ENHANCED LINK PREDICTION USING MACHINE LEARNING AND DEEP LEARNING TECHNIQUES

A Thesis Submitted
In Partial Fulfillment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

in

Artificial Intelligence

by

RAHUL JAGGI
(Roll No. 2K23/AFI/31)

Under the Supervision of
Dr. SANJAY KUMAR
(Asst Prof, Dept of Computer Science & Engineering)

To the
Department of Computer Science and Engineering

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042. India

May, 2025

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE'S DECLARATION

I, Rahul Jaggi, Roll No. 2K23/AFI/31 student of M.Tech (Artificial Intelligence), hereby certify that the work which is being presented in the thesis entitled "**Enhanced Link Prediction using Machine Learning and Deep Learning Techniques**" in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Artificial Intelligence 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 August 2023 to May 2025 under the supervision of Dr. Sanjay Kumar, Asst Prof, Dept of Computer Science and Engineering. 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.

Place: Delhi **Candidate's Signature**

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 our knowledge.

Signature of Supervisor **Signature of External Examiner**

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE

Certified that **Rahul Jaggi** (Roll No. 2K23/AFI/31) has carried out the research work presented in the thesis titled "**Enhanced Link Prediction using Machine Learning and Deep Learning Techniques**", for the award of Degree of Master of Technology from Department of Computer Science and Engineering, Delhi Technological University, Delhi under my supervision. The thesis embodies result of original work and the contents of the thesis do not form the basis for the award of any other degree for the candidate or submit else from the any other University /Institution.

Dr. Sanjay Kumar

(Supervisor)

Date

Enhanced Link Prediction using Machine Learning and Deep Learning Techniques

Rahul Jaggi

ABSTRACT

The ultimate aim of link prediction is to identify the possible potential connections in a network. The study on this topic has gained impetus as it results in efficiently saving resources, time, cost and effort to analyze future possibilities in a network. With the refined use of this technique, it significantly improves how the complex networks like social network analysis, biological networks, and recommendation systems are interpreted vis-à-vis experimental processes. In this paper, we propose a novel combination of five node centralities and four similarity measures with the aim of capturing both local and global features of networks. Consequently, feature vector made by integration of these five node centralities and four similarity indices are then passed through Machine Learning(ML) classifiers. By combination of results of different classifiers according to dynamic weighting scheme, the integrated classifier is then utilized for final link prediction. We have also analyzed the effect of varying thresholds on the ROC AUC and F1 scores and the same have been tabulated. This paper provides insights into the effectiveness of combining graph-theoretic features with ML models for accurate link prediction.

The understanding of time dependent dynamics in evolving network interactions is crucial for applications ranging across various domains. In this paper, we introduce TA-GC-LSTM (Temporal Adaptive Graph Convolutional Long Short-Term Memory) which uses deep learning framework with novel combination of models. This proposed model of ours, efficiently captures spatial dependencies through graph convolution, temporal sequences using LSTM, and gives selective importance to influential time steps through the attention mechanism. In contravention to traditional methods, which rely on static graph representations, TA-GC-LSTM dynamically learns the temporal evolution of node relationships, enhancing predictive accuracy in link prediction tasks. In our framework, we have carried out processing of datasets by binning interactions into fixed time windows, encoding unique nodes with learnable embeddings, and filtering sparse time steps to optimize computational efficiency. To validate our approach, we have tested the model on three real-world datasets and compared our model performance against Graph Convolution Embedded LSTM (GC-LSTM) and Temporal Graph Convolutional Network (T-GCN) as benchmarks across multiple evaluation metrics. Our results demonstrated that TA-GC-LSTM outperforms baseline models, achieving an AUC score of 93%, while maintaining computational efficiency, making it a robust solution for modelling evolving graph structures.

ACKNOWLEDGEMENTS

I have taken efforts in this implementation paper. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Dr. Sanjay Kumar** for his guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing this review paper. I would like to express my gratitude towards the **Head of the Department (Computer Science and Engineering, Delhi Technological University)** for their kind cooperation and encouragement which helped me in the completion of this research survey. I would like to express my special gratitude and thanks to all the Computer Science and Engineering staff for giving me such attention and time.

My thanks and appreciation also go to my colleague in writing the survey paper and the people who have willingly helped me out with their abilities.

Rahul Jaggi

CONTENTS

List of Tables

List of Figures

List of Abbreviations

ML	Machine Learning
DL	Deep Learning
RF	Random Forest
XGB	Extreme Gradient Boosting
LDA	Linear Discriminant Analysis
LR	Linear Regression
AUC	Area Under Curve
GCN	Graph Convolution Network
GNN	Graph Neural Network
LSTM	Long Short Term Memory
TA-GC-LSTM	Temporal Adaptive Graph Convolutional Long Short Term Memory
CN	Common Neighbour
JC	Jaccard Coefficient
AA	Adamic-Adar
SVM	Support Vector Machine
RNN	Recurrent Neural Network
TLP-NEGCN	Temporal Link Prediction via Network Embedding and Graph Convolutional Networks

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Link prediction and analysis is one of the fundamental task in complex network analysis that involves predicting future or missing links in a graph based network. Since its inception, the topic has remained in focus and continues gaining community wide attention. It is one of the popular research topics and has a wide application covering a large number of domains. The increasing complexity of networks, such as social networks, citation networks, and biological networks, has motivated research in this area due to its broad applicability and potential to improve decision making processes. Link prediction has been effectively utilized in various domains, such as social network growth prediction, protein-protein interaction analysis, and recommender systems, where identifying future relationships can significantly enhance performance and reduce costs.

Furthermore, in the real world networks, relationships evolve over time which implies that a temporal dimension gets associated with it. Therefore, this necessitates a separate analytical approach compared to static networks. So to fulfil this task, the concept of dynamic link prediction is utilized. It involves forecasting future connections that might get developed or the present links which might be missing. Thus, dynamic network link prediction is vital topic crucial for understanding modern complex networks.

Traditionally, only topological structures of the network were considered for link prediction, such as similarity indices and matrix factorization which primarily relied on static networks. This limited their ability to model real world dynamism efficiently and effectively [1], [2]. As a result, traditional approaches were not able to capture complex patterns in dynamic networks resulting in limited accuracy and applicability. Thus, dynamic network link prediction is vital topic crucial for understanding modern complex networks.

In our thesis, we have implemented two separate novel models based on ML and DL approaches respectively. In the ML domain, we have demonstrated implementation of an integrated approach that combines three ML models to enhance link prediction accuracy. Specifically, we have employed RF, XGB, and LDA as base classifiers and integrated their prediction probabilities by using dynamically determined optimal weights through LR. This weighted stacking framework that is generated, dynamically ensures that the integrated model effectively balances performance across multiple evaluation metrics, including AUC and F1-score [9].

In contrast, deep learning models, particularly GNNs, have shown the ability to learn intricate node relationships by capturing higher order dependencies. Building on this foundation, our research introduced a novel approach called TA-GC-LSTM model, which integrates GCNs, Attention mechanism, and LSTM networks to improve

link prediction in dynamic networks. Our model captures spatial dependencies using GCNs, models temporal sequences with LSTM, and selectively focuses on influential historical time steps through a Temporal Attention Module. GC-LSTM architecture with an adaptive attention mechanism, enhances prediction accuracy, and computational efficiency.

1.2 MOTIVATION

India, as one of the world's most geopolitically significant nations, has long been a target of terrorist activities orchestrated primarily by cross-border groups. From major attacks in metropolitan areas to insurgencies in border regions, the evolving nature of terrorism in India has made early detection and prevention more challenging than ever before. Terrorist networks operating in and around the nation, often function through covert, decentralized, and dynamic modes of communication, making their detection using conventional surveillance techniques increasingly difficult.

In this context, the proactive prediction of hidden or probable links that can be established in future within suspected networks becomes a strategic advantage for national security agencies. Link prediction, when applied to communication graphs, financial transactions, social networks, or other intelligence databases, can assist in identifying unobserved associations or planning stages of coordinated attacks. For instance, uncovering a potential connection between a known militant and a previously unmonitored individual can allow for timely investigation and preemptive measures.

These networks are engineered to evade detection—connections are often indirect, communication is minimal, and participants deliberately avoid centrality. Consequently, there is a pressing need for intelligent models capable of learning from both structured patterns and temporal dynamics, even under conditions of incomplete or noisy data.

It is the vast and varied data landscape, ranging from call records and financial flows to social media activity, offers immense potential for constructing actionable network graphs. However, the real value lies in building models that can sift through this data to highlight emerging threats before they manifest. The integration of enhanced link prediction models into India's national security infrastructure could help intelligence agencies prioritize leads, allocate resources, and neutralize threats proactively.

1.3 OBJECTIVES

a) To carry out Literature Survey on evolution of link prediction.

b) To study and use diverse real world datasets for model validation and prediction.

c) To implement ML based static link prediction model for enhanced link prediction using integrated classifier.

d) To implement DL based dynamic link prediction model for enhanced link prediction using TA-GC-LSTM models.

1.4 PROBLEM FORMULATION

Two formulations of the link prediction task have been elaborated below: -

(a) **Static Link Prediction**. Static link prediction treats the network as a snapshot frozen in time, focusing on identifying missing connections within the existing network structure. The problem formulation involves removing a random set of links from the network and then aiming to predict these missing connections based solely on the remaining topological structure. The proposed integrated ML classifier will target this problem.

Let us assume a simple graph network depicted mathematically as G(V,E), where 'V' denotes the vertices set and 'E' denotes the edges in the graph. Mathematically, the universal set 'U' should contain a total of V(V−1)/2 edges. Therefore, the difference i.e. |U|−|E| provide us with a set of links that are not present at this juncture but some of these links may appear in the near future. The aim of link prediction is to find these edges in a static framework.

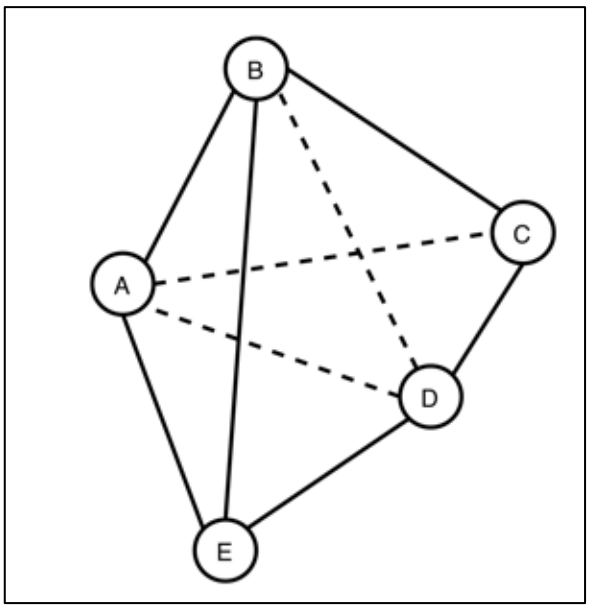

Figure 1.1 : Find Missing Links (AC, AD and BD)

(b) **Dynamic Link Prediction**. Dynamic link prediction incorporates temporal evolution, predicting future connections based on historical network states and temporal patterns. The problem formulation differs in the context of time.

Given a graph G[t, t'] containing edges up to time t measured within a fixed time [t, t'], output a ranked list 'L' of potential links (not present in G[t, t']) that are predicted to appear in the future time window G[t+1, t+1'].

