.

Fig. 3.3b shows the changes in AUC with respect to the sample size k , experiment with different values of $k = 10, 20, 30, 40, 50, 60$ to observe how the number of sampled neighbors affects the performance of the GSVAELP model. The result indicates that the highest AUC is achieved when the sample size is $k = 30$, while the lowest AUC is observed when the sample size is $k = 60$. Overall, the experiment shows that increasing the sample size has no effect on AUC. For our model, testing the AUC with different learning rates varies from 0.0001 to 0.0008. The best AUC occurs when the learning rate is set to be 0.0003 as shown in Fig.3.3c, whereas due to the large number of edges and node features in the data set, the accurate dropout rate argument is to be 0.8 as shown in Fig.3.3d. Overall, our GSVAELP model demonstrates superior link prediction performance when properly explored hyperparameters are used.

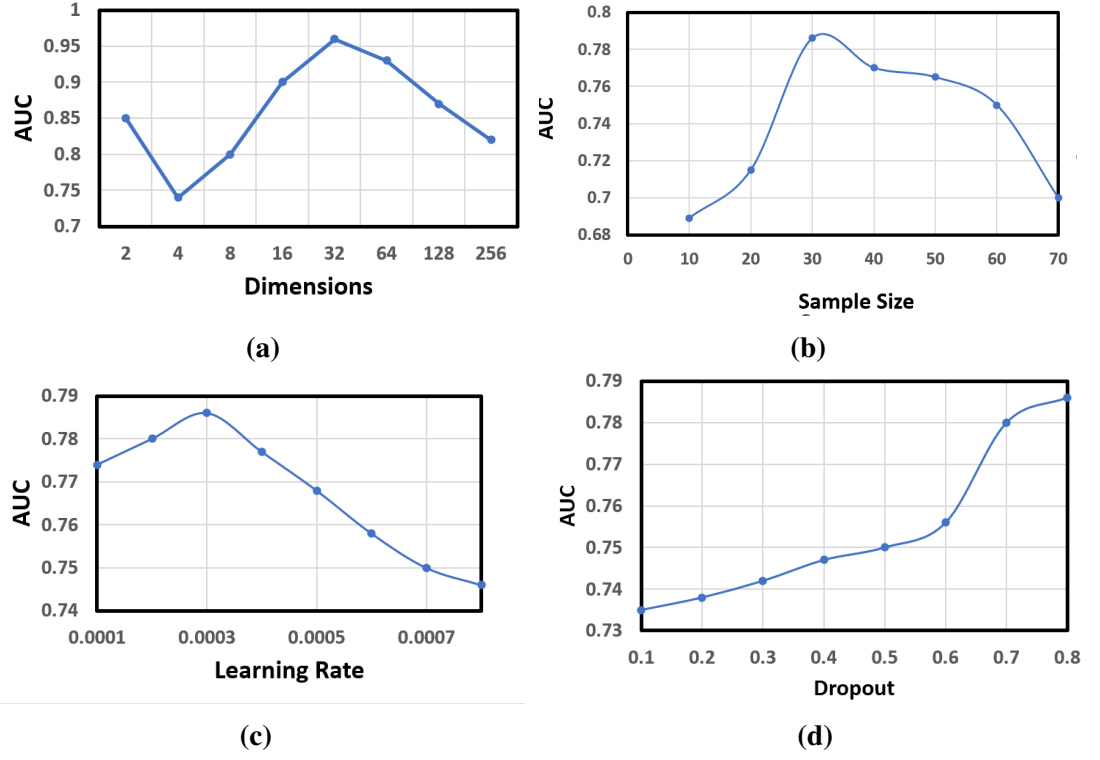


Fig. 3.3: Parameter Sensitivity on DBLP Dataset

3.4.3 Compative Performance Study

To validate the effectiveness of the proposed model GSVAELP, we compared its performance with some baseline and existing methods. Since many link prediction algorithms are stochastic, we ran the experiments 100 times and calculated the average results to obtain evaluation metrics. This approach provides a more comprehensive and reliable assessment of performance. The comparative performance study is shown in Tab. 3.2 to Tab. 3.9. The best result obtained for each dataset is shown and emphasized in bold.

For the Caltech network dataset, Tab.3.2 illustrates the results obtained. It can be clearly seen that the GSVAELP model outperformed in terms of precision of 0.921 and AUC score of 0.914, whereas the EMLP model performed well in terms of recall of 0.863 and accuracy 0.875. However, the LAELP model has achieved an F1-score of 0.893, compared to other link prediction models.

Table 3.2: Overall Performance Metrics of Caltech Dataset

Models	Precision	Recall	F1score	AUC	Accuracy
DeepWalk	0.642	0.623	0.655	0.738	0.674
node2vec	0.773	0.834	0.792	0.780	0.791
LINE	0.646	0.593	0.603	0.569	0.628
RFLP	0.723	0.734	0.720	0.727	0.728
LRLP	0.643	0.676	0.598	0.760	0.540
GBLP	0.728	0.743	0.730	0.820	0.747
LAELP	0.893	0.744	0.893	0.882	0.744
HSLP	0.681	0.672	0.673	0.701	0.682
NC-LGBM	0.726	0.701	0.711	0.782	0.755
EMLP	0.872	0.863	0.842	0.896	0.875
NSMLLP	0.867	0.932	0.891	0.862	0.837
GSVAELP	0.921	0.855	0.799	0.914	0.761

Tab.3.3 shows the results of the PolBlogs dataset, in this table our proposed model GSVAELP performs extremely well in terms of precision, recall, and AUC score. The higher AUC indicates better performance. The NC-LGBM model significantly performs well in terms of F1-score. On the other hand, the NSMLLP model achieved a good accuracy score as compared to other models.

From Tab. 3.4 on the DBLP dataset, we can observe the performance of different models. From the obtained results, it can be seen that the proposed GSVAELP model has the highest recall and AUC value, indicating its utility. On the other hand, models like LAELP have a good precision value, the GBLP performs well in terms of accuracy, and the model NSMLLP has a good F1-score.

Tab. 3.5 presents performance metrics of various models on the FB-TV show dataset. The proposed model GSVAELP outperformed all other methods with the highest precision of 0.967 and recall of 0.960 among all models. It achieved an AUC of 0.954 and an accuracy of 0.750, whereas the model LAELP closely followed the recall of 0.960. In terms of AUC, GBLP is the best-performing model, and the best accuracy is achieved by the NC-LGBM model.

Tab. 3.6 shows the results on the Delaunay N12 dataset. With a precision of 0.63

Table 3.3: Overall Performance Metrics of PolBlogs Dataset

Models	Precision	Recall	F1-score	AUC	Accuracy
DeepWalk	0.521	0.591	0.552	0.581	0.562
node2vec	0.630	0.723	0.712	0.707	0.746
LINE	0.746	0.683	0.753	0.769	0.694
RFLP	0.766	0.756	0.761	0.773	0.774
LRLP	0.734	0.732	0.687	0.838	0.647
GBLP	0.786	0.857	0.791	0.867	0.784
LAELP	0.952	0.953	0.783	0.947	0.767
HSLP	0.722	0.743	0.733	0.742	0.721
NC-LGBM	0.761	0.845	0.801	0.859	0.792
EMLP	0.730	0.802	0.764	0.817	0.746
NSMLLP	0.842	0.867	0.800	0.800	0.823
GSVAELP	0.967	0.955	0.792	0.952	0.748

and a recall of 0.963, our proposed model, GSVAELP, has demonstrated outstanding performance. Additionally, the proposed model achieves the highest AUC of 0.951 among all models. However, the GBLP has performed well in terms of an F1-score of 0.800. LRLP and NC-LGBM both achieve the highest accuracy of 0.812 among the existing methods.

The Tab. 3.7 evaluates the performance metrics on the Hamsterster social network dataset. The models GBLP and NC-LGBM both achieve the highest F1-score of 0.796. LAELP demonstrated excellent performance with a precision of 0.829. Overall as per the accuracy and the recall, the proposed GSVAELP model is the best performer. It achieved an accuracy of 0.834 and a recall of 0.878.

For the public figure dataset, the Tab. 3.8 shows the performance metrics of various models. In baseline methods, the notable performance includes the GBLP with the highest F1-score of 0.800 and LAELP has the highest precision of 0.99. The EMLP model has the highest accuracy of 0.823 among all the methods of link prediction, whereas the proposed model GSVAELP outperformed all other models with the highest recall of 0.954 and the AUC 0.926, indicating the best performance metrics.

The overall performance metrics of the Wiki vote dataset as shown in Tab. 3.9. The

Table 3.4: Overall Performance Metrics of DBLP Dataset

Models	Precision	Recall	F1-score	AUC	Accuracy
DeepWalk	0.597	0.655	0.603	0.673	0.578
node2vec	0.678	0.767	0.722	0.743	0.700
LINE	0.556	0.508	0.579	0.511	0.694
RFLP	0.797	0.733	0.763	0.763	0.762
LRLP	0.712	0.689	0.657	0.718	0.720
GBLP	0.786	0.857	0.791	0.867	0.787
LAELP	0.963	0.786	0.947	0.767	0.780
HSLP	0.724	0.752	0.738	0.718	0.720
NC-LGBM	0.817	0.732	0.773	0.829	0.774
EMLP	0.794	0.845	0.811	0.817	0.784
NSMLLP	0.867	0.912	0.822	0.796	0.772
GSVAELP	0.961	0.925	0.788	0.960	0.786

RFLP model achieved the highest accuracy of 0.795, and the model LAELP showed a higher F1-score of 0.794. Among some existing methods, the EMLP model has performed well in terms of precision of 0.828. The proposed model GSVAELP demonstrated the highest recall of 0.842 and AUC 0.812. In summary, the proposed model GSVAELP demonstrates better performance on all datasets.

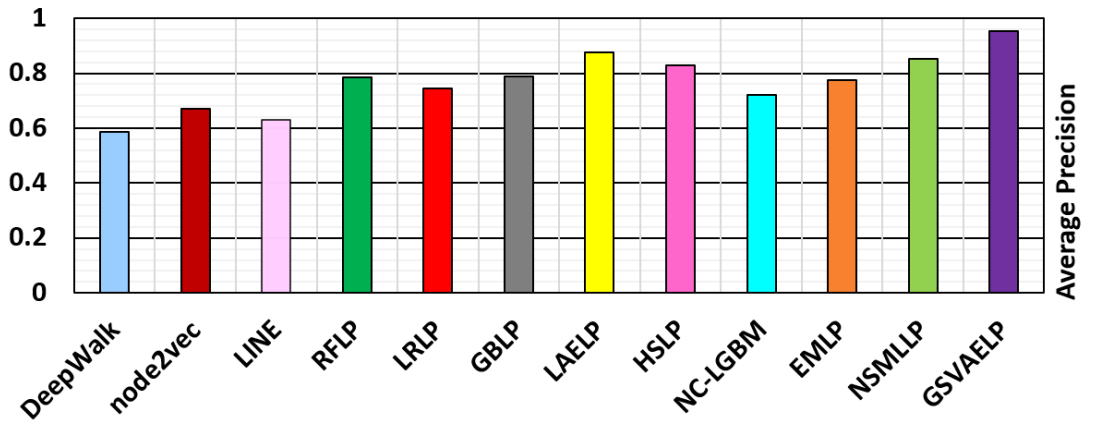
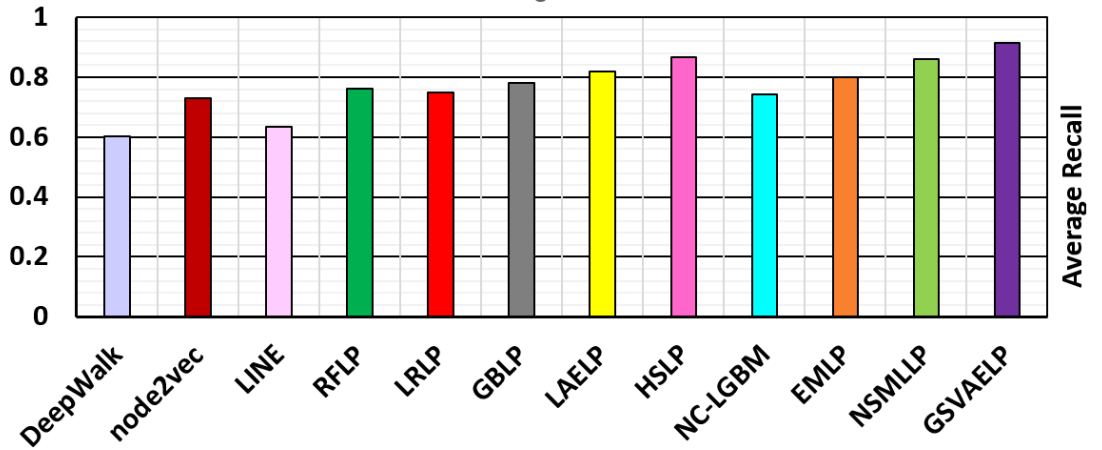
**Fig. 3.4:** Average Precision for each method on all eight datasets

Table 3.5: Overall Performance Metrics of FB-TV show dataset

Models	Precision	Recall	F1-score	AUC	Accuracy
DeepWalk	0.654	0.734	0.712	0.581	0.592
node2vec	0.730	0.760	0.742	0.689	0.623
LINE	0.702	0.754	0.687	0.722	0.690
RFLP	0.926	0.881	0.901	0.907	0.908
LRLP	0.742	0.773	0.687	0.931	0.514
GBLP	0.959	0.877	0.916	0.965	0.904
LAELP	0.962	0.960	0.791	0.949	0.746
HSLP	0.756	0.733	0.700	0.732	0.711
NC-LGBM	0.912	0.852	0.823	0.934	0.922
EMLP	0.925	0.860	0.891	0.897	0.898
NSMLLP	0.910	0.872	0.883	0.895	0.723
GSVAELP	0.967	0.960	0.794	0.954	0.750

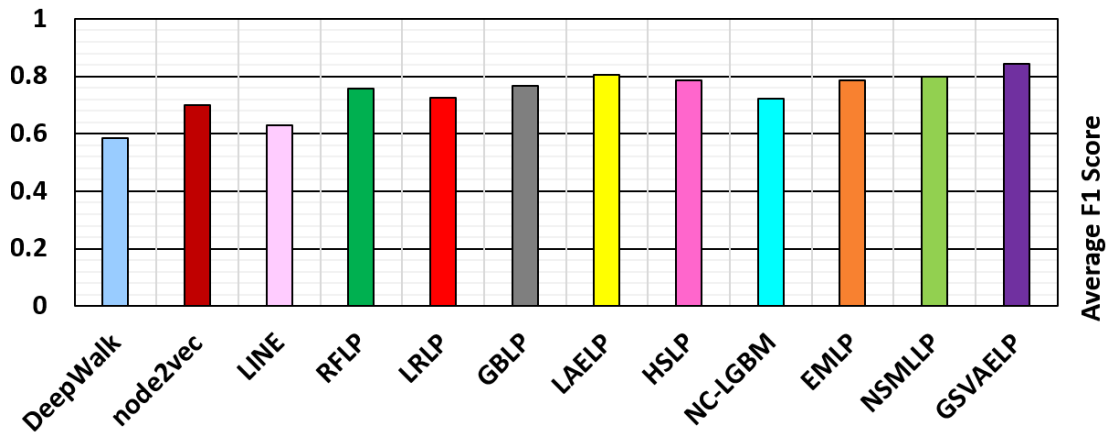
**Fig. 3.5:** Average Recall for each method on all eight datasets

3.4.4 Average Performance of each comparing method of link prediction across all datasets

To gain a better comparative analysis, we calculated the average precision, average recall, and average F1-score of all methods obtained in Tab. 3.2 to Tab. 3.9 for various datasets. The average results of precision, average recall, and average F1-score are depicted in Fig. 3.4, Fig. 3.5, and Fig. 3.6, respectively. The x -axis represents all the models, and the y -axis shows the average of precision, recall, and F1-score. From the results, it is clearly shown that the proposed GSVAELP model has outperformed

Table 3.6: Overall Performance Metrics of Delaunay-N12 Dqataset

Models	Precision	Recall	F1-score	AUC	Accuracy
DeepWalk	0.533	0.554	0.501	0.571	0.552
node2vec	0.664	0.752	0.705	0.686	0.680
LINE	0.589	0.722	0.672	0.698	0.700
RFLP	0.846	0.724	0.780	0.796	0.795
LRLP	0.833	0.756	0.792	0.824	0.812
GBLP	0.835	0.768	0.800	0.825	0.808
LAELP	0.962	0.945	0.788	0.945	0.746
HSLP	0.742	0.785	0.752	0.722	0.724
NC-LGBM	0.788	0.825	0.796	0.784	0.782
EMLP	0.823	0.780	0.806	0.812	0.812
NSMLLP	0.895	0.843	0.765	0.822	0.754
GSVAELP	0.963	0.963	0.793	0.951	0.749

**Fig. 3.6:** Average F1-score for each method on all eight datasets

significantly compared to the other models of link prediction. The average precision of the GSVAELP model is 0.922, whereas some other methods, like LAELP, achieved an average precision of 0.877, and HSLP obtained an average precision of 0.872 across all the datasets. Fig. 3.4 shows the average recall of the proposed GSVAELP model is 0.915 highest among all the other models, the LAELP model demonstrated the average recall of 0.818 whereas in Fig. 3.5 represents the average F1-score of the proposed GSVAELP model is 0.842. Finally, it is revealed that the proposed GSVAELP model outperforms other methods in terms of the highest average precision, average recall, and average F1-score.

Table 3.7: Overall Performance Metrics of Hamsterster Dataset

Models	Precision	Recall	F1-score	AUC	Accuracy
DeepWalk	0.523	0.556	0.520	0.578	0.676
node2vec	0.598	0.667	0.623	0.684	0.692
LINE	0.623	0.670	0.669	0.680	0.682
RFLP	0.740	0.758	0.772	0.750	0.755
LRLP	0.794	0.823	0.787	0.754	0.780
GBLP	0.745	0.848	0.796	0.772	0.786
LAELP	0.829	0.854	0.752	0.780	0.767
HSLP	0.767	0.740	0.723	0.745	0.756
NC-LGBM	0.730	0.834	0.796	0.758	0.797
EMLP	0.812	0.845	0.787	0.734	0.821
NSMLLP	0.815	0.842	0.756	0.797	0.734
GSVAELP	0.820	0.878	0.776	0.800	0.834

3.4.5 Performance Measures

Performance measures are used to show the performance of the classification model. Here, key evaluation parameters such as precision, recall, accuracy and loss are used to analyze the system's performance. Precision: Precision measures the accuracy of positive predictions. It is determined by the ratio of correctly classified positive samples to all samples predicted as positive.

$$Precision = \frac{TP}{TP + FP} \quad (3.17)$$

Recall: Recall measures the model's capability to detect all instances of the positive class, representing the ratio of correctly classified positive samples out of the total positive samples.

$$Recall = \frac{TP}{TP + FN} \quad (3.18)$$

Accuracy: Accuracy metrics evaluate the overall correctness of the model's predictions by considering all the samples.

Table 3.8: Overall Performance Metrics of Public Figure Dataset

Models	Precision	Recall	F1-score	AUC	Accuracy
DeepWalk	0.635	0.578	0.612	0.623	0.692
node2vec	0.655	0.658	0.622	0.686	0.667
LINE	0.569	0.564	0.523	0.605	0.611
RFLP	0.745	0.778	0.645	0.723	0.757
LRLP	0.789	0.812	0.799	0.812	0.796
GBLP	0.835	0.768	0.800	0.825	0.808
LAELP	0.966	0.950	0.723	0.900	0.762
HSLP	0.723	0.765	0.760	0.712	0.735
NC-LGBM	0.736	0.832	0.783	0.774	0.767
EMLP	0.865	0.893	0.887	0.820	0.823
NSMLLP	0.820	0.823	0.752	0.852	0.774
GSVAELP	0.956	0.954	0.773	0.926	0.820

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.19)$$

Where, True Positive (TP) is the total number of correctly predicted positive instances, True Negative (TN) is the total number of correctly predicted negative instances, False Positive (FP) is the total number of incorrectly predicted positive instances (when the actual class is negative) and False Negative (FN) is total number of incorrectly predicted negative instances (when the actual class is positive). Finally, the loss is a quantitative measure that indicates the dissimilarity between predicted and actual values, enabling optimization of the model during the training process.

3.5 Summary

Link Prediction in complex networks, like social networks, involves predicting the future links among the unconnected nodes and has many real-life applications. In this paper, we present a synergistic model, named GSVAELP, that combines graphSAGE embeddings with Variational Autoencoders (VAE) for link prediction. In the proposed model, feature extraction was performed using graphSAGE, which generated embed-

Table 3.9: Overall Performance Metrics of Wiki Vote Dataset

Models	Precision	Recall	F1-score	AUC	Accuracy
DeepWalk	0.580	0.545	0.520	0.596	0.623
node2vec	0.623	0.673	0.678	0.658	0.650
LINE	0.627	0.584	0.569	0.675	0.666
RFLP	0.724	0.731	0.723	0.740	0.795
LRLP	0.712	0.723	0.745	0.802	0.783
GBLP	0.728	0.834	0.745	0.789	0.725
LAELP	0.812	0.826	0.794	0.799	0.747
HSLP	0.645	0.754	0.723	0.725	0.734
NC-LGBM	0.723	0.820	0.745	0.734	0.768
EMLP	0.828	0.823	0.765	0.789	0.820
NSMLLP	0.812	0.825	0.742	0.723	0.734
GSVAELP	0.823	0.842	0.787	0.812	0.792

dings for each node in the network. These node embeddings were created as low-dimensional vectors, with the goal of capturing structural and relational information about the network’s nodes. The Variational autoencoders (VAE) were employed to refine the node embeddings generated by graphSAGE. This phase involved training the VAE to acquire a latent space representation for the nodes, ultimately leading to the creation of a continuous and meaningful latent space. Next, a binary classifier was used to predict forthcoming connections within the networks. Finally, necessary hyperparameter studies were conducted to determine the optimal values for the various hyperparameters used in the different phases of the model. To demonstrate the effectiveness of the proposed model, we conducted a comparative experiment using eight diverse real-world network datasets. The experiment’s findings suggested the efficacy of the proposed model compared to other models. This research contributes a wide range of link prediction methodologies and provides a solid foundation for future investigations into the dynamics of complex network interactions. The future extension of this paper may be to enable the model to handle dynamic graphs where the structure of the graph changes over time.

Chapter 4

MetaLP-EC: A Meta-Learning Framework for Temporal Link Prediction with Evolving Centrality and GCN Embeddings

This chapter introduces MetaLP-EC—a Meta-Learning framework for Temporal Link Prediction that integrates Evolving Centrality measures with Graph Convolutional Network (GCN) embeddings. The proposed model learns from historical snapshots, adapts quickly to new temporal patterns, and incorporates dynamic node influence into the prediction process. By combining structural learning, temporal dynamics, and meta-learning optimization, MetaLP-EC aims to provide accurate and efficient link prediction in evolving network environments. MetaLP-EC differs from existing approaches by jointly integrating meta-learning, evolving centrality, and GCN-based embeddings within a unified framework. While many temporal link prediction models focus on sequential modeling or snapshot-based embeddings, MetaLP-EC learns task-level representations that adapt across time steps. This enables the model to handle concept drift, network evolution, and temporal sparsity more effectively than traditional temporal or static embedding-based methods.

4.1 Background

Link prediction plays a pivotal role in understanding and forecasting the evolution of complex networks such as social, communication, biological, and citation networks [97]. While traditional link prediction methods focus on static snapshots of networks, real-world networks are inherently dynamic their nodes and edges evolve. Accurately modeling these temporal changes is essential for predicting future interactions, detecting community evolution, and understanding long-term behavioral trends within the network[98].

Temporal networks introduce unique challenges to link prediction due to their non-stationary nature and the need to model structural patterns and their evolution across time[99]. Static graph embedding techniques like node2vec[100] and DeepWalk[27] capture structural information effectively but fall short in learning temporal dynamics. Moreover, most existing models treat all nodes uniformly, overlooking the varying importance of nodes based on their centrality or influence in the network[31]. Graph Convolutional Networks were selected due to their ability to effectively capture local structural dependencies while propagating information across the network. In MetaLP-EC, GCNs generate temporally informed node embeddings by integrating evolving centrality features with neighborhood aggregation. This combination allows the model to encode both structural topology and temporal importance, making GCNs well-suited for dynamic link prediction tasks.

We propose MetaLP-EC, a novel Meta-Learning-based framework for Link Prediction with Evolving Centrality awareness to address these limitations. The key idea behind MetaLP-EC is to leverage meta-learning principles to generalize link prediction knowledge across multiple time snapshots while incorporating time-sensitive centrality features that reflect each node’s evolving importance in the network. Centrality measures such as degree, closeness, betweenness, and Jaccard neighborhood coeffi-

cient provide insights into the roles and influence of nodes within the network structure. Yet, their dynamic nature is often overlooked in existing models. The primary contributions of this work are:

1. An innovative meta-learning approach that leverages time snapshots to enhance the prediction of temporal links.
2. The implementation of dynamic centrality-aware embeddings ensures that the temporal significance of nodes is captured in their learned representations.
3. The development of an end-to-end model, MetaLP-EC, combines GCN-based embedding, centrality evolution modeling, and meta-learning with LSTM-based classification.
4. Comprehensive testing on standard temporal network datasets demonstrates superior performance compared to baseline methods.

4.2 Proposed Methodology

This study proposes a novel framework for temporal link prediction, termed Temporal MetaLP-EC (Meta-Learning with Evolving Centrality), which integrates centrality-aware temporal dynamics into a meta-learning-based link prediction paradigm. The model is designed to learn temporal evolution patterns of node interactions by combining time-aware centrality measures, dynamic graph embeddings, and meta-learning to predict future links with high adaptability and generalization across time windows.

The flowchart illustrates the overall architecture of the proposed MetaLP-EC framework for link prediction in dynamic networks. It begins with temporal graph snapshots representing the evolving structure of the network over time. From these snapshots, centrality measures such as Temporal Betweenness Centrality (TBC), Temporal Closeness Centrality (TCC), and Temporal Jaccard Neighborhood Centrality (TJNC) are

computed to quantify the importance and influence of nodes in a temporal context. These centrality scores are then incorporated into a Graph Convolutional Network (GCN), which generates temporal embeddings by capturing structural and temporal information across the graph snapshots. To address class imbalance, the dataset is balanced before training the model. An LSTM classifier is then employed to learn temporal dependencies from the embedding sequences of node pairs. Finally, the trained model performs link prediction, forecasting potential future links in the evolving network. The Fig. 4.1 enables the MetaLP-EC model to integrate temporal dynamics, node centrality, and deep learning techniques for accurate and interpretable link prediction.

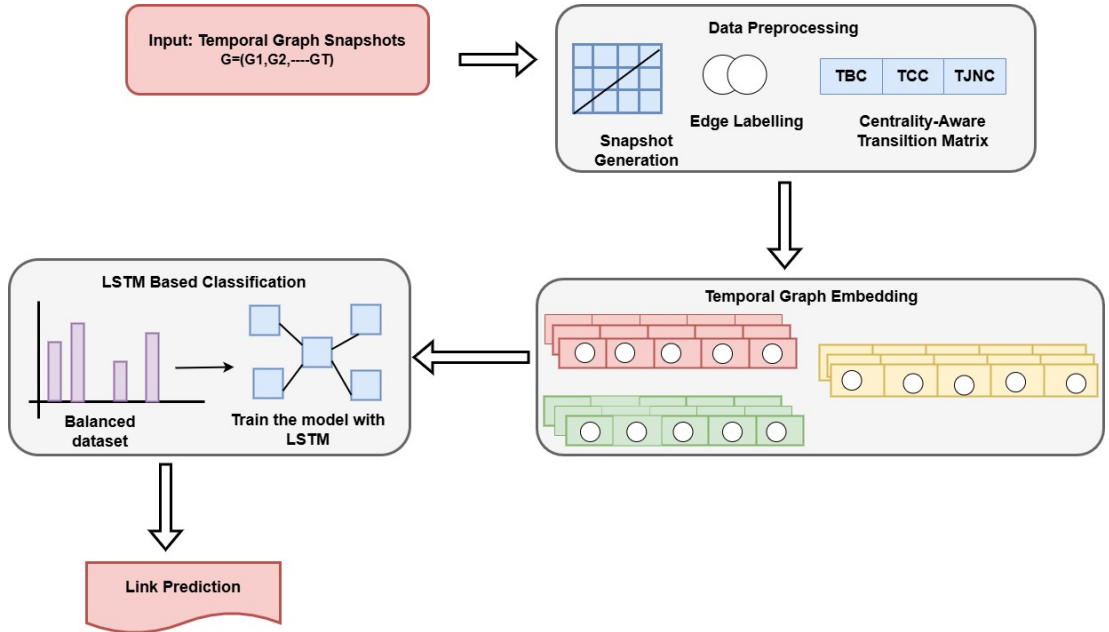


Fig. 4.1: Workflow of the MetaLP-EC Framework for Temporal Link Prediction.

4.2.1 Temporal Graph Modelling

The dynamic network is represented as a sequence of temporal snapshots $G = (G_1, G_2, \dots, G_T)$ where each snapshot $G_t = (V_t, E_t)$ denotes the state of the graph at time t . These snapshots capture evolving topological patterns and the appearance/disappearance of links over time.

4.2.2 Evolving Centrality-Aware Transition Matrix

To capture the temporal significance of nodes, evolving centrality metrics are computed for each graph snapshot G_t . These included.

1. **Temporal Betweenness Centrality (TBC)**: It measures the control of a node over time-evolving shortest paths.
2. **Temporal Closeness Centrality (TCC)**: It captures how close a node is to others across the dynamic network.
3. **Temporal Jaccard Neighborhood Coefficient (TJNC)**: It quantifies temporal co-occurrence similarity of node pairs. A centrality-aware transition matrix, denoted as P_t , is computed for each G_t using these temporal centrality scores. This matrix influences the node sampling process in embedding algorithms, encouraging transitions towards structurally and temporally important nodes. The probability of transitioning from node i to node j is as follows.

$$P_t = \alpha \cdot NormCentrality_t(j) + (1 - \alpha) \cdot \frac{1}{deg(i)} \quad (4.1)$$

Where $\alpha \in [0, 1]$ controls the influence of centrality, and $NormCentrality_t(j)$ is the normalized centrality of node j at time t .

4.2.3 Temporal Graph Embedding using GCN

Temporal networks exhibit non-stationary behavior, where patterns of link formation evolve. To address this challenge, the proposed model adopts a meta-learning framework wherein each graph snapshot G_t is treated as an individual learning task. This strategy enables the model to learn generalizable temporal patterns of link formation across multiple time steps and quickly adapt to unseen temporal states using limited supervision.

1. **Meta Training Phase:** For each $t \in \{1, 2, \dots, T - 1\}$ a meta-learner is trained using embeddings Z_t centrality features, and true link labels.
2. **Meta-Testing Phase:** When a new snapshot G_t is presented, the model rapidly adapts using a small number of labeled examples and predicts future links with minimal retraining. This meta-learning approach enables the classifier to generalize to previously unseen temporal states and adapt efficiently.

4.2.4 LSTM-based Classification for Link Prediction

: In this proposed work, we employ a (LSTM) network as the exclusive classifier to understand the temporal changes in node representations for predicting links in temporal networks. LSTM is particularly adept at handling sequential data, making it an excellent option for learning from graph embeddings that change over time. While conventional classifiers consider static node features or embeddings at a single timestamp, LSTM can model how a node’s structural and centrality-related properties evolve.

Before classification, we addressed the class imbalance inherent in link prediction tasks by applying dataset-balancing techniques. This ensured a fair and unbiased training process. The balanced dataset was partitioned into training and testing sets following an 80:20 ratio. Specifically, 80% of the data was used to train the model, while the remaining 20% was reserved for evaluating its predictive performance. This setup enables robust validation and ensures generalization across diverse temporal graph datasets. This sequential modeling allows for detecting subtle behavioral shifts in the network that may signal future link formations. For each node pair (u, v) , we construct time-series embedding sequences spanning the past k graph snapshots.

$$\mathbf{E}_u = [Z_{t-k}(u), \dots, Z_t(u)], \mathbf{E}_v = [Z_{t-k}(v), \dots, Z_t(v)] \quad (4.2)$$

Where $Z_t(u) \in \mathbb{R}^d$, is the embedding of node u at time t , obtained using the proposed

centrality-aware GCN. Each sequence captures the trajectory of node behavior over time. The LSTM learns temporal correlations in these sequences to determine whether a future link between u and v is likely.

1. Dual LSTM Block: Two independent LSTMs process the embedding sequences E_u and E_v respectively.

$$\mathbf{h}_u = LSTM(E_u), \mathbf{h}_v = LSTM(E_v) \quad (4.3)$$

The LSTMs independently encode temporal dependencies in each node's behavior and generate final hidden representations as $\mathbf{h}_u$ and $\mathbf{h}_v$.

2. Concatenation Layer: The outputs of the two LSTM branches are then concatenated to form a joint representation of the node pair as:

$$\mathbf{h}_{uv} = [h_u] || [h_v] \quad (4.4)$$

This concatenated vector captures the co-evolutionary temporal patterns between the two nodes. Finally, $\mathbf{h}_{uv}$ is passed through a fully connected dense layer followed by a sigmoid activation function to estimate the probability of link formation.

3. Dense + Sigmoid Layer: A fully connected layer followed by a sigmoid activation predicts the link probability. The prediction is computed as:

$$y_{uv} = \sigma(\mathbf{w} \cdot \mathbf{h}_{uv} + b) \quad (4.5)$$

Here, $y_{uv} \in [0, 1]$ represents the probability of a future link between nodes u and v . where $\mathbf{W}$ is the weight matrix of the dense layer. The learnable parameters transform the concatenated LSTM output $\mathbf{h}_{uv}$ into a scalar value. The b is the bias term; it helps the model fit the data more flexibly, and the last σ is the

Table 4.1: Basic Statistics of Selected Dynamic Networks

Dataset	$ V $	$ E_t $	$\bar{d}$	$d_{\max}$	Timespan (days)
Dynamic PPI (D-PPI)[101]	8,000	90.5K	22.6	1,102	365
FB-FORUM[102]	899	50.5K	669.8	5,177	164.49
WIKI-TALK[103]	2,394	100.1K	83.5	5,230	28.0
Dynamic Email Network[104]	986	332.3K	674.0	7,898	803

activation function, which maps the output to a probability value between 0 and 1.

$$\mathcal{L} = -\frac{1}{N} \sum_{(u,v)} [y_{uv} \log(\hat{y}_{uv}) + (1 - y_{uv}) \log(1 - \hat{y}_{uv})] \quad (4.6)$$

The LSTM model is trained using binary cross-entropy loss. $\mathcal{L}$ is the average loss over all node pairs (uv) in your training data. N is the total number of training samples. This work enables the model to learn and predict dynamic interactions within evolving networks.