The adjacency matrix A_t for each time step is defined as:

$$A_t(i,j) = \begin{cases} 1, & if (i,j) \in E_t \\ 0, & otherwise \end{cases}$$

The proposed TA-GC-LSTM model based on DL framework will target this problem.

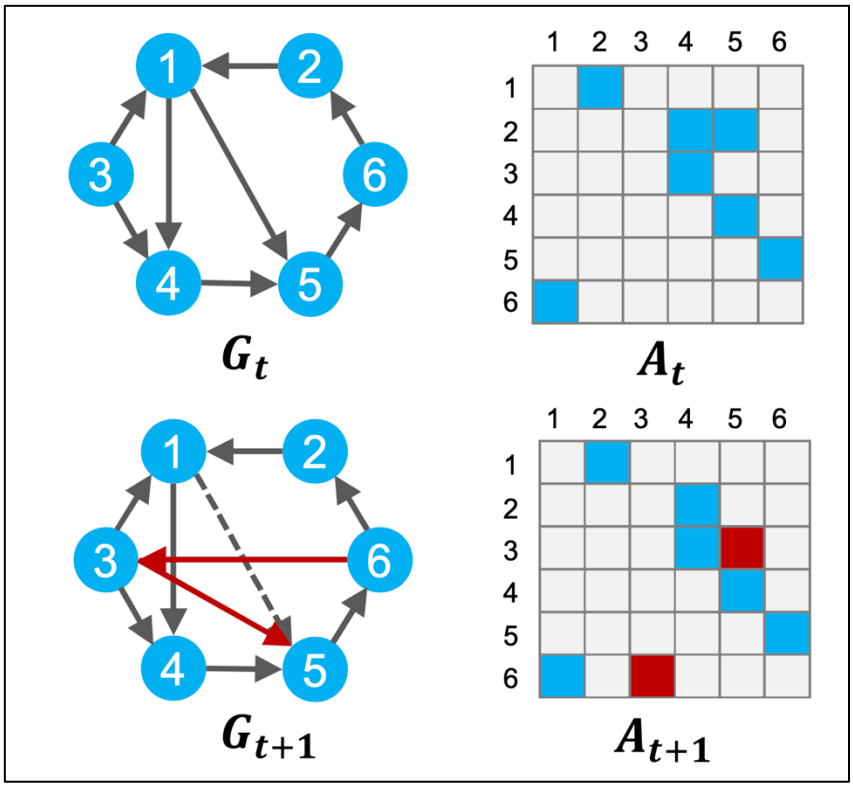

Figure 1.2 : Illustration of Adjacency Matrices at different time steps

1.5 THESIS STRUCTURE

The thesis is structured into five distinct chapters as follows:

- Chapter 1 introduces the topic, outlines the motivation behind the research, sets forth the objectives of the study and defines the problem statement formally.

- Chapter 2 offers a comprehensive review of the existing literature on methods for link prediction and their various forms. It also highlights the research gaps which were found and endeavoured to be addressed with the proposed models.

- Chapter 3 is dedicated to the proposed methodology and its results for ML based Integrated classifier model for enhance link prediction.

- Chapter 4 is dedicated to the proposed methodology and its results for DL based TA-GC-LSTM model for dynamic link prediction.

- Chapter 5 concludes the study with discussion on potential future research and probable additions.

CHAPTER 2

RELATED WORK

Link prediction has emerged as a critical research area within the broader field of network analysis, with applications spanning from social media recommendations and biological interactions to cybersecurity and counter-terrorism. Over the past decade, researchers have developed a variety of methods to address the problem of predicting missing or future links in both static and dynamic graphs.

2.1 TRADITIONAL METHODS

Link prediction in various networks has been on centre stage due to its applications in various domains, including social networks, terrorist networks, field of biology, product recommendation systems, etc. The primary goal here is to infer potential future or missing connections in a network depending on its existing structure. Since the very start, numerous approaches have been proposed, which can be generally classified into similarity based methods, probability based models, embedding techniques, and ML based approaches.

The earliest link prediction methods were based on simple topological similarity indices that measure the chances of a link formation between two nodes. Local measures such as CN, JC and AA are commonly used due to their computational efficiency. CN counts the shared neighbours between two nodes, while JC normalizes this count by the size of the union of their neighbourhoods. AA further refines Common Neighbours by weighting rare neighbours more heavily [3].

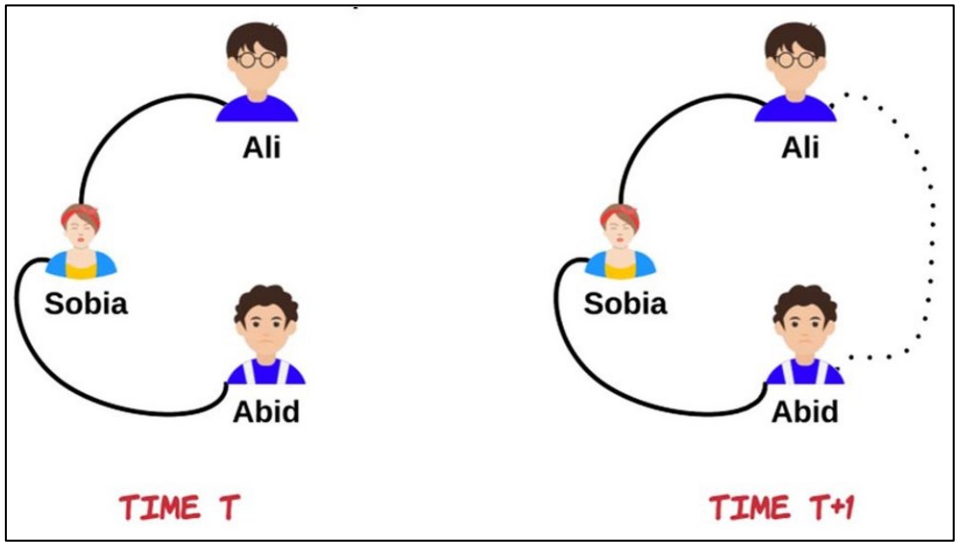

Figure 2.1: Link Prediction Example

Although the above mentioned methods are simple and computationally inexpensive, they primarily rely on local information.

Global similarity measures, on the flip side, overcome this limitation. Katz Index provides a score based on the path count between two nodes with more weight to shorter paths. However, the main limitation is that it incurs a high computational cost. Rooted Random Walk calculates the probability of reaching one node from another through a random walk. In essence, global measures provide better results but at the cost of high computational load.

2.2 ML BASED METHODS FOR STATIC LINK PREDICTION

Rahman et al. [5] considered the activities of the users and the mutual neighbours to bring out the local and global link prediction algorithm to calculate the similarity indices. ML approaches to link prediction involve extracting graph features and using them to train classifiers.

Early approaches focused on manually engineered features, such as node degree, clustering coefficient, and shortest path length. These features were used with classifiers like LR and SVMs. However, the reliance on handcrafted features limited the flexibility and scalability of these models [6]. A community relationship strength index (CRS) to find out the closeness between groups based on the similarity of nodes and topological information was proposed by Li et al. [7]. Ghorbanzadeh et al. [4] proposed a hybrid approach wherein, even if there are no common neighbours among nodes, they still have a chance of probable connection in the future [9]. In Table I, summarized information for recent advancements in the researches on this topic have been provided along with characteristics and limitations.

Table I. Recent Important Related Works(ML based Link Prediction)

Ser No.	Author(s) and Year	Method(s)	Characteristics	Limitations
(a)	Devi et al., 2020 [39]	Link prediction model based on measurement of geodesic distance	The model primarily relies on geodesic distance (shortest path between any two nodes) as a key predictor. Assumes that nodes with shorter geodesic distances have a higher probability of forming links.	Overreliance on Geodesic Distance: Focusing primarily on geodesic distance may overlook other significant topological features, such as common neighbours, clustering coefficients, or node centrality measures.
(b)	Berahmand et al. ,2021 [40]	CSADW based LP (Combination of Structural and Attributed deep Walk)	It is based on the notion that the probability of a connection between two nodes is more if they have more structure and attribute similarity.	The method relies on the idea that the probability of a connection between nodes is more if they are nearby in the network or share similar attributes, which may not always hold in complex networks

Ser No.	Author(s) and Year	Method(s)	Characteristics	Limitations
(c)	Kumar et al., 2022 [41]	NC-LGBM based LP (Node Centrality with Light Gradient Boosting Machine)	Proposed a combination based on using various node centralities. The feature vector generated is then passed through LGBM classifier for accurate link prediction.	Focusing primarily on node centrality metrics may overlook other significant topological features and the broader structural context of the network, potentially limiting predictive performance.
(d)	Chen et al. ,2022 [42]	EMLP (Ensemble model for link prediction based on graph embedding)	Combines multiple graph embedding techniques (e.g., node2vec, DeepWalk, GCN, GAT, etc.).Leverages diverse representations to capture different structural and relational properties.	The ensemble approach introduces additional hyperparameters, necessitating careful tuning to achieve optimal performance, which can be time-consuming and computationally intensive.

Modern ML techniques like RF and XGB, can automatically learn complex patterns from a diverse set of features. A link prediction method named LPXGB using the XGB ML technique employed the use of varied features to represent the dataset for binary classification problems [9]. These models are well suited for link prediction tasks, achieving high accuracy by combining multiple decision trees or boosting weak learners.

2.3 DL BASED METHODS FOR DYNAMIC LINK PREDICTION

Dynamism in link prediction has been a topic of pivotal focus in graph-based machine learning, with varied applications spanning across domains like social networks, biological systems, and communication infrastructures. The whole idea is to carry out analysis of a dynamic graph network and predict missing and future links.

While early methods primarily laid emphasis on static networks, researchers progressively recognized the necessity of modelling both spatial and temporal dependencies in order to adapt to dynamic networks.

In this sub-section, we have reviewed key advancements in three primary areas namely, static link prediction, dynamic network modelling, and spatio-temporal graph neural networks.

With the advent of deep learning, embedding-based methods became prominent, enabling learning of low-dimensional representations of nodes that captured structural properties. Techniques such as DeepWalk [10], Node2Vec [11], and LINE [12] used random walks and neural language models to generate node embeddings. More recently, Integrated GraphSAGE and variational autoencoder (GSVAELP) developed a deep learning based model for link prediction by using graphSAGE (graph sample and aggregation) and Variational Autoencoders (VAE) [13]. However, these approaches were inherently static and lacked the ability to capture evolving relationships in dynamic networks. The fixed representations of the above mentioned techniques, however, limited their application in real world scenarios where network structures are dynamic.

Dynamic networks have continuous and frequent changes, wherein nodes and edges keep appearing and disappearing as time elapses. Older static approaches proved insufficient for capturing such evolving structures, which led to the exploration of dynamic network modelling methods. These techniques can be categorised into two prominent categories, namely, temporal graph embedding techniques and RNN based models.

2.3.1 Temporal Graph Embedding Methods

Temporal graph embedding methods maintain temporal consistency along with preservation of topological information. Dynamic Triad [14] use static embeddings by enforcing temporal overlap, whereas Continuous-Time Dynamic Network Embeddings (CTDNE) [15] use time dependent random walks to learn temporal node representations. Although these methods gained impetus, but could not explicitly capture interdependencies across different time steps. Due to this, it limited their overall effectiveness in long-term link prediction tasks.

2.3.2 Recurrent Neural Network (RNN)Based Models

Spatio-Temporal Graph Neural Networks (STGNNs) enhanced the modelling of temporal aspect by integrating RNNs with GNNs. The T-GCN [16] combined GCNs with GRUs for traffic prediction, which enabled them for spatial and temporal pattern recognition tasks. Similarly, to improve temporal sequence learning the Graph Convolutional Recurrent Network (GCRN) [17] employed LSTM cells with GCNs to improve upon the time dependent learning. These models improved the overall experience in link prediction tasks, but the fixed time window sizes and the absence of attention mechanisms limited their scope of application in complex and evolving networks.

2.3.3 Spatio-Temporal GNNs

STGNNs address both the requirements in dynamic networks. These models combined graph convolutions for spatial learning with RNN based architectures or temporal attention mechanisms to model dynamic patterns effectively.

a) GC-LSTM:

GC-LSTM [18] integrated GCNs with LSTM cells, which resulted in improved memory retention for long term dependencies. The inherent nature of LSTM's sequential learning capabilities empowered GC-LSTM to provide superior performance in capturing complex temporal aspects. However, it's lack of a temporal attention mechanism restricted its ability to selectively choose important historical time steps, leading to susceptibility to dilution of information over long sequences.

b) Temporal Attention Mechanisms:

A significant improvement with the incorporation of temporal attention mechanisms allowed models to assign weights on the fly to historical time steps based on their relevance and contribution to the current state. Temporal Attention Memory (TAM) networks [19] and Spatio-Temporal Graph Attention Networks (ST-GAT) [20] exhibited superior performance by selectively emphasizing on influential past interactions. Although many advantages got accrued with this concept, due to their predominant focus on time steps and no explicit focus and integration on spatial relationships, it lead to suboptimal results in link prediction tasks.

c) Graph Convolutional Temporal Attention Mechanisms:

The combination of the above two architectures, led to the exploration of Graph Convolution with Temporal Attention mechanisms to enhance the task of link prediction. For event base networks, Temporal Graph Attention Networks (T-GAT) [21] used temporal attention along with Graph Attention Networks (GATs), whereas Adaptive Spatio-Temporal Graph Convolution Networks (ASTGCN) [22] incorporated both spatial and temporal attention for traffic forecasting. But these models although effective in their respective domains, were primarily designed for not exactly meant for link prediction tasks.

More recently, TLP-NEGCN transformed matrix into lower dimensional vector representations for the nodes of the network initially with the use of graph embedding with self-clustering (GEMSEC). These embeddings were then fed into GCNs across timestamps in the dataset [23].

2.3.4 Cross Domain Temporal Deep Learning

Recent works in various domains have demonstrated the versatility and efficiency of combining sequential models like LSTM, Bi-LSTM, and attention-based architectures for time-sensitive predictions. These architectures, though applied in

non-graph settings, validate the effectiveness of temporal learning strategies that closely align with our motivations in dynamic link prediction.

Ghosh et al. [24] introduced a model combining GloVe embeddings with LSTM for emotion detection. Their work emphasized how selective integration of shallow and deep models can enhance learning efficiency without compromising on semantic representation. Similarly, in the domain of sentiment analysis, Yadav and Pal [25] applied Bi-RNN and Bi-LSTM frameworks to classify Amazon reviews. Their findings highlighted the importance of capturing bidirectional temporal dependencies, further justifying our inclusion of LSTM modules in TA-GC-LSTM.

In event processing, F-DES (Fast and Deep Event Summarization) demonstrated how temporal attention and deep summarization could capture evolving storylines in textual event streams, showing the benefits of temporal compression and contextual focus [26]. This principle finds a direct echo in our temporal attention module, which selectively amplifies important time steps in evolving graphs.

Furthermore, summarization focused architectures like those studied by Sharma and Kaushal [27] in video summarization using deep learning leveraged sequential modelling to distil relevant content over time, an idea we borrow when filtering graph time steps through attention mechanisms.

Complementing this, a cloud based deep learning interface for text query-based event summarization and retrieval was presented in [28], demonstrating scalable implementations of attention-driven systems for large, evolving datasets reinforcing the importance of both efficiency and interpretability in time-aware models.

From a computational efficiency standpoint, Mehta and Roy [29] proposed an optimized ANN architecture tailored for small-scale problems, emphasizing minimal resource utilization. Inspired by such design goals, our use of Chebyshev polynomials in the GCN layer ensures scalability by avoiding expensive spectral decompositions. In Table II, summarized information for recent advancements in the researches on this topic have been provided along with characteristics and limitations.

Table II. Recent Important Related Works(DL based Link Prediction)

Ser No.	Author(s) and Year	Method(s)	Characteristics	Limitations
(a)	Yu et al., 2020 [16]	T-GCN	Integrates GCN with GRU for temporal modelling in traffic networks	GRUs struggle with long-term dependencies and is not tailored for link prediction
(b)	Rossi et al., 2020 [43]	Temporal Graph Network (TGN)	Uses memory modules and message passing for continuous-time graphs.	High computational complexity and is harder to scale
(c)	Chen et al., 2022 [18]	GC-LSTM	Embeds GCN into LSTM cells for dynamic network learning	Equal weight to all time steps but lacks temporal attention
(d)	Kumar et al., 2024 [23]	TLP-NEGCN	Uses self-clustering embeddings with GCN over time steps for link prediction	Relies on fixed-time binning, however, no attention or LSTM modelling

Lastly, recent innovations in human activity recognition such as those by Verma et al. [30] and Bansal et al. [31] used residual networks and fine tuning strategies to capture subtle temporal variations in sensor data. These models show how residual learning and attention can boost generalization and performance, especially when long-term dependencies are involved much like in dynamic graphs.

In a more recent advancement, Verma and Agarwal [32] introduced a hybrid deep learning model for road accident classification that integrated Capsule Recurrent Neural Networks (Capsule-RNNs) with an Improved Reptile Search Algorithm for optimization. Their architecture captured spatial hierarchies using capsule networks while maintaining sequential dependencies through RNNs demonstrating that spatial-

sequential fusion, when guided by effective attention or optimization strategies, can lead to substantial gains in predictive accuracy.

These studies, though addressing varied domains, provide strong foundational evidence supporting the core components of our proposed model especially the integration of GCNs for spatial learning, LSTM for sequence modelling, and temporal attention mechanisms for selective memory. The broader success of these components in other time sensitive tasks strengthens our motivation for adapting them in the context of dynamic link prediction.

2.4 RESEARCH GAPS

2.4.1 Overview

The advancement in the domain of link prediction in complex networks has evolved significantly over the past decade, yet several critical gaps persist that limit the effectiveness and applicability of existing approaches. Through comprehensive analysis of current methodologies, this research identifies and addresses key limitations across both static and dynamic link prediction paradigms.

2.4.2 Static Link Prediction

(a) Feature Fragmentation and Limited Integration

Traditional static link prediction methods suffer from a fundamental fragmentation problem where approaches focus on either local topological features (CN, AA, JC) or global structural properties (Katz Index, Random Walk measures) but rarely integrate both effectively. Local measures, while computationally efficient, fail to capture broader network topology and often yield suboptimal accuracy. Conversely, global measures provide better performance but incur prohibitive computational costs for large-scale networks.

(b) Static Weighting Limitations

ML approaches to link prediction, including ensemble methods using Random Forest, XGBoost, and other classifiers, typically employ static weighting schemes that cannot dynamically adapt to network specific patterns. The lack of adaptive weighting mechanisms results in suboptimal performance across heterogeneous network domains.

(c) Limited Scalability and Generalizability

Existing feature engineering approaches rely heavily on handcrafted similarity indices that lack adaptability across diverse network types. This limitation becomes particularly pronounced when dealing with networks of varying scales and structural properties, where fixed feature sets fail to capture domain-specific relationship patterns effectively.

2.4.3 Dynamic Link Prediction

(a) Temporal Oversimplification and Fixed Time Windows

Current dynamic link prediction methods suffer from temporal oversimplification, where models like GC-LSTM and T-GCN use fixed time windows or uniform attention across historical states. This approach fails to account for the varying importance of different time periods in network evolution, leading to information dilution over long sequences. The inability to adaptively focus on evolutionarily significant network states severely limits predictive accuracy in temporal networks with irregular interaction patterns.

(b) Long-Term Dependency Limitations

Recurrent architectures commonly used in dynamic link prediction, particularly GRU-based models like T-GCN, have a major limitation of vanishing gradients. This limitation becomes particularly problematic for networks with extended temporal sequences where early interactions significantly influence future link formation patterns.

(c) Computational Inefficiency in Spectral Methods

While spectral graph convolutions (such as ChebConv) provide effective spatial feature learning, they incur high computational costs for large dynamic graphs. The lack of efficient approximation methods for spectral operations limits the scalability of sophisticated graph neural network architectures to real-world large-scale temporal networks.

2.4.4 Comprehensive Gap Analysis

The identified research gaps reveal a clear need for:

(a) **Integrated Feature Learning**: Methods that systematically combine multi-scale graph-theoretic features with adaptive ensemble techniques.

(b) **Advanced Temporal Modelling**: Frameworks that incorporate sophisticated attention mechanisms for selective historical context processing

(c) **Unified Spatio-Temporal Architectures**: Models that effectively capture interdependencies between spatial network topology and temporal evolution

(d) **Scalable Implementation**: Approaches that balance computational efficiency with predictive accuracy for large-scale networks.

This comprehensive gap analysis provides the foundation for our researches in this domain, ensuring that the proposed solutions address real limitations in current link prediction methodologies.

CHAPTER 3

PROPOSED METHODOLOGY AND RESULTS FOR ML BASED INTEGRATED CLASSIFIER

In this chapter, we present our methodology used for predicting potential links in a given network using our ML based model.

3.1 Proposed Architecture: ML Integrated Classifier

The proposed model integrates multiple features, including node centrality measures and similarity indices, which are then used to train ML models for accurate link prediction. The overall methodology consists of various phases that have been enumerated below.