4.3 Experimental Setup and Datasets

We have executed the proposed algorithm alongside other leading algorithms to make a comparison using Python programming. The codes for all algorithms were written and executed in Collab Pro (Python). The experiments were conducted on a 64-bit desktop with an Intel(R) Core(TM) i5-8265U CPU @1.60 GHz, 1.80 GHz processor, and 8 GB RAM. We evaluated the performance of all methods on five real-world datasets. Table 4.1 presents a statistical overview of these datasets. A concise description of each dataset is provided in the following paragraphs.

4.3.1 Evaluation Metrics

The effectiveness of the proposed MetaLP-EC model is assessed through standard link prediction metrics. The AUC (Area Under the ROC Curve) evaluates the model’s capability to differentiate between connected and unconnected node pairs. Precision,

Table 4.2: Comparative Analysis on Dynamic PPI and FB-Forum Datasets

Method	Dynamic PPI				FB-Forum			
	AUC	Precision	Recall	F1-Score	AUC	Precision	Recall	F1-Score
Common Neighbors (CN)	0.682	0.612	0.579	0.595	0.651	0.584	0.553	0.568
Resource Allocation (RA)	0.695	0.627	0.593	0.609	0.667	0.597	0.566	0.581
SimRank	0.712	0.641	0.606	0.623	0.681	0.609	0.573	0.590
DeepWalk	0.789	0.711	0.693	0.702	0.755	0.698	0.679	0.688
node2vec	0.803	0.728	0.705	0.716	0.767	0.714	0.694	0.704
EMLP	0.842	0.765	0.751	0.758	0.811	0.752	0.820	0.742
LSNI	0.856	0.822	0.766	0.772	0.825	0.764	0.749	0.756
GC-LSTM	0.868	0.791	0.778	0.784	0.840	0.778	0.763	0.770
MetaLP-EC	0.878	0.819	0.803	0.813	0.837	0.807	0.788	0.797

Recall and F1-Score measure the accuracy and completeness of the predicted links. To assess the quality of the ranked prediction, Mean Reciprocal Rank (MRR) is used, which considers both the relevance and the position of correct predictions in the ranking. These metrics offer a thorough evaluation of classification accuracy and ranking quality in temporal contexts. The ranges between these metrics are within the $[0.1, 0.9]$ range. This range was chosen to comprehensively evaluate the model’s sensitivity to these parameters across diverse datasets.

4.4 Performance of the proposed method against other methods

To assess the strength and adaptability of the proposed MetaLP-EC model, we evaluate its performance compared to baseline and contemporary methods using four key temporal datasets. The models considered include traditional similarity measures (CN, RA, SimRank), graph embedding-based models (DeepWalk, node2vec), and cutting-edge temporal models (EMLP[38], LSNI[48], GC-LSTM[105]). The results are presented using four standard metrics: AUC, Precision, Recall, and F1-Score. The models that exhibit superior performance across the evaluation metrics are highlighted in bold for clarity and emphasis.

The proposed MetaLP-EC model’s performance is thoroughly assessed using four

Table 4.3: Comparative Analysis on Wiki-Talk and Dynamic Email Datasets

Method	Wiki-Talk				Dynamic Email			
	AUC	Precision	Recall	F1-Score	AUC	Precision	Recall	F1-Score
Common Neighbors (CN)	0.639	0.576	0.541	0.558	0.663	0.593	0.561	0.577
Resource Allocation (RA)	0.654	0.588	0.555	0.571	0.677	0.607	0.574	0.590
SimRank	0.671	0.602	0.566	0.583	0.692	0.619	0.586	0.602
DeepWalk	0.735	0.659	0.642	0.650	0.768	0.703	0.683	0.693
node2vec	0.751	0.672	0.656	0.664	0.779	0.718	0.697	0.707
EMLP	0.794	0.714	0.745	0.706	0.822	0.763	0.741	0.752
LSNI	0.807	0.726	0.711	0.718	0.836	0.776	0.756	0.766
GC-LSTM	0.833	0.738	0.722	0.730	0.849	0.818	0.769	0.778
MetaLP-EC	0.828	0.764	0.740	0.754	0.873	0.812	0.793	0.802

dynamic network datasets: Dynamic PPI, FB-Forum, Wiki-Talk, and Dynamic Email. Table 4.2 and Table 4.3 offer a comprehensive comparison of MetaLP-EC against both traditional heuristic methods (CN, RA, SimRank) and modern learning-based techniques (DeepWalk, node2vec, EMLP, LSNI, GC-LSTM) using standard evaluation metrics: AUC, Precision, Recall, and F1-Score. MetaLP-EC consistently surpasses baseline models across all datasets, particularly excelling in Recall and F1-Score, showcasing its proficiency in capturing the evolving structural patterns within temporal networks. For example, on the Dynamic PPI and FB-Forum datasets, MetaLP-EC attains the highest AUC and F1 scores, reflecting its strong discriminative and generalization capabilities. Likewise, in the Wiki-Talk and Dynamic Email datasets, MetaLP-EC sustains superior performance, especially in Precision and overall balanced F1-Score, underscoring its robustness and adaptability to diverse network dynamics. These findings confirm the effectiveness of the evolving-centrality-based meta-learning framework combined with GCN embeddings and an LSTM classifier.

The Fig.4.2 compares the average accuracy of different link prediction methods across four dynamic datasets: D-PPI, FB-Forum, Wiki Talk, and Dynamic Email. Traditional similarity-based techniques like Common Neighbors (CN), Resource Allocation (RA), and SimRank show relatively lower accuracy across all datasets, suggesting their limited ability to capture complex temporal patterns. Embedding-based methods such as DeepWalk and node2vec demonstrate moderate improvements, especially on the Wiki Talk and Dynamic Email datasets. More advanced techniques, includ-

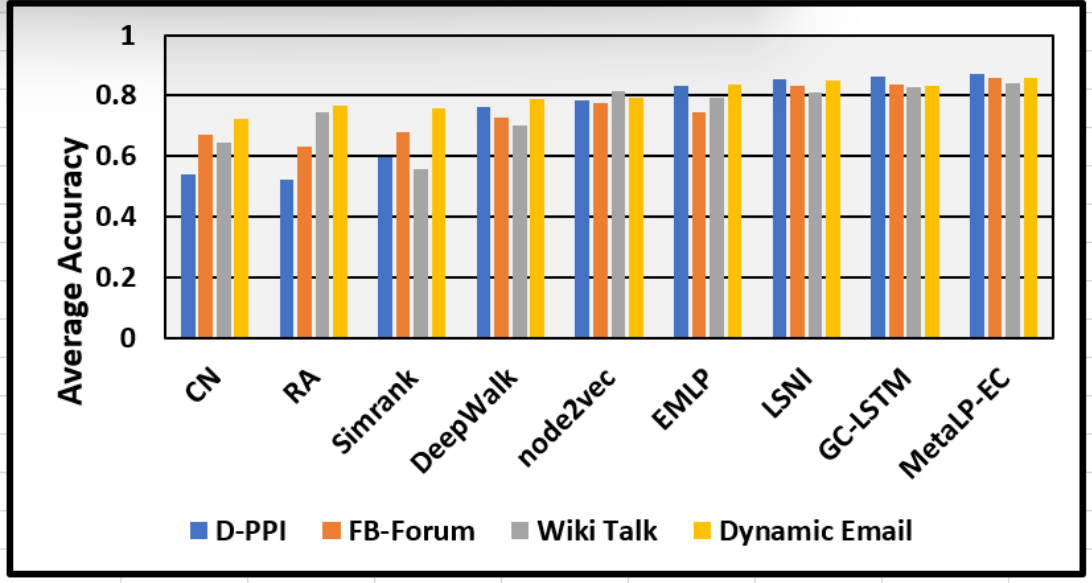


Fig. 4.2: Performance Evaluation of Link Prediction Methods Based on Average Accuracy

ing EMLP, LSNI, and GC-LSTM, consistently outperform earlier models, highlighting their effectiveness in modeling temporal and structural dependencies. Notably, the proposed MetaLP-EC framework achieves the highest or nearly the highest average accuracy across all datasets, showcasing its robustness and effectiveness in handling diverse temporal graphs. This underscores the benefit of integrating centrality-aware transition matrices and temporal graph embeddings with LSTM classifiers for enhanced link prediction.

4.5 Conclusion

In this study, we introduced MetaLP-EC, an innovative framework for temporal link prediction that combines evolving node centrality with a time-sensitive meta-learning approach. By utilizing Graph Convolutional Network (GCN) embeddings and capturing temporal dependencies through a classifier based on LSTM, the framework adeptly tracks the dynamic changes in node interactions. Integrating evolving centrality into the meta-learner enables the model to adjust to structural shifts over time, improving

prediction accuracy. Comprehensive experiments on various real-world dynamic networks reveal that MetaLP-EC consistently surpasses traditional similarity-based methods and recent deep learning-based techniques in standard evaluation metrics such as AUC, Precision, Recall, and F1-Score. These findings validate the effectiveness of merging temporal dynamics, centrality-driven transition matrices, and deep sequential modeling for reliable link prediction in evolving networks.

Chapter 5

MetaLP-DGI: Meta-Learning-Based Link Prediction With Centrality-Aware Deep Graph Infomax Embeddings

This chapter introduces a MetaLP-DGI model, a novel meta-learning-based link prediction framework that integrates centrality-aware transition matrices with Deep Graph Infomax (DGI) embeddings. The approach addresses key limitations of existing link prediction models, such as limited adaptability to varying network structures, underutilization of node centrality in guiding random walks, and the inability to generalize across diverse social network datasets. By incorporating graph centrality measures—including betweenness centrality, closeness centrality, and Jaccard coefficient centrality—into the transition matrix, the proposed framework enhances structural awareness during the embedding generation process. Furthermore, DGI ensures robust self-supervised representation learning, while meta-learning enables rapid adaptation to unseen networks with minimal retraining. MetaLP-DGI addresses network sparsity by maximising contrastive mutual information, encouraging informative embeddings even when explicit links are limited. The inclusion of centrality-aware features further compensates for missing structural information by emphasising influential nodes. Additionally, the meta-learning framework enhances robustness across heterogeneous network structures by learning common patterns shared across tasks, enabling effective

link prediction even in sparse or noisy networks.

5.1 Background

Complex network analysis is a key area of modern science, offering valuable tools to understand the structure and behavior of interconnected systems. These networks, which encompass social, biological, technological, and informational domains, are characterized by intricate relationships and non-trivial topological features such as clustering, community structures, and scale-free behavior [106, 107]. Analyzing complex networks offers valuable insights into the underlying principles governing their formation and evolution, enabling the discovery of patterns, the prediction of behaviors, and the identification of key elements within the system. Among the various tasks in network analysis, link prediction is a critical problem, as it seeks to anticipate future connections or infer missing ones based on existing data. This capability not only enhances our understanding of network growth and functionality but also supports a wide range of practical applications, from improving recommendation systems to uncovering hidden biological interactions [108]. Hence, link prediction is a fundamental task in network analysis, receiving substantial interest from researchers across various scientific domains. This task involves forecasting the likelihood of future or currently unobserved connections between nodes within a network [109, 48]. Traditional approaches to link prediction have been categorized into methodologies, such as similarity-based measures, probabilistic models, machine learning approaches, and embedding-based methods. Similarity-based measures rely on predefined metrics, such as common neighbors and the Jaccard coefficient, to estimate the potential for new links between node pairs [56]. Probabilistic methods utilize statistical inference to estimate the likelihood of edge formation, leveraging information derived from observed network structures. Machine learning methods, which incorporate features extracted from nodes and their relationships, have enhanced predictive performance.

More recently, network embedding-based approaches have become prominent due to their ability to capture complex patterns in network topology by embedding nodes into continuous vector spaces [110].

This study proposes a novel meta-learning-based link prediction framework integrating centrality measures into a higher-order multi-step connectivity matrix. This matrix is input for a Deep Graph Infomax (DGI) model to extract informative node embeddings. Subsequently, these embeddings are utilized by the CatBoost classifier to perform link prediction. By combining meta-learning principles, structural centrality information, and advanced machine-learning techniques. The proposed MetaLP-DGI model has strong potential for real-world applications where accurate link prediction is critical. In social media analytics, the model can forecast potential friendships or interactions based on users' network behavior, enabling more personalized content delivery, community detection, and improved recommendation systems. For example, platforms like Facebook or LinkedIn could leverage MetaLP-DGI to suggest meaningful connections by identifying latent patterns in users' networks. Similarly, in biological network analysis, particularly in protein-protein interaction (PPI) networks, the model can help predict unknown protein interactions, which is essential for understanding disease mechanisms and drug target discovery. By demonstrating high accuracy and generalization on diverse datasets such as Amazon (e-commerce co-purchase network) and PPI (biological network), the proposed method is both versatile and practical for a range of domains requiring robust link prediction capabilities. Our approach aims to improve predictive accuracy and generalization in diverse network settings. The main contributions of this proposed work are as follows:

- We develop a link prediction model to adapt to various network dynamics and improve generalization across different datasets. The meta-learning framework seamlessly incorporates centrality-based features, enabling the model to understand better the structural importance of nodes in link prediction tasks.

- The multi-step connectivity matrix is computed by utilizing meta-learning-based features such as closeness centrality, degree centrality, and betweenness centrality, enhancing the prediction of potential links in the network.
- Once the multi-step connectivity matrix is constructed, it can be utilized as input for the Deep graph Infomax (DGI) algorithm. The DGI algorithm generates node embeddings that leverage both direct and multi-hop relationships encoded within the matrix.
- The integration of DGI Embeddings replaces traditional node embeddings, which is more effective at capturing high-level graph structure and node relationships, improving link prediction accuracy.
- The updated embedding is then passed to the hybrid classifier Approach, which combines the strengths of CatBoost, a robust gradient-boosting classifier, with the meta-learning framework to optimize predictive performance.
- The proposed model is compared with several state-of-the-art models across eight distinct social network datasets. The results indicate that the proposed model outperforms its counterparts, showcasing its superiority.

5.2 Proposed Methodology

This section presents the proposed meta-learning-based link prediction model, MetaLP-DGI, which incorporates centrality-aware embeddings into the Deep Graph Infomax (DGI) framework. The proposed work aims to develop a robust link prediction model for social networks by leveraging Deep Graph Infomax (DGI) to generate node embeddings and apply meta-learning to dynamically adapt the model based on different regions within the graph. The embeddings generated using DGI will capture both local and global graph structures. The centrality-aware connectivity matrix will be em-

ployed to emphasize meaningful node connections, utilizing centrality measures such as degree centrality, betweenness centrality, and closeness centrality. These centrality scores, incorporated into the connectivity matrix, will guide the generation of the random walks and ultimately affect the learned embeddings.

The meta-learning component automatically tunes hyperparameters, such as the DGI random walk settings, and adjusts the ensemble classifier’s weights based on the graph’s properties. We use the CatBoost classifier for link prediction because it efficiently handles tabular data and manages categorical features without extra processing, making it a strong choice for this task. Before training the classifier, we must balance the dataset with an equal number of positive and negative samples. The well-labeled dataset is then input to the CatBoost machine learning algorithm to facilitate the final link prediction. We perform various hyperparameter studies to obtain the best values of different model parameters. The key novelty lies in using a meta-learner to optimize the weights in the connectivity matrix for random walks, making the model more responsive to different node characteristics and graph structures. The proposed model also intends to handle unbalanced data distributions or uncertainty inherent in link prediction tasks. The flowchart of our proposed MetaLPDGI-CatBoost is shown in Fig. 5.1. The following is a step-by-step explanation of the various phases of the proposed link prediction model, as discussed below.

5.2.1 Multi-step Connectivity Matrix Construction

In this phase, we create a dynamic higher-order connectivity matrix $M_{connect}$. The Higher-Order connectivity matrix approach effectively enhances link prediction tasks, particularly by capturing relationships between nodes that are not directly connected but linked through higher-order connections (i.e., via intermediate nodes). This approach enables a more comprehensive understanding of the network’s structure and enhances the model’s ability to predict potential links.

This allows the model to capture more global graph structures and distant dependencies. It combines the adjacency matrix and distinct centrality measures with weights learned by a meta-learning model. The higher-order connectivity matrix is a symmetric matrix of size $N^k \times N^k$, where N is the number of nodes in the graph and k is the order of the matrix.

The idea is to create the connectivity matrix dynamically, where the meta-learning algorithm learns how to adjust the matrix construction parameters based on node and graph features. The new higher-order connectivity matrix likely introduces additional features or captures different aspects of node relationships compared to traditional link prediction methods. This can lead to more informative representations of node pairs, further enhancing the model's predictive performance. It potentially improves the model's ability to capture subtle changes in connectivity patterns. A higher-order connectivity matrix is rooted in graph theory and Markov chain analysis. It represents the probability of transitioning from one node to another in a network. In link prediction, the connectivity matrix can capture the relationships between the nodes by quantifying the likelihood of node transition or providing a structured way to model dynamics and connectivity. For the meta-learning approach, we introduce the idea of weighting these matrices based on centrality:

$$M_{connect} = \beta_1 T_1 + \beta_2 T_2 + \dots \beta_n T_n \quad (5.1)$$

where β_i are the learned weights (through meta-learning) for each step i , and T_i is the i -order connectivity matrix. The centralities are integrated into the matrix weights β_i , allowing the model to adjust the importance of different orders of connectivity based on the characteristics of the nodes. The weights could be functions of the centrality measures. For example, higher centrality nodes could increase the weight of lower-order matrices (closer, direct connections). Lower centrality nodes might give more weight to higher-order matrices (longer-distance connections).

The centralities can be incorporated as follows:

$$\beta_i = \mathbf{f}(C_{deg}, C_{bet}, C_{clo}) \quad (5.2)$$

Where f is a function learned through meta-learning that determines how each centrality contributes to the weight of each order i .

$$M_{connect}(u, v) = \frac{A^k(u, v) \cdot f(C(u), C(j))}{\sum_j A^k(u, v) \cdot f(C(u), C(j))} \quad (5.3)$$

Here, the $M_{connect}(u, v)$ represents the meta-learning-based connectivity matrix between the nodes u, v , where $A^k(u, v)$ is the adjacency matrix entry for nodes u and v . The $f(C(u), C(j))$ is a function of the centrality measures of nodes u and v . Furthermore, $C(u)$ and $C(j)$ are the combination of centrality measures like C_{deg}, C_{bet} and C_{clo} . The denominator ensures that the rows of the matrix sum to 1, making it a valid connectivity matrix. This matrix provides a centrality-aware representation of the graph structure, highlighting direct and indirect multi-step connections between nodes.

5.2.2 Modified DGI with meta-learning connectivity matrix

In this phase, after getting the connectivity matrix between all the node pairs, we utilize the network embedding-based DGI algorithm to transform the high-dimensional network into a low-dimensional space. These embeddings provide a rich feature space reflecting the network topology or enhancing the model's ability to capture complex node relationships. It mainly focuses on network-specific properties and relational data, distinguishing it from traditional statistical methods that may not consider network structure's complex interactions and dependencies

The DGI method is employed to generate node embeddings that encode the structural information of the graph, allowing the model to identify complex patterns and interactions within the network. It learns a d -dimensional embedding for each node

by optimizing a skip-gram model to predict the neighboring nodes of each node in the random walks. Once the node embeddings are learned, we use these embeddings to predict the existence of the link between the nodes u, v .

Once the centrality-aware connectivity matrix $M_{connect}(u, v)$ is constructed, the DGI algorithm is applied to learn low-dimensional node embeddings. DGI optimizes the embeddings by maximizing mutual information between the graph's nodes and a global summary vector, capturing both local and global structural features: The equation below denotes the generated embedding for all nodes in the network by taking the embedding dimension (d), walk length (l), and the computed connectivity matrix as input.

$$Z(uv) = DGI(M_{connect}(u, v), d, l) \quad (5.4)$$

Here, $Z(uv)$ is the set of embedding for all nodes in the network, and it is a two-dimensional matrix of size $n \times d$ with n as the number of nodes, Where d is the embedding dimension and l is the walk length. The equation below denotes the generated embedding of dimension d for node u of the network.

When we pass the meta-learning-based connectivity matrix $M_{connect}$ as input, the embedding Z_u is now learned concerning this matrix. So, the embedding learning is influenced by centrality measures and similarity metrics, making the embeddings richer in capturing local and global structural information. Thus, the modified DGI loss function incorporating the meta-learning-based connectivity matrix is:

$$L_{DGI-meta} = -E_G [\log \sigma (z_u \mathbf{S}(M_{connect})) + \sum_{v \in V} \log \sigma (-z_v) \mathbf{S}(M_{connect})] \quad (5.5)$$

The equation represents the DGI loss function, and E_G refers to the expectation

over the graph G . This means the loss function is computed by taking the average of the expressions within the brackets across multiple nodes or instances of the graph. The first part $(\mathbf{S}(M_{connect}))$ represents the global summary vector that depends on the meta-learning-based connectivity matrix $M_{connect}$. It is computed as the aggregation of node embedding z_u over the entire graph. Where the $\log \sigma(\mathbf{S}(M_{connect}))$ indicates the sigmoid function evaluates the similarity between the embeddings of node u and the global summary vector. The second term $\sum_{v \in V} \log \sigma(-z_v) \mathbf{S}(M_{connect})$ helps the model discriminate between the real embedding and the negative node embeddings. We need to connect the node embedding generation process to integrate the loss function with the DGI embedding process in our proposed Meta-Learning-Based Link Prediction using the Centrality-Aware connectivity matrix model to maximize the likelihood of node co-occurrence in random walks.

The loss function helps guide the learning of embeddings so that structurally or topologically close nodes in the network (as determined by the centrality-aware connectivity matrix) have more similar embeddings. Here's how this fits together. After getting the DGI embedding of each vertex $u \in V$, we introduce the embedding for each edge (u, v) as the dot product of the embedding of node u and node v as follows.

$$Z[u, v] = z[u] \odot z[v] \quad (5.6)$$

$$z(u) \odot z(v) = (u_1 v_1, u_2 v_2, \dots, u_d v_d) \quad (5.7)$$

The element-wise dot product of the embeddings $Z(u)$ and $Z(v)$ for an edge (u, v) is represented in Eq. 5.6. Combining the DGI and CatBoost ensemble classifier enhances the model's robustness and generalization ability. The traditional link prediction methods, like probabilistic and similarity-based methods, may struggle with noisy or incomplete data, which may not generalize to unseen data or capture high-order network structures. After that, we must prepare the training and testing datasets by

combining the actual and non-existent links.

5.2.3 Boosting Algorithm for Link Prediction

Once the node embeddings Z_u and Z_v are learned, we apply an ensemble classifier, such as CatBoost, to predict the presence of links between node pairs. The idea behind boosting is to train weak learners sequentially and combine their predictions to form a robust model. For boosting, let consider $h_m(Z)$ represents the m weak learner-like decision tree and $Z[u, v] = Z_u \odot Z_v$ be the features vector of edge (u, v) as discussed in Eq. 5.6. The boosting algorithm iteratively learns weak learners by minimizing the following objective function for each learner m .

$$\mathcal{L}^{(m)} = \sum_{(u,v)} \ell(y_{uv}, F_{m-1}(\mathbf{Z}_{uv}) + \eta h_m(\mathbf{Z}_{uv})) \quad (5.8)$$

y_{uv} indicates whether a link exists between the node u and v . $F_{m-1}(\mathbf{Z}_{uv})$ is the prediction from the ensemble at iteration $m - 1$. The η represents the learning rate, and $h_m(\mathbf{Z}_{uv})$ is the prediction from the current weak learner (decision tree). ℓ is the loss function; now compute the binary loss entropy for link prediction.

$$\ell(y_{uv}, \hat{y}_{uv}) = -y_{uv} \log \hat{y}_{uv} - (1 - y_{uv}) \log(1 - \hat{y}_{uv}) \quad (5.9)$$

The Eq.5.9 represents the binary cross entropy loss function. The breakdown of the equation is as follows: y_{uv} is the actual label for the existence of the link between the nodes u and v , and $\hat{y}_{uv}$ is predicted. The probability produced by the model lies between 1 and 0. The $-y_{uv} \log \hat{y}_{uv}$ penalizes the model when the actual label is 1 and the predicted probability $\hat{y}_{uv}$ is low. While the other term $-(1 - y_{uv}) \log(1 - \hat{y}_{uv})$, penalizes the model when the actual label is 0, and the predicted probability $\hat{y}_{uv}$ is high.

The final prediction for link existence is computed using the aggregated output of

the boosting classifier. At each boosting step, the model's predictions are updated. After that, compute the evaluation metrics on distinct datasets.

$$F_m(Z_{uv}) = F_{m-1}(Z_{uv}) + \eta h_m(Z_{uv}) \quad (5.10)$$

The final link prediction for link existence is computed using the aggregated output of the classifier.

$$\hat{y}_{uv} = \sigma(F_m(emb_{uv})) \quad (5.11)$$

M is the total number of boosting iterations, and $\sigma(emb_{uv})$ is the sigmoid function to map the output to a probability.

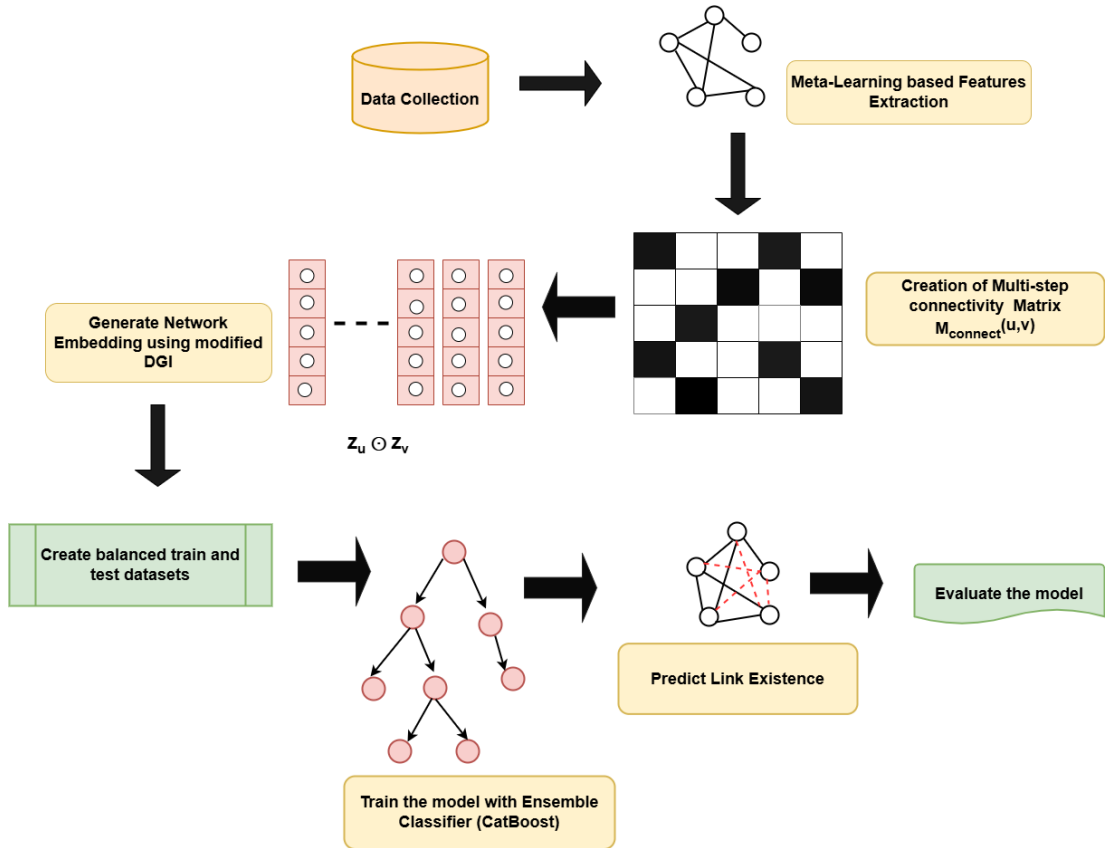


Fig. 5.1: Flowchart of the proposed meta-learning, deep graph infomax, and CatBoost-based link prediction model

Algorithm 1 Multi-step Connectivity Matrix + DGI + CatBoost for Link Prediction

Require: Graph $G = (V, E)$, Centrality measures $C_{deg}, C_{bet}, C_{close}$, Maximum number of steps k , Weights β for each order

Ensure: Predicted link existence between node pairs

- 1: **Step 1: Construct the Multi-step Connectivity Matrix**
- 2: **for** each pair of nodes u and v in V **do**
- 3: Compute adjacency matrix power $A^k(u, v)$ for k steps
- 4: For each k -th step, compute the centrality-based function $f(C(u), C(v))$
- 5: Calculate the multi-step connectivity matrix element:

$$M_{\text{connect}}(u, v) = \frac{A^k(u, v) \cdot f(C(u), C(v))}{\sum_j A^k(u, j) \cdot f(C(u), C(j))}$$

- 6: **end for**
- 7: **Step 2: Input M_{connect} into Data Graph Infomax (DGI)**
- 8: Initialize encoder network to learn node embeddings from M_{connect}
- 9: **for** each node u in V **do**
- 10: Compute the node embedding $h(u)$ using DGI
- 11: Optimization objective:

$$\text{Maximize: } L_{\text{DGI}} = \log D(h(u), h_{\text{global}}) - \log(1 - D(h(u), h_{\text{global}}))$$

- 12: **end for**
- 13: Obtain final node embeddings $h(u)$ for all $u \in V$
- 14: **Step 3: Prepare the Dataset for Link Prediction**
- 15: **for** each node pair (u, v) **do**
- 16: Use the learned embeddings $h(u)$ and $h(v)$ as features for link prediction
- 17: **end for**
- 18: Create a labeled dataset of node pairs (u, v) :
- 19: Positive samples: $(u, v) \in E$
- 20: Negative samples: Random pairs $(u, v) \notin E$
- 21: **Step 4: Train CatBoost Classifier for Link Prediction**
- 22: Split the dataset into training and test sets
- 23: Train CatBoost on the training set:
- 24: Input: Node embeddings $h(u), h(v)$ for each node pair
- 25: Output: Predicted probability y_{pred} of a link existing between u and v
- 26: Optimize CatBoost model based on metrics like accuracy, precision, recall, F1 score, AUC
- 27: **Step 5: Test and Evaluate the Model**
- 28: Use the trained CatBoost model to predict links in the test set
- 29: Evaluate model performance using precision, recall, F1 score, accuracy, and AUC
- 30: **Step 6: Output the predicted link probabilities for node pairs**

5.2.4 Algorithm Explanation

This subsection presents the algorithmic steps of our proposed MetaLPDGI-CatBoost model for link prediction. Consider a social network graph $G(V, E)$ that consists of the set of nodes V and edges E as input. We want to predict the links between the two nodes for the output. The step-by-step explanation of the proposed algorithm is discussed below:

1. **Construct the Multi-step Connectivity Matrix:** For each pair of nodes u and v , the algorithm computes the power of the adjacency matrix $A_{u,v}^k$, where k represents the number of steps in the graph traversal. After that, using the centrality measures like degree, betweenness, or closeness centrality, the function $\mathbf{f}(C(u), C(v))$ is calculated based on the centrality values of the nodes. The Multi-step Connectivity Matrix element for each node pair is computed as a weighted combination of adjacency powers and centrality functions, providing a refined measure of the relationship between nodes.
2. **Input the Connectivity Matrix into Data Graph Infomax (DGI):** After assigning the weights, compute the connectivity matrix $M_{connect}$ for each pair of nodes u, v by linearly combining the features using the meta-learning weights. The Multi-step Connectivity Matrix is then passed as input to DGI. DGI is an unsupervised algorithm that learns node embeddings by maximizing mutual information between the local node features and a global graph representation. The algorithm computes node embeddings Z_u by optimizing for each node u in the graph, using DGI's optimization objective.
3. **Prepare the Dataset for Link Prediction:** After obtaining the embeddings, a dataset is created where each node pair u, v is represented by the concatenation of their embeddings $Z(u)$ and $Z(v)$. We constructed positive samples (i.e., pairs with existing links) and negative samples (i.e., pairs without links) to form a

labeled dataset for classification.

4. **Train CatBoost Classifier for Link Prediction:** The CatBoost classifier is trained on the dataset using the node embeddings as input features. CatBoost is a gradient-boosting algorithm that works well on structured datasets. The classifier is trained to predict whether a link exists between node pairs by optimizing metrics like accuracy, precision, recall, F1 score, and AUC.
5. **Test and Evaluate the Model:** After training, the model is tested on unseen data, and its performance is evaluated using standard metrics for classification tasks.
6. **Output Predicted Links:** The final output is the predicted probability that a link exists between given node pairs, providing a solution for the link prediction task.
7. **Computational Complexity:** The computational complexity of the proposed MetaLP-DGI algorithm is analyzed by considering its three major components: the multi-step centrality-aware connectivity matrix, the Deep Graph Infomax (DGI) embedding module, and the CatBoost classifier. The construction of the multi-step connectivity matrix involves computing powers of the adjacency matrix up to a predefined step k and integrating centrality measures, resulting in a complexity of $O(k \cdot n \cdot d^k + nm)$, where n is the number of nodes, m is the number of edges, and d is the average degree of the graph. Among the centrality measures, betweenness centrality is the most computationally intensive algorithm in $O(nm)$ time. The DGI embedding step operates with a complexity of $O(mF + nF^2)$, where F is the feature dimension, leveraging sparse operations on graph structures. Finally, the CatBoost classifier, trained on s node pairs with $2F$ -dimensional input vectors, incurs an $O(sF \log s)$ complexity during training. The overall time complexity of the MetaLP-DGI pipeline is $O(k \cdot n \cdot d^k + nm + mF + nF^2 + sF \log s)$. This highlights the model's scalabil-

ity, especially for sparse graphs with controlled k , and demonstrates its practical efficiency for real-time or large-scale link prediction tasks.

Table 5.1: Brief Statistical Description of Various Social Network Datasets used in this study

Dataset	Total Nodes	Total Edges	Average Degree	Clustering Coefficient
Facebook [111]	4,039	88,234	43.7	0.605
Grid [112]	4,941	9,806	3.97	0.001
FB-TV [112]	7,469	103,731	27.77	0.136
Football [113]	115	613	10.65	0.403
Email-University [113]	1,133	5,451	9.62	0.220
Social-Wellesley [113]	1,466	13,888	18.94	0.254
Epinions [114]	75,879	508,837	13.41	0.137
PolBlogs [115]	1,490	16,715	22.43	0.320
Amazon [113]	7,650	238,163	22.43	0.320
Protein-Protein Interaction [116]	1,767	43,085	21.31	0.206

This paper utilizes ten real-world networks from diverse domains with unique topological structures. The detailed properties of these networks are provided in Table 5.1. The Facebook dataset contains 4,039 nodes and 88,234 edges. It has a high average degree of 43.7 and a significant clustering coefficient of 0.605, indicating strong community structures. The Grid network represents a power grid with 4,941 nodes and 9,806 edges. This network has a low average degree of 3.97 and a minimal clustering coefficient (0.001), reflecting a sparse and highly structured topology. The next dataset is the FB-TV, a social network of users from a TV-based community with 7,469 nodes and 103,731 edges. It has a moderate clustering coefficient of 0.136 and an average degree of 27.77, showing some degree of clustering. Furthermore, a network of football teams with 115 nodes and 613 edges represents matches between teams. It has a high clustering coefficient (0.403) and an average degree of 10.65, indicating that teams are grouped into highly interconnected communities. An email communication network within a university consisting of 1,133 nodes and 5,451 edges. This network has a clustering coefficient of 0.220 and an average degree of 9.62, capturing moderate clustering in communication patterns. A social network among Wellesley College students, with 1,466 nodes and 13,888 edges. It has an average degree of 18.94 and a clustering coefficient of 0.254, reflecting social groups within the community. The

Table 5.2: Evaluation Metrics for Link Prediction

Metric	Formula	Description
Prec	$\frac{TP}{TP+FP}$	Proportion of true positives among predicted positives.
Recall (Sensitivity)	$\frac{TP}{TP+FN}$	Proportion of true positives among actual positives.
F1 Score	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	Harmonic mean of precision and recall.
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Ratio of correct predictions to total instances.
AUC	$\int_0^1 \text{TPR}(x) dx$	Area under the ROC curve, indicating class separability.

Epinions is a sizeable online review platform network comprising 75,879 nodes and 508,837 edges. It has an average degree of 13.41 and a clustering coefficient of 0.137, with moderate clustering and high connectivity. A political blog network with 1,490 nodes and 16,715 edges. This network has an average degree of 22.43 and a clustering coefficient of 0.320, representing political communities with interconnected blogs. The Amazon dataset represents a product co-purchasing network where nodes correspond to products and edges indicate frequent co-purchase relationships. This dataset contains approximately 334,863 nodes and 925,872 edges. The PPI dataset is a biological network where nodes represent proteins and edges indicate experimentally validated physical interactions. This dataset comprises approximately 3,890 nodes and 76,584 edges, resulting in a relatively sparse yet complex graph.

5.3 Experimental Result and Analysis

This section presents a detailed analysis of the experiment we performed. To evaluate the performance of our proposed MetaLP-DGI approach in comparison to other link prediction algorithms across various datasets, we employed distinct evaluation metrics, as shown in Table 5.2. Precision (Prec) measures the proportion of correctly predicted links to the total number of predicted links. The following evaluation metric is Recall, which measures the propagation of actual positive links correctly identified by the

model. The F1 score is the harmonic mean of precision and Recall. Accuracy (Acc) measures how well a model correctly predicts the existing and non-existent links within the network. In the context of link prediction, the Area under the curve (AUC) is used to evaluate the classifiers' ability to distinguish between true and false links in a network, with higher AUC values indicating better model performance.

We compared the performance of the proposed link prediction model (MetaLP-DGI) against some recently proposed models such as link prediction model based on geodesic distance measure using various machine learning classifiers (LPGD-ML) [117], Ensemble model-based link prediction of complex networks("EMLP") [118], link prediction using deep walk (LP-Deep-Walk) [119], link prediction in complex networks using node centrality and light gradient boosting machine ("LP-LGBM") [120], link prediction based on local structure and node information along local paths ("LSNI") [48], and Gsvaelp: integrating graphsage and variational autoencoder for link prediction [121].

Further, we also performed an ablation study. As a part of the ablation study, we used all the strategies of the proposed model, namely, creating a connectivity matrix, feature extraction using DGI, labeling and balancing the positive and negative edge samples, and finally, classification using the CatBoost ensemble. Here, we used all steps of the proposed model except classification. Instead of classification by the CatBoost ensemble classifier, we utilize several baseline machine learning classifiers, including logistic regression (LR), Naive Bayes (NB), Support Vector Machine (SVM), and AdaBoost (ADB). We name these classifiers for link prediction as LPDGI-LR(link prediction using Logistic Regression), LPDGI-NB (link prediction using Naive Bayes), LPDGI-SVM (link prediction using Support Vector Machine), LPDGI-ADB (link prediction using AdaBoost), as the part of an ablation study. The results obtained from the proposed algorithm are compared with those from other comparing methods and are presented in tables from Table 5.3 to Table 5.8. We used Python programming in Colab Pro and its environment for simulations. The scikit-learn and networkx li-

braries are utilized for various machine learning classifiers, similarity measures, and popularity measures. All the experiments were carried out on personal computers with an Intel Core i5-1005G1 CPU with 8GB of RAM and a 64-bit operating system. Below, we present the hyperparameter studies for the proposed work on all datasets.

5.3.1 Hyperparamter Tunning

In this section, we conduct experiments to evaluate the performance of the proposed link prediction model with various hyperparameters. These include the embedding dimensions (d), walk length (l), seed size (k), and the number of layers (L) in the DGI embedding. Additionally, we examine the impact of the number of estimators (N-estimators) and the learning rate (eta) of the CatBoost classifier. We have used all ten datasets to perform this experiment and get the optimal results. We change the embedding dimensions from a series of [2, 4, 8, 16, 32, 64, 256], and the best AUC score is obtained when the embedding dimension (d) is 32, as shown in Fig. 5.2a. When the embedding dimension changes from 128 to 256, the MetaLP-DGI suddenly drops the AUC-Score. Fig. 5.2b shows the AUC change in the walk length (l) of DGI embedding, where the DGI model Walk length is defined as the series of random walks starting from the same node. The best AUC is obtained when the walk length is $w = 10$. The next hyperparameter is the Seed Size, which refers to the number of initial nodes (or starting points) selected in the social network to initiate link prediction or diffusion processes. The seed nodes serve as influential points from which information or relationships are expanded, assessed, or predicted within the network. For link prediction, increasing the seed size can potentially improve the accuracy of predictions by capturing more structural information and enhancing the ability to infer connections. In this chart, as the seed size $k = 30$ grows, the model generally shows improved performance (higher AUC values), indicating that a more extensive set of initial nodes enables the model to capture complex interactions in each network better, as shown in Fig. 5.2c.

The next important hyperparameter of DGI embedding is the number of layers (L). It typically refers to the depth of the neural network architecture used to encode node representations in the graph. Specifically, each layer in the DGI model represents a graph convolutional layer, such as a Graph Convolutional Network (GCN) layer, that aggregates information from neighboring nodes to capture structural patterns and node features. The Fig. 5.2d chart shows the relationship between the number of layers (L) in a model and the Area Under the Curve (AUC) performance metric across various datasets. The x-axis represents the number of layers (ranging from $L=1$ to $L=3$), while the y-axis shows the AUC values. Each line corresponds to a different dataset, such as Facebook, Grid, FB-TV, and others. As the number of layers increases, most datasets show a gradual improvement in AUC, indicating that deeper models generally lead to better link prediction performance in these datasets. However, the level of improvement varies by dataset, with some, like Email-University and Epinions, showing more significant gains than others, like Football and Grid. The highest AUC values we observed at $L=3$ for most datasets suggest that a three-layer model may offer the optimal complexity for this task. To achieve optimal performance with the proposed model, we conducted a series of experiments to fine-tune the hyperparameters of the CatBoost classifier. Figure 5.2e illustrates the relationship between accuracy and the learning rate (denoted as η). The η parameter, representing the learning rate, plays a critical role in controlling the speed of the gradient descent process in boosting algorithms. Its primary function is to mitigate the risk of overfitting by shrinking the feature weights, thereby making the boosting process more conservative. Our experiments indicate that the highest accuracy is achieved when η is set to 0.3. The following hyperparameter change in the CatBoost classifier is the N-estimators, which refers to the number of boosting rounds or decision trees to be built. It is the number of runs that the model will try to learn. We check the model's accuracy on different estimators, and the best result is obtained on $n = 300$, shown in Fig. 5.2f. The best combination of parameters to improve the accuracy is $n = 300, \gamma = 0, \eta = 0.3$. The γ and

Table 5.3: Experimental results of different link prediction methods based on Prec, Recall, F1 Score, Acc, and AUC for the Facebook dataset.

Models	Prec	Recall	F1 Score	Acc	AUC
LPDGI-LR	0.546 ± 0.02	0.584 ± 0.03	0.563 ± 0.01	0.542 ± 0.02	0.561 ± 0.02
LPDGI-NB	0.542 ± 0.03	0.39 ± 0.04	0.46 ± 0.02	0.52 ± 0.02	0.52 ± 0.01
LPDGI-SVM	0.534 ± 0.01	0.65 ± 0.02	0.58 ± 0.01	0.53 ± 0.02	0.54 ± 0.01
LPDGI-ADB	0.583 ± 0.01	0.756 ± 0.02	0.58 ± 0.02	0.742 ± 0.01	0.58 ± 0.01
LPDGI-RF	0.672 ± 0.02	0.69 ± 0.01	0.68 ± 0.02	0.66 ± 0.01	0.67 ± 0.02
LP-Deep-Walk	0.564 ± 0.03	0.55 ± 0.01	0.55 ± 0.02	0.56 ± 0.02	0.55 ± 0.01
LPGD-ML [117]	0.705 ± 0.01	0.61 ± 0.02	0.65 ± 0.01	0.66 ± 0.02	0.67 ± 0.01
EMLP [118]	0.682 ± 0.01	0.67 ± 0.02	0.67 ± 0.01	0.686 ± 0.02	0.67 ± 0.01
LP-LGBM [120]	0.725 ± 0.02	0.70 ± 0.01	0.711 ± 0.02	0.744 ± 0.02	0.75 ± 0.01
LSNI [48]	0.754 ± 0.01	0.742 ± 0.02	0.736 ± 0.01	0.773 ± 0.02	0.746 ± 0.01
GSVAELP [121]	0.715 ± 0.01	0.688 ± 0.02	0.703 ± 0.01	0.733 ± 0.02	0.752 ± 0.01
MetaLP-DGI	0.743 ± 0.01	0.749 ± 0.02	0.739 ± 0.01	0.740 ± 0.02	0.782 ± 0.01

the accuracy are inversely proportional to each other, which means the lower the value of gamma, the better the accuracy. In summary, optimizing these hyperparameters is essential to balance performance and computational efficiency in our proposed model. The hyper-tuning allows the model to capture network structures effectively, resulting in improved link prediction accuracy.

5.3.2 Comparative Performance Study

In this section, we compare the performance of our approach to that of other relevant algorithms using multiple evaluation metrics. We evaluate the performance of our proposed MetaLP-DGI approach and other methods considered for each dataset by using a variety of evaluation metrics. The results below highlight the evaluation metrics, with the best outcomes for each dataset emphasized in bold. In the tables, the "" symbol represents the standard deviation of the respective metric values across multiple runs or experimental trials. It indicates the variability or consistency of the model's performance. Table 5.3 shows the experimental results of different link prediction methods evaluated on the Facebook dataset. The proposed model MetaLP-DGI achieved the highest Recall (0.749 ± 0.02), F1 Score (0.739 ± 0.01), and AUC (0.782 ± 0.01). This model captures accurate links while balancing precision and recall, leading to high

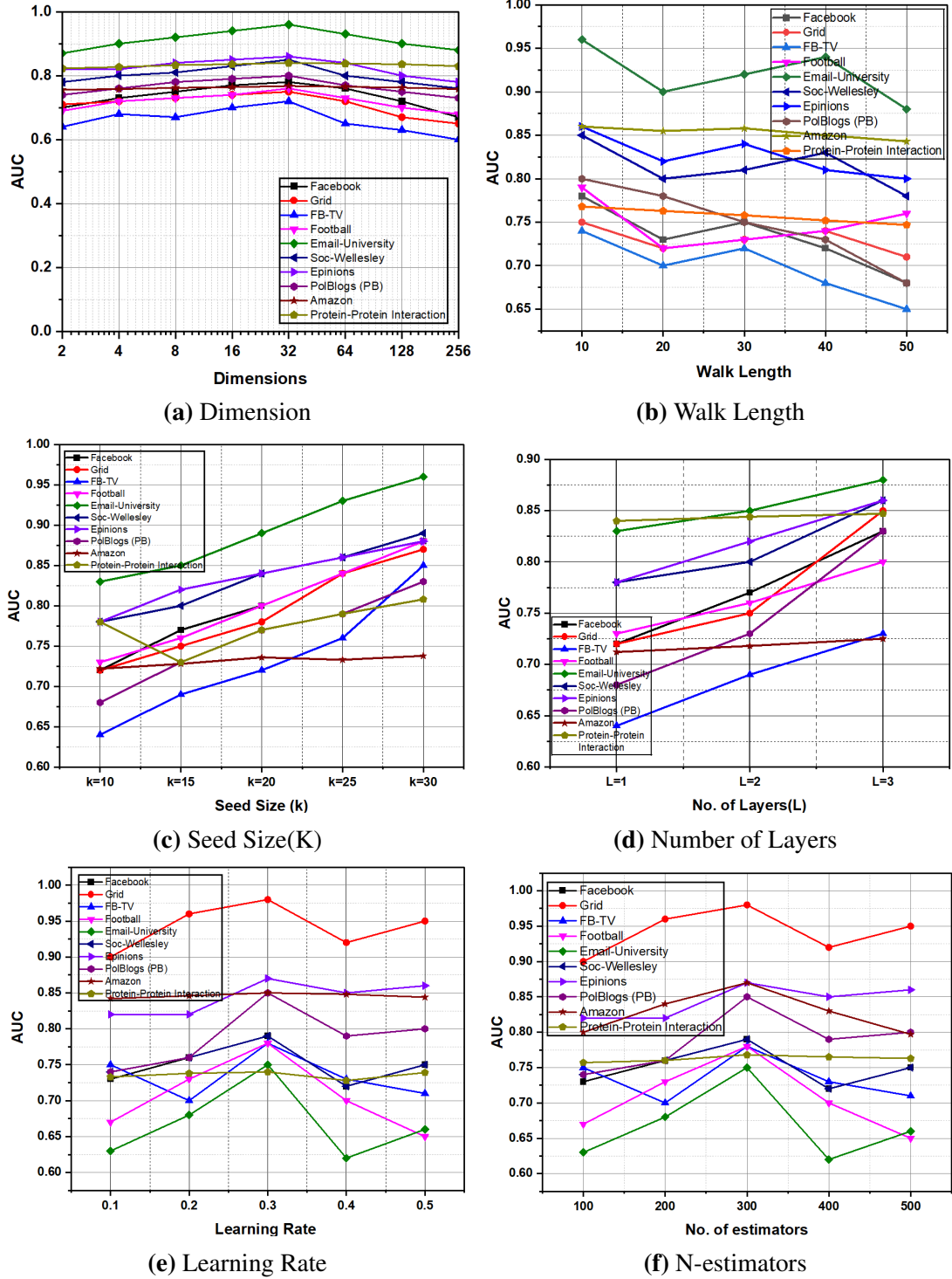


Fig. 5.2: Parameter Sensitivity on all Datasets

Table 5.4: Experimental results of different link prediction methods based on Prec, Recall, F1 Score, Acc, and AUC for the Email-University dataset.

Models	Prec	Recall	F1 Score	Acc	AUC
LPDGI-LR	0.662 ± 0.03	0.662 ± 0.02	0.644 ± 0.01	0.648 ± 0.01	0.724 ± 0.01
LPDGI-NB	0.735 ± 0.05	0.741 ± 0.01	0.723 ± 0.01	0.763 ± 0.02	0.732 ± 0.01
LPDGI-SVM	0.649 ± 0.011	0.741 ± 0.02	0.720 ± 0.01	0.728 ± 0.01	0.715 ± 0.02
LPDGI-ADB	0.834 ± 0.023	0.834 ± 0.01	0.765 ± 0.01	0.688 ± 0.01	0.765 ± 0.02
LPDGI-RF	0.714 ± 0.017	0.728 ± 0.01	0.73 ± 0.01	0.762 ± 0.01	0.754 ± 0.01
LP-Deep-Walk	0.782 ± 0.051	0.757 ± 0.02	0.77 ± 0.01	0.783 ± 0.01	0.748 ± 0.01
LPGD-ML [117]	0.742 ± 0.01	0.810 ± 0.01	0.782 ± 0.01	0.812 ± 0.01	0.768 ± 0.01
EMLP [118]	0.705 ± 0.01	0.765 ± 0.01	0.755 ± 0.01	0.775 ± 0.02	0.768 ± 0.01
LP-LGBM [120]	0.782 ± 0.01	0.815 ± 0.02	0.75 ± 0.01	0.712 ± 0.01	0.840 ± 0.01
LSNI [48]	0.800 ± 0.02	0.728 ± 0.01	0.821 ± 0.02	0.736 ± 0.01	0.85 ± 0.01
GSVAELP [121]	0.823 ± 0.023	0.806 ± 0.02	0.746 ± 0.01	0.805 ± 0.02	0.834 ± 0.01
MetaLP-DGI	0.817 ± 0.15	0.822 ± 0.04	0.782 ± 0.01	0.807 ± 0.02	0.862 ± 0.01

classification accuracy. Another top performer with a Precision of 0.725 ± 0.02 and a solid AUC of 0.75 ± 0.01 . It balances precision and recall well and achieves high accuracy, making it competitive with other strong models. MetaLP-DGI and LSNI are the standout models on this dataset, with MetaLP-DGI providing the best overall balance and generalization, as indicated by its superior recall and AUC. However, the LSNI model achieves a precision of 0.755, outperforming the other methods in this metric.

Table 5.4 presents the experimental results of various link prediction models applied to the Email-University dataset. Our proposed model, MetaLP-DGI, achieves the highest AUC (0.862 ± 0.01), which signifies its strong overall performance in distinguishing between positive and negative links. It also achieves a high Recall (0.822 ± 0.04), suggesting it is highly effective at identifying actual links, with decent Accuracy (0.807 ± 0.02) and Precision (0.817 ± 0.15). GSVAELP performs well in terms of Precision (0.823 ± 0.023), the highest among all models, meaning it has a low rate of false positives. It also shows a strong balance between Recall (0.806 ± 0.02), Accuracy (0.81 ± 0.02), and AUC (0.83 ± 0.01). LSNI excels in F1 Score (0.82 ± 0.02), indicating a good balance between precision and recall, reflecting robust performance in identifying true positives and minimizing false positives. LP-LGBM and LP-Deep-Walk also show strong results, particularly in AUC (0.84 ± 0.01) for LP-LGBM and

Table 5.5: Experimental results of different link prediction methods based on Prec, Recall, F1 score, Acc, and AUC for the FB-TV dataset.

Models	Prec	Recall	F1 Score	Acc	AUC
LPDGI-LR	0.50 ± 0.01	0.51 ± 0.02	0.50 ± 0.01	0.53 ± 0.01	0.52 ± 0.01
LPDGI-NB	0.52 ± 0.01	0.56 ± 0.01	0.54 ± 0.01	0.53 ± 0.02	0.51 ± 0.01
LPDGI-SVM	0.54 ± 0.02	0.52 ± 0.01	0.53 ± 0.02	0.56 ± 0.01	0.54 ± 0.02
LPDGI-ADB	0.58 ± 0.02	0.51 ± 0.02	0.54 ± 0.01	0.56 ± 0.01	0.57 ± 0.01
LPDGI-RF	0.57 ± 0.01	0.55 ± 0.01	0.56 ± 0.02	0.58 ± 0.01	0.57 ± 0.02
LP-Deep-Walk	0.60 ± 0.01	0.63 ± 0.02	0.61 ± 0.01	0.63 ± 0.02	0.60 ± 0.01
LPGD-ML [117]	0.63 ± 0.02	0.55 ± 0.02	0.59 ± 0.01	0.61 ± 0.02	0.61 ± 0.01
EMLP [118]	0.59 ± 0.01	0.62 ± 0.01	0.691 ± 0.02	0.62 ± 0.02	0.61 ± 0.01
LP-LGBM [120]	0.732 ± 0.01	0.69 ± 0.02	0.673 ± 0.01	0.65 ± 0.02	0.63 ± 0.01
LSNI [48]	0.72 ± 0.01	0.64 ± 0.02	0.68 ± 0.02	0.661 ± 0.01	0.69 ± 0.02
GSVAELP [121]	0.723 ± 0.01	0.64 ± 0.02	0.68 ± 0.02	0.711 ± 0.01	0.665 ± 0.02
MetaLP-DGI	0.719 ± 0.01	0.720 ± 0.01	0.684 ± 0.02	0.727 ± 0.01	0.703 ± 0.02

F1 Score (0.77 ± 0.01) for LP-Deep-Walk, indicating their moderate but reliable performance in link prediction. Models like LPDGI-ADB, EMLP, and LP-Deep-Walk perform moderately well, especially in certain metrics, making them suitable for specific cases depending on the desired balance of precision and recall.

Table 5.5 shows the evaluation comparison of The Fb-TV dataset. In this comparison, our proposed model MetaLP-DGI performs well with the highest AUC (0.703 ± 0.02) and the highest Accuracy (0.727 ± 0.01), indicating that it performs better than other models in distinguishing between positive and negative links. It also performed well in Recall (0.720 ± 0.01), making it the most accurate predictor of positive links. The LP-LGBM model showed the highest Precision (0.732 ± 0.01), which is highly reliable in predicting positive links, though it has a slightly lower Recall (0.69 ± 0.02) compared to MetaLP-DGI. On the other hand, the EMLP model performed well in terms of F1 score (0.691 ± 0.02), which balances both Precision and Recall. Its overall performance across metrics is competitive with other models but slightly lower in terms of AUC and Accuracy compared to MetaLP-DGI and LP-LGBM.

Table 5.6 presents the experimental results of various link prediction models applied to the Fb-Political dataset. MetaLP-DGI performs strongly across most metrics, achieving the best recall (0.69), F1 score (0.71), and accuracy (0.723), with an AUC

Table 5.6: Experimental results of different link prediction methods based on Prec, Recall, F1 score, Acc, and AUC for the Fb-Political dataset.

Models	Prec	Recall	F1 Score	Acc	AUC
LPDGI-LR	0.53 ± 0.01	0.48 ± 0.01	0.50 ± 0.01	0.54 ± 0.02	0.53 ± 0.01
LPDGI-NB	0.59 ± 0.01	0.60 ± 0.02	0.59 ± 0.01	0.62 ± 0.01	0.60 ± 0.01
LPDGI-SVM	0.54 ± 0.02	0.60 ± 0.01	0.57 ± 0.01	0.56 ± 0.02	0.54 ± 0.02
LPDGI-ADB	0.74 ± 0.01	0.67 ± 0.02	0.68 ± 0.01	0.67 ± 0.01	0.68 ± 0.02
LPDGI-RF	0.56 ± 0.01	0.52 ± 0.01	0.54 ± 0.01	0.56 ± 0.02	0.57 ± 0.01
LPGD-ML [117]	0.67 ± 0.02	0.58 ± 0.02	0.62 ± 0.01	0.65 ± 0.01	0.65 ± 0.02
LP-Deep-Walk	0.66 ± 0.01	0.63 ± 0.01	0.66 ± 0.01	0.64 ± 0.02	0.65 ± 0.01
EMLP [118]	0.63 ± 0.01	0.55 ± 0.01	0.59 ± 0.02	0.62 ± 0.01	0.61 ± 0.02
LP-LGBM [120]	0.66 ± 0.02	0.65 ± 0.01	0.65 ± 0.02	0.66 ± 0.02	0.66 ± 0.01
LSNI [48]	0.68 ± 0.03	0.67 ± 0.01	0.692 ± 0.01	0.658 ± 0.01	0.70 ± 0.02
GSVAELP	0.70 ± 0.01	0.67 ± 0.02	0.66 ± 0.01	0.67 ± 0.02	0.708 ± 0.01
MetaLP-DGI [121]	0.72 ± 0.02	0.69 ± 0.01	0.71 ± 0.01	0.723 ± 0.02	0.706 ± 0.01

of 0.706. LPDGI-ADB delivers the highest precision (0.74) with consistent performance across other metrics, though its recall and AUC are slightly lower than MetaLP-DGI. GSVAELP and LSNI also show strong performance, particularly in terms of precision, recall, and AUC. GSVAELP achieves an AUC of 0.708, the highest in this table, indicating good discrimination ability. Other models, such as LPDGI-NB and LP-LGBM, demonstrate moderate performance but fall behind compared to the top-performing methods. The performance standards determined on the Social-Wellesley network are shown in Table 5.7. The proposed MetaLP-DGI achieves the highest Recall (0.837), F1 score (0.824), and Accuracy (0.749), indicating its strong performance in correctly predicting links, with low variation across runs (standard deviations around 0.01–0.02). The Deep walk-based method is the baseline method of our proposed method. It helps us to calculate the network embedding of the graph; the LP-Deep-Walk and LSNI also perform well, with LP-Deep-Walk showing high Precision (0.742) and recall (0.806), and LSNI having F1 score (0.803) and recall (0.817) values. Table 5.8 represents the performance evaluation on the Epinions Dataset. Each model is evaluated based on several performance metrics. The proposed model MetaLP-DGI achieved the best results in terms of precision (0.823), recall (0.845), and accuracy (0.827). This suggests that the MetaLP-DGI model is the most effective for link pre-

diction on the Epinions dataset. The other models, such as LP-Deep-Walk and LPGD-ML, demonstrate strong performances across the board, particularly in accuracy, with LP-Deep-Walk achieving a high value of 0.832 and LPGD-ML achieving the highest F1 Score (0.837). The low standard deviation across all models indicates consistent performance over multiple runs, with MetaLP-DGI showing the most stable results.

Table 5.7: Experimental results of different link prediction models based on Prec, Recall, F1 score, Acc, and AUC for the Social-Wellesley dataset.

Models	Prec	Recall	F1 Score	Acc	AUC
LPDGI-LR	0.62 ± 0.02	0.70 ± 0.01	0.652 ± 0.01	0.65 ± 0.01	0.68 ± 0.01
LPDGI-NB	0.65 ± 0.01	0.72 ± 0.02	0.672 ± 0.01	0.64 ± 0.02	0.67 ± 0.01
LPDGI-SVM	0.64 ± 0.02	0.718 ± 0.01	0.663 ± 0.02	0.67 ± 0.01	0.69 ± 0.02
LPDGI-ADB	0.66 ± 0.01	0.73 ± 0.02	0.690 ± 0.01	0.68 ± 0.01	0.68 ± 0.02
LPDGI-RF	0.65 ± 0.01	0.75 ± 0.01	0.703 ± 0.01	0.69 ± 0.02	0.67 ± 0.01
LP-Deep-Walk	0.742 ± 0.02	0.806 ± 0.01	0.748 ± 0.02	0.725 ± 0.01	0.729 ± 0.02
LPGD-ML [117]	0.67 ± 0.01	0.729 ± 0.01	0.78 ± 0.01	0.732 ± 0.02	0.738 ± 0.01
EMLP [118]	0.65 ± 0.02	0.72 ± 0.01	0.68 ± 0.01	0.64 ± 0.02	0.66 ± 0.01
LP-LGBM [120]	0.69 ± 0.01	0.76 ± 0.01	0.72 ± 0.01	0.735 ± 0.01	0.763 ± 0.02
LSNI [48]	0.722 ± 0.01	0.817 ± 0.01	0.803 ± 0.01	0.736 ± 0.02	0.751 ± 0.01
GSVAELP [121]	0.727 ± 0.01	0.831 ± 0.01	0.817 ± 0.02	0.723 ± 0.02	0.742 ± 0.01
MetaLP-DGI	0.740 ± 0.01	0.837 ± 0.01	0.824 ± 0.01	0.749 ± 0.01	0.755 ± 0.02

Table 5.8: Experimental results of different link prediction methods based on Prec, Recall, F1 score, Acc, and AUC for the Epinions Dataset.

Models	Prec	Recall	F1 Score	Acc	AUC
LPDGI-LR	0.66 ± 0.02	0.74 ± 0.01	0.71 ± 0.01	0.75 ± 0.01	0.72 ± 0.01
LPDGI-NB	0.56 ± 0.01	0.62 ± 0.02	0.60 ± 0.02	0.66 ± 0.02	0.69 ± 0.01
LPDGI-SVM	0.71 ± 0.02	0.74 ± 0.01	0.72 ± 0.02	0.69 ± 0.02	0.71 ± 0.01
LPDGI-ADB	0.74 ± 0.01	0.87 ± 0.02	0.77 ± 0.02	0.75 ± 0.02	0.73 ± 0.01
LPDGI-RF	0.72 ± 0.01	0.76 ± 0.01	0.73 ± 0.01	0.79 ± 0.01	0.77 ± 0.02
LP-Deep-Walk	0.78 ± 0.01	0.82 ± 0.01	0.78 ± 0.01	0.832 ± 0.01	0.77 ± 0.02
LPGD-ML [117]	0.78 ± 0.02	0.832 ± 0.02	0.837 ± 0.01	0.75 ± 0.02	0.802 ± 0.01
EMLP [118]	0.69 ± 0.02	0.78 ± 0.01	0.753 ± 0.01	0.74 ± 0.01	0.783 ± 0.02
LP-LGBM [120]	0.73 ± 0.01	0.74 ± 0.01	0.697 ± 0.01	0.73 ± 0.02	0.788 ± 0.01
LSNI [48]	0.810 ± 0.02	0.88 ± 0.01	0.827 ± 0.01	0.818 ± 0.01	0.807 ± 0.02
GSVAELP [121]	0.786 ± 0.02	0.88 ± 0.01	0.833 ± 0.01	0.828 ± 0.02	0.815 ± 0.01
MetaLP-DGI	0.823 ± 0.01	0.845 ± 0.01	0.827 ± 0.01	0.823 ± 0.01	0.827 ± 0.01

For the PolBlogs Dataset, Table 5.9 depicts the performance evaluation of the various models. In baseline methods, the notable performance includes the LPDGI-ADB and LP-Deep-Walk, with the highest F1 score of 0.823, whereas the LP-LGBM model

Table 5.9: Experimental results of different link prediction methods based on Prec, Recall, F1 score, Acc, and AUC for the PolBlogs Dataset.

Models	Prec	Recall	F1 Score	Acc	AUC
LPDGI-LR	0.58 ± 0.02	0.64 ± 0.01	0.64 ± 0.02	0.72 ± 0.01	0.68 ± 0.01
LPDGI-NB	0.65 ± 0.01	0.68 ± 0.02	0.64 ± 0.01	0.72 ± 0.01	0.70 ± 0.02
LPDGI-SVM	0.75 ± 0.01	0.78 ± 0.02	0.75 ± 0.02	0.78 ± 0.01	0.73 ± 0.01
LPDGI-ADB	0.817 ± 0.02	0.78 ± 0.01	0.812 ± 0.02	0.80 ± 0.01	0.80 ± 0.02
LPDGI-RF	0.75 ± 0.01	0.829 ± 0.02	0.823 ± 0.01	0.817 ± 0.02	0.79 ± 0.01
LP-Deep-Walk	0.834 ± 0.01	0.812 ± 0.01	0.819 ± 0.01	0.80 ± 0.01	0.80 ± 0.02
LPGD-ML [117]	0.825 ± 0.01	0.738 ± 0.01	0.78 ± 0.02	0.81 ± 0.01	0.814 ± 0.01
EMLP [118]	0.78 ± 0.02	0.807 ± 0.01	0.79 ± 0.01	0.79 ± 0.01	0.811 ± 0.02
LP-LGBM [120]	0.75 ± 0.01	0.832 ± 0.01	0.80 ± 0.02	0.79 ± 0.01	0.77 ± 0.01
LSNI [48]	0.827 ± 0.01	0.784 ± 0.02	0.793 ± 0.01	0.764 ± 0.01	0.804 ± 0.02
GSVAELP [121]	0.844 ± 0.01	0.795 ± 0.02	0.79 ± 0.01	0.740 ± 0.01	0.815 ± 0.02
MetaLP-DGI	0.857 ± 0.01	0.827 ± 0.01	0.819 ± 0.02	0.831 ± 0.01	0.829 ± 0.02

showed the highest recall value of 0.832. On the other hand, our proposed method, MetaLP-DGI, outperformed all other models with the highest precision of 0.857 and the Accuracy of 0.831, indicating the best performance metrics. The proposed method exhibits a competitive advantage by producing favorable results across diverse performance metrics under evaluation. The results also indicated that the selected node embedding produces the best results when integrated with machine learning classifiers other than XGB, like logistic regression, Random Forest, Support vector machine, Gradient, Adaboost, etc. Furthermore, our proposed method, MetaLP-DGI, consistently outperforms all chosen networks regarding AUC. Combining node embedding with ensemble machine-learning classifiers is responsible for our proposed strategy's consistent and better performance.

Table 5.10 showcases the evaluation of different link prediction techniques on the Amazon dataset. The findings reveal that MetaLP-DGI stands out with the highest Precision (0.855), F1 Score (0.828), and AUC (0.830), indicating its superior performance over other models. EMLP records the top Recall (0.833), while GSVAELP achieves the best Accuracy (0.828), highlighting that various models excel in distinct performance areas. Traditional embedding-based methods like LP-DeepWalk (AUC: 0.81) and LPGD-ML (AUC: 0.833) show competitive results, yet MetaLP-DGI sur-

Table 5.10: Experimental results of different link prediction methods based on Precision, Recall, F1 score, Accuracy, and AUC for the Amazon Dataset

Models	Prec	Recall	F1 Score	Acc	AUC
LPDGI-LR	0.62 ± 0.02	0.66 ± 0.01	0.64 ± 0.02	0.71 ± 0.01	0.69 ± 0.01
LPDGI-NB	0.67 ± 0.01	0.69 ± 0.02	0.66 ± 0.01	0.73 ± 0.01	0.71 ± 0.02
LPDGI-SVM	0.74 ± 0.01	0.79 ± 0.02	0.76 ± 0.02	0.79 ± 0.01	0.74 ± 0.01
LPDGI-ADB	0.81 ± 0.02	0.79 ± 0.01	0.813 ± 0.02	0.81 ± 0.01	0.81 ± 0.02
LPDGI-RF	0.76 ± 0.01	0.827 ± 0.02	0.826 ± 0.01	0.821 ± 0.02	0.80 ± 0.01
LP-Deep-Walk	0.835 ± 0.01	0.818 ± 0.01	0.821 ± 0.01	0.81 ± 0.01	0.81 ± 0.02
LPGD-ML [117]	0.828 ± 0.01	0.742 ± 0.01	0.785 ± 0.02	0.82 ± 0.01	0.833 ± 0.01
EMLP [118]	0.79 ± 0.02	0.833 ± 0.01	0.795 ± 0.01	0.807 ± 0.01	0.812 ± 0.02
LP-LGBM [120]	0.76 ± 0.01	0.817 ± 0.01	0.81 ± 0.02	0.805 ± 0.01	0.78 ± 0.01
LSNI [48]	0.830 ± 0.01	0.788 ± 0.02	0.798 ± 0.01	0.765 ± 0.01	0.807 ± 0.02
GSVAELP [121]	0.846 ± 0.01	0.798 ± 0.02	0.795 ± 0.01	0.828 ± 0.01	0.817 ± 0.02
MetaLP-DGI	0.855 ± 0.01	0.830 ± 0.01	0.828 ± 0.02	0.825 ± 0.01	0.830 ± 0.02

Table 5.11: Experimental results of different link prediction methods based on Prec, Recall, F1 Score, Acc, and AUC for the PPI Dataset.