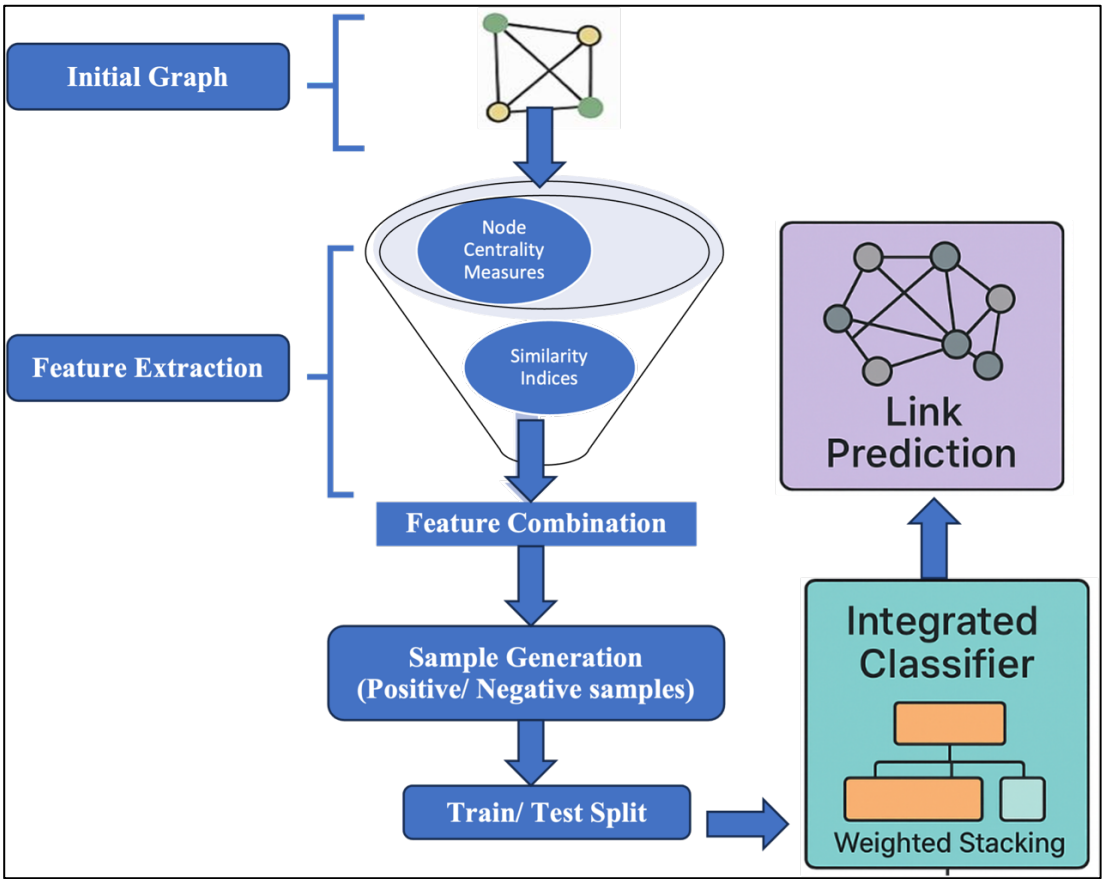

Figure 3.1: Link Prediction using Integrated Classifier

3.1.1 Feature Extraction

The quality and accuracy of link prediction depends significantly on the quality of the features used. In our approach, we extract two types of features: node centrality

measures and similarity indices, which accrue benefits of local and global structural properties of the graph dataset.

(a) Node Centrality Measures

Node centrality measures the popularity by quantifying the topological positioning of the nodes in the network. For each pair of vertex (u, v), we have extracted the five centrality scores mentioned below. For each edge (u, v), the centrality-based features are combined by concatenating the centrality values of both nodes.

We have utilized the following five node centrality measures that provides a wholesome picture with the combination of local and global centrality measures to capture various structural properties of the graph. A brief description along with the formulas are enumerated below:

(i) Clustering Coefficient:

The clustering coefficient C(v) measures the likelihood that a node's neighbours are interconnected. A high value of this coefficient indicates the presence of communities in a graph. However, on the flip side it does not account for the overall connectivity of the graph which indicates that the node with high clustering coefficient may still be poorly connected with the other parts of the graph [8]. It is defined as given by (1) :

$$C(v) = \frac{2 \cdot |\{(u,w) \in E : u,w \in N(v)\}|}{d(v) \cdot (d(v)-1)} \quad (1)$$

where N(v) is the neighbourhood of node v , and d(v) is its degree.

(ii) Katz Centrality:

Katz centrality determines the comparative influence of a node by not only considering the immediate neighbourhood but the entire network structure. It assigns a centrality score to each node based on the premise that any linkages to densely connected nodes weigh more to the overall score of a node than connections to sparsely connected nodes. Therefore, the nodes with high Katz score reflect that they are not only well connected, rather also connected to other influential nodes [33]. The Katz centrality is mathematically expressed as given by (2) :

$$C = \beta(I + \alpha A + \alpha^2 A^2 + \alpha^3 A^3 + \cdots) \quad (2)$$

where, A is the adjacency matrix, α is the decay factor, β is the bias term.

(iii) Radiality Coefficient:

The radiality coefficient measures the closeness of a node to all other nodes and quantifies how easily a node can reach other nodes. It considers the shortest path distances between nodes while incorporating the concept of reachability within a maximum possible distance in the network.

A substantial score means that a node can reach a large portion of the graph quickly. However, for bigger networks computational cost increases because of calculation of shortest paths between all pairs of nodes.

The radiality coefficient is mathematically expressed as given by (3) :

$$R(v) = \frac{\sum_{u \in V}(d_{max} - d(v,u))}{|V|-1} \quad (3)$$

where d_{max} is the maximum shortest path distance in the graph.

(iv) Extended Coreness(EC):

It is a way of identifying the core part of the network. The coreness of a node means the spatial positioning and is the largest k -value of the k -core of the graph. This implies that the particular node has at least k neighbours.

Extended coreness [34] includes not only the immediate degree of a node but also the broader structural properties of the graph. Highly connected dense graphs will have nodes with high extended coreness. As the graph size increases, the calculation for this metric may be computationally expensive.

Mathematically, EC centrality of a node is given by (4) :

$$C_e(v) = k(v) + \alpha \sum_{u \in N(v)}^{n} k(u) \quad (4)$$

where, *k(v)* is the core number from k-core decomposition, N(v) represents the set of neighbours of node *v*, *k(u)* is the core number of a neighbor *u* , α is a scaling factor (typically between 0 and 1) that adjusts the contribution of the neighbour's coreness.

(v) Vote Rank:

Vote rank is an iterative algorithm that ranks nodes based on their potential to spread influence in the network. Each node votes for its neighbours, and nodes with the highest votes are considered more influential. The voting process occurs iteratively, with previously selected nodes having reduced influence in subsequent rounds to prevent selecting similar or nearby nodes. Unlike many other centrality

measures that identify only a single most important node, Vote Rank selects a set of influential nodes. By reducing the influence of previously selected nodes, Vote Rank avoids selecting nodes that are too close to each other, ensuring better network coverage. Vote Rank excels in applications where the goal is to maximize the spread of information or influence, such as viral marketing, epidemic control, and rumour spreading.

(b) Similarity Measures

Similarity indices measure the chances of a link formation based on their structural similarity. Four similarity indices have been computed for each node pair (u, v). The combination of these similarity indices provides a comprehensive representation of the potential relationship between node pairs. The following four similarity indices have been employed in our framework:

(i) Preferential Attachment:

Preferential Attachment algorithm [35] calculates the similarity score based on the degree of nodes *u* and *v* respectively. It works on the concept of 'Rich gets Richer'. When a new node joins a network, it doesn't connect randomly to other nodes. Instead, it is more likely to link to other nodes that already have a high number of connections.
This index assumes that higher degree nodes are more likely to form new links. The mathematical expression is given by (5):

$$PA(u, v) = d(u) \cdot d(v) \tag{5}$$

where *d(u)* and *d(v)* are the degrees of nodes.

(ii) Rooted Random Walk:

This index calculates the probability of reaching a specific node through a random walk starting from a given node. A rooted random walk starts at a designated root node in a graph. The walk progresses by randomly choosing one of the neighbouring nodes to move to. The probability of moving to a neighbour 'u' from the current node 'v' is typically given by (6):

$$P(v \rightarrow u) = \frac{1}{deg(v)} \tag{6}$$

where *deg(v)* is the degree of node.

(iii) Laplacian Similarity:

Laplacian similarity is derived from the graph Laplacian matrix L , where the similarity is computed based on their spectral representation. Unlike simple similarity measures based on direct connections, Laplacian similarity captures the overall structure of the graph, making it more robust for complex networks. However, computing the eigenvectors of the graph Laplacian can be expensive for large graphs, making it less practical for massive networks. Let 'U' be the eigenvector matrix of the Laplacian, and let u_i and u_j represent the eigenvector components corresponding to nodes *i* and *j* . The similarity between nodes is given by (7) :

$$Sim_{Laplacian}(i,j) = \boldsymbol{u}_i^{\top}\boldsymbol{u}_j \tag{7}$$

(iv) Pearson Correlation:

Pearson correlation gives out the correlation between the two variables. It quantifies the linear relationship between two variables. It is given by (8) :

$$PC(u,v) = \frac{\sum_{i\in N(u)\cap N(v)}(A_{ui}-\mu_u)(A_{vi}-\mu_v)}{\sqrt{\sum_{i\in N(u)}(A_{ui}-\mu_u)^2}\cdot\sqrt{\sum_{i\in N(v)}(A_{vi}-\mu_v)^2}} \tag{8}$$

where μ_u and μ_v are the mean of the degrees of nodes.

3.1.2 Sample Generation:

Once the features are extracted, the samples are generated for training the ML models. This involves creating both positive as well as negative samples. Labels are assigned to each sample, with 1 for positive samples and 0 for negative samples. To ensure a balanced dataset, equal number of negative and positive samples were generated.

3.1.3 Feature Combination:

For each sample (positive or negative), the extracted centrality and similarity features were then concatenated to form a combined feature vector.

3.1.4 Model Training and Prediction:

ML classifiers are fed with the feature vector formed from the combined centrality and similarity features described above to predict potential links. Following ML models have been employed in our framework :

(a) RF:

RF [36] is a ML technique that builds multiple sets of decision trees and a combination of prediction of each of the tree is considered to make a final decision. Each such decision tree is trained on a random part of the dataset and uses different features at each split. This randomness helps prevent overfitting. Because it averages the results of many trees, RF is known for its worthwhile performance in classification tasks and can handle huge amounts of data effectively.

(b) XGB:

XGB is another efficient tree based model that enhance the quality of results from decision trees through multiple iterations. Its speed, scalability and the ability to improve the results of the traditional decision trees makes it a popular ML model. Furthermore, it has the ability to handle large datasets.

(c) LDA:

LDA is a classical dimensionality reduction method which is employed for classification problems involving multiple classes. LDA separates data for multiple classes through dimensionality reduction technique [37].

3.1.5 Integrated Classifier for Prediction

The final step in the process is to predict the links using the integrated classifier. In our approach, we integrate three ML classifiers to enhance link prediction performance. The final prediction is made by the proposed integrated classifier, wherein, the predictions from RF, XGB, and LDA are combined by associating dynamically learned optimal weights through Logistic Regression (LR) as the meta-classifier [9]. Using the learned weights from the LR meta-classifier, the final probability score for each instance is computed by (9) :

$$\Theta[i] = \frac{RF(T[i])\cdot wt_{RF} + XGB(T[i])\cdot wt_{XGB} + LDA(T[i])\cdot wt_{LDA}}{wt_{RF} + wt_{XGB} + wt_{LDA}} \tag{9}$$

Where wt_{RF}, wt_{XGB}, wt_{LDA} are the learned weights and RF(T[i]), XGB(T[i]), LDA(T[i]) represent the probability predictions from the respective classifiers each instance. The final binary classification decision is made using an optimized threshold which is given by (10) :

$$Prediction[i] = \begin{cases} 1 & if \Theta[i] \geq best\_threshold \\ 0 & otherwise \end{cases} \tag{10}$$

where *best_threshold* is dynamically selected to maximize the F1-score

3.2 EXPERIMENTAL SETUP

3.2.1 Software Requirements

(a) Operating System : Ubuntu 20.04/22.04 LTS, Windows 10/11, or macOS

(b) Python Version : Python 3.7 or higher.

3.2.2 Hardware Requirements

(a) Processor: Multi-core CPU (Intel i5/i7/i9, AMD Ryzen 5/7/9 or equivalent)

(b) Memory (RAM):

(i) Minimum: 8 GB
(ii) Recommended: 16–32 GB

(c) Graphics Processing Unit (GPU):

(i) GPU is optional but can accelerate certain ML tasks; CPU is sufficient for most experiments.

3.2.3 Major Libraries/ Packages

(a) scikit-learn: An open-source machine learning library also known as sklearn. It provides a wide range of tools for various tasks such as classification, regression, clustering, and dimensionality reduction.

(b) xgboost: It is a library optimized distributed and gradient boosting which provides parallel tree boosting.

(c) Numpy: A fundamental package for carrying out various computations on the experimental data in Python, and provides support for arrays and matrices.

(d) Pandas: It is a library that provides various data structures and tools for data analysis which is used for manipulation of mathematical and time series data.

(e) Matplotlib: It is a plotting library for Python which renders useful visualizations for the model and helps in visual analysis of the data points.

3.3 DATASET DESCRIPTION

3.3.1 Rationale for Datasets Selection

(a) Diversity of Scale: Ranges from moderate to large-scale networks.

(b) Interaction Heterogeneity: Includes social, expertise-based, and organizational communication patterns.

(c) Public Availability: All datasets are part of the Stanford Network Analysis Platform (SNAP), ensuring reproducibility.

3.3.2 Overview of Dataset Characteristics

The experimental validation of the integrated classifier model utilized three real-world network datasets spanning collaboration networks and social networks. These datasets were selected for their structural diversity and relevance to link prediction tasks. In Table I, summarized information for all datasets has been provided.

(a) CA-HepTh [38]: This dataset represents a collaboration network of authors in the field of High Energy Physics Theory. Each node represents an author, and an edge indicates co-authoring at least one paper.

(b) CA-GrQc [38]: This dataset represents a collaboration network of authors in the field of General Relativity and Quantum Cosmology. Nodes represent authors, and edges indicates co-authors relationships.

(c) Facebook [38]: This dataset represents an anonymized Facebook friendship network. Nodes correspond to Facebook users, and edges represent friendship relationships between users. In Table I, summarized information for all datasets has been provided.

TABLE III. VARIOUS DATASETS FOR INTEGRATED CLASSIFIER

S. No.	Dataset	**Type**	**Nodes**	**Edges**	**Domain**
(i)	**CA-HepTh**	Collaboration Network	9875	25,973	High Energy Physics
(ii)	**CA-GrQc**	Collaboration Network	5241	14,484	General Relativity and Cosmology
(iii)	**Facebook**	Social Network	4039	88,234	Social Network

3.4 COMPLEXITY ANALYSIS

The computational complexity of feature extraction and model training varies based on the methods used.

Feature extraction includes similarity measures like Preferential Attachment (O(1)), Rooted Random Walk (O(n)), Laplacian Similarity ($O(n^2)$), and Pearson Correlation (O(n)), where Laplacian Similarity is the most

expensive due to matrix operations.

Model training complexity differs across classifiers: Random Forest ($O(T m \log m)$), XGB ($O(T m \log m)$), where T is the number of estimators), and LDA ($O(n d^2)$), with XGBoost being the most computationally expensive. Optimizations such as feature selection, reducing XGB estimators, and using simpler similarity measures can improve efficiency.

3.5 RESULT ANALYSIS

The ML based integrated classifier approach combining five node centralities and four similarity indices with a dynamically weighted ensemble of RF, XGB, and LDA consistently outperformed individual classifiers across all tested datasets (CA-HepTh, CA-GrQc, Facebook). The integrated model achieved the highest AUC and F1-scores, demonstrating its ability to leverage both local and global structural features for more reliable link prediction. The dynamic weighting mechanism further ensured stable performance across varying data splits and thresholds, highlighting the model's robustness and generalizability. These results affirm that a unified feature extraction and ensemble learning framework can address the limitations of traditional methods, providing enhanced scalability and interpretability for static networks

The performance of the models were measured using AUC and F1-score. Across all datasets, the Integrated Classifier consistently outperformed the individual models, demonstrating its robustness in link prediction. The combination of various models as per dynamic weighting scheme resulted in more stable F1-scores for all datasets. The results for the same have been tabulated in table III below. These results have been tabulated for a fixed threshold of 0.5 for binary classification. Also, analysis of how the AUC and F1 scores vary with different values of thresholds for all the models was carried out across datasets and the sample results for CA-GrQc dataset have been given in Fig. 1 and Fig. 2. Likewise, similar results were achieved for all datasets. Further analysis was carried out based on capturing model performance by splitting the datasets into training and test sets according to a split ratio varied from 50% to 95% [9].

TABLE IV. RESULTS ACROSS DATASETS

Ser No.	Model	CA-GrQc		CA-HepTh		Facebook	
		AUC	F1	AUC	F1	AUC	F1
(a)	RF	0.981	0.937	0.962	0.915	0.991	0.9614
(b)	XGBoost	0.979	0.932	0.965	0.913	0.991	0.9613
(c)	LDA	0.967	0.903	0.959	0.902	0.969	0.8939
(d)	Integrated Classifier	0.982	0.938	0.967	0.918	0.992	0.9592

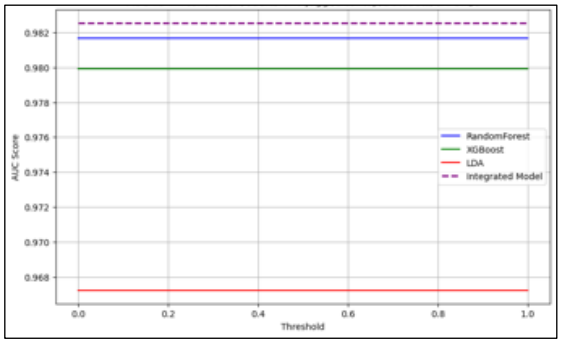

Figure 3.2. AUC vs Threshold

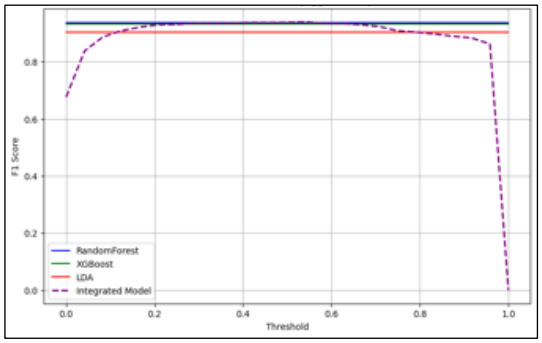

Figure 3.3. F1 vs Threshold

Best threshold value was found across various splits. The plots generated have been given in Figures 3, 4 and 5. For illustration purpose, bar graph giving performance summary of all models at 70% split ratio with best threshold parameters for CA-GrQc dataset has been given in Fig. 6.

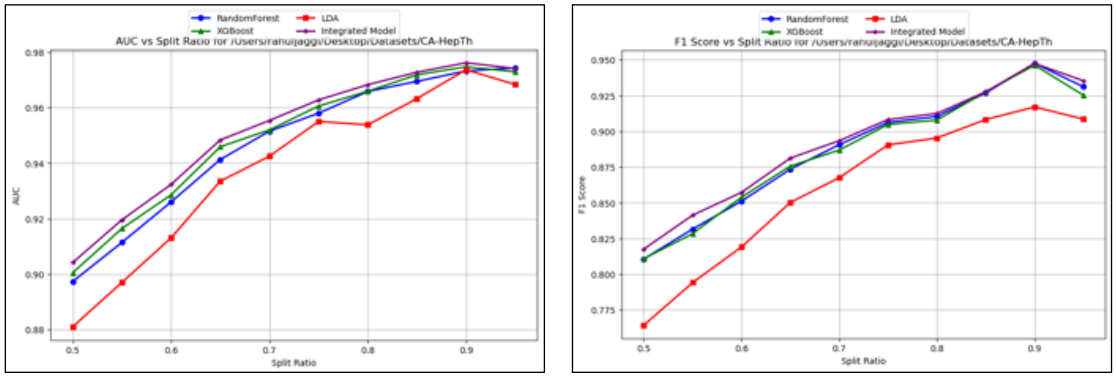

Figure 3.4. CA-HepTh(AUC and F1 score plots)

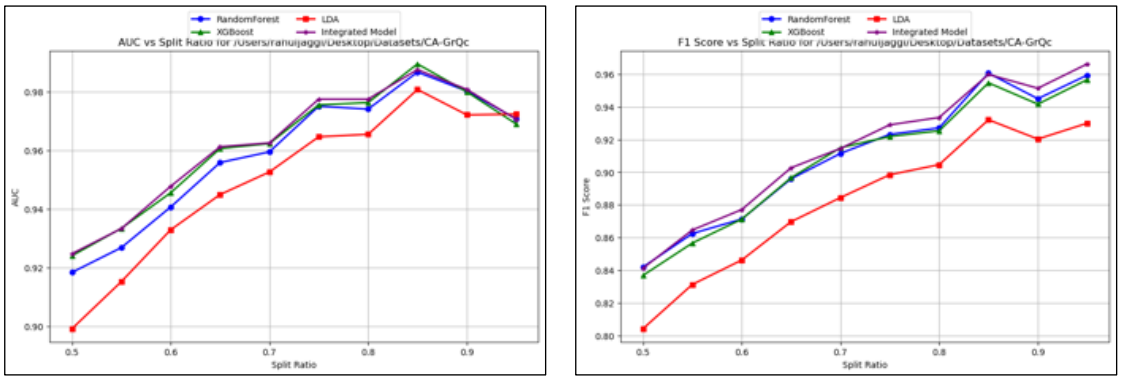

Figure 3.5. CA-GrQc(AUC and F1 score plots)

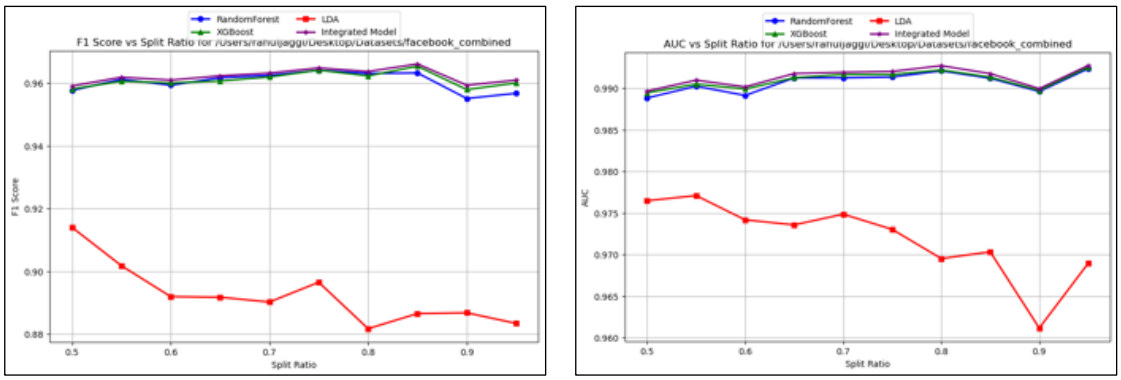

Fig. 3.6. Facebook(AUC and F1 score plots)