Models	Prec	Recall	F1	Acc	AUC
LPDGI-LR	0.608 ± 0.02	0.65 ± 0.01	0.63 ± 0.02	0.704 ± 0.01	0.683 ± 0.01
LPDGI-NB	0.636 ± 0.01	0.68 ± 0.02	0.65 ± 0.01	0.72 ± 0.01	0.707 ± 0.02
LPDGI-SVM	0.723 ± 0.01	0.78 ± 0.02	0.752 ± 0.02	0.781 ± 0.01	0.73 ± 0.01
LPDGI-ADB	0.80 ± 0.02	0.78 ± 0.01	0.828 ± 0.02	0.80 ± 0.01	0.80 ± 0.02
LPDGI-RF	0.765 ± 0.01	0.830 ± 0.02	0.824 ± 0.01	0.820 ± 0.02	0.79 ± 0.01
LP-DeepWalk	0.832 ± 0.01	0.816 ± 0.01	0.820 ± 0.01	0.81 ± 0.01	0.81 ± 0.02
LPGD-ML [117]	0.827 ± 0.01	0.740 ± 0.01	0.784 ± 0.02	0.824 ± 0.01	0.814 ± 0.01
EMLP [118]	0.784 ± 0.02	0.807 ± 0.01	0.792 ± 0.01	0.794 ± 0.01	0.811 ± 0.02
LP-LGBM [120]	0.753 ± 0.01	0.834 ± 0.01	0.810 ± 0.02	0.79 ± 0.01	0.77 ± 0.01
LSNI [48]	0.829 ± 0.01	0.786 ± 0.02	0.796 ± 0.01	0.762 ± 0.01	0.806 ± 0.02
GSVAELP [121]	0.818 ± 0.01	0.796 ± 0.02	0.792 ± 0.01	0.740 ± 0.01	0.816 ± 0.02
MetaLP-DGI	0.826 ± 0.01	0.835 ± 0.01	0.826 ± 0.02	0.834 ± 0.01	0.830 ± 0.02

passes them in most metrics. Ensemble learning-based models such as LP-LGBM and LPDGI-RF also deliver strong outcomes but slightly lag behind MetaLP-DGI. These results underscore the effectiveness of deep learning-based approaches, particularly MetaLP-DGI, in enhancing link prediction accuracy on the Amazon dataset.

Table 5.11 displays the experimental outcomes of different link prediction techniques applied to the Protein-Protein Interaction (PPI) dataset, assessed through Precision (Prec), Recall, F1 Score, Accuracy (Acc), and Area Under the Curve (AUC). The data reveals that MetaLP-DGI attains the highest Recall (0.835), Accuracy (0.834),

and AUC (0.830), highlighting its robust predictive performance for PPI networks. LSNI records the top Precision (0.829), whereas LPDGI-ADB achieves the leading F1 Score (0.828), illustrating that various models excel in distinct metrics. Traditional embedding-based models like LP-DeepWalk (AUC: 0.81) and LPGD-ML (AUC: 0.814) perform commendably. Yet, MetaLP-DGI surpasses them in most metrics, underscoring the efficacy of centrality-aware transition matrices and DGI embeddings in link prediction. Models such as LP-LGBM and LPDGI-RF also perform admirably but are slightly outperformed by MetaLP-DGI. These results emphasize that deep learning-based methods, particularly MetaLP-DGI, provide superior predictive accuracy compared to traditional and ensemble-based machine learning models.

5.3.3 Relative Information Gain

In the context of link prediction, a relative information gain refers to the improvement in the desired performance metric that the model achieves compared to the existing and baseline models. It determines how effectively the proposed model predicts links within a network. Hence, the improvement in performance measure that a model achieves for a task compared to other models is known as the information gain. Along with the evaluation metrics like Precision (Prec), Recall, F1 score, Accuracy (Acc), and Area under the curve (AUC) score, we have also computed the relative information gain of our proposed model. Table 5.12 depicts the relative information gain achieved by the proposed model compared to the best baseline and recent literature regarding AUC for various datasets. The columns indicate the percentage improvement achieved by the proposed model over the best baseline methods and the most recent literature method for each dataset. For example, on the Facebook dataset, the proposed approach shows a 14.2% improvement over the best baseline LPDGI-RF and 2.1% over the best recent LSNI method. Even in more complex and large-scale networks like Amazon and PPI (Protein-Protein Interaction), it shows consistent gains over both traditional

Table 5.12: Relative Gain in AUC of the proposed approach against the best baseline and recent findings

Datasets	Against best Baseline (%)	Against best Recent literature (%)
Facebook	14.2(LPDGI-RF)	2.1 (LSNI)
Email-University	2.65 (LP-DeepWalk)	5.18 (EMLP)
Hamsterster	9.79 (LPDGI-RF)	2.71 (LSNI)
FB-TV	11.5 (LPDGI-ADB)	3.5 (LP-LGBM)
FB-Political	5.71 (LPDGI-NB)	7.3 (LPGD-ML)
Social-Wellesley	14.6 (LPDGI-ADB)	4.7 (LSNI)
Epinions	6.45 (LP-DeepWalk)	2.63 (LPGD-ML)
PolBlogs	12.4 (LPDGI-ADB)	3.73 (LP-LGBM)
Amazon	2.94 (LPGD-ML)	1.63 (GSVAELP)
Protein-Protein Interaction (PPI)	5.06 (LPDGI-RF)	2.68 (LSNI)

baselines and recent approaches, validating the model’s generalizability and robustness in both social and biological contexts. These results emphasize the superiority of MetaLP-DGI in enhancing link prediction performance across a diverse range of real-world datasets.

5.3.4 Quantify Efficiency

This section provides a comprehensive evaluation of the proposed MetaLP-DGI model in terms of computational complexity, training time, inference time, and memory usage, compared to some baseline and recent state-of-the-art link prediction models as depicted in Table 5.13. The training time refers to the model’s total time (in seconds) to learn from the training data and optimize its parameters. A shorter training time typically reflects computational efficiency and faster convergence. In our case, this metric helps evaluate how quickly MetaLP-DGI and baseline models can learn link prediction patterns from the graph structure and node features.

The inference time indicates how long (in seconds) it takes for the trained model to make predictions on unseen data. This is a critical factor in real-time or large-scale applications where rapid decision-making is essential. Lower inference time

Table 5.13: Model Efficiency Comparison in terms of Training Time, Inference Time, and Memory Usage

Model	Training Time (s)	Inference Time (s)	Memory Usage (MB)
LP-DeepWalk	35.2	4.8	220
LPDGI-ADB	48.5	6.1	310
EMLP	62.4	7.9	350
MetaLP-DGI	54.3	5.5	295

suggests better runtime performance during deployment. The last is the memory usage, which measures the peak amount of RAM (in megabytes) used by the model during training and inference. Empirical results presented in Table 5.13 demonstrate that while MetaLP-DGI incurs slightly higher training time due to its ensemble and meta-learning components, it achieves competitive inference speed and stable memory utilization, both scalable and efficient. These observations reinforce the practical utility of our model, particularly in real-world settings where both accuracy and runtime performance are critical.

5.3.5 Average performance of each comparing method of link prediction on all datasets

Furthermore, to assess the effectiveness of the proposed link prediction model, we evaluate the average F1 score and average AUC of the proposed method on each dataset against those of all other comparison methods. The bar graphs presented in Figs. 5.3 and 5.4 illustrate a comparative performance analysis of various link prediction models based on the average F1 score and average AUC (Area Under the Curve), respectively. These metrics are essential for assessing the models' effectiveness in predicting links within a network, with a focus on both discrimination power and balanced accuracy. Below is a detailed description of each chart and its implications.

1. Average F1 Score: Fig. 5.3 showcases the models' performance in terms of the average F1 Score. This metric balances precision (correctly predicted links

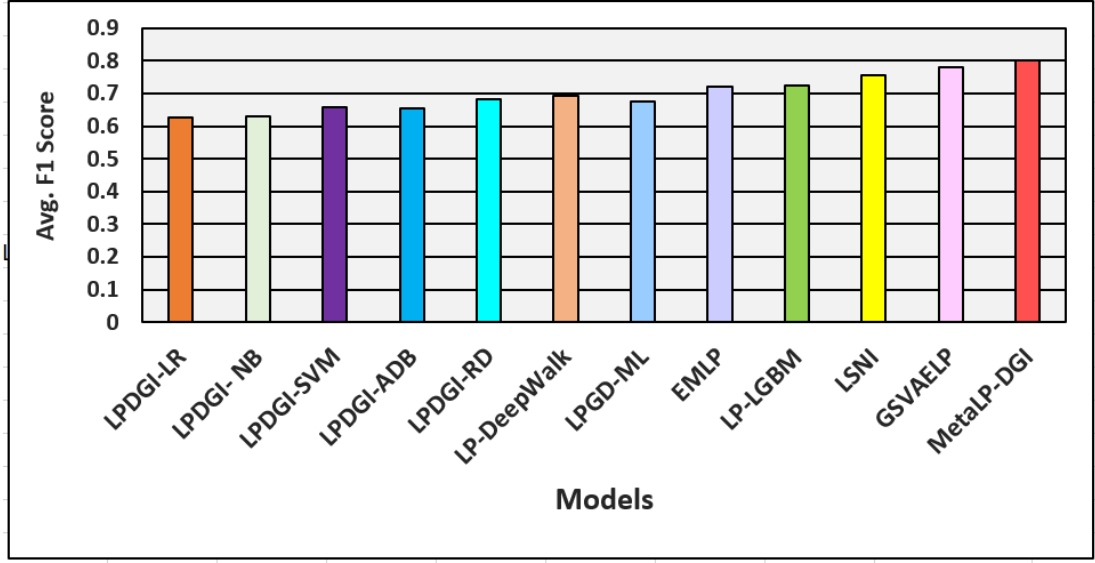


Fig. 5.3: Average F1 Score for each model on all the datasets

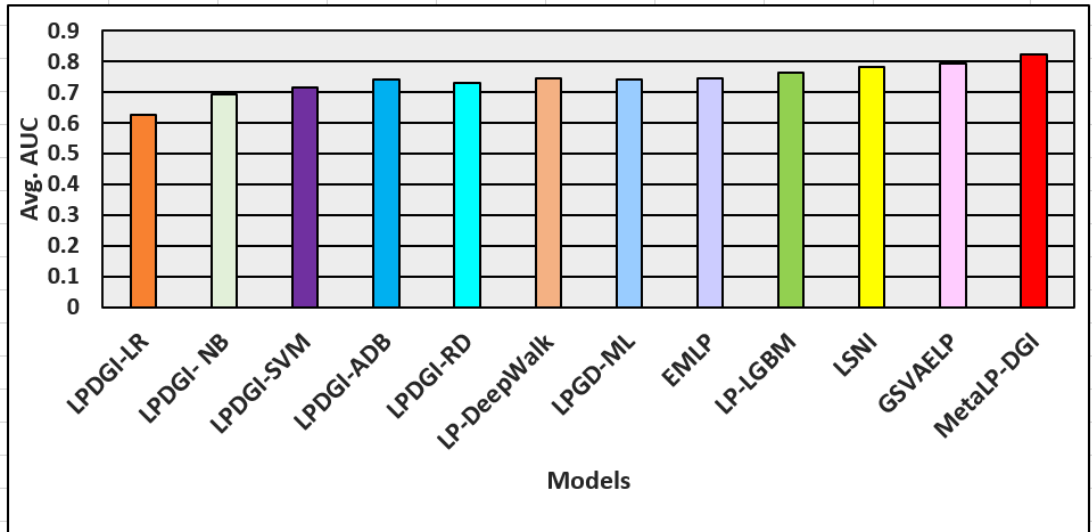


Fig. 5.4: Average AUC score for each model on all the datasets

among predicted positives) and recall (correctly predicted links among actual links). A higher F1 Score indicates better performance in terms of both precision and recall, which is crucial for models applied to datasets with class imbalance. Similar to the AUC chart, MetaLP-DGI exhibits the highest F1 Score, followed closely by GSVAELP and LSNI. This reflects MetaLP-DGI's consistency in achieving a balance between precision and recall, further validating its performance for accurate link prediction. The F1 Score results underscore MetaLP-

DGI's capability to maintain high accuracy while minimizing false positives and false negatives. This is particularly valuable for applications where false predictions can lead to significant implications, such as social network analysis or recommendation systems. The strong performance of MetaLP-DGI in both AUC and F1 Score suggests that its architecture effectively balances the exploration of the graph structure (through DGI embeddings) and classifying links with a minimal bias towards either class.

2. Average AUC (Area Under the Curve): Fig. 5.4 presents the average AUC values for each model, where AUC measures the model's ability to distinguish between positive (existing links) and negative (non-existent links) cases. A higher AUC value indicates a better model in terms of correctly ranking edges as true links or non-links. The MetaLP-DGI model achieves the highest AUC score among all models, closely followed by GSV AELP and LSNI, highlighting its strong capability in accurately distinguishing link pairs. Traditional machine learning-based models like LPDGI-LR (Logistic Regression) and LPDGI-NB (Naive Bayes) exhibit relatively lower AUC scores, indicating that these methods might be less capable of capturing complex patterns in the graph structure compared to the more sophisticated approaches like MetaLP-DGI. The results indicate that the MetaLP-DGI framework effectively captures both local and global graph structures, potentially due to its integration of centrality measures and Deep Graph Infomax (DGI) embeddings. This high AUC score highlights MetaLP-DGI's robustness in link prediction tasks, making it a suitable choice for networks requiring high precision in edge classification.

5.4 Conclusion

Link prediction is a key research problem in social network analysis and network science, aiming to identify connections that are likely to exist but have not yet materialized and may soon appear within the network. This paper introduced an improved link prediction model using a new connectivity matrix, DGI graph embedding, and CatBoost ensemble classifier. First, a new connectivity matrix was introduced by extracting pairs of nodes from the adjacency matrix of the input graph, followed by calculating similarity measures between pairs of nodes. The DGI embedding was adopted as part of the representation learning of nodes. We introduced edge embedding by taking the dot product of the embeddings of nodes at both ends. The necessary data balancing between training and test datasets is performed before the CatBoost ensemble classifier computes the edge prediction. Additionally, we conducted thorough hyperparameter studies to identify optimal values for various model parameters. We then compared the performance of the proposed model with other link prediction models, including recently introduced approaches, across six different social network datasets. The results revealed that the proposed model performed better. Some real-world networks are dynamic, with edges forming and dissolving over time. The proposed link prediction model performed well on static networks, but this model may struggle to adapt to the temporal dynamics, leading to less accurate predictions in dynamic environments. The future extension of this work may be to develop effective embedding methods on attributed networks and explore link prediction in temporal and dynamic networks, which involve incorporating time-dependent features and adapting the model to evolving network structures.

5.5 Summary

This paper introduces MetaLP-DGI, a powerful meta-learning-based framework designed to improve link prediction in complex and dynamic networks. Instead of relying on a single learning model, the approach combines Deep Graph Infomax (DGI) embeddings with centrality-aware structural information, ensuring that both global network influence and local connectivity patterns are effectively captured. By incorporating meta-learning, the framework learns how to adapt quickly across different datasets and network conditions, making it more generalizable and resilient than traditional models. Extensive experiments conducted on multiple benchmark datasets show that MetaLP-DGI consistently outperforms existing link prediction techniques in terms of accuracy, precision, recall, F1-score, and AUC. Overall, the study demonstrates that integrating self-supervised graph embeddings with centrality information and meta-learning can significantly enhance the predictive capability and robustness of link prediction systems. All things considered, MetaLP-DGI provides a reliable, scalable, and structure-aware method for link prediction in intricate social, biological,

Chapter 6

Link Prediction in Social Networks: A Hybrid Approach with Graph Embedding and Ensemble Learning

This chapter presents a hybrid link prediction framework that integrates graph embedding algorithms with ensemble learning classifiers. The proposed approach leverages multiple embedding methods to capture diverse structural and semantic patterns in the network, which are then fused and passed to an ensemble of machine learning models for final prediction. By integrating representation learning with ensemble-based decision-making, the framework aims to improve accuracy, robustness, and adaptability across diverse social network datasets. While traditional link prediction methods often rely on hand-crafted similarity metrics, recent advances in graph embedding techniques have enabled automated representation learning that captures both structural and attribute information of nodes. However, embeddings derived from a single model may fail to capture all relevant patterns, leading to suboptimal predictions. On the other hand, ensemble learning provides a robust framework for combining the strengths of multiple models, thereby reducing variance and enhancing generalization.

6.1 Background

Social networks model social interactions and interpersonal relationships and have gained considerable attention over the years in various real-world applications. These networks have emerged as influential tools, revolutionizing the ways individuals connect, communicate, and exchange information. These networks are intricate and present numerous challenges for research and business, including tasks such as detecting community structures [122], abnormal behavior detection [123], information diffusion, making social recommendations [124], predicting links, and more [6, 125]. As a result, the fundamental goal of the link prediction model is to uncover the mechanisms underlying network formation while detecting potential latent connections that may emerge in the future.

LP enables researchers to understand the growth, evolution, and structure of social networks over time. By analyzing patterns and features in the network data, link prediction seeks to predict future links that have not yet been established [126]. With various real-world applications, link prediction is essential in both practical and theoretical domains. For instance, it is used to predict missing links in protein-protein interaction (PPI) networks [101], in online social networks, e-commerce platforms for recommendation systems, and many other areas [127].

Traditional link prediction techniques primarily rely on handcrafted similarity measures, which often fail to capture the intricate structural dependencies within complex networks[128]. To address this limitation, we propose LPKtz2vec-XGBoost, a novel link prediction framework that integrates Katz similarity matrix, node2vec embeddings [100] and the XGBoost ensemble classifier to enhance predictive performance.

The proposed approach leverages the Katz similarity matrix to quantify the influence of all possible paths between node pairs, capturing both local and global structural properties. Unlike conventional similarity metrics that consider only direct or two-hop

neighbors, the Katz index accounts for higher-order connectivity patterns, making it well-suited for complex networks.

To generate meaningful node representations, we employ node2vec, a powerful self-supervised embedding method that maximizes mutual information between local and global representations of the graph. This ensures that the learned embeddings preserve both structural and contextual information, leading to more expressive feature representations.

This precisely labeled and balanced dataset is then input into the XGBoost classifier, which plays a crucial role in the final link prediction process. To achieve optimal performance, we thoroughly investigate and fine-tune the various hyperparameters of the proposed model. To summarize, this work has made the following main contributions.

- (I) Incorporation of Katz Similarity: Captures higher-order connectivity patterns for a more informed transition probability matrix.
- (II) node2vec-based Node Embeddings: It preserves both local and global network structures using self-supervised learning.
- (Iii) Balanced Training Strategy: Mitigates class imbalance issues by incorporating both existing and non-existing edge samples.
- (Iv) XGBoost for Link Prediction: Ensures robust and scalable classification of potential links.

6.2 Proposed Methodology

This section outlines the proposed model for predicting links based on the Katz similarity matrix and the integration of network embedding-based node2vec and XGBoost ensemble classifier for link prediction. The Katz similarity matrix is constructed us-

ing the adjacency matrix and captures the global structure of the network by considering all paths between node pairs with exponential decay. The transition matrix is replaced with the Katz similarity matrix, which is then passed to the node2vec algorithm for representation learning. The node2vec method converts the network into a compact representation while maintaining the structural integrity of the entire network. These vectors are then used as input to the link prediction model. The XGBoost machine learning algorithm utilizes the generated embedding for each edge, along with its corresponding positive or negative edge label, as input to perform the ultimate link prediction task. The superiority of our proposed approach lies in its ability to incorporate sophisticated feature extraction and representation learning techniques, allowing the model to capture both local and global structure information of nodes, leading to more accurate link prediction compared to methods relying solely on basic similarity measures. The proposed model also aims to handle unbalanced data distributions and uncertainties inherent in link prediction tasks. We further optimize the performance of our proposed model by conducting hyperparameter tuning to enhance accuracy and ensure optimal predictive capability. This section provides a detailed, sequential explanation of the distinct stages within the suggested model. The overall proposed work as shown in Fig.6.1

6.2.1 Creation of Katz Similarity Matrix

Let's examine a social network dataset represented by $G = (V, E)$, where V denotes the total number of nodes, and E signifies the collection of edges within G . After collecting the network dataset, we retrieved the connected and unconnected pairs of the nodes in the input network. Let A be the adjacency matrix, where an edge $(e_{u,v})$ between nodes u and v is represented as 1, otherwise 0, as described in the equation 6.1.

$$e_{uv} = \begin{cases} 1, & e_{uv} \in E \\ 0, & e_{uv} \in E^U - E \end{cases} \quad (6.1)$$

Here, E^U represents the complete set of all possible edges for a graph G . The Katz similarity matrix is computed to capture both direct and indirect connections between nodes, taking into account all paths between nodes with an attenuation factor to downweight longer paths. It is defined as:

$$Ktz = (I - \alpha A)^{-1} - I \quad (6.2)$$

where Ktz is the Katz similarity matrix, I is the identity matrix, A is the adjacency matrix, and α is the attenuation factor, controlling the impact of longer paths. A larger α considers more indirect connections, while a smaller α focuses more on direct links.

The Katz similarity matrix captures global connectivity patterns and enhances the ability of the model to infer missing links by leveraging both local and long-range node relationships. This structured representation provides an enriched feature space that helps improve link prediction accuracy.

6.2.2 Representation Learning using node2vec Embedding

After obtaining the Katz similarity matrix, we utilize the node2vec algorithm to transform the high-dimensional network into a low-dimensional space. The node2vec algorithm improves the model's capacity to identify intricate node connections by optimizing the shared information between local and global node representations.

The node2vec algorithm learns node embeddings that capture structural information by contrasting positive node representations with negative samples drawn from a corrupted version of the graph. This enables the model to uncover latent features crucial for link prediction. Once the node embeddings have been generated, they are

utilized to forecast the possible existence of linkages between nodes.

The equation below denotes the generated embedding for all nodes in the network by taking the embedding dimension (d), Katz similarity matrix, and hyperparameters as input:

$$emb = node2vec(Ktz, d) \quad (6.3)$$

In this context, emb denotes the collection of embeddings for all nodes within the network. This collection is represented as a two-dimensional matrix with dimensions $n \times d$, where n signifies the total number of nodes and d represents the dimensionality of the embedding.

After obtaining the node2vec embeddings for each vertex $u \in V$, we introduce the embedding for each edge (u, v) as the dot product of the embeddings of node u and node v :

6.2.3 Creation and Balancing of Training and Testing Datasets

To train and evaluate the link prediction model, we split the network into training G_{Train} and testing G_{Test} with an 80% : 20% ratio. Balanced datasets are created by combining positive (existing edges) and negative (non-existing edges) samples. These balanced datasets ensure fair evaluation and prevent overfitting. The final training and testing datasets combine shuffled positive and negative samples, ensuring an equal distribution for fair model evaluation.

6.2.4 Training XGBoost Ensemble Classifier

After preparing the well-labeled training and testing dataset in the previous step, we utilize the XGBoost ensemble machine classifier for classification in link prediction. XGBoost is known for its high performance and scalability, making it well-suited for capturing complex patterns and interactions within network data.

The classifier learns a function $f(x)$ that maps the feature vector:

$$f(x) = [(emb)[u], (emb)[v], (emb)[u] \odot (emb)[v]] \quad (6.4)$$

Here, $emb[u]$ and $emb[v]$ are the generated embeddings of nodes u and v , and $emb[u] \odot emb[v]$ represents the embedding of edge uv . The binary cross-entropy loss function $L(f)$ serves as the optimization target for XGBoost during its training process:

$$L(f) = -[y \log(p) + (1 - y) \log(1 - p)] \quad (6.5)$$

In this equation, y denotes the actual label (either 0 or 1), while p signifies the estimated likelihood of a connection existing between nodes. The ultimate objective function is established as:

$$\text{obj}(\theta) = L(f) + \lambda |w|^2 \quad (6.6)$$

where $\lambda |w|^2$ is the regularization term to prevent overfitting. This ensures that the classifier generalizes well across different datasets, making it highly effective for link prediction.

6.3 Social Network Datasets

To evaluate the proposed model, we have used various real-life networks belonging to multiple social and information networks. Details of these networks are shown in Table6.1:

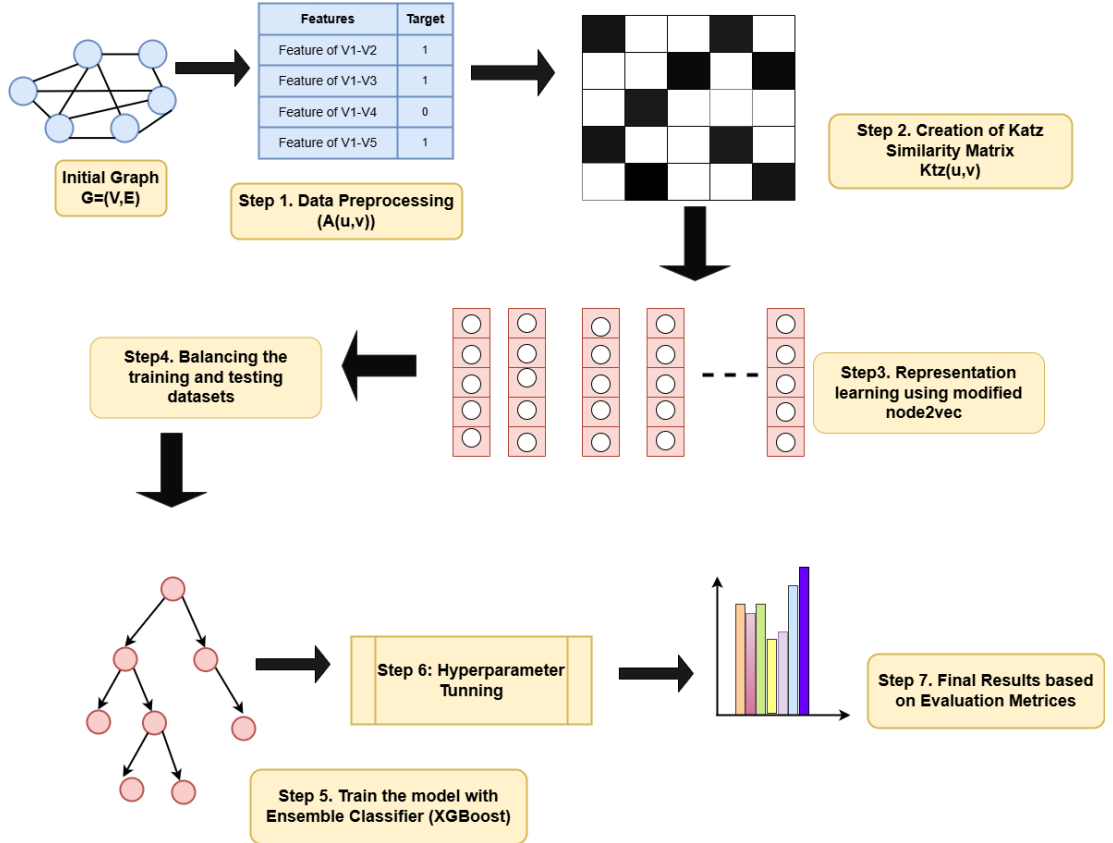


Fig. 6.1: Feature extraction of using node2vec model with Ensemble machine learning classifiers

Table 6.1: Details of Social Network Datasets

Dataset	Description	Nodes	Edges
FB-Food	Facebook pages dataset with vertices representing pages and edges indicating connections [129].	620	2102
Email-University	Email social network visualizing the email-university link structure. I will schedule some time for us to connect.[104]	1133	5449
Hamsterster	Facebook friendship dataset with nodes representing people and edges indicating friendships.[130]	2235	90952
FB-TV	Facebook dataset with vertices representing pages and edges indicating connections[131].	3930	17262

6.4 Experimental Result and Analysis

This section examines the tests conducted on various social network datasets to evaluate the proposed LPKtz2vec-XGBoost link prediction technique. To evaluate and

compare the model against various link prediction algorithms, several performance metrics are utilized, including Precision(Pre), Recall, F1 score, Accuracy, and AUC. Precision measures correctly predicted links, Recall measures correctly identified positive links, and the F1 score balances Precision and Recall. Accuracy evaluates the model's ability to predict both existing and non-existent links, while AUC indicates how well the model distinguishes true from false links. The models that exhibit superior performance across the evaluation metrics are highlighted in bold for clarity and emphasis.

The LPKtz2vec-XGBoost model was compared with several recent approaches, including EMLP[118], LPGD-ML[127], HSLP[128], LP-LGBM[86], and LP-DeepWalk[131]. An ablation study assessed the model's performance without XGBoost classification, using Logistic Regression, Naive Bayes, SVM, and AdaBoost instead. These variations, named LPn2vec-LR, LPn2vec-NB, LPn2vec-SVM, and LPn2vec-ADB, were also evaluated, and the results are presented in tables comparing the proposed model with these baseline methods.

6.4.1 Comparative Performance Study

In this section, we employ a range of evaluation metrics to compare our approach to the performance of other similar algorithms, as shown in Tables 6.2 and Table 6.3.

Table 6.2 presents the evaluation metrics for both datasets. The LPKtz2vec-XGBoost model outperformed several baseline approaches, including LP-LGBM, HSLP, and LP-Deep Walk, when applied to the Facebook Food dataset. This model achieved superior results with a recall of 0.70, F1 score of 0.76, AUC of 0.78, and accuracy of 0.79. In comparison, the EMLP model demonstrated a precision of 0.755. For the Email-University network, the LPKtz2vec-XGBoost model outperforms others with an AUC of 0.93, Accuracy of 0.92, and Precision of 0.93. The LPGD-ML model, however, performs best regarding Recall (0.97), and the highest F1 score is achieved

by EMLP (0.85), likely due to its ability better to capture network details through the selected node embeddings.

Table 6.2: Outcomes of various link prediction techniques applied to the FB-food and Email-University datasets.

Models	FB-food Dataset					Email-University Dataset				
	Prec	Recall	F1	Acc	AUC	Prec	Recall	F1	Acc	AUC
LPn2vecec-LR	0.54	0.58	0.56	0.54	0.56	0.66	0.62	0.64	0.68	0.70
LPn2vecec-NB	0.54	0.39	0.46	0.52	0.52	0.75	0.71	0.73	0.73	0.72
LPn2vecec-SVM	0.53	0.65	0.58	0.53	0.54	0.69	0.71	0.70	0.72	0.71
LPn2vecec-ADB	0.58	0.59	0.58	0.57	0.58	0.90	0.93	0.83	0.86	0.90
LPn2vecec-RD	0.67	0.69	0.68	0.66	0.67	0.74	0.72	0.73	0.76	0.75
LP-Deep-Walk	0.56	0.55	0.55	0.56	0.55	0.78	0.77	0.77	0.78	0.78
LPGD-ML	0.68	0.61	0.65	0.66	0.67	0.94	0.97	0.85	0.89	0.86
HSLP	0.68	0.67	0.67	0.68	0.67	0.70	0.80	0.75	0.77	0.76
LP-LGBM	0.72	0.73	0.71	0.74	0.75	0.83	0.81	0.75	0.84	0.84
EMLP	0.75	0.72	0.75	0.77	0.76	0.88	0.80	0.88	0.86	0.85
LPKtz2vec-XGBoost	0.75	0.71	0.76	0.74	0.78	0.85	0.90	0.87	0.92	0.93

Table 6.3: Outcomes of various link prediction techniques applied to the Hamsterster and FB-TV datasets.

Models	Hamsterster Dataset					FB-TV Dataset				
	Prec	Recall	F1	Acc	AUC	Prec	Recall	F1	Acc	AUC
LPn2vecec-LR	0.65	0.64	0.64	0.65	0.73	0.50	0.51	0.50	0.53	0.52
LPn2vecec-NB	0.69	0.62	0.65	0.66	0.67	0.52	0.56	0.54	0.53	0.51
LPn2vecec-SVM	0.73	0.59	0.65	0.68	0.69	0.54	0.52	0.53	0.56	0.54
LPn2vecec-ADB	0.69	0.63	0.66	0.67	0.68	0.58	0.51	0.54	0.56	0.57
LPn2vecec-RD	0.73	0.70	0.71	0.75	0.76	0.57	0.55	0.56	0.58	0.57
LP-Deep-Walk	0.61	0.64	0.62	0.62	0.63	0.60	0.63	0.61	0.63	0.60
LPGD-ML	0.90	0.54	0.68	0.73	0.78	0.63	0.55	0.59	0.61	0.61
HSLP	0.71	0.67	0.69	0.72	0.71	0.59	0.62	0.60	0.62	0.61
LP-LGBM	0.76	0.59	0.66	0.73	0.77	0.73	0.69	0.71	0.65	0.63
EMLP	0.89	0.67	0.76	0.78	0.80	0.72	0.64	0.68	0.65	0.69
LPKtz2vec-XGBoost	0.93	0.63	0.77	0.75	0.82	0.69	0.71	0.69	0.72	0.70

Table 6.3 presents the evaluation performance of the Hamsterster network and FB-TV data sets. In the Hamsterster data set, the LPKtz2vec-XGBoost model exhibits exceptional performance, boasting a precision of 0.93, an F1 score of 0.77, and an AUC of 0.82. Currently, the EMLP model delivers robust results, with an F1 score of 0.77 and an accuracy of 0.78. This high level of performance enables the proposed LPKtz2vec-XGBoost approach to employ ensemble classifiers in the model-training process. In the evaluation comparison of the FB-TV dataset, our proposed

model LPKtz2vec-XGBoost performs well with an accuracy of 0.72 and an AUC of 0.7. The LP-LGBM-based approach for link prediction demonstrates strong performance, achieving a precision of 0.73 and an F1 score of 0.71. This strategy enhances the effectiveness of our proposed LPKtz2vec-XGBoost method, enabling it to more efficiently capture the network intricacies.

6.5 Conclusion

The proposed LPKatz2Vec-XGB is a novel link prediction approach that integrates the Katz similarity matrix, node2vec embeddings, and XGBoost ensemble classifier. The Katz similarity matrix effectively captures both local and global structural similarities between node pairs, enhancing transition probabilities for random walks. The node2vec algorithm then generates high-quality, low-dimensional embeddings while preserving network topology. Finally, the XGBoost classifier is trained on these enriched embeddings to accurately predict missing or future links in the network. The integration of node2vec embeddings transforms high-dimensional network data into a low-dimensional feature space, preserving structural and relational information to identify intricate patterns within the network. The XGBoost classifier further improves performance by handling non-linear relationships and imbalanced datasets. The model's performance was evaluated against several link prediction approaches, including recently introduced methods, across diverse social network datasets, demonstrating its effectiveness and robustness. The future extension of this work may be to develop effective embedding methods on attributed networks and explore link prediction in temporal and dynamic networks, which involve incorporating time-dependent features and adapting the model to evolving network structures.

6.6 Summary

This paper presents a hybrid framework designed to enhance the accuracy and reliability of link prediction in social networks. The approach intelligently combines graph embedding techniques with ensemble learning to better understand hidden structural patterns and relationships between nodes. First, meaningful graph embeddings are generated to capture both local and global network characteristics. These embeddings are then fed into an ensemble of classifiers, allowing the model to learn more diverse patterns and reduce bias from any single method. Through extensive experiments on multiple real-world social network datasets, the proposed model demonstrates significant improvements over traditional link prediction techniques in terms of accuracy and robustness. Overall, the study demonstrates that integrating graph embedding with ensemble learning can significantly improve the quality of link prediction, making it applicable to various domains, including recommendation systems, community detection, and network evolution analysis.

Chapter 7

Enhancing Link Prediction through Graph Attention Network and Linear Discriminant Analysis

This chapter presents a novel link prediction framework that integrates the Graph Attention Network (GAT) with Linear Discriminant Analysis (LDA) to improve both feature representation and classification performance in complex networks. The approach addresses the limitations of conventional link prediction methods that often fail to differentiate between subtle structural variations in highly connected or noisy graphs. The GAT component enhances node embeddings by dynamically attending to neighbors with varying importance, enabling the model to capture fine-grained structural dependencies. Subsequently, LDA is employed as a dimensionality reduction and class-separability enhancement technique, ensuring that the embeddings are more discriminative for the link prediction task. While deep graph representation learning methods such as GCN and GraphSAGE have shown promising results in link prediction, they typically assume equal weighting for all neighboring nodes or rely on fixed aggregation schemes. This uniform treatment often neglects the heterogeneous importance of neighbors in real-world networks. Moreover, high-dimensional embeddings learned by deep models can contain redundant or noisy features, which hinder classifier performance.

7.1 Background

Link prediction (LP) plays a vital role in complex network analysis, which aims to predict future links based on the existing structure and characteristics of the network[132]. LP has evolved into an extensively researched topic within the complex network science community. Its prominence stems from its significant theoretical importance in understanding the mechanisms of complex network evolution and its broad applicability across various scientific disciplines[133]. Hence, LP enables the identification of connections, offering insights into the network's growth patterns. LP has wide-ranging applications in social networks, biological networks, and recommendation systems[1]. Some of the real-life applications of LP are recommending new friends on social media[134], finding spreader nodes, predicting potential collaborations in scientific research, forecasting disease outbreaks, suggesting new links in transportation systems, offering product recommendations[135] in e-commerce, recommending movies or music, and many others.

A wide range of link prediction (LP) methods has been discussed in the literature, focusing on forecasting the probability of future links between nodes in a network[136]. Link prediction techniques are divided into various categories, such as the Likelihood-based approach, which evaluates the similarity between two disconnected nodes and assumes a higher likelihood of a link between more similar nodes. Formally, likelihood-based link prediction methods propose that the chance of their interaction grows when two nodes are connected to comparable or adjacent nodes within the network. Additionally, recent studies have explored deep learning and machine learning-based approaches for link prediction. Beyond the previously discussed techniques, graph embedding algorithms within machine learning have recently been applied to address link prediction challenges[86].

In the proposed link prediction model, we utilize a graph attention network (GAT)

to learn the embeddings of the nodes by selectively aggregating information from neighboring nodes. During message passing, the adaptive attention system of GAT dynamically determines the importance of surrounding nodes, enabling adaptive information aggregation based on relevance. In addition, the suggested model combines node centrality metrics, such as hubs and authority centrality (HAC), and local interaction density (LID), with Jaccard similarity and GAT. Initializing a weight matrix integrating structural and topological information, the model combines network properties obtained from similarity measures and node centrality into a feature matrix. Thus, by using GAT to produce node embeddings, the model can develop rich representations that consider both local and global contexts.

The resulting embeddings, paired with positive or negative labels for nodes at edge endpoints, form a well-defined dataset. Additionally, LDA is applied to the labeled dataset for classification purposes. The effectiveness of the model is validated across eight datasets, demonstrating superior performance compared to the baseline and recent models. The summary of the proposed work is listed below.

- (i) We present a link prediction model by adopting similarity measures, node centrality, graph attention network (GAT), and linear discriminant analysis (LDA).
- (ii) The proposed model utilizes a GAT to compute embeddings based on the combined feature representation. This feature combination enhances the model's predictive capability.
- (iii) The learned representation and model interoperability allow a better understanding of the network's underlying structural patterns and relationships. The learned embeddings are then used with LDA for link prediction.
- (iv) We investigate the optimal values of various model parameters through hyperparameter studies to improve the results of the proposed model.
- (v) The proposed model is evaluated and compared to various existing link predic-

tion methods, as determined by experiments conducted on eight social network datasets.

7.2 Preliminary

This section outlines some fundamental concepts that are relevant in this paper. Let us consider an undirected attributed social network graph $G = (V, E, F)$, where $V = \{V_1, V_2, V_3, \dots, V_N\}$, is the set of nodes in the graph, N is the total number of vertices(nodes), E is the set of links(edges) in the graph, and $F = \{f_1, f_2, f_3, \dots, f_d\}$ implies the initial features associated with the nodes. $D = |F|$ represents the dimensionality of the features for each node in the graph. The topological configuration of the graph is captured by the adjacency matrix $A \in \mathbb{R}^{N \times N}$, while the feature matrix of each node in the graph is represented as $F \in \mathbb{R}^{N \times d}$.

7.2.1 Definition 1: Measuring Structural Similarity (MSS)

We utilize the Jaccard similarity proposed by [14] to measure the structural similarity between the node pairs. The Jaccard Similarity between the two nodes u and v in a graph is a measure of similarity based on their neighbors. It normalizes the size of common neighbors. The similarity score is calculated as.

$$sim(u, v) = \frac{|N_u \cap N_v|}{|N_u \cup N_v|} \quad (7.1)$$

The similarity measures between the nodes u and v is shown in Eq. 7.1, N_u , and N_v are the set of neighbors of a nodes u and v . Here, $|N_u \cap N_v|$ is the size of intersection and $|N_u \cup N_v|$ is the size of union. The coefficient ranges between 0 and 1. A value closer to 1 indicates a higher similarity, and 0 implies no common neighbors and, therefore, no similarity.

7.2.2 Definition 2: Measuring Topological Similarity (MTS)

In graph analytics, centrality is an important term to identify the importance of the nodes. **Local Interaction Density (LID)**[137] centrality measures the relative importance of different nodes in a graph. It is the proportion of several interacting nodes inside the node's subgraph. The LID score of the nodes u is expressed in Eq. 7.2 as:

$$LID_u = \frac{|E_{N_{int}(u)}|}{|V_{N_{int}(u)}|} \quad (7.2)$$

where the number of interactive edges is denoted by $|E_{N_{int}(u)}|$, which means the edges within the subgraph of a node u signify the interactions among the nodes in the neighborhood of that particular node. In contrast, the $V_{N_{int}(u)}$ is the number of nodes that are engaged in interactions with each other within the neighborhood of node u . For normalized versions, sometimes LID is scaled to $[0,1]$ using maximum values across the graph.

7.2.3 Hubs and authorities centrality (HAC)

: In network analysis, hubs, and authorities are used to identify the importance of the nodes. A hub can be a node that sends out many links, while authorities can be nodes that receive many links. This is the foundation for the hubs and authorities method, also known as the hyperlink-induced topic search (HITS). The relationship between the hubs and authorities is represented in Eq. 7.3 and Eq. 7.4.

$$Auth(u) = A^T(\mathbf{Hub}(u)) \quad (7.3)$$

$$Hub(u) = A\mathbf{Auth}(u) \quad (7.4)$$

The adjacency matrix is symbolized by A . In contrast, the transpose of the adjacency matrix is represented by A^T . Since both hub and authority scores are computed us-

ing matrix multiplications and iterative power methods, their values are always non-negative if initialized with non-negative values. If a node is isolated (has no incoming or outgoing links), then its hub and authority scores will be 0.

$$\text{hub}(u) \geq 0, \quad \text{auth}(u) \geq 0$$

The hub and authority scores are normalized in most algorithms so that the maximum score across all nodes is 1.

7.2.4 Definition 3: Weight Matrix

A complex network $G = (V, E, F)$ is turned into a weighted graph $G = (V, E, F, W)$ by combining Jaccard similarity, local interaction density, and hubs and authorities centrality. Integrating structural and topological similarities between node pairs helps enhance the reliability of the link weight between two nodes.

$$W(u, v) = \begin{cases} \beta \times MSS(u, v) + (1 - \beta) \times MTS(u, v), & \text{if } A(u, v) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (7.5)$$

Where $\beta \in [0, 1]$ is the weighting coefficients that control the influence of local interaction density and hubs-authorities centrality in the overall weight. $A(u, v)$ is the adjacency matrix, indicating if there is an edge between the nodes u, v , labeled as 1, otherwise 0.

7.3 Proposed Work

This section details the proposed model for link prediction by integrating similarity and node centrality measures using a graph attention network (GAT) embedding with linear discriminant analysis (LDA). In short, our model is called SNGAT-LDA. The

proposed work consists of three stages. The first stage involves feature extraction, where features such as Jaccard similarity, local interaction density (LID), and centrality of hubs and authority centrality (HAC) are integrated into a comprehensive feature extraction process. These features encompass a node's local, global, topological, and relative influence over others. Subsequently, a weight matrix is computed to derive and combine the topological and attributed similarity matrices. This weight matrix signifies the likelihood of moving on to the next node. The next stage is generating an embedding through the Graph Attention Network (GAT). In this phase, the node embeddings are computed using a GAT with feature matrix F and the weight matrix W . This process captures the rich representation of nodes. GAT utilizes attention mechanisms to assign weights to neighboring nodes while generating embeddings dynamically. The third stage is the loss function and model training. The generated embeddings are then passed to the linear discriminant analysis (LDA) for link prediction to obtain performance measures. LDA is trained on the generated embeddings to discern the potential relationships among links. The trained LDA model predicts future links using the computed node embeddings. This sequential integration leverages GAT's representation power for embedding and LDA's discriminative capability for classification, thereby improving link prediction performance in complex networks. This integration allows the model to not only focus on relevant neighboring nodes (via GAT) but also enhance discriminative learning (via LDA), leading to improved classification and link prediction performance.

7.3.1 Features Extraction

The proposed model intends to capture the structural and topological configuration between the node pairs by incorporating the links' weights, which signify the connections' strength and similarity. for this, we compute the probability $P(u, v)$ of transitioning from node u to its previous node (v) in a network. The transition probability

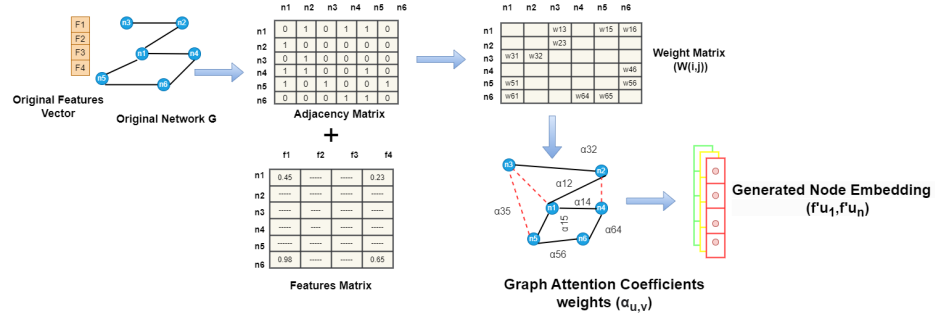
is calculated only if the edge (u, v) exists in the set of edge E . If the edge exists, the transition probability is given by the weight of the edge $w_{u,v}$ divided by the sum of the weights of all edges connected to the node u , denoted as $\sum_{v \in \tau(i)} w_{uv}$. This normalizes the transition probability among all possible connections of the node u . If there is no edge (u, v) , the transition probability is set to zero. Calculating transition probabilities for existing edges helps in modeling the current state of the network. This is important because predicting future links heavily relies on understanding the current connections and the influence flow between nodes. The equation 1.6 essentially captures how likely it is to move from one node to another based on the weights of the connecting edges, considering only existing edges in the network.

$$P(u, v) = \begin{cases} \frac{w_{uv}}{\sum_{v \in \tau(i)} w_{uv}}, & \text{if } (u, v) \in E \\ 0, & \text{Otherwise} \end{cases} \quad (7.6)$$

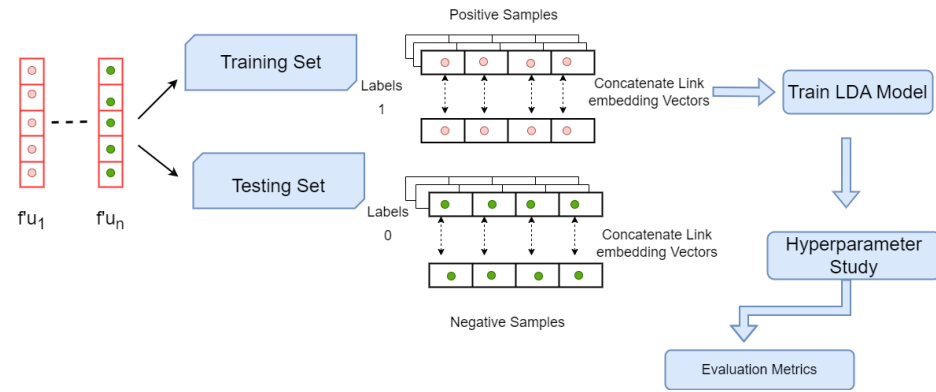
The weight matrix is paired with the graph attention network in a later stage to refine the embeddings. These probabilities are used as features in machine-learning models to help learn patterns that govern existing and future connections. By incorporating the weights and transition probabilities, the model learns which types of connections are more likely to occur. Hence, the extracted features enable the model to learn more informative and discriminative node embeddings, leading to more accurate predictions of future links in the network.

7.3.2 Generating Embedding using GAT

In this phase, we use the feature matrix F and the weight matrix W as input to generate node embeddings using GAT. Network embedding techniques help transform the high-dimensional network into a low-dimensional space. Regarding link prediction, network embedding models operate well because they assume that every node is connected to an unobserved feature vector. The interaction among these latent features



(a) Generated features using GAT from similarity measure, node centralities, and initial weight matrix.



(b) Proposed Link prediction model using GAT generated embedding and LDA.

Fig. 7.1: Overview of the proposed method with feature generation and link prediction model.

is used to calculate the likelihood of a relationship. The proposed model combines the principle of network representation learning with the concept of deep learning attention mechanisms. The fundamental concept behind the GAT involves assigning weights to the immediate neighbors of the nodes within a network using a masked attention mechanism. We aim to change the GAT method by introducing a new weight matrix, calculated by the similarity and centrality of node pairs. The node centralities typically offer vital insights into a node's significance by examining the network structure, interconnections among vertices, and various key topological characteristics. The weights will be assigned to the links by applying the structural and topological features, and the updated weight matrix extracts the local and global characteristics, making the GAT more adaptable to diverse network structures. Hence, the combination of node

similarity and centralities improves the attention mechanism of GAT.

In the proposed model, GAT takes the weight matrix W , the combination of the graph's structural and topological structure, as input. This matrix captures the importance of different features in the graph. The GAT is designed to have the expressive capacity needed to transform input features into sophisticated output features effectively. The output after using the GAT will be $f' = \{f'_1, f'_2, f'_3 \dots f'_d\}$, $f'_u \in \mathbb{R}^{N \in d'}$. In the first step, we generate the attention coefficient α_{uv} between the nodes u and v , by utilizing the masked self-attention mechanism a_u shown in Eq. 7.7.

$$\alpha_{u,v} = a_u(W'f_u, W'f_v) \quad (7.7)$$

The above equation indicates the attention importance of node feature's u to node v where f_u and f_v are the original features vector of node u v , respectively. Now, we calculate the attention score of nodes $\alpha_{u,v}$ by utilizing the adjacent nodes u, v as an equation 7.8.

$$\alpha_{u,v} = \sigma(a^{-T} \cdot [W'f_u || W'f_v]), v \in (Nu1, Nu2) \quad (7.8)$$

Here, σ indicates the activation function (ReLU), and a^T is the transpose of the learnable attention weight vector a . The concatenation of transformed feature vectors is represented as $W'f_u$ and $W'f_v$ where $Nu1, Nu2$ denotes the set of immediate neighbors or the set of similar nodes for node v , respectively.

$$\alpha_{u,v} = \text{softmax}(e_{u,v}) = \frac{\exp(e_{u,v})}{\sum_{p \in Nu1} \exp(e_{u,p}) + \sum_{q \in Nu2} \exp(e_{u,q})} \quad (7.9)$$

We normalize the attention coefficient using the softmax function in Eq. 7.9 to access the significance of the graph's first-order neighbor and similar nodes. The $\exp(e_{u,p})$ and the $\exp(e_{u,q})$ are the sum of the exponentiated attention scores for all first-order neighbors in $Nu1$ and $Nu2$. We employ multi-head attention mechanisms,

as shown in Eq. 7.10, to maintain the consistency of node embedding vectors.

$$f'_u = \prod_k^{k=1} \sigma \left(\sum_{p \in Nu1} \alpha_{up}^k \cdot W'_p f'_p + \sum_{q \in Nu2} \alpha_{uq}^k \cdot W'_q f'_q \right) \quad (7.10)$$

In the above equation, the final node embedding vector is generated by averaging the embedding vectors of node u across the output layer. Here, k represents the number of attention heads. The normalized attention coefficients of node u to the neighboring node p are denoted as α_{up}^k , and W'_k is the transferable learnable matrix acquired from the k^{th} attention mechanisms, respectively.

$$f'_u = \sigma \left(\frac{1}{k} \sum_{p \in Nu1} \alpha_{up}^k \cdot W'_p f'_p + \sum_{q \in Nu2} \alpha_{uq}^k \cdot W'_q f'_q \right) \quad (7.11)$$

Finally, the aggregation process of multi-head graph attention layers is represented in Eq. 7.11, where the multi-head attention layers embedding is passed for the link prediction tasks.

7.3.3 Model Training and Loss function

We employ linear discriminant analysis (LDA) for classification tasks in the proposed model. In the context of SNGAT-LDA, LDA refers to Linear Discriminant Analysis, a supervised dimensionality reduction technique used to improve classification performance. LDA further refines these embeddings by projecting them into a space where different links (positive and negative) are better separated. A labeled data set containing positive and negative samples is required before training the LDA classifier. Positive samples indicate the presence of edges, while negative samples indicate their absence. Labels are assigned to each sample based on whether an edge exists between a pair of nodes. Positive and negative samples are partitioned into training and testing sets. Next, we generate embedding vector representations of links by merging the embedding vectors of the relevant node pairs in both the training and testing sets. After

that, these embedding features are used as input samples in the training set for linear discriminant analysis to identify potential relationships among links. Then, we utilize the trained LDA to predict the potential links by inputting the sample's embedding vectors within the test set.

$$y_{(u,v)} = \begin{cases} 1, & \text{The link exists between } (u, v) \\ 0, & \text{otherwise} \end{cases} \quad (7.12)$$

in the Eq. , $y_{u,v}$ represents the corresponding label vector. A value of 1 for $y_{u,v}$ indicates that the node pair (u, v) is connected by an edge, while a value of 0 signifies the absence of an edge.

$$L(P, y_{u,v}) = -\frac{1}{B} \sum_{u,v=1}^B [y_{u,v} \log(p_{u,v}) + (1 - y_{u,v}) \log(1 - \sigma'(p_{u,v}))] \quad (7.13)$$

The cross-entropy loss $L(P, y_{u,v})$ is minimized for link prediction, as mentioned in Eq. 7.13. It assesses the variation between the predicted probabilities and the actual binary labels. Here, B denotes the count of node pairs in the training set, $p_{u,v}$ corresponds to predicted probability, $y_{u,v}$ is the actual label (either 0 and 1), and σ' is the activation function, namely the rectified linear unit (ReLU).

The entire layout of the proposed SNGAT-LDA model is depicted in Fig. 7.1a and Fig. 7.1b. Fig.7.1a illustrates generating the node embeddings for a graph using graph attention networks—a visual representation of the graph showing the connections between the nodes. The graph's adjacency matrix indicates the existing and non-existing links between the nodes u, v , where the input features for each node in the network are labeled as f_1, f_2, f_3 , and f_4 . The feature matrix integrates the graph's similarity structure and topological structure of the graph. These features are passed through to update the weight matrix $W(u, v)$ by integrating $[sim, LID, Hub, Auth]$. GAT takes

a Weight Matrix W as input, a combination of distinct features. The attention mechanism computes coefficient α for each edge, indicating the importance of the edge. The final node embeddings f'_{u_1}, f'_{u_n} are produced by averaging the feature vectors across the multi-head attention layers, resulting in a new feature representation for each node.

The second phase of the proposed model, as shown in Fig.7.1b, illustrates the training and evaluation process for link prediction using LDA. The generated node embeddings f'_{u_1}, f'_{u_n} are passed for the training and testing phase. The embedding vectors for links are obtained by concatenating the embeddings of the corresponding node pairs. A labeled dataset with a balanced number of positive and negative samples is desired to employ machine learning techniques. After constructing this labeled dataset, the dataset is categorized into training and testing sets. The LDA model is trained using the training set. The trained LDA model predicts the potential links using the sample embedding vectors within the test set. Finally, the efficacy of the proposed model is evaluated using evaluation metrics such as precision, recall, F1 score, AUC, and relative information gain.

Algorithm 2 SNGAT-LDA Link Prediction Model

Require: Graph $G = (V, E, F)$ where V is the set of nodes, E is the set of edges, and F represents node features

Ensure: Predicted future links

- 1: $MSS(u) \leftarrow$ Compute Local Interaction Density using Eq. 7.1
 - 2: $MTS(u) \leftarrow$ Compute LID, Hub, and Authority Centrality using Eq. ?? and 7.3
 - 3: $\mathbf{F} \leftarrow [sim, LID, Hub, Auth]$ Combine features.
 - 4: $\mathbf{W}(u, v) \leftarrow$ Update weight matrix using MSS and MTS using Eq. 7.4
 - 5: $\alpha_{u,v} \leftarrow$ Compute attention weights via GAT with softmax using Eq. 7.5
 - 6: $\mathbf{f}'_u \leftarrow$ Generate final embeddings via multi-head attention using Eq. 7.11
 - 7: $D^{train}, D^{test} \leftarrow$ Create balanced dataset.
 - 8: $f'_u \leftarrow$ Train model with embeddings to predict future links as per Eq. 7.3.3
 - 9: $fL(P, y_{u,v}) \leftarrow$ Optimize cross-entropy loss.
 - 10: $SNGAT - LDA \leftarrow$ Train model and tune hyperparameters
 - 11: **return** Performance metrics
-

7.3.4 Algorithmic Explanation

The algorithm 2 depicts the overall steps of the proposed link prediction model named SNGAT-LDA. The description of each step is enumerated as follows.

1. In the first step, after taking the input, we compute the updated feature matrix by combining all the features (Jaccard Similarity, LID, Hubs, and authority scores) into a single feature matrix. The size of this matrix is $N \times d$, where N represents the total number of nodes and d is the dimensionality of the features.
2. After generating the feature matrix, we create a weight matrix W using the structural and topological features of the graph. This matrix will be used to capture the importance of different features in the graph.
3. The updated weight matrix $W(u, v)$ is passed as input to the graph attention network to compute the node embeddings. GAT uses the attention mechanism to assign different weights to different nodes. This step ensures that the attention coefficients sum up to one for each node.
4. Generate the final node embeddings by averaging the vectors of each node u across the output layer of the multi-head attention mechanism as $f' = \{f'_1, f'_2, f'_3 \dots f'_d\}$, $f'_u \in \mathbb{R}^{N \times d'}$.
5. Assemble a balanced dataset containing the same number of positive and negative samples and then split the dataset into the training and testing sets as D^{train}, D^{test} .
6. We use the generated node embeddings f' to train a Linear Discriminant Analysis (LDA) model for link prediction. LDA is a classification technique that identifies the linear combination of features that most effectively distinguishes between the classes, such as existing or non-existing links.

7. Train the LDA model using the generated node embeddings. This involves learning the relationship between the embeddings and the presence of links in the training data.
8. During the training, the model minimizes the cross-entropy loss between the predicted links P and the ground truth labels $y_{u,v}$ discussed in Eq. 7.13.
9. Once the model is trained, a hyperparameter study is carried out to identify the best values for the model parameters.
10. Use the trained $SNGAT - LDA$ model to predict the existence of future links based on the node embedding in the test data.
11. The concluding step is to measure the model's effectiveness using the test dataset and provide the performance metrics such as precision (Pre), recall(Rec), F1 score, and AUC.

7.4 Dataset and Evaluation Metrics

To assess the effectiveness of our proposed method, we performed experiments on a wide variety of real-world complex networks. These networks varied in size, sparsity, and complexity. Different social media platforms collect them. We selected these networks because they accurately model real-user social interactions. Table 7.1 represents the basic statistical details of the social network dataset.

1. Cora Dataset [138]: The Cora dataset is derived from the Cora research paper, which contains academic papers from machine learning. It comprises 2,708 scientific publications and 5,429 links.
2. Citeseer Dataset[139]: This dataset is a widely used benchmark dataset in the field of graph representation learning and machine learning. The dataset comprises 3,312 scientific publications in one of six classes and contains 4,732 links.

Table 7.1: Topological measures of real-world datasets

Network	N	E	Avg. Degree	Density	Assortativity
Cora	2,708	5,429	4	0.0014	-0.0656
Citeseer	3,312	4,732	2	0.00085	-0.0048
Pubmed	19,717	44,338	5	0.0002	-0.0082
Facebook	4,039	88,234	8	0.000715	-0.0688
Wiki	889	2,914	6	0.00073	-0.0028
Hamsterster	2,426	16,630	13	0.0056	0.0478
PolBlogs	1,222	19,201	25	0.0171	-0.1961
Email	32,430	54,395	3	0.000103	-0.3816
Twitter	81,306	1,768,149	44	0.000535	-0.242

3. Pubmed Dataset [140]: The Pubmed dataset is derived from the Pubmed scientific publication citation network. The PubMed consists of a network where nodes correspond to academic papers and edges represent citations. It consists of 19717 scientific publications and 44338 links.
4. Facebook [102]: It is a social network dataset that represents a snapshot of users' interactions, where nodes correspond to users and edges denote friendships between pairs of nodes. The dataset includes 4,039 nodes and 88,234 edges.
5. Wiki [140]: This network illustrates the voting activity on Wikipedia from its inception until January 2008. Nodes in the network represent Wikipedia users, and edges illustrate their voting interactions. The network comprises 889 nodes and 2,914 edges.
6. Hamster[140]: It is a social network dataset representing user interactions on the Hamster website. The vertices represent the users, and the links signify their interactions with each other. The dataset contains 2,426 nodes and 16,630 links.
7. PolBlogs[115]: It is a directed network that captures the interaction between the US political blog pages. The nodes in the polBlogs dataset represent individual political blog pages, and the edges indicate direct interactions between pairs of blog pages. The dataset comprises 1,222 nodes and 19,201 directed edges.

8. Email[141]: The network comprises an email-based system of connections among significant European research institutions. The vertices represent the users, and the edges show their interactions. The network comprises 32430 nodes, and there are 54395 edges.
9. Twitter[142]: The Twitter dataset is a large-scale social network graph that represents user interactions, such as follows or mentions. It is characterized by a sparse topology, high heterogeneity, and low density. It consists of 81,306 nodes and 1,768,149 edges.

7.4.1 Evaluation Metrics

Link prediction can be viewed as a binary classification problem. The assessment of this binary classification process is typically represented by a confusion matrix. Within this matrix, specific terms are employed:

1. Precision: This metric evaluates the accuracy of positive class predictions (in this case, the presence of links) by calculating the proportion that are correctly identified. It can be determined using Eq.7.14.

$$Prec = \frac{TruePositive(TP)}{TP + FalsePositive} \quad (7.14)$$

2. Recall: This measure identifies all positive instances (here, the existence of links) within the dataset. It is calculated using Eq.7.15.

$$Recall = \frac{TP}{TP + FalseNegative(FN)} \quad (7.15)$$

3. F1-Measure: This metric represents the harmonic average of precision and re-

call. It is computed using the following equation:

$$F1 - score = \frac{2 \cdot Prec \cdot Recall}{Prec + Recall} \quad (7.16)$$

4. Area Under the Curve (AUC): The AUC metric assesses link prediction techniques by examining the scores given to unobserved connections. It integrates the ROC (Receiver Operating Characteristic) curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold levels. The AUC is given by:

$$AUC - ROC = \int_0^1 TPR(FPR) d(FPR) \quad (7.17)$$

It measures a classifier's ability to distinguish between positive and negative samples. A higher AUC value indicates better classification performance, with $AUC = 1$ being a perfect classifier and $AUC = 0.5$ indicating random guessing.

5. Accuracy: This metric represents the ratio of correct predictions to the total number of predictions. It is calculated using the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7.18)$$

7.5 Results and Analysis

This section presents the experimental simulations and analyzes the resulting data. To conduct a thorough comparison, we evaluated the performance of our proposed algorithm against traditional link prediction algorithms like Common Neighbors (CN) [88], Adamic Adhar (AA) [143], Katz, and Resource Allocation (RA) [86], and SimRank[144]. We also used popular embedding algorithms like LINE, DeepWalk[5] [91], node2vec [100], and GAT [92] as baseline algorithms. We also employed our extracted node

centrality-based features with Random Forest and support vector machine, denoted as NC-RF (node centrality with random forest) and NC-SVM (Node centrality with support vector machine) as comparison models. To ensure the robustness of the proposed model, SNGAT-LDA, we compared its performance against some recently proposed popular link prediction models. These include EMLP (an ensemble model for link prediction based on graph embedding) [53], HSLP (H-index-based link prediction) [143], LP-LGBM [82], GC-LSTM (graph convolution embedded LSTM for dynamic link prediction) [54], an interlayer-link prediction method based on edge-weighted embedding (SWLEV) proposed by [145], [146], link prediction based on local structure and node information along local paths (LSNI) [48], and GSVAELP (integrating graph-SAGE and variational autoencoder for link prediction).

The overall performance of the proposed model compared to various benchmark models on all datasets is presented in Tables 7.2 to Tables 7.7.

For the proposed SNGAT-LDA model, we also fine-tuned the hyperparameters. In our experiments, the feature dimensions are set as follows: the hidden layer has 32 dimensions, the input layer has 64 dimensions, and the output layer has 16 dimensions, respectively, and the number of attention heads is 2. The SNGAT-LDA uses an Adam optimizer with a distinct learning rate of 0.3 and 0.001, with an early stopping strategy to avoid overfitting. The epoch is 200 for model training, and all the experiments are performed ten times. The Colab pro-Python programming language is used to execute the entire code. While implementing our algorithm, we utilized some Python libraries like NetworkX and Scikit learns library for machine learning classification.

7.5.1 Hyperparameter Analysis

A hyperparameter study is conducted to determine the optimal ratio of training and testing datasets. Figures 7.2a, 7.2b, 7.2c, and 7.2d investigate the optimal training-testing dataset ratio across four benchmark datasets: Cora, Citeseer, Pubmed, and

Table 7.2: Accuracy of proposed SNGAT-LDA Model on different hidden layers of GAT over several networks

No. of Hidden Layers	Cora	Citeseet	Pubmed	Facebook	Wiki	Hamster	PolBlogs	Email	Twitter
1	0.824	0.816	0.832	0.843	0.812	0.853	0.734	0.805	0.812
2	0.834	0.828	0.843	0.855	0.823	0.862	0.786	0.819	0.840
3	0.830	0.822	0.838	0.850	0.818	0.857	0.776	0.813	0.832

Facebook. To determine this optimal allocation, we systematically varied the splitting ratio from 55% to 95%, incrementing by 5% at each step. The resulting Area Under the Curve (AUC) values, presented in Figures 7.2a to 7.2d demonstrate a consistent trend: AUC performance improves as the training data proportion increases, reaching a peak around an 80% training split. This observation suggests that a 55% split leads to underfitting, likely due to insufficient training data for the model to effectively capture network features. Conversely, exceeding a 90% split appears to induce overfitting, resulting in diminished performance. Consequently, the analysis highlights the critical importance of balancing training and testing data proportions to achieve optimal model generalization. Using 80% data for training instead of 90% ensures a sufficient test set for a robust evaluation. The other reason is that a smaller test set (e.g., 10%) may not adequately represent unseen connections, leading to an over-optimistic performance. With 20 % test data, the model is tested on a more diverse set of potential links, improving its generalization ability. Hence, based on the observed AUC peak around an 80% training split, we conclude that an 80:20 training-testing ratio offers the optimal balance for model performance across the tested datasets considered.

7.5.1.1 Other Parameter Analysis

This section examines the impact of different parameter configurations used in the experiments, including the number of hidden layers and the dimensionality of the node embeddings used in the proposed model *SNGAT – LDA*.