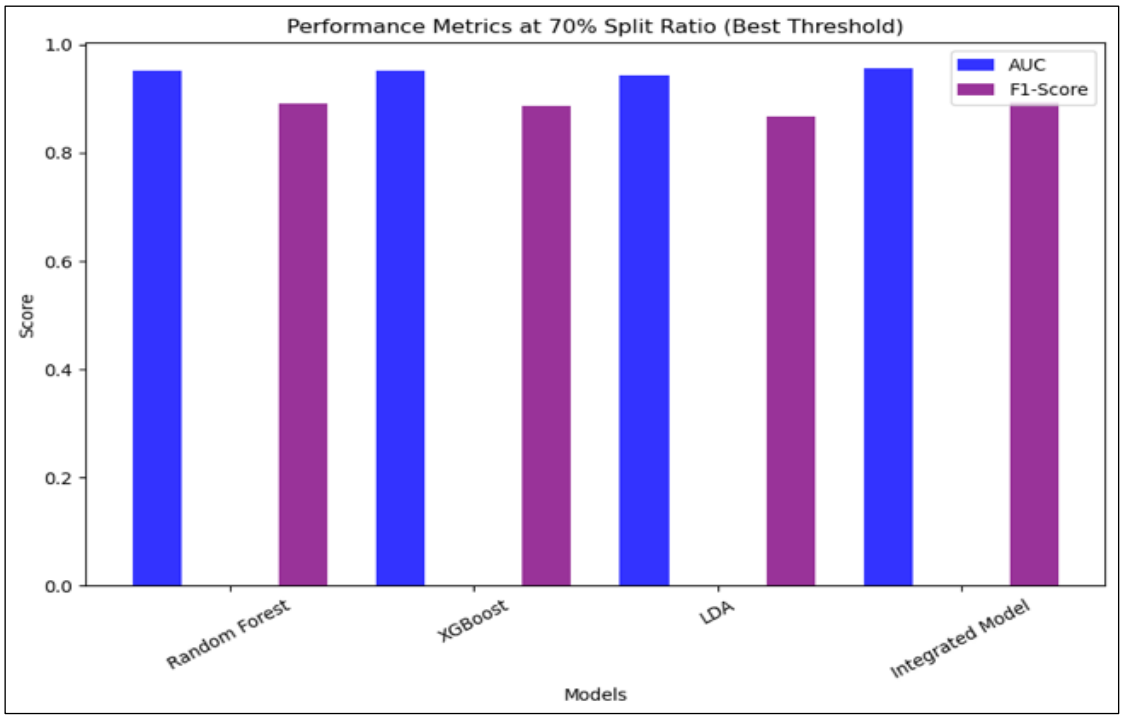

Fig. 3.7. Performance Summary (CA-HepTh)

Threshold selection is a critical aspect of binary classification models, as different thresholds affect how the classifier distinguishes between positive and negative links. To evaluate model stability across different dataset sizes, we analyze AUC at various split ratios (50% to 95%) [9]. This helps determine whether the models maintain their ranking performance as the available training data increases. Across all split ratios, the Integrated Model achieves the highest AUC, reaching 0.970, 0.974 and 0.993 at 95% split ratio for CA-GrQc, CA-HepTh and Facebook datasets respectively.

CHAPTER 4

PROPOSED METHODOLOGY AND RESULTS FOR DL BASED TA-GC-LSTM MODEL

In this chapter, we present our methodology used for dynamic link prediction through our model. The proposed model integrates Graph Convolution, LSTM and temporal attention aspects to effectively capture both spatial and temporal dependencies in graphs. By using the temporal attention mechanism, the model assigns different levels of importance to past time steps, helping it to learn complex patterns and long-term dependencies more effectively.

4.1 Proposed Architecture: DL Model

TA-GC-LSTM model comprises of three main components:

(a) Graph Convolution Layer
(b) Temporal Attention Layer
(c) LSTM Module

The above components enable the model to learn spatio-temporal patterns and make it suitable for dynamic link prediction.

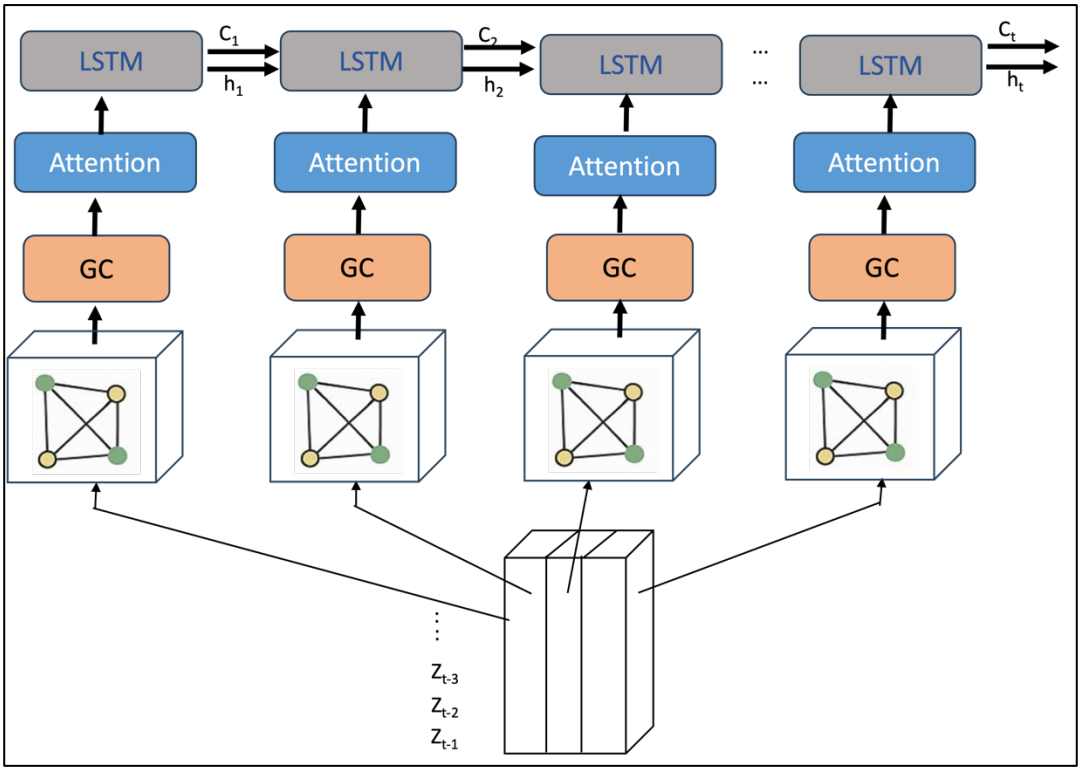

Figure 4.1: Dynamic Link Prediction using TA-GC-LSTM

4.1.1 Graph Convolution Layer

Once the graph snapshots have been prepared, the next step is to extract spatial

dependencies using GCNs. In order to improve computational efficiency, rather than traditional GCNs, which are inherently spectral graph convolutions, Chebyshev Graph Convolutions were used. The Chebyshev approximation enables efficient convolution without explicitly computing the Laplacian eigenvalues. The mathematic equation for the convolution operation is given by (11):

$$H_t^{(k)} = \sum_{k=0}^{K-1} \theta_k \cdot T_k(\tilde{L}) \cdot X_t \tag{11}$$

Where, $\tilde{L} = \frac{2L}{\lambda_{max}} - I$ is the Laplacian of the graph post scaling, L = D - A is the graph Laplacian, $T_k(\cdot)$ denotes the Chebyshev polynomial of order k , K is the number of Chebyshev polynomial terms and θ_k are learnable weights.

4.1.2 Temporal Sequence Modelling using LSTM

The spatial embeddings extracted through the graph convolution layers as described above are extracted and their temporal evolution are modelled using LSTM network. At each time step t, the LSTM receives the concatenated input $[H_t, C_t]$ from the Graph Convolution and Temporal Attention Layers, and computes the hidden and cell states. The LSTM cell equations are given by (12) to (17):

(a) Input, Ouput and Forget gates:

$$i_t = \sigma(W_i[H_t, C_t, h_{t-1}] + b_i) \tag{12}$$
$$f_t = \sigma(W_f[H_t, C_t, h_{t-1}] + b_f) \tag{13}$$
$$o_t = \sigma(W_o[H_t, C_t, h_{t-1}] + b_o) \tag{14}$$

(b) Cell States and Hidden State:

$$\tilde{c_t} = tanh(W_c[H_t, C_t, ht - 1] + b_c) \tag{15}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c_t} \tag{16}$$
$$h_t = o_t \odot tanh(c_t) \tag{17}$$

4.1.3 Temporal Attention Mechanism

The temporal attention mechanism selectively assigns importance to past time steps, allowing the model to focus on most influential historical interactions. The temporal attention mechanism is formulated as follows:
Given hidden states $H = [h_1, h_2, \dots, h_T]$, attention weights α_t are computed as given by (18) to (20):

$$e_t = v_a^T \cdot tanh(W_a H_t + b_a) \tag{18}$$

$$\alpha_t = \frac{exp(e_t)}{\sum_{k=1}^{T} exp(e_k)} \tag{19}$$

$$H^{'} = \sum_{t=1}^{T} \alpha_t \cdot H_t \tag{20}$$

Where, W_a, b_a, v_a are trainable attention parameters and H' is the weighted sum of past hidden states.

4.1.4 Feature Vector

From the node embeddings, the model then derives the edge representations. For an edge between two nodes, the equation is given by (21):

$$E_{ij} = [h_i \parallel h_j] \tag{21}$$

Where, ‖ represents concatenation. This forms the final feature vector for each candidate edge.

The context vector C_t is then concatenated with the current hidden state and fed into the LSTM module for temporal modeling.

4.1.5 Final Prediction Layer

The edge embeddings i.e. the feature vector are then mapped to respective probabilities by passing the feature vector through a Fully Connected (FC) layer.

$$\hat{y}_{ij} = \sigma(W \cdot E_{ij} + b) \tag{22}$$

where, W, b are learnable parameters and y_{ij} represents the probability of an edge forming.

4.1.6 Loss Function and Optimization

The model is trained using the Binary Cross-Entropy (BCE) loss function. In order to ensure faster convergence, the optimization is performed using Adam optimizer. The equation is given by (23):

$$\mathcal{L} = -\sum_{(i,j)} \left[y_{ij} \cdot log(\hat{yij}) + (1 - yij) \cdot log(1 - \hat{y}_{ij}) \right] \tag{23}$$

where, $\hat{y}_{ij}$ is the predicted probability and y_{ij} is the ground truth.

4.2 EXPERIMENTAL SETUP

4.2.1 Software Requirements

(a) Operating System : Ubuntu 20.04/22.04 LTS, Windows 10/11, or macOS

(b) Python Version : Python 3.7 or higher.

4.2.2 Hardware Requirements

(a) Processor: Multi-core CPU (Intel i5/i7/i9, AMD Ryzen 5/7/9 or equivalent)

(b) Memory (RAM):

(i) Minimum: 8 GB
(ii) Recommended: 16–32 GB

(c) Graphics Processing Unit (GPU):

(i) For Dynamic Link Prediction:

(aa) NVIDIA GPU with CUDA support (e.g., RTX 3060/3090, A100, or equivalent)

(bb) Minimum 8 GB VRAM; 12 GB or more recommended for large datasets and deep models

(cc) CUDA Toolkit 11.7+ for PyTorch compatibility

4.2.3 Major Libraries/ Packages

(a) PyTorch: An open-source machine learning library favoured for its ease of use and efficiency in creating and training neural networks.

(b) PyTorch Geometric: It is a library built upon PyTorch to easily write and train Graph Neural Networks (GNNs) for a wide range of applications related to structured data..