Table. 7.2 illustrates the impact of hidden layers on the proposed model (SNGAT-LDA). Through experimentation, we determined that the optimal AUC values were

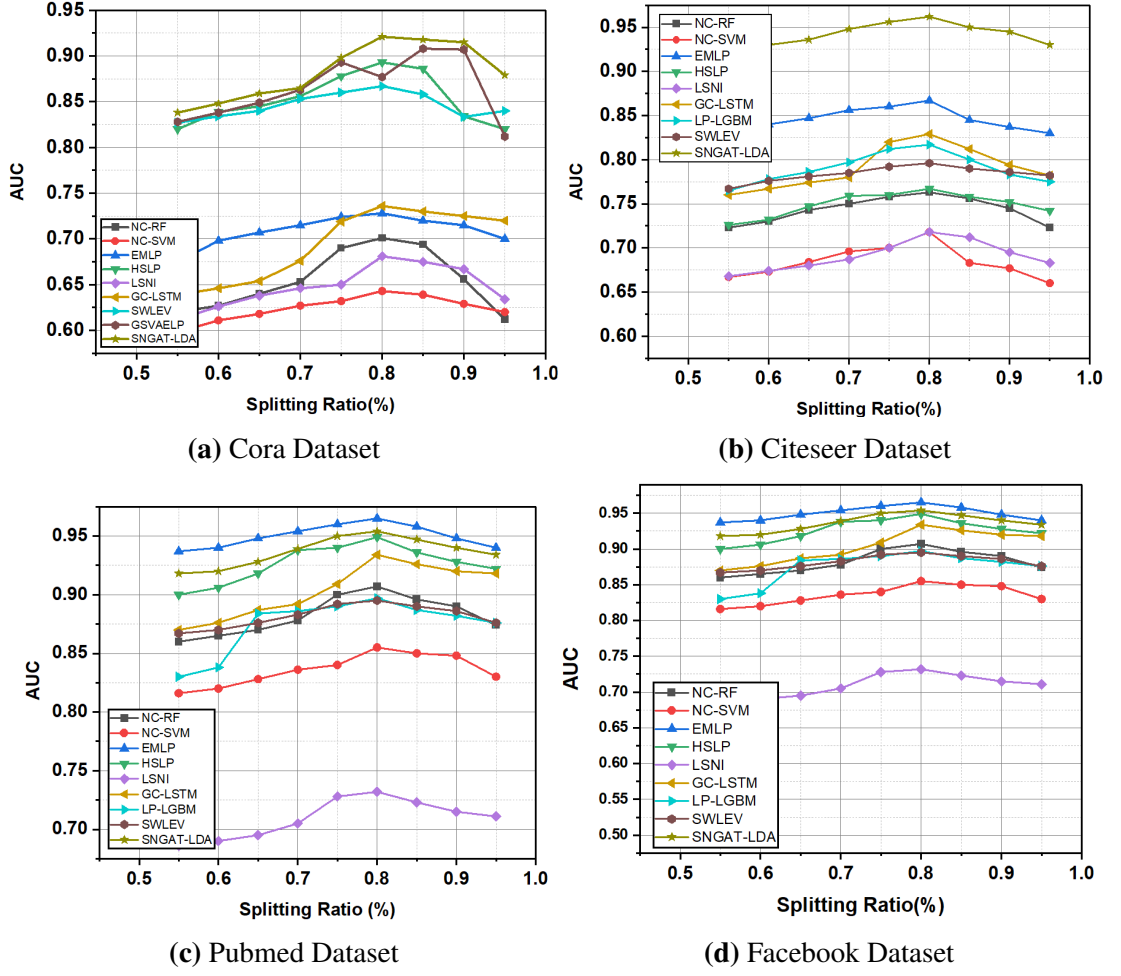
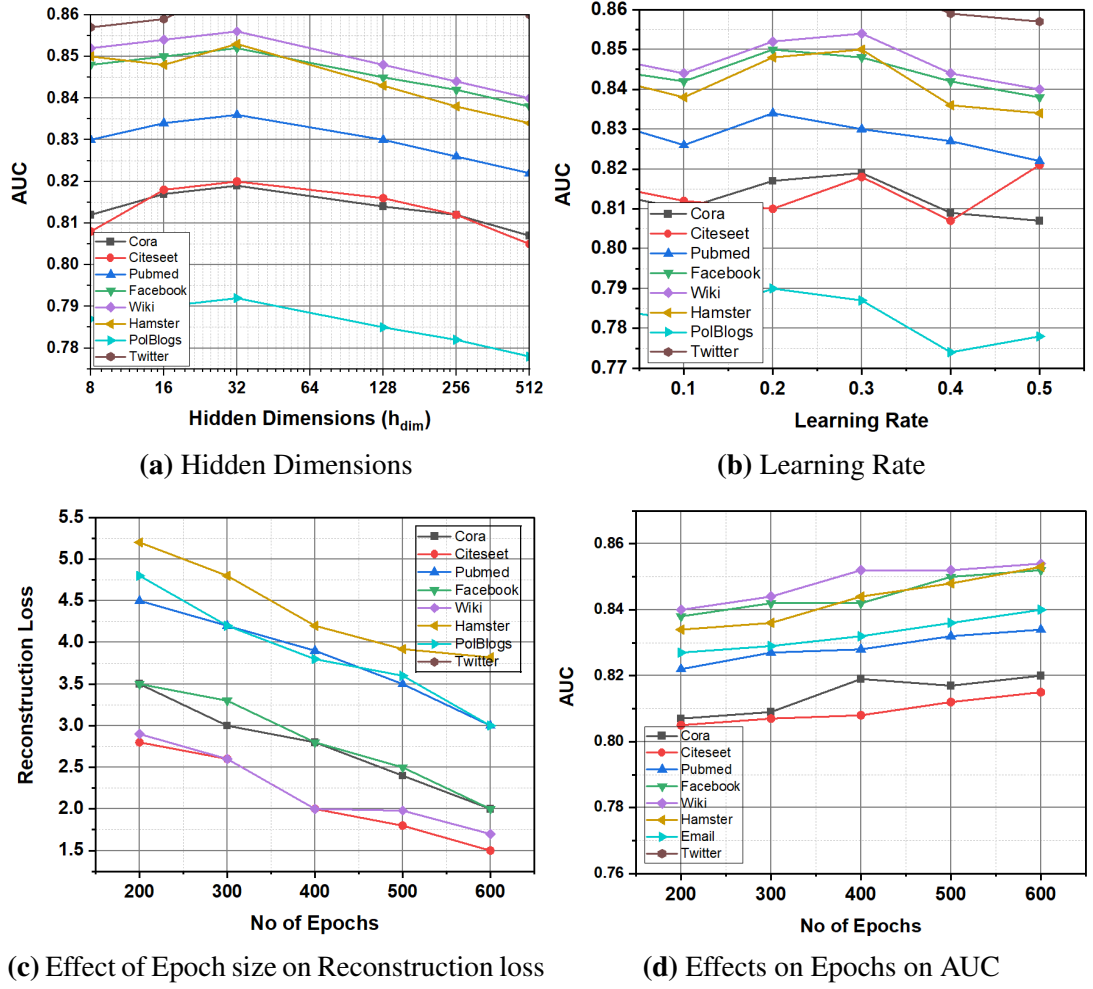


Fig. 7.2: a–d: Study of training and testing dataset ratio to obtain optimal allocation for training and testing. AUC outcomes for various datasets using different link prediction models, with the allocation of training data from 55% to 95% in 5% steps and the remaining for testing.

achieved with a configuration comprising two hidden layers. Further increasing the number of hidden convolution and attention layers beyond two does not change the results. The number of nodes in the input graph can be used to adjust a parameter called the embedding dimensions. Fig. 7.3a shows the effect of dimensionality h_{dim} on AUC in various data sets. The optimal performance for the Cora, Citeseer, Facebook, and Email datasets is achieved at $h_{dim} = 128$, and for other datasets such as Pubmed, Wiki, Hamster, and PolBlogs, the best AUC was found at $h_{dim} = 256$.

Fig. 7.3b shows the variation in AUC scores across different learning rates for mul-

**Fig. 7.3:** Various hyperparameter studies

multiple datasets. The x-axis represents the learning rate values ranging from 0.05 to 0.5, and the y-axis shows the AUC scores ranging from 0.77 to 0.86. The learning rate is a key hyperparameter in machine learning and deep learning algorithms that controls how much the model's weights are updated during training. Wiki and Hamster consistently achieve the highest AUC scores, peaking around a learning rate of 0.3. Facebook and Email also perform well with the same learning rate of 0.3. The reduction in reconstruction loss with an increase in the number of epochs is shown in Fig. 7.3c. For all datasets, reconstruction loss decreases as the number of epochs increases from 200 to 600, which indicates improved learning and better reconstruction of the graph structure over time. The reconstruction loss is minimized across all datasets at epoch 600.

The improvement in AUC for increasing the number of epochs is shown in Fig. 7.3d. The number of epochs affects the performance of the learning algorithm. Increasing the number of epochs 600 enables the algorithm to learn more intricate patterns within the data, which may improve performance.

The conducted hyperparameter analysis demonstrates that model performance, particularly AUC scores, is significantly influenced by key parameters such as hidden dimension size, learning rate, and number of training epochs. The embedding dimensions h_{dim} were found optimal with a value of 128 for the first four datasets and 256 for the other four datasets. A learning rate of 0.3 consistently delivered high AUC scores across most datasets, indicating a well-balanced learning pace. Finally, 600 training epochs were selected as the optimal point where reconstruction loss and AUC stabilized, ensuring thorough model convergence without overfitting. These tuned hyperparameters contribute to the robust and consistent model performance across various graph datasets.

7.5.2 Comparative Performance Study

This section compares the performance of the proposed model with existing link prediction models in social networks and some baseline methods in each dataset, using a variety of evaluation metrics. Since most link prediction techniques are stochastic, we conducted our experiments ten times, averaging the findings for each evaluation parameter to provide a more thorough study. We compare our model with traditional methods and recent methods for link prediction. The traditional methods, like CN, AA, Katz, RA, and SimRank, rely solely on the local topology of the graph and do not incorporate node features or learned representations. Modern approaches, including DeepWalk, node2vec, LINE, GAT, GC-LSTM, GSVAELP, and LP-LGBM, offer substantial improvements by learning latent representations of nodes through neighborhood sampling, attention mechanisms, or deep neural architectures. By comparing

Table 7.3: Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on Cora dataset

Algorithms	Precision	Recall	F1 score	AUC
CN	0.551	0.567	0.526	0.534
AA	0.582	0.553	0.523	0.526
Katz	0.622	0.602	0.543	0.545
RA	0.541	0.567	0.534	0.518
SimRank	0.528	0.545	0.537	0.567
LINE	0.653	0.624	0.643	0.642
Deepwalk	0.655	0.667	0.6626	0.648
node2vec	0.682	0.678	0.676	0.708
GAT	0.684	0.662	0.674	0.712
NC-RF	0.724	0.783	0.773	0.753
NC-SVM	0.736	0.756	0.675	0.672
EMLP	0.754	0.786	0.783	0.746
HSLP	0.678	0.745	0.723	0.732
LSNI	0.723	0.752	0.745	0.733
GC-LSTM	0.792	0.837	0.824	0.819
LP-LGBM	0.751	0.794	0.782	0.723
SWLEV	0.804	0.819	0.815	0.805
GSVAELP	0.814	0.839	<u>0.832</u>	0.822
SNGAT-LDA	<u>0.821</u>	<u>0.840</u>	0.831	<u>0.916</u>

the two classes of algorithms, we demonstrate that SNGAT-LDA effectively captures latent node dependencies while maintaining high predictive accuracy across all evaluation metrics. It strikes a balance between interpretability and performance, outperforming both traditional and deep models on these datasets.

Table 7.3 compares different models based on evaluation metrics of the Cora network dataset, including precision (Pre), recall (Rec), F1 score, and AUC. Each row displays the various algorithms, and each column represents a distinct metric. The numbers in the table indicate the overall performance on each metric. The numbers highlighted and underlined show the best performance for each metric. For example, the highest precision score in the precision column is for the algorithm *SNGAT – LDA*, with a value of 0.821. This indicates that it is highly accurate in predicting links, resulting in fewer false positives.

GSVAELP has the highest F1-score (0.832) suggests that it maintains a good bal-

Table 7.4: Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on Citeseer dataset

Algorithms	Precision	Recall	F1 score	AUC
CN	0.540	0.643	0.567	0.543
AA	0.564	0.626	0.545	0.541
Katz	0.572	0.652	0.572	0.551
RA	0.542	0.647	0.543	0.523
SimRank	0.624	0.653	0.645	0.613
LINE	0.622	0.652	0.631	0.633
DeepWalk	0.654	0.728	0.718	0.715
node2vec	0.661	0.743	0.724	0.722
GAT	0.761	0.821	0.782	0.812
NC-RF	0.802	0.784	0.765	0.769
NC-SVM	0.78	0.765	0.773	0.754
EMLP	0.784	0.794	0.764	0.802
HSLP	0.722	0.772	0.728	0.752
LSNI	0.815	0.832	<u>0.846</u>	0.806
GC-LSTM	0.828	0.847	0.798	<u>0.823</u>
LP-LGBM	0.820	0.807	0.822	0.763
SWLEV	0.822	0.833	0.820	0.786
GSVAELP	0.834	0.849	0.833	0.818
SNGAT-LDA	<u>0.846</u>	<u>0.852</u>	0.842	0.820

ance between precision and recall, making it an effective model overall. A higher F1-score indicates that the model performs well in correctly identifying links while minimizing false positives and false negatives. When comparing recall and AUC scores, it is clear that our proposed model, *SNGAT – LDA*, outperforms all other models, achieving values of 0.843 and 0.826.

The Performance metrics for the Citeseer dataset are shown in Table 7.4. The proposed graph attention-based model *SNGAT – LDA* has outperformed in terms of precision and recall with values of 0.846 and 0.852, while for the F1 and AUC score, the highest F1 score 0.846 is for the model *LSNI*, it effectively balances precision and recall, ensuring that both false positives and false negatives are minimized and the highest AUC score gives the GC-LSTM model with a value of 0.823. The Graph Convolutional Network (GCN) component extracts high-quality node embeddings by aggregating information from neighboring nodes, while the LSTM component facili-

Table 7.5: Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on Pubmed dataset

Algorithms	Precision	Recall	F1 score	AUC
CN	0.652	0.557	0.512	0.521
AA	0.654	0.563	0.516	0.524
Katz	0.662	0.573	0.534	0.53
RA	0.624	0.567	0.543	0.512
SimRank	0.718	0.688	0.693	0.640
LINE	0.678	0.558	0.637	0.664
DeepWalk	0.738	0.758	0.747	0.745
node2vec	0.727	0.778	0.753	0.765
GAT	0.785	0.883	0.872	0.834
NC-RF	0.772	0.693	0.685	0.764
NC-SVM	0.747	0.792	0.756	0.723
EMLP	0.822	0.852	0.822	0.810
HSLP	0.785	0.8382	0.832	0.765
LSNI	0.765	0.734	0.741	0.722
GC-LSTM	0.850	0.872	0.872	0.847
LP-LGBM	0.806	0.833	0.826	0.823
SWLEV	0.778	0.798	0.782	0.780
GSVAELP	0.850	0.894	<u>0.875</u>	0.835
SNGAT-LDA	<u>0.852</u>	<u>0.894</u>	0.863	<u>0.854</u>

tates the learning of sequential patterns in dynamic networks.

Table 7.5 shows the results of the Pubmed dataset. This table shows that our proposed model $SNGAT - LDA$ has outperformed all baselines and existing models regarding precision, recall, and AUC score. The $SNGAT - LDA$ model has a precision of 0.852. This indicates that of all the links predicted as positive, approximately 85.2% are correct. Comparatively, the $SNGAT - LDA$ model's precision is competitive with other models like $GC - LSTM$, with the precision of 0.850 and EMLP model 0.822, whereas the $SNGAT - LDA$ model demonstrates a high recall score of 0.894. The proposed model, SNGAT-LDA, achieves an AUC score of 0.854, demonstrating strong performance by distinguishing between positive and negative instances. It outperforms other models, such as GAT (0.883), NC-SVM (0.792), and node2vec (0.778), indicating its effectiveness in capturing a higher proportion of true positive links.

Table 7.6: Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on Facebook dataset

Algorithms	Precision	Recall	F1 score	AUC
CN	0.652	0.557	0.512	0.521
AA	0.654	0.563	0.516	0.524
Katz	0.662	0.573	0.534	0.53
RA	0.624	0.567	0.543	0.512
SimRank	0.644	0.587	0.623	0.602
LINE	0.678	0.558	0.637	0.664
DeepWalk	0.720	0.733	0.729	0.773
node2vec	0.727	0.778	0.753	0.765
GAT	0.785	0.883	0.872	0.834
NC-RF	0.772	0.693	0.685	0.764
NC-SVM	0.747	0.792	0.756	0.723
EMLP	0.822	0.852	0.822	0.802
HSLP	0.785	0.838	0.832	0.765
LSNI	0.683	0.721	0.718	0.711
GC-LSTM	<u>0.858</u>	0.872	0.872	0.847
LP-LGBM	0.806	0.833	0.826	0.823
SWLEV	0.816	0.825	0.822	0.807
GSVAELP	0.854	0.894	0.863	0.835
SNGAT-LDA	0.843	<u>0.877</u>	<u>0.875</u>	<u>0.874</u>

Table 7.6 shows the results obtained for the Facebook network dataset. Here, the graph attention-based model *SNGAT-LDA* achieves a precision of 0.843; compared to the other models, it falls between the precision value of *GSVAELP* 0.854 and *GC-LSTM* 0.858. The recall value of *SNGAT-LDA* is 0.877, the highest among all models, in terms of the F1 and AUC score. The proposed model *SNGAT-LDA* gives the best results 0.875, and 0.854. Table. 7.8 presents the evaluation metric for different link prediction models on the Wiki dataset. The proposed model, *SNGAT-LDA*, achieves the highest precision at 0.867, outperforming other models such as *GC-LSTM* (0.858), *GSVAELP* (0.863), and *LP-LGBM* (0.825). While *LP-LGBM* excels in recall with a score of 0.859, *SNGAT-LDA* demonstrates balanced performance, evidenced by an F1 score of 0.864. Furthermore, *SNGAT-LDA* leads in AUC with a top score of 0.852, underscoring its effectiveness in overall predictive accuracy.

Table 7.6 shows the results obtained for the Facebook network dataset. Here, the

Table 7.7: Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on Wiki dataset

Algorithms	Precision	Recall	F1 score	AUC
CN	0.632	0.627	0.627	0.53
AA	0.625	0.631	0.631	0.612
Katz	0.654	0.673	0.673	0.617
RA	0.545	0.662	0.662	0.608
SimRank	0.574	0.602	0.588	0.610
LINE	0.688	0.682	0.673	0.680
DeepWalk	0.753	0.812	0.766	0.783
node2vec	0.767	0.826	0.822	0.778
GAT	0.822	0.854	0.854	0.821
NC-RF	0.842	0.813	0.813	0.812
NC-SVM	0.776	0.802	0.802	0.802
EMLP	0.764	0.792	0.792	0.784
HSLP	0.623	0.723	0.723	0.721
LSNI	0.732	0.745	0.738	0.732
GC-LSTM	0.858	0.854	0.847	0.838
LP-LGBM	0.825	<u>0.859</u>	0.844	0.754
SWLEV	0.846	0.833	0.830	0.835
GSVAELP	0.863	0.852	0.862	0.840
SNGAT-LDA	<u>0.867</u>	0.857	<u>0.864</u>	<u>0.852</u>

graph attention-based model *SNGAT-LDA* achieves a precision of 0.843; compared to the other models, it falls between the precision value of *GSVAELP* 0.854 and *GC-LSTM* 0.858. The recall value of *SNGAT-LDA* is 0.877, the highest among all the models. Regarding the F1 and AUC score, the proposed model *SNGAT-LDA* gives the best results of 0.875 and 0.854.

Table. 7.7 presents the evaluation metric for different link prediction models on the Wiki dataset. The proposed model *SNGAT-LDA* achieves the highest precision of 0.867, while the other models, including *GC-LSTM* 0.858, *GSVAELP* 0.863, and *LP-LGBM* 0.825, also have a good precision score, whereas, in terms of recall, the *LP-LGBM* model gives the best result 0.859. The F1 score of the *SNGAT-LDA* model is 0.864, indicating its balanced performance in terms of precision and recall. In terms of AUC, our proposed model, *SNGAT-LDA*, has the highest AUC score of 0.852

Table. 7.8 shows the results obtained for the Hamster dataset. Here, the *EMLP*

Table 7.8: Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on the Hamsterster dataset

Algorithms	Precision	Recall	F1 score	AUC
CN	0.567	0.567	0.527	0.540
AA	0.582	0.553	0.518	0.628
Katz	0.621	0.568	0.523	0.612
RA	0.556	0.554	0.52	0.585
SimRank	0.606	0.578	0.565	0.563
LINE	0.569	0.657	0.635	0.623
DeepWalk	0.627	0.648	0.635	0.630
node2vec	0.678	0.772	0.765	0.733
GAT	0.823	0.872	0.822	0.814
NC-RF	0.775	0.784	0.776	0.726
NC-SVM	0.723	0.782	0.766	0.811
EMLP	<u>0.850</u>	0.812	0.826	0.776
HSLP	0.782	0.762	0.752	0.734
LSNI	0.821	0.783	0.815	0.820
GC-LSTM	0.848	0.864	0.832	<u>0.850</u>
LP-LGBM	0.793	0.753	0.743	0.765
SWLEV	0.768	0.781	0.771	0.780
GSVAELP	0.842	0.856	0.824	0.836
SNGAT-LDA	0.846	<u>0.886</u>	<u>0.843</u>	0.839

model outperformed in precision by 0.850; the other models, like SNGAT-LDA 0.846 and GC-LSTM 0.848, also achieved good precision. Our proposed model SNGAT-LDA has the highest recall of 0.886 and f1 score of 0.843. The GC-LSTM model gives the best AUC of 0.850. GC-LSTM excels in this ranking due to its deep learning-based feature extraction and sequence modeling capability. Overall, the graph attention-based model SNGAT-LDA demonstrates competitive performance in precision, recall, F1 score, and AUC score compared to other models listed in the table.

Table 7.9 presents the evaluation metrics for different link prediction models on the PolBlogs dataset. The table shows that our proposed model gives the best precision score of 0.782, while other models, such as NC-RF, NC-SVM, and LP-LGBM, also demonstrate relatively high precision values above 0.750. The GC-LSTM model shows the highest recall of 0.788, followed by SNGAT-LDA and GSVAELP with recall values above 0.77. In terms of F1 score, SNGAT-LDA achieves an F1 score of 0.780, which

Table 7.9: Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on the PolBlogs dataset

Algorithms	Precision	Recall	F1 score	AUC
CN	0.623	0.642	0.633	0.638
AA	0.665	0.655	0.644	0.664
Katz	0.628	0.647	0.638	0.641
RA	0.652	0.659	0.655	0.651
SimRank	0.712	0.684	0.675	0.664
LINE	0.663	0.641	0.632	0.635
DeepWalk	0.683	0.645	0.676	0.656
node2vec	0.653	0.665	0.661	0.664
GAT	0.743	0.738	0.736	0.745
NC-RF	0.772	0.764	0.768	0.765
NC-SVM	0.757	0.743	0.734	0.723
EMLP	0.780	0.774	0.764	0.772
HSLP	0.748	0.755	0.742	0.754
LSNI	0.665	0.687	0.674	0.670
GC-LSTM	0.775	<u>0.788</u>	0.778	0.776
LP-LGBM	0.765	0.772	0.769	0.774
SWLEV	0.717	0.737	0.815	0.805
GSVAELP	0.775	0.782	0.769	0.764
SNGAT-LDA	<u>0.782</u>	0.784	<u>0.780</u>	<u>0.785</u>

is the highest among the listed models, whereas the other EMLP, GSVAELP, GC-LSTM, and LP-LGBM also demonstrate competitive F1 scores of 0.760. The last evaluation metric is the AUC. The SNGAT-LDA shows the highest AUC values of 0.785, indicating superior discriminative performance compared to other models.

The obtained results from the Email dataset are shown in Table 7.10; the results show that the EMLP model gives the highest precision score of 0.836, while the other models, GSVAELP, GAT, and GC-LSTM, also show good precision scores. Whereas our graph-attention-based approach, SNGAT-LDA, outperformed all other models with a recall and the AUC score of 0.846 and 0.844. The GC-LSTM model shows the highest F1 score of 0.839, while the SNGAT-LDA also demonstrates a competitive F1 score of 0.836. Our proposed SNGAT-LDA model constantly shows strong performance across precision, recall, F1, and AUC scores, making it a promising model for link prediction.

Table 7.10: Evaluation Metrics for different link prediction models based on Precision, Recall, F1 score, and AUC on the Email Dataset

Algorithms	Precision	Recall	F1 score	AUC
CN	0.618	0.576	0.611	0.547
AA	0.644	0.625	0.633	0.63
Katz	0.639	0.644	0.64	0.637
RA	0.573	0.626	0.622	0.632
SimRank	0.647	0.628	0.633	0.627
LINE	0.723	0.711	0.723	0.717
DeepWalk	0.745	0.763	0.752	0.738
node2vec	0.765	0.743	0.74	0.742
GAT	0.816	0.774	0.785	0.774
NC-RF	0.745	0.756	0.749	0.751
NC-SVM	0.728	0.763	0.747	0.743
EMLP	<u>0.836</u>	0.765	0.763	0.76
HSLP	0.768	0.753	0.755	0.763
LSNI	0.742	0.733	0.728	0.716
GC-LSTM	0.831	0.843	<u>0.839</u>	0.835
LP-LGBM	0.781	0.774	0.778	0.783
SWLEV	0.722	0.747	0.766	0.780
GSVAELP	0.820	0.827	0.823	0.827
SNGAT-LDA	0.823	<u>0.846</u>	0.836	<u>0.844</u>

The evaluation metrics outlined in Table 7.11 offer a detailed comparison of traditional, classical, and advanced link prediction models applied to the Twitter dataset, showcasing their performance in terms of Precision, Recall, F1 Score, and AUC. The proposed SNGAT-LDA method stands out with the highest precision (0.878) and AUC (0.906), highlighting its strong ability to accurately predict true links and effectively differentiate between positive and negative cases. On the other hand, GSVAELP achieves the highest recall (0.859), reflecting its proficiency in identifying the majority of actual links, while also maintaining competitive F1 and AUC scores. Overall, SNGAT-LDA consistently outperforms all other baseline models, confirming its effectiveness in capturing complex graph structures and managing real-world sparse social networks, such as Twitter.

Table 7.11: Evaluation metrics for different link prediction models based on Precision, Recall, F1 Score, and AUC on the Twitter dataset

Algorithms	Precision	Recall	F1 Score	AUC
CN	0.551	0.567	0.526	0.534
AA	0.582	0.553	0.523	0.526
Katz	0.622	0.602	0.543	0.545
RA	0.541	0.567	0.534	0.518
SimRank	0.528	0.545	0.537	0.567
LINE	0.653	0.624	0.643	0.642
DeepWalk	0.655	0.667	0.663	0.648
node2vec	0.682	0.678	0.676	0.708
GAT	0.684	0.662	0.674	0.712
NC-RF	0.724	0.783	0.773	0.753
NC-SVM	0.736	0.756	0.675	0.672
EMLP	0.754	0.786	0.783	0.746
HSLP	0.678	0.745	0.723	0.732
LSNI	0.723	0.752	0.745	0.733
GC-LSTM	0.792	0.837	0.824	0.819
LP-LGBM	0.751	0.794	0.782	0.723
SWLEV	0.804	0.819	0.815	0.805
GSVAELP	0.847	<u>0.859</u>	0.848	0.886
SNGAT-LDA	<u>0.878</u>	0.857	0.865	<u>0.906</u>

7.5.3 Average Performance achieved by each LP model on all eight different datasets

This section summarizes the average performance of different link prediction models across all eight datasets used in this study to evaluate the overall effectiveness of the proposed model. The average accuracy of the proposed model compared to the other models can be found in Fig. 7.4. The x axis represents the various link prediction models evaluated. The y - $axis$ shows the average accuracy, the proportion of correctly predicted instances out of the total number of instances. A higher accuracy indicates an overall improved performance of the model. The results suggest that the proposed model SNGAT-LDA has superior performance to other models, as shown by the average accuracy for each model on all the datasets. Facebook, Wiki, and Hamsterster datasets obtain the highest average accuracy. The average accuracy of our proposed

model SNGAT-LDA is 0.836 on the Cora dataset. In contrast, the competitive models, including GC-LSTM and GSVAELP, also perform well with accuracy scores of 0.822 and 0.818, respectively. The average accuracy on the Facebook dataset, the proposed SNGAT-LDA model achieves an average accuracy of 0.854, the highest of the other models. GSVAELP follows closely with an accuracy of 0.835, demonstrating its robust performance in the link prediction task.

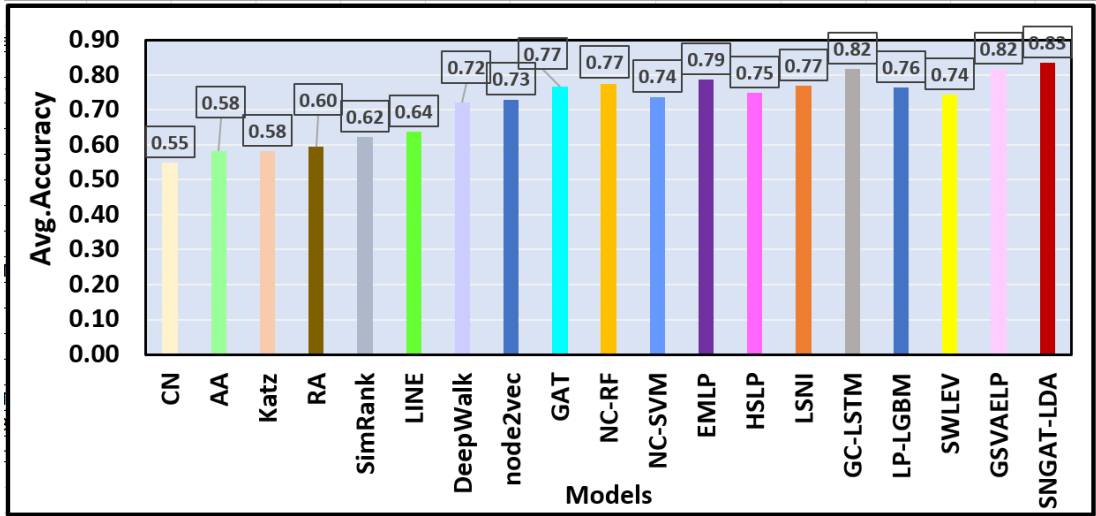


Fig. 7.4: Comparison of Average Accuracy across different link prediction models on all the datasets

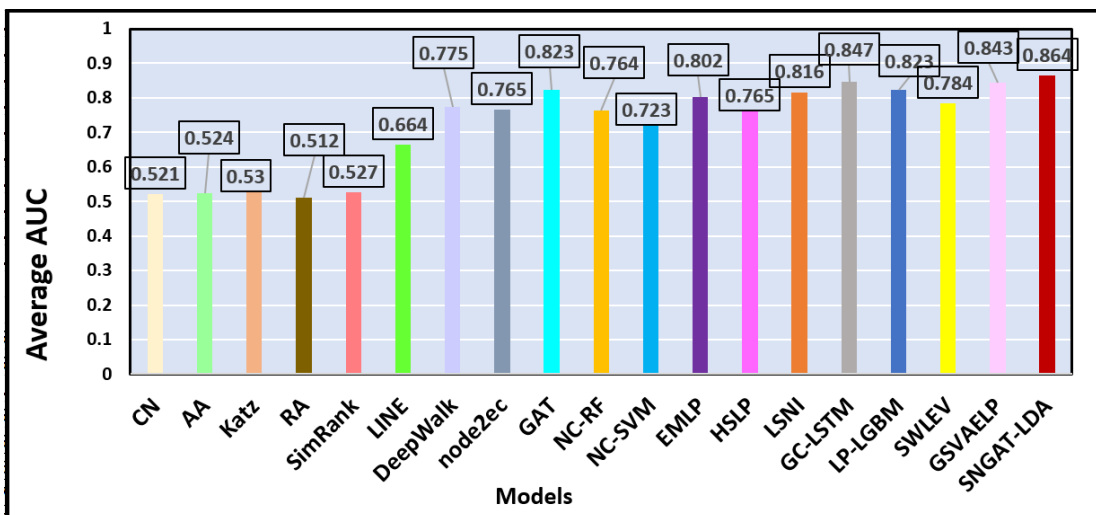


Fig. 7.5: Comparison of Average AUC Scores across different link prediction models on all the datasets

Fig. 7.5 compares the average AUC scores of various link prediction models, demonstrating their effectiveness in predicting missing links in a network. The x-axis lists different models, while the y-axis represents the AUC score ranging from 0 to 1. Traditional heuristic-based models such as CN, AA, Katz, and RA exhibit lower AUC scores, indicating limited predictive performance. In contrast, more advanced graph embedding and learning-based models, such as GAT, GC-LSTM, LP-LGBM, LSNI, SWLEV, and GSVAELP, show improved results. Among all models, the SNGAT-LDA model achieves the highest AUC score, highlighting its superior ability to capture complex network structures and predict links more accurately. This performance gain suggests that the proposed SNGAT-LDA integrates structural and feature-based learning to enhance link prediction accuracy.

7.5.4 Relative Information Gain

The improvement in an algorithm's performance metric is called the information gain. We computed information gain achieved by the proposed model using metrics such as precision, recall, F1 score, and AUC metrics. The information gained for the proposed model over the other models under consideration is shown in Fig. 7.6. This illustrates the percentage improvement of our proposed work over the best performance achieved by other models under consideration in terms of precision, recall, F1 score, and AUC.

7.5.5 Limitations and Future Scope

While the proposed SNGAT-LDA model demonstrates strong performance across various datasets, it has certain limitations. Firstly, the current architecture is optimized for static graphs and does not natively support temporal or dynamic graph structures. Handling streaming or time-evolving data would require additional components, such as temporal attention mechanisms or dynamic node updating strategies. Secondly, the model's reliance on pre-computed node embeddings may limit its adaptability to

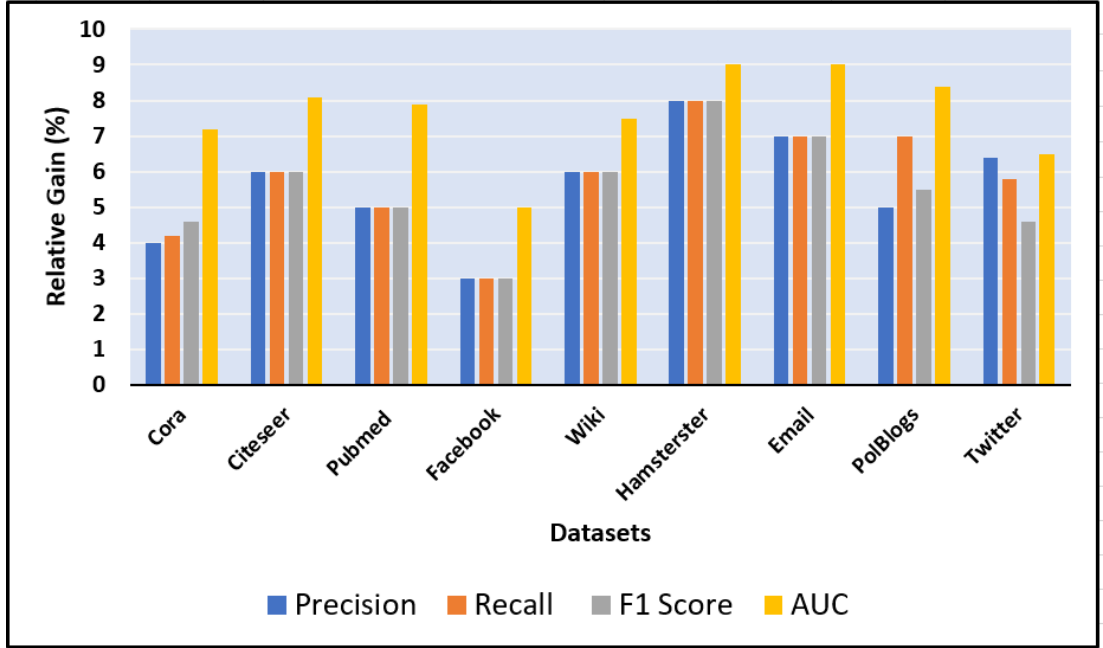


Fig. 7.6: The relative improvement of the proposed model in precision, recall, F1 score, and AUC when compared to the chosen models for each dataset.

inductive settings where unseen nodes are introduced during inference. Lastly, the computational cost associated with attention mechanisms and LDA integration can be significant for large-scale networks. Future work will focus on extending the model to include structurally diverse and sparse networks, such as Twitter graphs, sexual contact networks, and citation subgraphs with extreme sparsity, and exploring inductive link prediction using advanced deep-learning variants. We also plan to explore lightweight approximations of the attention mechanism for enhanced scalability.

7.6 Conclusion

LP is an essential task in network analysis that focuses on identifying potential future connections within a network by examining its existing structure and characteristics. This paper presents an advanced link prediction model that integrates similarity measures, node centrality, a Graph Attention Network (GAT), and Linear Discriminant Analysis (LDA). The proposed model, SNGAT-LDA, combines structural and topolog-

ical information through a graph attention network embedding, significantly improving the accuracy and robustness of link prediction. Key contributions of the SNGAT-LDA model include the integration of Jaccard similarity, LID, and HUBS centrality into a comprehensive feature matrix. By utilizing these metrics, the model assigns weight matrices to links, reflecting the strength and similarity of connections between nodes. This approach overcomes the limitations of traditional GAT, which primarily relies on first-order neighbor information, by implementing an advanced attention mechanism that allocates distinct weight coefficients to various similar nodes. The generated embeddings are then passed through Linear Discriminant Analysis (LDA) for link prediction. The performance of SNGAT-LDA was evaluated across eight different social and other network datasets, demonstrating superior results compared to the baseline and some recently introduced models. Additionally, studies on hyperparameters were conducted to determine the optimal parameter settings, thereby ensuring the best performance of the proposed model.

7.7 Summary

The paper proposes a novel hybrid model for the link prediction task that combines Graph Attention Networks (GAT) with Linear Discriminant Analysis (LDA) to improve prediction accuracy in complex networks. The approach begins by constructing a feature matrix that integrates multiple topological and similarity metrics—such as Jaccard similarity, local interaction density, and hubs/authority centrality—to capture structural and relational information among nodes. Using this feature set, node embeddings are learned through a GAT model, which employs multi-head attention to effectively encode both local and global neighborhood information into the embeddings. These learned embeddings are then used to create a balanced labeled dataset for link prediction, where edge pairs are classified as existing or non-existing links.

To perform the classification, the authors apply LDA, which helps distinguish between positive and negative link samples based on the embedding distribution. Extensive experiments conducted across eight different real-world datasets show that this combined GAT + LDA model significantly outperforms various baseline and recent link prediction methods in terms of predictive performance, demonstrating the effectiveness of integrating attention-based representation learning with a classical discriminative classifier for link prediction tasks.

Chapter 8

UnifiedAttri2Vec-LSTM:Unified Link Prediction: Leveraging Multi-Algorithm Embeddings and Attri2Vec

8.1 Background

Online social networks have become essential for modern society, facilitating rapid communication, information exchange, and building relationships. Usually, these networks are modeled as graphs, with nodes standing in for individuals or objects and edges showing their connections or interactions [147]. In recent years, network science and social network analysis have been the epicenter of numerous research problems within these graphs, such as community detection, link prediction, fake news detection [148], rumor detection [149], influential node identification [150], and many others. Link prediction (LP) is a significant area of study within network science and social networks, focusing on predicting the likelihood of future or absent connections between nodes by analyzing the current network structure [45].

Link prediction (LP) is critical to understanding network dynamics and can help predict new relationships or partnerships. Its applications span various domains, including e-commerce recommendation systems, suggesting friends on social media platforms, and predicting protein-protein interactions in biological networks [10]. Nu-

merous approaches have emerged over time to address the LP problem. These can be broadly categorized into traditional methods, such as similarity-based and probabilistic techniques. More recent approaches include machine learning and embedding-based methods, which take advantage of various aspects of the network [151]. These aspects include local neighborhood structure, global network topology, and node attributes to predict link formation [152]. Traditional approaches often rely solely on structural information or node attributes, limiting their ability to capture the full context of interactions.

Recent advancements in network embedding techniques, such as DeepWalk [153], Node2Vec [154], Graph Convolutional Networks (GCN)[155], GraphSAGE [30], and Graph Attention Networks (GAT) [156], have improved LP performance by learning low-dimensional vector representations of nodes. These methods encode structural information effectively but often underutilize or oversimplify node attributes, limiting their ability to model rich semantic information in attributed networks. In many real-world scenarios, links between nodes are influenced not only by network topology but also by the similarity or complementarity of node attributes. For example, in citation networks such as Cora, two research papers may establish a connection not because they share common neighbors but because they address similar research topics based on their textual features.

To address these limitations, we propose UnifiedAttri2Vec-LSTM, a novel link prediction framework that hierarchically fuses node attribute embeddings and structural neighborhood embeddings to generate robust and task-specific unified representations. Our approach leverages Attri2Vec to map high-dimensional attribute information into a latent space while simultaneously capturing structural context from node neighborhoods. The embeddings are combined through a learnable weighted fusion mechanism, ensuring that both attribute-driven and structure-driven information contribute meaningfully to the final representation. A pairwise edge embedding is then generated using the Hadamard product and fed into a Long Short-Term Memory (LSTM) classi-

fier, which captures complex dependencies across feature dimensions, enhancing link prediction performance. To address these limitations, we propose UnifiedAttri2Vec-LSTM, a novel link prediction framework that hierarchically fuses node attribute embeddings and structural neighborhood embeddings to generate robust and task-specific unified representations. Our approach leverages Attri2Vec to map high-dimensional attribute information into a latent space while simultaneously capturing structural context from node neighborhoods. The embeddings are combined through a learnable weighted fusion mechanism, ensuring that both attribute-driven and structure-driven information contribute meaningfully to the final representation. A pairwise edge embedding is then generated using the Hadamard product and fed into a Long Short-Term Memory (LSTM) classifier, which captures complex dependencies across feature dimensions, enhancing link prediction performance. The main contributions of this study are summarized below.

Chapter 9

Conclusion, Future Scope and Social Impact

This chapter summarizes the research conducted on link prediction in social networks, focusing on the development of advanced frameworks, such as GSVAELP, MetaLP-DGI, and MetaLP-EC, as well as hybrid approaches that integrate graph embedding and ensemble learning. This reflects the effectiveness of the proposed methods in capturing complex network structures, incorporating node centrality, and improving predictive accuracy through extensive experimental validation on benchmark datasets. The chapter also discusses future work, including the exploration of transformer-based graph models, temporal attention mechanisms, and the integration of heterogeneous and multimodal network data for more comprehensive link prediction. Finally, it highlights the societal impact of this research in enhancing applications such as friend recommendation systems, knowledge graph completion, e-commerce personalization, and biological network analysis, ultimately contributing to better connectivity, knowledge discovery, and decision-making in various domains.

9.1 Conclusion

This thesis presents a series of innovative solutions for the problem of link prediction in complex networks by progressively integrating structural learning, attribute learning, centrality awareness, attention mechanisms, ensemble learning, and meta-learning.

Through six research contributions—GSVAELP, MetaLP-DGI, Hybrid Graph Embedding with Ensemble Learning, MetaLP-EC, GAT-LDA based prediction, and Unified Attri2Vec-based embedding integration—this work establishes a holistic and scalable framework that addresses the limitations of traditional link prediction such as cold-start, sparsity, temporal evolution, weak generalization, and poor heterogeneity handling.

The first contribution, GSVAELP, introduced a hybrid GraphSAGE-VAE model that leveraged local neighborhood aggregation with probabilistic latent-space embedding, successfully capturing both structural dependencies and latent relational patterns. This laid the foundation for robust structure-and-attribute-aware link prediction. The second study, MetaLP-DGI, introduced centrality-aware Deep Graph Infomax with meta-learning, enhancing embedding quality by incorporating influential node characteristics while improving generalization across heterogeneous networks. The third work, Hybrid Graph Embedding and Ensemble Learning, demonstrated that combining multiple embeddings with ensemble classifiers significantly improves predictive consistency and reduces model bias.

To address temporal link prediction, the fourth work, MetaLP-EC, introduced an evolving centrality-based learning mechanism that dynamically captures network evolution using GCN embeddings and meta-learning, effectively addressing time-aware link prediction challenges. The fifth contribution, GAT-LDA-based enhancement, revealed that integrating attention mechanisms with Linear Discriminant Analysis improves class separability in embedding space, leading to more discriminative link prediction. Lastly, the proposed Unified Link Prediction with Multi-Algorithm Embeddings and Attri2Vec provided a generalized fusion mechanism that synthesizes embeddings from Node2Vec, DeepWalk, GCN, GraphSAGE, and GAT using Attri2Vec, showcasing high adaptability across varied network types.

Collectively, these contributions address critical challenges in link prediction—sparsity, attribute integration, dynamic evolution, heterogeneity, class imbalance, and weak em-

bedding quality—while introducing novel paradigms like centrality-aware learning, unified embedding fusion, and meta-learning-driven adaptation. The proposed frameworks have demonstrated superior performance across real-world networks with consistent improvements in AUC, Precision, Recall, and F1-score.

In conclusion, this thesis not only advances the state of link prediction by integrating multiple learning paradigms but also opens new research directions in unified graph representation, interpretability, dynamic multilayer networks, and few-shot link prediction. The research establishes a strong foundation for future exploration of personalized recommendation systems, biological network analysis, temporal-social interaction modeling, and intelligent cyber-physical systems.

9.2 Future Scope

Future investigations can further enhance this research in several promising directions. First, new models can focus on multilayer, heterogeneous, and knowledge graphs, where nodes represent diverse entities such as people, products, proteins, and institutions, and edges capture multiple types of relationships. This would enable link prediction in more realistic and complex ecosystems like healthcare, finance, and social combined networks.

Second, temporal link prediction can be advanced using Graph Transformers, Neural ODEs, and diffusion models to better capture evolving connectivity patterns and dynamic centrality changes over time. Incorporating incremental and lifelong learning would allow models to update predictions without retraining from scratch as graphs grow.

Third, integrating explainable AI (XAI) with graph learning models would help identify why certain links are predicted, making them more trustworthy for critical applications like medical diagnosis, financial risk analysis, and cybersecurity.

Fourth, developing privacy-preserving and federated link prediction frameworks

would allow models to learn from decentralized graph data without compromising sensitive information—crucial for healthcare, smart cities, banking, and government networks.

Fifth, few-shot and zero-shot link prediction techniques can be explored for cold-start scenarios, where limited or no historical links are available, enabling better performance in sparse and emerging networks.

Finally, integrating graph learning with Large Language Models (LLMs) could lead to hybrid Graph-LLM systems, capable of performing reasoning, interpretation, and even answering complex questions based on graph structures, thereby opening a path toward intelligent network-aware AI agents.

9.3 Social Impact

The proposed link prediction research has strong societal relevance and broad interdisciplinary impact. In healthcare, it can help identify hidden connections among diseases, genes, and patients, enabling early disease prediction, personalized treatment, and improved medical research collaboration. It also supports hospital resource allocation and patient referral systems by forecasting network interactions. In transportation and smart cities, link prediction can enhance intelligent traffic routing, detect emerging connectivity patterns between locations, optimize public transport networks, and support accident risk prediction. It helps improve urban planning and enhances informed decision-making for smart mobility.

In education and research communities, it fosters academic collaboration by predicting likely co-authorships, identifying expert networks, and recommending research partnerships. It strengthens digital learning platforms by predicting learner interactions, peer support, and topic evolution networks. In social media and digital communities, it enhances friend recommendation, builds healthier online communities, and promotes meaningful interactions while reducing isolation, misinformation, and echo

chambers. For the e-commerce and financial sectors, link prediction supports product recommendations, customer behavior prediction, fraud detection, and risk assessment, thereby improving user experience, trust, and economic growth. In cybersecurity, it identifies abnormal or malicious links, predicts potential data breaches, and secures communication networks by detecting suspicious relationships in fraud, crime, or intrusion networks. In environmental and disaster management, it helps model and predict relationships in climate data, weather networks, and natural disaster propagation patterns, supporting decision-making for early warning and resource distribution.

Overall, this work contributes to a smarter, interconnected, and data-driven society, improving collaboration, decision-making, safety, and well-being across various domains.

References

Bibliography

- [1] N. N. Daud, S. H. Ab Hamid, M. Saadoon, F. Sahran, and N. B. Anuar, “Applications of link prediction in social networks: A review,” *Journal of Network and Computer Applications*, vol. 166, p. 102716, 2020.
- [2] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos, “Fast and accurate link prediction in social networking systems,” *Journal of Systems and Software*, vol. 85, no. 9, pp. 2119–2132, 2012.
- [3] V. Martínez, F. Berzal, and J.-C. Cubero, “A survey of link prediction in complex networks,” *ACM computing surveys (CSUR)*, vol. 49, no. 4, pp. 1–33, 2016.
- [4] P. Szyman and D. Barbucha, “Link prediction in time-evolving organizational social networks,” *Procedia Computer Science*, vol. 225, pp. 2816–2825, 2023.
- [5] K. Berahmand, E. Nasiri, M. Rostami, and S. Forouzandeh, “A modified deep-walk method for link prediction in attributed social network,” *Computing*, vol. 103, no. 10, pp. 2227–2249, 2021.
- [6] H. Wu, C. Song, Y. Ge, and T. Ge, “Link prediction on complex networks: an experimental survey,” *Data science and engineering*, vol. 7, no. 3, pp. 253–278, 2022.
- [7] Y. Yao, T. Cheng, X. Li, Y. He, F. Yang, T. Li, Z. Liu, and Z. Xu, “Link prediction based on the mutual information with high-order clustering structure of

- nodes in complex networks,” *Physica A: Statistical Mechanics and its Applications*, vol. 610, p. 128428, 2023.
- [8] S. Kumar, B. Panda, and D. Aggarwal, “Community detection in complex networks using network embedding and gravitational search algorithm,” *Journal of Intelligent Information Systems*, vol. 57, no. 1, pp. 51–72, 2021.
- [9] N. N. Daud, S. H. Ab Hamid, M. Saadoon, F. Sahran, and N. B. Anuar, “Applications of link prediction in social networks: A review,” *Journal of Network and Computer Applications*, vol. 166, p. 102716, 2020.
- [10] J. Yang and Y. Wu, “Link prediction based on depth structure in social networks,” *International Journal of Machine Learning and Cybernetics*, vol. 15, no. 10, pp. 4639–4657, 2024.
- [11] Y. Liu, S. Liu, F. Yu, and X. Yang, “Link prediction algorithm based on the initial information contribution of nodes,” *Information Sciences*, vol. 608, pp. 1591–1616, 2022.
- [12] L. Yao, L. Wang, L. Pan, and K. Yao, “Link prediction based on common-neighbors for dynamic social network,” *Procedia Computer Science*, vol. 83, pp. 82–89, 2016.
- [13] L. A. Adamic and E. Adar, “Friends and neighbors on the web,” *Social networks*, vol. 25, no. 3, pp. 211–230, 2003.
- [14] P. Jaccard, “Étude comparative de la distribution florale dans une portion des alpes et des jura,” *Bull Soc Vaudoise Sci Nat*, vol. 37, pp. 547–579, 1901.
- [15] M. Mitzenmacher, “A brief history of generative models for power law and log-normal distributions,” *Internet mathematics*, vol. 1, no. 2, pp. 226–251, 2004.
- [16] G. Salton, “Modern information retrieval,” (*No Title*), 1983.

-
- [17] T. Zhou, L. Lü, and Y.-C. Zhang, “Predicting missing links via local information,” *The European Physical Journal B*, vol. 71, no. 4, pp. 623–630, 2009.
- [18] S. Anand, Rahul, A. Mallik, and S. Kumar, “Integrating node centralities, similarity measures, and machine learning classifiers for link prediction,” *Multimedia tools and applications*, vol. 81, no. 27, pp. 38 593–38 621, 2022.
- [19] L. Katz, “A new status index derived from sociometric analysis,” *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.
- [20] L. Lü, C.-H. Jin, and T. Zhou, “Similarity index based on local paths for link prediction of complex networks,” *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 80, no. 4, p. 046122, 2009.
- [21] R. Guimerà and M. Sales-Pardo, “Missing and spurious interactions and the reconstruction of complex networks,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 52, pp. 22 073–22 078, 2009.
- [22] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [23] J. M. Keller, M. R. Gray, and J. A. Givens, “A fuzzy k-nearest neighbor algorithm,” *IEEE transactions on systems, man, and cybernetics*, no. 4, pp. 580–585, 1985.
- [24] S. R. Safavian and D. Landgrebe, “A survey of decision tree classifier methodology,” *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.
- [25] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013.
- [26] S. Mitra and S. K. Pal, “Fuzzy multi-layer perceptron, inferencing and rule generation,” *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 51–63, 1995.

-
- [27] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.
- [28] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.
- [29] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “Line: Large-scale information network embedding,” in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.
- [30] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” *Advances in neural information processing systems*, vol. 30, 2017.
- [31] Y. Xiu, X. Liu, K. Cao, B. Chen, and W. K. V. Chan, “An extended self-representation model of complex networks for link prediction,” *Information Sciences*, vol. 662, p. 120254, 2024.
- [32] N. M. A. Ibrahim and L. Chen, “Link prediction in dynamic social networks by integrating different types of information,” *Applied Intelligence*, vol. 42, no. 4, pp. 738–750, 2015.
- [33] S. Kumar, A. Mallik, A. Khetarpal, and B. S. Panda, “Influence maximization in social networks using graph embedding and graph neural network,” *Information Sciences*, vol. 607, pp. 1617–1636, 2022.
- [34] L. Si, L. Li, H. Luo, and Z. Ma, “Link prediction in multiplex social networks: An information transmission approach,” *Chaos, Solitons & Fractals*, vol. 189, p. 115683, 2024.

-
- [35] W. Cai, X. Chang, and P. Yang, "Link prediction in multilayer social networks using reliable local random walk and boosting ensemble classifier," *Chaos, Solitons & Fractals*, vol. 188, p. 115530, 2024.
- [36] N. Talasu, A. Jonnalagadda, S. S. A. Pillai, and J. Rahul, "A link prediction based approach for recommendation systems," in *2017 international conference on advances in computing, communications and informatics (ICACCI)*. IEEE, 2017, pp. 2059–2062.
- [37] S. Li, X. Song, H. Lu, L. Zeng, M. Shi, and F. Liu, "Friend recommendation for cross marketing in online brand community based on intelligent attention allocation link prediction algorithm," *Expert Systems with Applications*, vol. 139, p. 112839, 2020.
- [38] Y.-L. Chen, C.-H. Hsiao, and C.-C. Wu, "An ensemble model for link prediction based on graph embedding," *Decision Support Systems*, vol. 157, p. 113753, 2022.
- [39] J. Liu, L. Song, G. Wang, and X. Shang, "Meta-hgt: Metapath-aware hypergraph transformer for heterogeneous information network embedding," *Neural Networks*, vol. 157, pp. 65–76, 2023.
- [40] D. Garompolo and V. Inzillo, "A gcnn-lstm framework for link prediction in dynamic iiot networks," *Internet of Things*, vol. 29, p. 101455, 2025.
- [41] A. K. Singh and K. Lakshmanan, "Pihnb: Popularity, interests, location used hidden naive bayesian-based model for link prediction in dynamic social networks," *Neurocomputing*, vol. 461, pp. 562–576, 2021.
- [42] R. Karami, S. M. Vahidipour, and A. Rezvanian, "Seal+: A subgraph-enhanced framework for link prediction with graph neural networks," *Journal of Industrial Information Integration*, vol. 44, p. 100802, 2025.

-
- [43] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy, “Make new friends, but keep the old: recommending people on social networking sites,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2009, pp. 201–210.
- [44] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *arXiv preprint arXiv:1710.10903*, 2017.
- [45] S. Kumar, A. Mallik, and B. Panda, “Link prediction in complex networks using node centrality and light gradient boosting machine,” *World Wide Web*, vol. 25, no. 6, pp. 2487–2513, 2022.
- [46] T. Li, J. Zhang, S. Y. Philip, Y. Zhang, and Y. Yan, “Deep dynamic network embedding for link prediction,” *IEEE Access*, vol. 6, pp. 29 219–29 230, 2018.
- [47] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A. J. Smola, “Distributed large-scale natural graph factorization,” in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 37–48.
- [48] T. Li, R. Zhang, B. Niu, Y. Yao, J. Ma, J. Jiang, and Z. Zhao, “Link prediction based on local structure and node information along local paths,” *The Computer Journal*, vol. 67, no. 1, pp. 45–56, 2024.
- [49] P. I. Fierens and L. Chaves Rêgo, “A similarity-based assortativity measure for complex networks,” *Journal of Complex Networks*, vol. 12, no. 2, p. cnae010, 2024.
- [50] G. Chen, C. Xu, J. Wang, J. Feng, and J. Feng, “Nonnegative matrix factorization for link prediction in directed complex networks using pagerank and asymmetric link clustering information,” *Expert Systems with Applications*, vol. 148, p. 113290, 2020.

-
- [51] X. Mo, J. Pang, and Z. Liu, “Effective link prediction with topological and temporal information using wavelet neural network embedding,” *The Computer Journal*, vol. 64, no. 3, pp. 325–336, 2021.
- [52] B. Rozemberczki, C. Allen, and R. Sarkar, “Multi-scale attributed node embedding,” *Journal of Complex Networks*, vol. 9, no. 2, p. cnab014, 2021.
- [53] Y.-L. Chen, C.-H. Hsiao, and C.-C. Wu, “An ensemble model for link prediction based on graph embedding,” *Decision Support Systems*, vol. 157, p. 113753, 2022.
- [54] J. Chen, X. Wang, and X. Xu, “Gc-lstm: Graph convolution embedded lstm for dynamic network link prediction,” *Applied Intelligence*, pp. 1–16, 2022.
- [55] E. Nasiri, K. Berahmand, and Y. Li, “Robust graph regularization nonnegative matrix factorization for link prediction in attributed networks,” *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 3745–3768, 2023.
- [56] M.-Y. Zhou, F. Wang, Z. Chen, J. Wu, G. Liu, and H. Liao, “Weak link prediction based on hyper latent distance in complex network,” *Expert Systems with Applications*, vol. 238, p. 121843, 2024.
- [57] H. Wang, Y. Teng, L. Qin, X. Guo, and P. Hu, “A multiple attention layer-shareable method for link prediction in multilayer networks,” *ACM Transactions on Knowledge Discovery from Data*, vol. 19, no. 2, pp. 1–22, 2025.
- [58] X. Jiang, Z. Yu, C. Hai, H. Liu, X. Wu, and T. Ward, “Dnformer: Temporal link prediction with transfer learning in dynamic networks,” *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 3, pp. 1–21, 2023.
- [59] Z. Qiu, J. Wu, W. Hu, B. Du, G. Yuan, and P. S. Yu, “Temporal link prediction with motifs for social networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 3, pp. 3145–3158, 2021.

-
- [60] P. Zhao, L. You, M. Wang, and Q. Peng, "Multilayer network link prediction considering multiple correlation features," *Expert Systems with Applications*, p. 127700, 2025.
- [61] X. Li and H. Chen, "Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach," *Decision Support Systems*, vol. 54, no. 2, pp. 880–890, 2013.
- [62] M. Fire, L. Tenenboim-Chekina, R. Puzis, O. Lesser, L. Rokach, and Y. Elovici, "Computationally efficient link prediction in a variety of social networks," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 1, pp. 1–25, 2014.
- [63] I. Elkabani and R. A. Aboo Khachfeh, "Homophily-based link prediction in the facebook online social network: a rough sets approach," *Journal of Intelligent Systems*, vol. 24, no. 4, pp. 491–503, 2015.
- [64] C. Dai, L. Chen, B. Li, and Y. Li, "Link prediction in multi-relational networks based on relational similarity," *Information Sciences*, vol. 394, pp. 198–216, 2017.
- [65] J. Chen, J. Zhang, X. Xu, C. Fu, D. Zhang, Q. Zhang, and Q. Xuan, "E-lstm-d: A deep learning framework for dynamic network link prediction," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3699–3712, 2019.
- [66] K. Li, L. Tu, and L. Chai, "Ensemble-model-based link prediction of complex networks," *Computer Networks*, vol. 166, p. 106978, 2020.
- [67] A. Zareie and R. Sakellariou, "Similarity-based link prediction in social networks using latent relationships between the users," *Scientific reports*, vol. 10, no. 1, p. 20137, 2020.

-
- [68] S. Anand, Rahul, A. Mallik, and S. Kumar, “Integrating node centralities, similarity measures, and machine learning classifiers for link prediction,” *Multimedia tools and applications*, vol. 81, no. 27, pp. 38 593–38 621, 2022.
- [69] S. Mishra, S. S. Singh, A. Kumar, and B. Biswas, “Elp: Link prediction in social networks based on ego network perspective,” *Physica A: Statistical Mechanics and its Applications*, vol. 605, p. 128008, 2022.
- [70] S. P. Tripathi, R. K. Yadav, and A. K. Rai, “Network embedding based link prediction in dynamic networks,” *Future Generation Computer Systems*, vol. 127, pp. 409–420, 2022.
- [71] E. P. Barracchia, G. Pio, A. Bifet, H. M. Gomes, B. Pfahringer, and M. Ceci, “Lp-robin: link prediction in dynamic networks exploiting incremental node embedding,” *Information Sciences*, vol. 606, pp. 702–721, 2022.
- [72] Z. Zhao, Z. Gou, Y. Du, J. Ma, T. Li, and R. Zhang, “A novel link prediction algorithm based on inductive matrix completion,” *Expert Systems with Applications*, vol. 188, p. 116033, 2022.
- [73] X. Liu, X. Li, G. Fiumara, and P. De Meo, “Link prediction approach combined graph neural network with capsule network,” *Expert Systems with Applications*, vol. 212, p. 118737, 2023.
- [74] H. Wang, Z. Cui, R. Liu, L. Fang, and Y. Sha, “A multi-type transferable method for missing link prediction in heterogeneous social networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 11, pp. 10 981–10 991, 2023.
- [75] D. Shu, T. Chen, M. Jin, C. Zhang, M. Du, and Y. Zhang, “Knowledge graph large language model (kg-llm) for link prediction,” *arXiv preprint arXiv:2403.07311*, 2024.

-
- [76] R. Nie, G. Wang, Q. Liu, and C. Peng, “Link prediction for attribute and structure learning based on attention mechanism,” *Applied Soft Computing*, p. 113268, 2025.
- [77] S. Anand, Rahul, A. Mallik, and S. Kumar, “Integrating node centralities, similarity measures, and machine learning classifiers for link prediction,” *Multimedia tools and applications*, vol. 81, no. 27, pp. 38 593–38 621, 2022.
- [78] S. Li, N. Cai, and Z. Yu, “Constructing friendship in social networks for precise peer influence marketing by consensus link prediction algorithm,” *Multimedia tools and applications*, vol. 79, pp. 7649–7668, 2020.
- [79] L. Zhang, M. Zhao, and D. Zhao, “Bipartite graph link prediction method with homogeneous nodes similarity for music recommendation,” *Multimedia Tools and Applications*, vol. 79, pp. 13 197–13 215, 2020.
- [80] B. Kaya, “A hotel recommendation system based on customer location: a link prediction approach,” *Multimedia Tools and Applications*, vol. 79, pp. 1745–1758, 2020.
- [81] K. Abbas, A. Abbasi, S. Dong, L. Niu, L. Yu, B. Chen, S.-M. Cai, and Q. Hasan, “Application of network link prediction in drug discovery,” *BMC bioinformatics*, vol. 22, no. 1, p. 187, 2021.
- [82] S. Kumar, A. Mallik, and B. Panda, “Link prediction in complex networks using node centrality and light gradient boosting machine,” *World Wide Web*, vol. 25, no. 6, pp. 2487–2513, 2022.
- [83] P. Cui, X. Wang, J. Pei, and W. Zhu, “A survey on network embedding,” *IEEE transactions on knowledge and data engineering*, vol. 31, no. 5, pp. 833–852, 2018.

-
- [84] H. Cai, V. W. Zheng, and K. C.-C. Chang, “A comprehensive survey of graph embedding: Problems, techniques, and applications,” *IEEE transactions on knowledge and data engineering*, vol. 30, no. 9, pp. 1616–1637, 2018.
- [85] P. Wang, B. Xu, Y. Wu, and X. Zhou, “Link prediction in social networks: the state-of-the-art,” *arXiv preprint arXiv:1411.5118*, 2014.
- [86] A. Kumar, S. S. Singh, K. Singh, and B. Biswas, “Link prediction techniques, applications, and performance: A survey,” *Physica A: Statistical Mechanics and its Applications*, vol. 553, p. 124289, 2020.
- [87] D. Malhotra and R. Goyal, “Link prediction in complex networks using information-theoretic measures,” *Journal of Complex Networks*, vol. 8, no. 4, p. cnaa035, 2020.
- [88] L. Yao, L. Wang, L. Pan, and K. Yao, “Link prediction based on common-neighbors for dynamic social network,” *Procedia Computer Science*, vol. 83, pp. 82–89, 2016.
- [89] R. Rossi and N. Ahmed, “The network data repository with interactive graph analytics and visualization,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.
- [90] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton, “Gemsec: Graph embedding with self clustering,” in *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, 2019, pp. 65–72.
- [91] W. Zhou, J. Gu, and Y. Jia, “h-index-based link prediction methods in citation network,” *Scientometrics*, vol. 117, pp. 381–390, 2018.
- [92] W. Gu, F. Gao, X. Lou, and J. Zhang, “Link prediction via graph attention network,” *arXiv preprint arXiv:1910.04807*, 2019.

-
- [93] K. Li, L. Tu, and L. Chai, “Ensemble-model-based link prediction of complex networks,” *Computer Networks*, vol. 166, p. 106978, 2020.
- [94] S. Kumar, A. Mallik, and B. Panda, “Link prediction in complex networks using node centrality and light gradient boosting machine,” *World Wide Web*, vol. 25, no. 6, pp. 2487–2513, 2022.
- [95] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “Line: Large-scale information network embedding,” in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.
- [96] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” *Advances in neural information processing systems*, vol. 30, 2017.
- [97] G. Zare, N. J. Navimipour, M. Hosseinzadeh, and A. Sahafi, “Network link prediction via deep learning method: A comparative analysis with traditional methods,” *Engineering Science and Technology, an International Journal*, vol. 56, p. 101782, 2024.
- [98] A. K. Rai, S. P. Tripathi, and R. K. Yadav, “A novel similarity-based parameterized method for link prediction,” *Chaos, Solitons & Fractals*, vol. 175, p. 114046, 2023.
- [99] J. Wu, L. He, T. Jia, and L. Tao, “Temporal link prediction based on node dynamics,” *Chaos, Solitons & Fractals*, vol. 170, p. 113402, 2023.
- [100] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

-
- [101] I. A. Kovács, K. Luck, K. Spirohn, Y. Wang, C. Pollis, S. Schlabach, W. Bian, D.-K. Kim, N. Kishore, T. Hao *et al.*, “Network-based prediction of protein interactions,” *Nature communications*, vol. 10, no. 1, p. 1240, 2019.
- [102] J. Leskovec and J. Mcauley, “Learning to discover social circles in ego networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [103] A. Paranjape, A. R. Benson, and J. Leskovec, “Motifs in temporal networks,” in *Proceedings of the tenth ACM international conference on web search and data mining*, 2017, pp. 601–610.
- [104] R. Rossi and N. Ahmed, “The network data repository with interactive graph analytics and visualization,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.
- [105] J. Chen, X. Wang, and X. Xu, “Gc-lstm: Graph convolution embedded lstm for dynamic network link prediction,” *Applied Intelligence*, vol. 52, no. 7, pp. 7513–7528, 2022.
- [106] W. Rebhi, N. Ben Yahia, and N. Bellamine Ben Saoud, “Stable communities detection method for temporal multiplex graphs: heterogeneous social network case study,” *The Computer Journal*, vol. 64, no. 3, pp. 418–431, 2021.
- [107] H. Gul, F. Al-Obeidat, A. Amin, and F. Moreira, “Analyzing complex networks: Extracting key characteristics and measuring structural similarities,” *Expert Systems*, vol. 41, no. 1, p. e13470, 2024.
- [108] N. N. Daud, S. H. Ab Hamid, M. Saadoon, F. Sahran, and N. B. Anuar, “Applications of link prediction in social networks: A review,” *Journal of Network and Computer Applications*, vol. 166, p. 102716, 2020.

-
- [109] A. Kumar, S. S. Singh, K. Singh, and B. Biswas, "Link prediction techniques, applications, and performance: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 553, p. 124289, 2020.
- [110] H. Wu, C. Song, Y. Ge, and T. Ge, "Link prediction on complex networks: an experimental survey," *Data science and engineering*, vol. 7, no. 3, pp. 253–278, 2022.
- [111] J. Leskovec and J. Mcauley, "Learning to discover social circles in ego networks," *Advances in neural information processing systems*, vol. 25, 2012.
- [112] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton, "Gemsec: Graph embedding with self clustering," in *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, 2019, pp. 65–72.
- [113] R. Rossi and N. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.
- [114] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the semantic web," in *International semantic Web conference*. Springer, 2003, pp. 351–368.
- [115] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 us election: divided they blog," in *Proceedings of the 3rd international workshop on Link discovery*, 2005, pp. 36–43.
- [116] M. Agrawal, M. Zitnik, and J. Leskovec, "Large-scale analysis of disease pathways in the human interactome," in *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, vol. 23, 2018, p. 111.