(c) Numpy: A fundamental package for carrying out various computations on the experimental data in Python, and provides support for arrays and matrices.

(d) Pandas: It is a library that provides various data structures and tools for data analysis which is used for manipulation of mathematical and time series data.

(e) Matplotlib: It is a plotting library for Python which renders useful visualizations for the model and helps in visual analysis of the data points.

4.3 DATASET DESCRIPTION

4.3.1 Rationale for Datasets Selection

(a) Diversity of Scale: Ranges from moderate to large-scale networks.

(b) Interaction Heterogeneity: Includes social, expertise-based, and organizational communication patterns.

(c) Public Availability: All datasets are part of the Stanford Network Analysis Platform (SNAP), ensuring reproducibility.

(d) Temporal Granularity: Varied timestamp resolutions test model adaptability to different time scales.

4.3.2 Overview of Dataset Characteristics

The experimental validation of the TA-GC-LSTM model utilized three real-world temporal network datasets, each representing distinct domains of interaction. These datasets were selected for their dynamic nature, enabling rigorous evaluation of spatio-temporal link prediction capabilities. In Table II, summarized information for all datasets has been provided.

(a) **CollegeMsg [38]:** The CollegeMsg dataset captures the communication dynamics of a facebook like online social network comprising 1,899 nodes representing users and 59,835 directed edges representing private messages exchanged between users. Each edge is timestamped, allowing for the analysis of temporal communication patterns.

(b) **sx-mathoverflow [38**]: The MathOverflow dataset captures user interactions on the MathOverflow Q&A platform, consisting of 24,818 nodes representing users and 506,550 directed edges indicating user-to-user interactions such as questions, answers, and comments. Each edge is timestamped, enabling the study of temporal communication patterns.

(c) **email-Eu-core-temporal [38]:** The Email-Eu-core-temporal dataset captures email communications consisting of 1,005 nodes representing individuals and 25,571 directed edges indicating emails sent between them. Each edge is timestamped, enabling dynamic analysis of interactions over a period of 803 days. The temporal nature of this dataset makes it suitable for studying link prediction, community evolution, and information diffusion in dynamic networks.

TABLE V. VARIOUS DATASETS FOR TA-GC-LSTM

Dataset	Type	Nodes	Edges	Domain
CollegeMsg	Directed, Temporal	1899	59,835	Messages on a Facebook like platform
Sx-mathoverflow	Directed, Temporal	24818	506550	Comments, questions, and answers on Math Overflow
Email-Eu-core-temporal	Directed, Temporal	1005	25,571	E-mails between users at research institution

4.4 COMPLEXITY ANALYSIS

For feature extraction and model training, the computational complexity of the TA-GC-LSTM model was analysed to find out its efficiency in link prediction tasks.

In the feature extraction phase, the node degree features were computed and the adjacency matrices were transformed into sparse representation. For a given adjacency matrix of size N x N, O(E) is the time complexity, where N is the number of nodes and E is the number of edges. Furthermore, the conversion of adjacency matrices into sparse edge lists requires O(E) operations per time step, leading to an overall time complexity of feature extraction as O(TE), where T represents the number of different time steps.

For the training phase of the model, the Chebyshev Graph Convolution (ChebConv) requires O(KTEF + TN) operations to aggregate information from the neighbourhood, where K is the Chebyshev polynomial order, F is the feature dimension. The LSTM-based sequence modelling introduced an additional complexity of O(TN + TNHT), where H represents the hidden dimension. So, the total time complexity per training epoch is the summation of above mentioned complexities.

4.5 RESULT ANALYSIS

The performance of the proposed model and its comparative analysis with baseline models, including GC-LSTM and T-GCN has been enumerated in the following paragraphs. For all datasets, interactions were binned into fixed time windows, creating dynamic adjacency matrices which captured the evolution of network structures over time. Node features were derived using degree based embeddings, and sparse adjacency matrices were converted into graph structures suitable for graph neural networks. The dataset was split into training, validation and testing with a ratio of 8:1:1. Adam optimizer with a learning rate of 0.001 was employed for training. Furthermore, in order to prevent overfitting, early stopping was implemented. We have also compared the performance to baseline models like GCN-LSTM and T-GCN, wherein our model demonstrated a well-balanced trade-off between computational efficiency and predictive performance, leveraging temporal attention to capture long-term dependencies while maintaining scalability for large dynamic graphs. In Tables

II to IV, performance comparison across different models for different metrics has been provided.

TABLE VI. AUC SCORES ACROSS DATASETS

Ser No.	Model	Collegemsg	Mathoverflow	Email-EU
(i)	**GC-LSTM**	0.925	0.908	0.919
(ii)	**T-GCN**	0.924	0.903	0.915
(iii)	**TA-GC-LSTM**	0.931	0.945	0.943

TABLE VII. ACCURACY ACROSS DATASETS

Ser No.	Model	Collegemsg	Mathoverflow	Email-EU
(i)	**GC-LSTM**	0.8427	0.8250	0.8408
(ii)	**T-GCN**	0.8395	0.8185	0.8372
(iii)	**TA-GC-LSTM**	0.8613	0.8650	0.8617

TABLE VIII. F1 SCORES ACROSS DATASETS

Ser No.	Model	Collegemsg	Mathoverflow	Email-EU
(i)	**GC-LSTM**	0.8398	0.8200	0.8447
(ii)	**T-GCN**	0.8354	0.8105	0.8357
(iii)	**TA-GC-LSTM**	0.8661	0.8700	0.8683

Graphical plots for the evaluation metrics were generated for visualizing one shot comparison of all the models. The same have been given in Figures 1, 2 and 3 below.

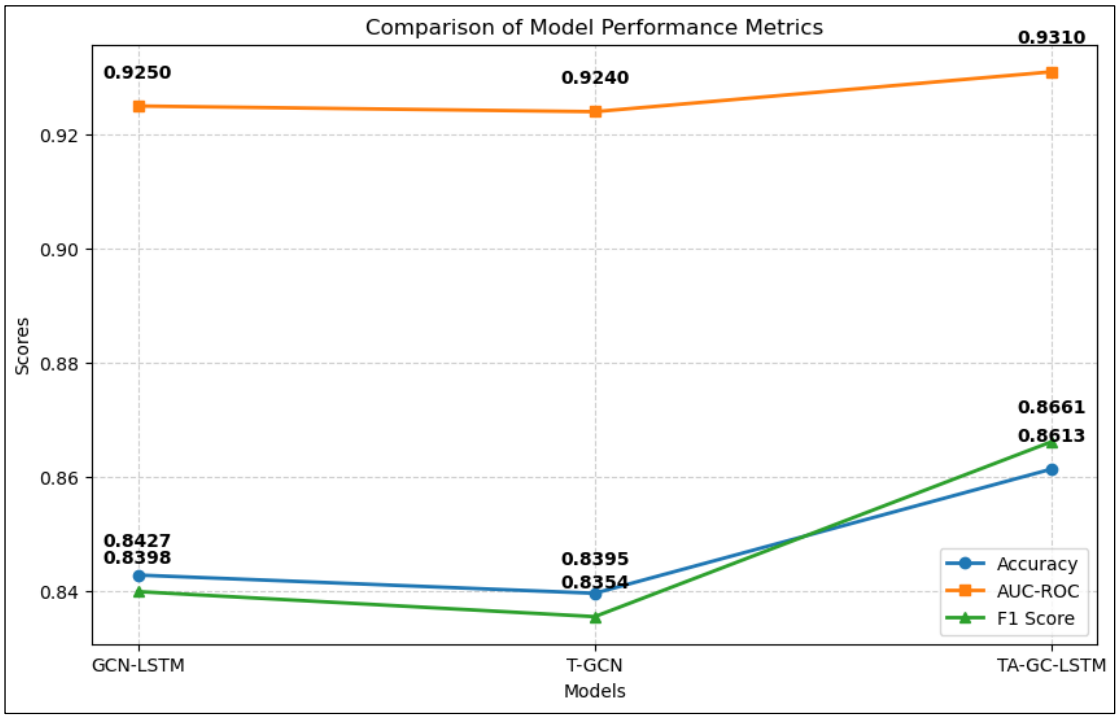

Fig. 4.2. Performance Summary(CollegMsg Dataset)

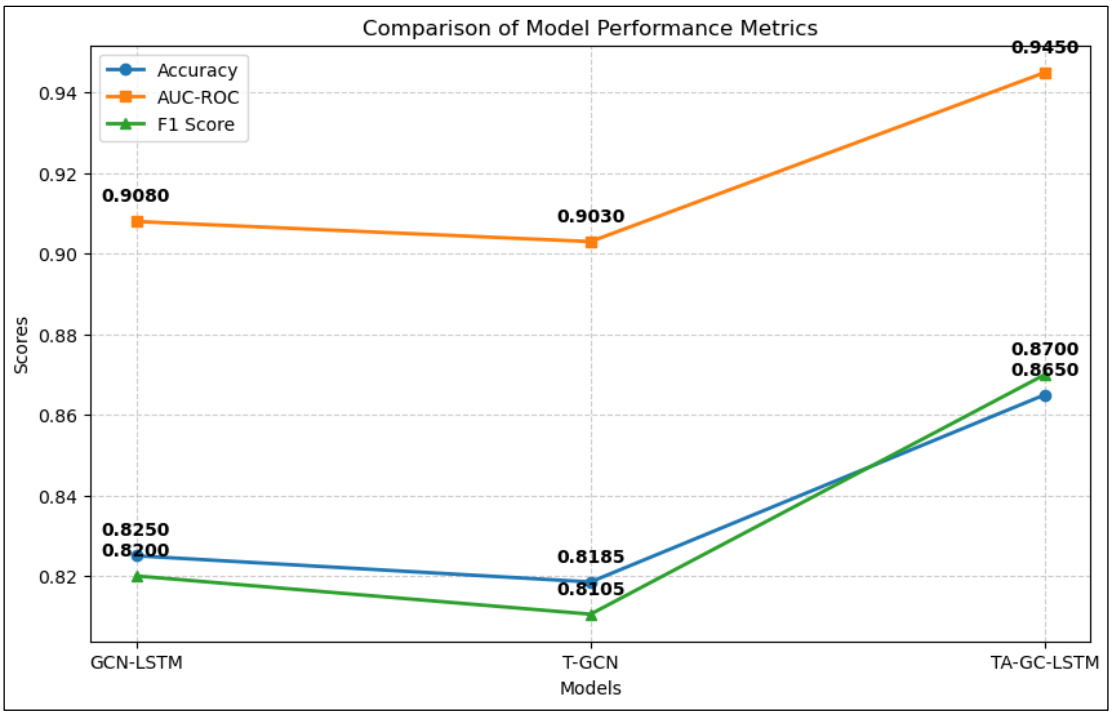

Figure 4.3. Performance Summary (MathOverflow)