-
- [117] S. Jayachitra Devi and B. Singh, “Link prediction model based on geodesic distance measure using various machine learning classification models,” *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 5, pp. 6663–6675, 2020.
- [118] K. Li, L. Tu, and L. Chai, “Ensemble-model-based link prediction of complex networks,” *Computer Networks*, vol. 166, p. 106978, 2020.
- [119] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.
- [120] S. Kumar, A. Mallik, and B. Panda, “Link prediction in complex networks using node centrality and light gradient boosting machine,” *World Wide Web*, vol. 25, no. 6, pp. 2487–2513, 2022.
- [121] F. Ziya and S. Kumar, “Gsvaelp: integrating graphsage and variational autoencoder for link prediction,” *Multimedia Tools and Applications*, pp. 1–27, 2024.
- [122] A. Bouyer, P. Shahgholi, B. Arasteh, and E. B. Tirkolaee, “Local core expanding-based label diffusion and local deep embedding for fast community detection algorithm in social networks,” *Computers and Electrical Engineering*, vol. 119, p. 109502, 2024.
- [123] D. Mavaluru, A. Mubarakali, B. R. Narapureddy, J. Ramakrishnan, R. John, N. Ravishankar, and P. Karthika, “Deep convolutional neural network based real-time abnormal behavior detection in social networks,” *Computers and Electrical Engineering*, vol. 111, p. 108987, 2023.
- [124] K. Bok, C. Lee, and J. Yoo, “Recommending similar users using moving patterns in mobile social networks,” *Computers & Electrical Engineering*, vol. 77, pp. 47–60, 2019.

-
- [125] A. Kumar, A. Mallik, and S. Kumar, “Tlp-negcn: temporal link prediction via network embedding and graph convolutional networks,” *IEEE transactions on computational social systems*, vol. 11, no. 3, pp. 4454–4464, 2024.
- [126] S. Kumar, A. Mallik, and B. Panda, “Link prediction in complex networks using node centrality and light gradient boosting machine,” *World Wide Web*, vol. 25, no. 6, pp. 2487–2513, 2022.
- [127] S. Jayachitra Devi and B. Singh, “Link prediction model based on geodesic distance measure using various machine learning classification models,” *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 5, pp. 6663–6675, 2020.
- [128] W. Zhou, J. Gu, and Y. Jia, “h-index-based link prediction methods in citation network,” *Scientometrics*, vol. 117, no. 1, pp. 381–390, 2018.
- [129] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton, “Gemsec: Graph embedding with self clustering,” in *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, 2019, pp. 65–72.
- [130] A. C. Khosroshahi, S. T. Afshord, B. Zarei, and B. Arasteh, “Influence maximization in social networks using improved genetic algorithm,” *IEEE Access*, 2025.
- [131] K. Berahmand, E. Nasiri, M. Rostami, and S. Forouzandeh, “A modified deep-walk method for link prediction in attributed social network,” *Computing*, vol. 103, pp. 2227–2249, 2021.
- [132] H. Gul, F. Al-Obeidat, A. Amin, and F. Moreira, “Analyzing complex networks: Extracting key characteristics and measuring structural similarities,” *Expert Systems*, vol. 41, no. 1, p. e13470, 2024.

-
- [133] M. Liu and V. Jannesari, “Tlfls: link prediction in multilayer social networks using trustworthy lévy-flight semi-local random walk,” *Journal of Complex Networks*, vol. 12, no. 4, 2024.
- [134] Z. Wang, J. Liao, Q. Cao, H. Qi, and Z. Wang, “Friendbook: a semantic-based friend recommendation system for social networks,” *IEEE transactions on mobile computing*, vol. 14, no. 3, pp. 538–551, 2014.
- [135] L. Berkani, S. Belkacem, M. Ouafi, and A. Guessoum, “Recommendation of users in social networks: A semantic and social based classification approach,” *Expert Systems*, vol. 38, no. 2, p. e12634, 2021.
- [136] W. Zhang, “An analysis and prediction model based on complex network time series,” *Expert Systems*, vol. 40, no. 5, p. e13132, 2023.
- [137] Y. Qi and J. Luo, “Prediction of essential proteins based on local interaction density,” *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 13, no. 6, pp. 1170–1182, 2015.
- [138] R. A. Rossi and N. K. Ahmed, “The network data repository with interactive graph analytics and visualization,” in *AAAI*, 2015. [Online]. Available: <https://networkrepository.com>
- [139] R. Geisberger, P. Sanders, and D. Schultes, “Better approximation of betweenness centrality,” in *2008 Proceedings of the Tenth Workshop on Algorithm Engineering and Experiments (ALENEX)*. SIAM, 2008, pp. 90–100.
- [140] R. Rossi and N. Ahmed, “The network data repository with interactive graph analytics and visualization,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.

-
- [141] J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graph evolution: Densification and shrinking diameters,” *ACM transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, pp. 2–es, 2007.
- [142] P. Boldi, M. Rosa, M. Santini, and S. Vigna, “Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks,” in *Proceedings of the 20th international conference on World Wide Web*, 2011, pp. 587–596.
- [143] H. Cai, V. W. Zheng, and K. C.-C. Chang, “A comprehensive survey of graph embedding: Problems, techniques, and applications,” *IEEE transactions on knowledge and data engineering*, vol. 30, no. 9, pp. 1616–1637, 2018.
- [144] G. Jeh and J. Widom, “Simrank: a measure of structural-context similarity,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 538–543.
- [145] H. Hu, S. Zhang, and Y. Wang, “An interlayer link prediction method based on edge-weighted embedding,” *Complexity*, vol. 2023, no. 1, p. 3541437, 2023.
- [146] F. Ziya and S. Kumar, “Gsvaelp: integrating graphsage and variational autoencoder for link prediction,” *Multimedia Tools and Applications*, pp. 1–27, 2024.
- [147] L. Si, L. Li, H. Luo, and Z. Ma, “Link prediction in multiplex social networks: An information transmission approach,” *Chaos, Solitons & Fractals*, vol. 189, p. 115683, 2024.
- [148] M. Azzeh, A. Qusef, and O. Alabboushi, “Arabic fake news detection in social media context using word embeddings and pre-trained transformers,” *Arabian Journal for Science and Engineering*, vol. 50, no. 2, pp. 923–936, 2025.

-
- [149] S. Bajpai and A. Chaturvedi, "Toward detecting rumor initiator in online social networks using ontology-driven model," *Arabian Journal for Science and Engineering*, vol. 50, no. 2, pp. 661–677, 2025.
- [150] K. Hajarathaiah, M. K. Enduri, S. Anamalamudi, and A. R. Sangi, "Algorithms for finding influential people with mixed centrality in social networks," *Arabian Journal for Science and Engineering*, vol. 48, no. 8, pp. 10 417–10 428, 2023.
- [151] F. Ziya and S. Kumar, "Metalp-dgi: Meta-learning-based link prediction with centrality-aware deep graph infomax embeddings," *Concurrency and Computation: Practice and Experience*, vol. 37, no. 21-22, p. e70211, 2025.
- [152] Y.-L. Chen, C.-H. Hsiao, and C.-C. Wu, "An ensemble model for link prediction based on graph embedding," *Decision Support Systems*, vol. 157, p. 113753, 2022.
- [153] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.
- [154] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.
- [155] X. Liu, J. Chen, and Q. Wen, "A survey on graph classification and link prediction based on gnn," *arXiv preprint arXiv:2307.00865*, 2023.
- [156] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.
- [157] B. Pandey, P. K. Bhanodia, A. Khamparia, and D. K. Pandey, "A comprehensive survey of edge prediction in social networks: Techniques, parameters and challenges," *Expert Systems with Applications*, vol. 124, pp. 164–181, 2019.

-
- [158] Z. Wang, Y. Lei, and W. Li, “Neighborhood attention networks with adversarial learning for link prediction,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3653–3663, 2020.
- [159] Z. Guo, F. Wang, K. Yao, J. Liang, and Z. Wang, “Multi-scale variational graph autoencoder for link prediction,” in *Proceedings of the Fifteenth ACM international conference on web search and data mining*, 2022, pp. 334–342.
- [160] E. C. Mutlu and T. A. Oghaz, “Review on graph feature learning and feature extraction techniques for link prediction,” *arXiv preprint arXiv:1901.03425*, p. 38, 2019.
- [161] A. Kumari, R. K. Behera, B. Sahoo, and S. P. Sahoo, “Prediction of link evolution using community detection in social network,” *Computing*, vol. 104, no. 5, pp. 1077–1098, 2022.
- [162] M. Kumar, S. Mishra, and B. Biswas, “Pwaf: Path weight aggregation feature for link prediction in dynamic networks,” *Computer Communications*, vol. 191, pp. 438–458, 2022.
- [163] K. Yang, Y. Liu, Z. Zhao, X. Zhou, and P. Ding, “Graph attention network via node similarity for link prediction,” *The European Physical Journal B*, vol. 96, no. 3, p. 27, 2023.
- [164] D. Zhang, J. Yin, X. Zhu, and C. Zhang, “Attributed network embedding via subspace discovery,” *Data Mining and Knowledge Discovery*, vol. 33, no. 6, pp. 1953–1980, 2019.
- [165] Z. He, J. Zhu, S. Qian, J. Chai, and D. Koutra, “Linkgpt: Teaching large language models to predict missing links,” *arXiv preprint arXiv:2406.04640*, 2024.
- [166] L. Chen, Y. Zhou, Q. Wei, B. Sun, R. Zhao, C. Zhang, and X. Wang, “Elpgpt: Large language models enhancing link prediction in electrical knowl-

- edge graph,” *International Journal of High Speed Electronics and Systems*, vol. 34, no. 01, p. 2540115, 2025.
- [167] G. Li, Z. Pan, F. Cai, W. Chen, and L. Cheng, “A structure-feature dynamic decoupling gnn architecture for link prediction,” *Knowledge-Based Systems*, p. 113883, 2025.
- [168] E. Nasiri, K. Berahmand, and Y. Li, “Robust graph regularization nonnegative matrix factorization for link prediction in attributed networks,” *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 3745–3768, 2023.
- [169] H. Tan, Y. Liu, X. Liu, L. Hu, and Z. He, “Interpretable link prediction,” *Chaos, Solitons & Fractals*, vol. 191, p. 115928, 2025.
- [170] E. Nasiri, K. Berahmand, and Y. Li, “A new link prediction in multiplex networks using topologically biased random walks,” *Chaos, Solitons & Fractals*, vol. 151, p. 111230, 2021.
- [171] S. K. Gupta and D. P. Singh, “Seed community identification framework for community detection over social media,” *Arabian Journal for Science and Engineering*, vol. 48, no. 2, pp. 1829–1843, 2023.
- [172] S. S. Singh, S. Muhuri, and V. Srivastava, “Bt-lpd: B+ tree-inspired community-based link prediction in dynamic social networks,” *Arabian Journal for Science and Engineering*, vol. 49, no. 3, pp. 4039–4060, 2024.
- [173] S. J. Devi and B. Singh, “Link prediction model based on the topological feature learning for complex networks,” *Arabian Journal for Science and Engineering*, vol. 45, no. 12, pp. 10 051–10 065, 2020.
- [174] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.

- [175] S. Hasanzadeh, S. M. Fakhrahmad, and M. Taheri, "Based recommender systems: A proposed rating prediction scheme using word embedding representation of reviews," *The Computer Journal*, vol. 65, no. 2, pp. 345–354, 2022.
- [176] Q. Mi, X. Wang, and Y. Lin, "A double attention graph network for link prediction on temporal graph," *Applied Soft Computing*, vol. 136, p. 110059, 2023.
- [177] X. Ren, "Link prediction using extended neighborhood based local random walk in multilayer social networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 2, p. 101931, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S131915782400020X>
- [178] W. Almansoori, S. Gao, T. N. Jarada, A. M. Elsheikh, A. N. Murshed, J. Jida, R. Alhajj, and J. Rokne, "Link prediction and classification in social networks and its application in healthcare and systems biology," *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 1, no. 1, pp. 27–36, 2012.
- [179] D. Mumin, L.-L. Shi, and L. Liu, "An efficient algorithm for link prediction based on local information: considering the effect of node degree," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 7, p. e6289, 2022.
- [180] J. Wang, Y. Luo, Z. Du, P. Li, and X.-K. Xu, "Cwiiiif: A novel algorithm for identifying influential nodes in multilayer networks," *IEEE Transactions on Computational Social Systems*, 2025.
- [181] P. M. Chuan, L. H. Son, M. Ali, T. D. Khang, L. T. Huong, and N. Dey, "Link prediction in co-authorship networks based on hybrid content similarity metric," *Applied Intelligence*, vol. 48, no. 8, pp. 2470–2486, 2018.
- [182] G. Zare, N. J. Navimipour, M. Hosseinzadeh, and A. Sahafi, "Network link prediction via deep learning method: A comparative analysis with traditional meth-

- ods,” *Engineering Science and Technology, an International Journal*, vol. 56, p. 101782, 2024.
- [183] F. Aghabozorgi and M. R. Khayyambashi, “A new similarity measure for link prediction based on local structures in social networks,” *Physica A: Statistical Mechanics and its Applications*, vol. 501, pp. 12–23, 2018.
- [184] M. Liu and V. Jannesari, “Tlfs1: link prediction in multilayer social networks using trustworthy lévy-flight semi-local random walk,” *Journal of Complex Networks*, vol. 12, no. 4, p. cnae026, 2024.
- [185] X. Liu and X. Li, “A social network link prediction method based on stacked generalization,” *The Computer Journal*, vol. 65, no. 10, pp. 2693–2708, 2022.
- [186] Y. Yao, Y. He, Z. Huang, Z. Xu, F. Yang, J. Tang, and K. Gao, “Deep non-negative matrix factorization with edge generator for link prediction in complex networks,” *Applied Intelligence*, vol. 54, no. 1, pp. 592–613, 2024.
- [187] R. Yang, C. Yang, X. Peng, and A. Rezaeipannah, “A novel similarity measure of link prediction in multi-layer social networks based on reliable paths,” *Concurrency and Computation: Practice and Experience*, vol. 34, no. 10, p. e6829, 2022.
- [188] N. Sonti, R. Mulpuri Santhi Sri, and V. R. Pamulapati, “Social network based link correlation using graph neural network with deep learning architectures for feature vectors prediction and classification,” *Concurrency and Computation: Practice and Experience*, vol. 36, no. 14, p. e8090, 2024.
- [189] W. Yu, J. Fu, Y. Zhao, H. Shi, X. Chen, S. Shen, and X.-Z. Gao, “Link prediction in bipartite networks via deep autoencoder-like nonnegative matrix factorization,” *Applied Soft Computing*, vol. 169, p. 112616, 2025.

- [190] A. K. Menon and C. Elkan, “Link prediction via matrix factorization,” in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II* 22. Springer, 2011, pp. 437–452.
- [191] X. Ma, P. Sun, and Y. Wang, “Graph regularized nonnegative matrix factorization for temporal link prediction in dynamic networks,” *Physica A: Statistical mechanics and its applications*, vol. 496, pp. 121–136, 2018.
- [192] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A. J. Smola, “Distributed large-scale natural graph factorization,” in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 37–48.

Appendices

.1 List of Prime Research Papers Selected in this SLR

ID	Title	Year	Ref.
S1	A Survey of Link Prediction in Complex Networks	2016	[3]
S2	A comprehensive survey of edge prediction in social networks: Techniques, parameters and challenges	2019	[157]
S3	Applications of link prediction in social networks: A review	2020	[1]
S4	Neighborhood Attention Networks With Adversarial Learning for Link Prediction	2020	[158]
S5	Ensemble-model-based link prediction of complex networks	2020	[66]
S6	Integrating node centralities, similarity measures, and machine learning classifiers for link prediction	2022	[68]
S7	Link prediction in complex networks using node centrality and light gradient boosting machine	2021	[86]
S8	Influence maximization in social networks using transfer learning via graph-based LSTM	2022	[33]
S9	Multi-Scale Variational Graph AutoEncoder for Link Prediction	2022	[159]
S10	Graph attention network via node similarity for link prediction	2022	[10]
S11	Link Prediction on Complex Networks: An Experimental Survey	2022	[6]
S12	A modified DeepWalk method for link prediction in attributed social network	2021	[5]
S13	Review on Graph Feature Learning and Feature Extraction Techniques for Link Prediction	2019	[160]
S14	Prediction of link evolution using community detection in social network	2022	[161]

ID	Title	Year	Ref.
S15	node2vec: Scalable Feature Learning for Networks	2016	[100]
S16	Link prediction based on the mutual information with high-order clustering structure of nodes in complex networks	2023	[7]
S17	Link prediction approach combined graph neural network with capsule network .	2023	[73]
S18	An ensemble model for link prediction based on graph embedding	2022	[38]
S19	PWAF:Path Weight Aggregation Feature for link prediction in dynamic networks	2022	[162]
S20	Link Prediction Based On Local Structure And Node Information Along Local Paths	2022	[48]
S21	Link prediction in multilayer social networks using reliable local random walk and boosting ensemble classifier	2024	[35]
S22	GC-LSTM: graph convolution embedded LSTM for dynamic network link prediction	2022	[105]
S23	DICN: Similarity-based LP with direct/indirect neighbors. Considers latent relationships via second-order neighbors	2020	[67]
S24	Skip-gram edge embedding + LinkVec + LGBM	2022	[70]
S25	Lp-robin: link prediction in dynamic networks exploiting incremental node embedding	2022	[71]
S26	A novel link prediction algorithm based on inductive matrix completion	2022	[72]
S27	Meta-HGT: Metapath-aware HyperGraph Transformer for heterogeneous information network embedding.	2023	[39]
S28	An extended self-representation model of complex networks for link prediction	2024	[31]

ID	Title	Year	Ref.
S29	Link prediction in multiplex social networks: An information transmission approach	2024	[147]
S30	Algorithms for finding influential people with mixed centrality in social networks	2023	[150]
S31	Graph attention network via node similarity for link prediction.	2023	[163]
S32	Attributed network embedding via subspace discovery.	2019	[164]
S33	Linkgpt: Teaching large language models to predict missing links	2024	[165]
S34	ELPGPT: Large Language Models Enhancing Link Prediction in Electrical Knowledge Graph	2024	[166]
S35	A survey on graph classification and link prediction based on gnn.	2023	[155]
S36	A Structure-Feature Dynamic Decoupling GNN architecture for link prediction	2025	[167]
S37	Robust graph regularization nonnegative matrix factorization for link prediction in attributed networks	2023	[168]
S38	Interpretable link prediction	2025	[169]
S39	A new link prediction in multiplex networks using topologically biased random walks .	2021	[170]
S40	Seed community identification framework for community detection over social media	2021	[171]
S41	Bt-lpd: B+ tree-inspired community-based link prediction in dynamic social networks	2024	[172]
S42	Neighborhood attention networks with adversarial learning for link prediction	2020	[158]

ID	Title	Year	Ref.
S43	Link prediction based on the mutual information with high-order clustering structure of nodes in complex networks	2023	[7]
S44	Review on graph feature learning and feature extraction techniques for link prediction	2019	[160]
S45	Link prediction model based on the topological feature learning for complex networks	2019	[173]
S46	Semi-supervised classification with graph convolutional networks.	2016	[174]
S47	Based recommender systems: A proposed rating prediction scheme using word embedding representation of reviews.	2020	[175]
S48	Link prediction in complex networks using information-theoretic measures.	2020	[87]
S49	Link prediction based on common-neighbors for dynamic social network	2016	[88]
S50	A double attention graph network for link prediction on temporal graph.	2023	[176]
S51	Link prediction using extended neighborhood-based local random walk in multilayer social networks	2024	[177]
S52	Predicting missing links via local information	2009	[17]
S53	Line: Large-scale information network embedding	2015	[95]
S54	Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach	2013	[61]
S55	Link prediction in dynamic social networks by integrating different types of information	2015	[32]
S56	Link prediction in multi-relational networks based on relational similarity	2017	[64]

ID	Title	Year	Ref.
S57	E-LSTM-D: A deep learning framework for dynamic network link prediction	2019	[65]
S58	Knowledge graph large language model (KG-LLM) for link prediction	2024	[75]
S59	A novel similarity-based parameterized method for link prediction	2023	[98]
S60	Temporal link prediction based on node dynamics	2023	[99]
S61	Link prediction for attribute and structure learning based on attention mechanism	2025	[76]
S62	SEAL+: A subgraph-enhanced framework for link prediction with graph neural networks	2025	[42]
S63	Link prediction and classification in social networks and its application in healthcare and systems biology	2012	[178]
S64	Link prediction in multiplex social networks: An information transmission approach	2024	[34]
S65	Link prediction in multilayer social networks using reliable local random walk and boosting ensemble classifier	2020	[35]
S66	Bipartite graph link prediction method with homogeneous nodes similarity for music recommendation	2022	[79]
S67	An efficient algorithm for link prediction based on local information: considering the effect of node degree	2022	[179]
S68	Analyzing complex networks: Extracting key characteristics and measuring structural similarities	2021	[106]
S69	Analyzing complex networks: Extracting key characteristics and measuring structural similarities	2024	[107]

ID	Title	Year	Ref.
S70	CWIIIF: A Novel Algorithm for Identifying Influential Nodes in Multilayer Networks	2021	[180]
S71	Link prediction in co-authorship networks based on hybrid content similarity metric	2021	[181]
S72	Network link prediction via deep learning method: A comparative analysis with traditional methods	2024	[182]
S73	A new similarity measure for link prediction based on local structures in social networks	2018	[183]
S74	TLFSL: link prediction in multilayer social networks using trustworthy Levy-flight semi-local random walk	2024	[184]
S75	A Social Network Link Prediction Method Based on Stacked Generalization	2022	[185]
S76	Deep non-negative matrix factorization with edge generator for link prediction in complex networks	2024	[186]
S77	A novel similarity measure of link prediction in multi-layer social networks based on reliable paths	2022	[187]
S78	Social network based link correlation using graph neural network with deep learning architectures for feature vectors prediction and classification	2024	[188]
S79	Local core expanding-based label diffusion and local deep embedding for fast community detection algorithm in social networks	2024	[122]
S80	Link prediction in bipartite networks via deep autoencoder-like nonnegative matrix factorization	2025	[189]
S81	Link prediction via matrix factorization	2011	[190]

ID	Title	Year	Ref.
S82	Graph regularized nonnegative matrix factorization for temporal link prediction in dynamic networks	2018	[191]
S83	Temporal link prediction with motifs for social networks.	2021	[59]
S84	Distributed large-scale natural graph factorization	2013	[192]
S85	A multiple attention layer-shareable method for link prediction in multilayer networks	2025	[57]

PUBLICATIONS

Published/ Accepted

SCI-INDEXED JOURNALS

1. Fatima Ziya, Sanjay Kumar, "GSVAELP: integrating graphSAGE and variational autoencoder for link prediction.,"**Multimedia Tools and Applications** Volume 84 (31 August 2024),doi:<https://doi.org/10.1007/s11042-024-20123-z>.
2. Fatima Ziya, Sanjay Kumar,"MetaLP-DGI: Meta-Learning-Based Link Prediction With Centrality-Aware Deep Graph Infomax Embeddings,"**Concurrency and Computation, Wiley** 37(21-22)(31 July 2025),doi:<https://doi.org/10.1002/cpe.70211>.(Impact Factor 1.5).
3. Fatima Ziya, Sanjay Kumar, "Enhancing Link Prediction through Graph Attention Network and Linear Discriminant Analysis", **Expert System,Wiley** doi: <https://doi.org/10.1111/exsy.70114>. **(Impact Factor 2.3)**

Communicated/Under Review

SCI-INDEXED JOURNALS

1. Fatima Ziya, Sanjay Kumar “Unified Link Prediction: Leveraging Multi-Algorithm Embeddings and Attri2Vec”, **Journal of Chaos, Solitons and Fractals**, June,2025 (under revisions) (Minor Revisions Submitted)(**Impact Factor 5.6**)
2. Fatima Ziya, Sanjay Kumar "Integrating Structure and Similarity-Aware Graph Convolutional Network with Cross-Layer Representation learning in Multiplex

Networks”, **IEEE Transactions on Computational Social Systems**,(under review).

SCOPUS-INDEXED CONFERENCES

1. Fatima Ziya and Sanjay Kumar, "Link Prediction in Social Networks: A Hybrid Approach with Graph Embedding and Ensemble Learning," **8th International Conference on Innovative Computing and Communication (ICICC-2025)** Shaheed Sukhdev College of Business Studies, New Delhi, India, 2025.doi: https://doi.org/10.1007/978-981-96-6883-0_39. **(Published)**.
2. Fatima Ziya and Sanjay Kumar, "MetaLP-EC: A Meta-Learning Framework for Temporal Link Prediction with Evolving Centrality and GCN Embeddings," 6th International Conference on Data Analytics and Management (ICDAM-2025) Venue London Metropolitan University, United Kingdom June 13, 2025, https://doi.org/10.1007/978-3-032-03769-5_12. **(Published)**.



GSVAELP: integrating graphSAGE and variational autoencoder for link prediction

Fatima Ziya¹ · Sanjay Kumar¹ 

Received: 4 November 2023 / Revised: 10 July 2024 / Accepted: 18 August 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

Link prediction (LP) plays a crucial role in network science, which forecasts potential connections or relationships between nodes or entities within the network. Link prediction has found many applications, such as suggesting new connections to users in social networks, personalized recommendations in e-commerce, predicting new routes in transportation networks, and many others. This paper introduces a deep learning-based link prediction model in social networks by leveraging graphSAGE (graph sample and aggregation) and Variational Autoencoders (VAE). The proposed work starts by utilizing a graphSAGE to generate node embeddings of the input network by sampling and aggregating information from neighborhood nodes. The generated embeddings are sufficiently expressive to capture the local and global network structure. Further, we adopt VAE to learn a latent space representation of the graphSAGE embeddings. The VAE helps to refine the node embeddings and learn a meaningful latent space representation of the input data, which can be useful for the downstream link prediction task. The encoder's output (latent space) can capture important features that aid link prediction. Finally, we train a logistic regression classifier using the latent representations from the VAE as features to predict the upcoming links in the network. The necessary hyperparameter studies are performed to obtain the optimal values of the various model parameters. The experiments and simulations conducted on eight different real-world network datasets illustrate the effectiveness of the proposed link prediction model. Additionally, we evaluate the average performance of each comparative link prediction method across all datasets to assess the efficacy of the proposed approach.

Keywords Network embedding · GraphSAGE (Graph Sample and Aggregation) · Link prediction · Social networks · Variational autoencoders (VAE)

Sanjay Kumar
sanjay.kumar@dtu.ac.in

Fatima Ziya
Fatimazziya@gmail.com

¹ Department of Computer Science and Engineering, Delhi Technological University, Main Bawana Road, New Delhi 110042, Delhi, India

RESEARCH ARTICLE

MetaLP-DGI: Meta-Learning-Based Link Prediction With Centrality-Aware Deep Graph Infomax Embeddings

Fatima Ziya | Sanjay Kumar 

Computer Science and Engineering, Delhi Technological University, New Delhi, India

Correspondence: Sanjay Kumar (sanjay.kumar@dtu.ac.in)**Received:** 26 November 2024 | **Revised:** 8 May 2025 | **Accepted:** 10 July 2025**Funding:** This work was supported by the Science and Engineering Research Board (SERB) (Grant No. CRG/2023/007610).**Keywords:** connectivity matrix | deep graph infomax (DGI) | link prediction | meta-learning | network embedding | social networks

ABSTRACT

Link prediction is a fundamental task in social and complex network analysis, focused on forecasting the likelihood of unseen or future connections between nodes. Accurate link prediction can enhance understanding of network dynamics, reveal hidden structures, and improve recommendations in social and information networks. This paper proposes a novel Meta-Learning-Based Link Prediction model that utilizes a Centrality-Aware connectivity matrix and incorporates Deep Graph Infomax (DGI) embeddings with the CatBoost classifier. The connectivity matrix is constructed using node centrality measures like closeness centrality, degree centrality, and betweenness centrality by capturing the network's local and global structural properties. The DGI embedding algorithm efficiently learns the network's latent features, while the CatBoost classifier is employed to enhance prediction performance. To address the challenge of imbalanced datasets in social networks, we apply downsampling to create balanced training and testing datasets, ensuring robust model learning. Our framework demonstrates improved accuracy, scalability, and adaptability compared to traditional link prediction methods. Extensive experiments on real-world social network datasets show that the proposed model achieves superior performance in link prediction tasks, making it a promising approach for various network analysis applications.

1 | Introduction

Complex network analysis is a key area of modern science, offering valuable tools to understand the structure and behavior of interconnected systems. These networks, which encompass social, biological, technological, and informational domains, are characterized by intricate relationships and non-trivial topological features such as clustering, community structures, and scale-free behavior [1, 2]. Analyzing complex networks offers valuable insights into the underlying principles governing their formation and evolution, enabling the discovery of patterns, the prediction of behaviors, and the identification of key elements within the system. Among the various tasks in network analysis,

link prediction is a critical problem, as it seeks to anticipate future connections or infer missing ones based on existing data. This capability not only enhances our understanding of network growth and functionality but also supports a wide range of practical applications, from improving recommendation systems to uncovering hidden biological interactions [3]. Hence, link prediction is a fundamental task in network analysis, receiving substantial interest from researchers across various scientific domains. This task involves forecasting the likelihood of future or currently unobserved connections between nodes within a network? [4]. Traditional approaches to link prediction have been categorized into methodologies, such as similarity-based measures, probabilistic models, machine learning approaches, and

ORIGINAL ARTICLE

Enhancing Link Prediction Through Graph Attention Network and Linear Discriminant Analysis

Fatima Ziya | Sanjay Kumar 

Department of Computer Science and Engineering, Delhi Technological University, Delhi, India

Correspondence: Sanjay Kumar (sanjay.kumar@dtu.ac.in)**Received:** 25 October 2024 | **Revised:** 25 July 2025 | **Accepted:** 9 August 2025**Funding:** This work was supported by the Science and Engineering Research Board (SERB) (Grant No. CRG/2023/007610).**Keywords:** complex network | graph attention network (GAT) | linear discriminant analysis (LDA) | link prediction | network embedding | node centrality | social networks

ABSTRACT

Link prediction (LP) is a prominent research topic in network science and complex network analysis, focused on predicting future connections between unconnected node pairs by examining network topology and related characteristics. This paper introduces an advanced LP model combining the graph attention network (GAT) with Jaccard similarity and node centrality measures, such as local interaction density (LID) and hubs and authority centrality (HAC). Initially, a weight matrix incorporating structural and topological information is created using a feature matrix derived from similarity measures and node centrality. Then, node embeddings are generated using GAT, allowing the model to learn detailed representations that consider local and global contexts. GAT's multi-head attention mechanism enables the model to focus on various aspects of the node neighbourhood, capturing diverse structural information. A well-defined dataset is created from these embeddings, representing nodes at the endpoints of edges labelled as positive or negative. Linear discriminant analysis (LDA) is applied to this well-balanced and labelled dataset to perform classification tasks, leading to accurate link prediction. The model's performance is evaluated across eight different datasets, and the obtained results reveal the proposed model's superiority over several baseline and recently proposed LP models.

1 | Introduction

Link prediction (LP) plays a vital role in complex network analysis, which aims to predict future links based on the existing structure and characteristics of the network (Gul et al. 2024). LP has evolved into an extensively researched topic within the complex network science community. Its prominence stems from its significant theoretical importance in understanding the mechanisms of complex network evolution and its broad applicability across various scientific disciplines (Liu and Jannesari 2024). Hence, LP enables the identification of connections, offering insights into the network's growth patterns. LP has wide-ranging applications in social networks, biological networks, and recommendation systems (Daud et al. 2020). Some of the

real-life applications of LP are recommending new friends on social media (Wang et al. 2014), finding spreader nodes, predicting potential collaborations in scientific research, forecasting disease outbreaks, suggesting new links in transportation systems, offering product recommendations (Berkani et al. 2021) in e-commerce, recommending movies or music and many others.

A wide range of LP methods has been discussed in the literature, focusing on forecasting the probability of future links between nodes in a network (Zhang 2023). LP techniques are divided into various categories, such as the likelihood-based approach, which evaluates the similarity between two disconnected nodes and assumes a higher likelihood of a link between more similar nodes. Formally, likelihood-based LP methods propose that the chance

Abbreviations: ANA, anti-nuclear antibodies; APC, antigen-presenting cells; IRF, interferon regulatory factor.

© 2025 John Wiley & Sons Ltd.

ICICC-2025



INTERNATIONAL CONFERENCE ON INNOVATIVE COMPUTING AND COMMUNICATION (ICICC-2025)

Certificate

This is to certify that **Prof. / Dr. / Mr. / Ms. Fatima Ziya** is a presenter/co-author of the paper titled **Link Prediction in Social Networks: A Hybrid Approach with Graph Embedding and Ensemble Learning** in the **8th International Conference on Innovative Computing and Communication (ICICC-2025)**, organized by Shaheed Sukhdev College of Business Studies, University of Delhi, New Delhi, India in association with the National Institute of Technology Patna, India and the University of Valladolid, Spain on **14th-15th February 2025**.

Poonam Verma
Principal, SSCBS,
University of Delhi,
New Delhi

Prabhat Kumar
General Chair
National Institute of Technology,
Patna

A K Singh
Technical Program Chair
National Institute of Technology,
Kurukshetra

ICDAM-2025



LONDON
METROPOLITAN
UNIVERSITY



PORTALEGRE
POLYTECHNIC
UNIVERSITY
POLITECNICO
DE PORTALEGRE



Springer

INTERNATIONAL CONFERENCE ON DATA ANALYTICS & MANAGEMENT (ICDAM - 2025)

Certificate

This is to certify that **Prof. /Dr. / Mr. / Ms. Fatima Ziya** is a presenter/co-author of the paper titled **MetaLP-EC: A Meta-Learning Framework for Temporal Link Prediction with Evolving Centrality and GCN Embeddings** presented at the **6th International Conference on Data Analytics and Management (ICDAM-2025)**, organized jointly by London Metropolitan University, London, United Kingdom in association with the WSG University, Bydgoszcz, Poland; Portalegre Polytechnic University, Portugal & SGGW Management Institute, Poland, Europe during **13th – 15th June 2025**.

Prof. (Dr.) Bal Virdee
London Metropolitan University, UK
Convener

Prof. (Dr.) Abhishek Swaroop
Bhagwan Parshuram Institute of Technology, India
Publication Chair

7% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 10 words)

Exclusions

- 1 Excluded Source
- 138 Excluded Matches

Match Groups

-  **300 Not Cited or Quoted 7%**
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 3%  Internet sources
- 7%  Publications
- 2%  Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

-  **Replaced Characters**
44 suspect characters on 20 pages
Letters are swapped with similar characters from another alphabet.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Author Biography



Fatima Ziya

Research Scholar,
Department of Computer Science & Engineering
Delhi Technological University, Delhi, India
Email: Fatimazziya@gmail.com

Fatima Ziya received her B.Tech. degree in information technology in 2016 from Jamia Hamdard University, India, Delhi. She completed her Master of Technology from the Ambedkar Institute of Advanced Telecommunication and Research (now known as NSUT East Campus), India, Delhi, in 2019. She is pursuing her PhD from Delhi Technological University, New Delhi, India. She has 5+ years of teaching experience. Her research interests include machine learning, deep learning, and artificial intelligence.