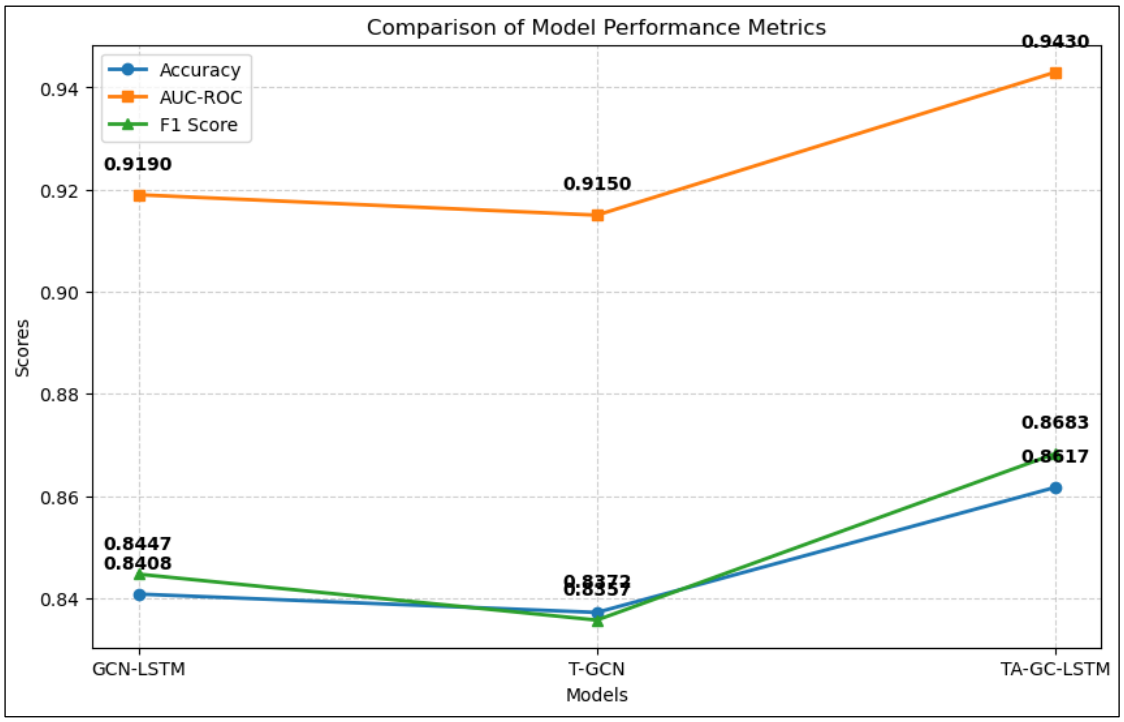

Figure 4.4. Performance Summary (Email-EU)

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

Despite the advancements in link prediction, most existing works focus on either local features individually or global features of the network, but combination of features provide a more robust approach.

Few studies have attempted to integrate diverse centrality measures and similarity indices into a unified framework. This gap motivated our proposed approach, which combines various graph-theoretic features, including clustering coefficient, Katz centrality, and vote rank, with ML classifiers for robust link prediction.

By leveraging both local and global structural properties, our method addresses the limitations of existing techniques and offers improved scalability for large networks.

Similarly, in the DL domain, by integrating GCN, LSTM and Temporal Attention Mechanism, our model effectively captured both spatial and temporal dependencies in evolving graph structures.

The temporal attention module allowed the model to selectively focus on past interactions that had a significant influence on future connections, improving predictive accuracy while maintaining computational efficiency.

Our results demonstrated that TA-GC-LSTM outperformed existing baselines, including GCN-LSTM and T-GCN, with the evaluation metrics being AUC, F1-score, precision, and recall, indicating its superiority in learning dynamic network representations

5.3 FUTURE WORK

Our study acknowledges that we have done our analysis on only 3 datasets for each of the models, which are relatively small compared to large-scale real-world networks like Twitter, DBLP, and LinkedIn. While these datasets provide valuable insights, they may not fully capture the complexity of large networks.

Future work should focus on testing our approach on bigger datasets to evaluate its scalability and real-world applicability. Link prediction can be a powerful tool in counter-terrorism by helping security agencies detect hidden connections between individuals and predict potential threats. Terrorist networks often operate in secret, making it difficult to track their activities using traditional methods.

By analyzing social interactions, financial transactions, and communication patterns, link prediction can uncover relationships that might otherwise go unnoticed. This can help in identifying new recruits, tracking suspicious activities on the dark web, and preventing coordinated attacks.

Additionally, it can improve intelligence sharing between agencies by connecting fragmented data. To ensure reliability, techniques like SHAP and LIME can be used to explain why certain links are predicted, making the model more interpretable and useful for real-world security operations.

Furthermore, self-supervised learning techniques could be incorporated to minimize the need for labeled data, making the model more applicable in scenarios where ground truth link information is sparse. Furthermore, exploring adaptive temporal binning methods that dynamically adjust time intervals based on activity levels in the network could further refine the model's temporal learning capabilities.

REFERENCES

[1] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," Physica A: Statistical Mechanics and its Applications, vol. 390, no. 6, pp. 1150-1170, 2011.

[2] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," Journal of the American Society for Information Science and Technology, vol. 58, no. 7, pp. 1019–1031, 2007.

[3] Veeramallu, M. S., Mallu, H. R. (2024). Link Prediction in Social Networks: A Review. IEEE Conference on Emerging Technologies.

[4] Ghorbanzadeh H, Sheikhahmadi A, Jalili M, Sulaimany S (2021) A hybrid method of link prediction in directed graphs. Expert Systems with Applications. 165:113896

[5] M. S. Rahman, L. R. Dey, S. Haider, M. A. Uddin, and M. Islam, ''Link prediction by correlation on social network,'' in Proc. 20th Int. Conf. Comput. Inf. Technol. (ICCIT), Dhaka, Bangladesh, Dec. 2017.

[6] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[7] L. Li, S. Fang, S. Bai, S. Xu, J. Cheng, and X. Chen, ''Effective link prediction based on community relationship strength,'' IEEE Access, vol. 7, pp. 43233–43248, 2019.

[8] M. E. J. Newman, *Networks: An Introduction*, 2nd ed. Oxford University Press, 2018.

[9] 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. 10, 2022.

[10] 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 (KDD), 2014, pp. 701-710.

[11] 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 (KDD), 2016, pp. 855-864.

[12] 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 (WWW), 2015, pp. 1067-1077.

[13] F. Ziya and S. Kumar, "GSVAELP: Integrating GraphSAGE and Variational Autoencoder for Link Prediction," *Multimedia Tools and Applications*, 2024. doi: 10.1007/s11042-024-20123-z.

[14] S. Zhou, J. Yang, H. Wang, X. Ren, and Y. Sun, "DynamicTriad: A dynamic graph embedding framework for social network evolution," in *Proceedings of the 22nd International Conference on World Wide Web (WWW)*, 2018, pp. 117-126.

[15] A. Nguyen, T. Dinh, and M. Thai, “Continuous-time dynamic network embeddings,” in Proceedings of the 2nd International Workshop on Learning Representations for Big Networks (BigNet), 2018.

[16] W. Yu, H. Qian, Y. Hu, X. Ma, and L. Xie, “T-GCN: A temporal graph convolutional network for traffic prediction,” IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 9, pp. 3848-3858, Sep. 2020.

[17] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” Int. Conf. Learn. Representations (ICLR), Vancouver, Canada, Apr. 2018.

[18] Chen, J., Wang, X. & Xu, X. GC-LSTM: graph convolution embedded LSTM for dynamic network link prediction. Appl Intell 52, 7513–7528 (2022). https://doi.org/10.1007/s10489-021-02518-9

[19] P. Velickovic et al., “Graph attention networks,” in Proceedings of ICLR, 2018.

[20] W. Huang et al., “Adaptive sampling towards fast graph representation learning,” in Proceedings of NeurIPS, 2018, pp. 4563-4572.

[21] H. Xu et al., “T-GAT: Learning time-aware graph attention for risk prediction,” in AAAI, 2022, pp. 6265-6272.

[22] B. Guo et al., “ASTGCN: Attention-based spatio-temporal graph convolutional networks for traffic forecasting,” in IJCAI, 2019, pp. 4057-4063.

[23] A. Kumar, A. Mallik and S. Kumar, "TLP-NEGCN: Temporal Link Prediction via Network Embedding and Graph Convolutional Networks," in IEEE Transactions on Computational Social Systems, vol. 11, no. 3, pp. 4454-4464, June 2024, doi: 10.1109/TCSS.2024.3367231.

[24] A. Ghosh, S. Sharma, and R. Gupta, “Selective shallow models strength integration for emotion detection using GloVe and LSTM,” Multimedia Tools and Applications, vol. XX, no. XX, pp. XX–XX, 2023.

[25] S. Yadav and A. Pal, “Bi-RNN and Bi-LSTM based text classification for Amazon reviews,” Journal of Information Science, vol. XX, pp. XX–XX, 2023.

[26] H. Singh, R. Kumar, and A. Srivastava, “F-DES: Fast and deep event summarization,” Knowledge-Based Systems, vol. XX, no. XX, 2023.

[27] M. Sharma and R. Kaushal, “Video summarization using deep learning techniques: A detailed analysis and investigation,” ACM Computing Surveys, vol. XX, no. XX, 2024.

[28] S. Kumar, N. Gupta, and A. Malhotra, “Text query-based summarized event searching interface system using deep learning over cloud,” Future Generation Computer Systems, vol. XX, 2024.

[29] K. Mehta and J. Roy, “Computationally efficient ANN model for small-scale problems,” Journal of Computational Intelligence, vol. XX, no. XX, 2023.

[30] D. Verma, R. Verma, and M. Singh, “Predictive analytics for recognizing human activities using residual network and fine-tuning,” IEEE Access, vol. XX, pp. XX–XX, 2024.

[31] R. Bansal and T. Gupta, “End-to-end residual learning-based deep neural network model deployment for human activity recognition,” Applied Soft Computing, vol. XX, 2024.

[32] Verma, R., & Agarwal, M. M. (2025). Enhancing Road Accident Classification with Capsule Recurrent Neural Network and Improved Reptile Search Algorithm. IETE Journal of Research, 1–20. https://doi.org/10.1080/03772063.2025.2464239

[33] J. Zhan, S. Gurung, and S. P. K. Parsa, “Identification of top-K nodes in large networks using Katz centrality,” *Journal of Big Data*, vol. 4, no. 16, 2017.

[34] Bae J, Kim S (2014) Identifying and ranking influential spreaders in complex networks by neighborhood coreness. Physica A Stat Mechan Appl 395:549–59

[35] R. S. Ahmad Zareie, “Similarity-based link prediction in social networks using latent relationships between the users,” 2020.

[36] Breiman L (2001) Random forests. Machine learning 45(1):5–32

[37] S. Ji and J. Ye, "Generalized Linear Discriminant Analysis: A Unified Framework and Efficient Model Selection," in IEEE Transactions on Neural Networks, vol. 19, no. 10, pp. 1768-1782, Oct. 2008, doi: 10.1109/TNN.2008.2002078.

[38] J. Leskovec and A. Krevl, *SNAP Datasets: Stanford Large Network Dataset Collection*. [Online]. Available: http://snap.stanford.edu/data.

[39] Jayachitra Devi, S. and Singh, B., 2020. Link prediction model based on geodesic distance measure using various machine learning classification models. Journal of Intelligent and Fuzzy Systems, 38(5), pp.6663-6675.

[40] Berahmand, K., Nasiri, E., Rostami, M. and Forouzandeh, S., 2021. A modified DeepWalk method for link prediction in attributed social network. Computing, 103, pp.2227-2249.

[41] Kumar, S., Mallik, A. and Panda, B.S., 2022. Link prediction in complex networks using node centrality and light gradient boosting machine. World Wide Web, 25(6), pp.2487-2513.

[42] Chen, Y.L., Hsiao, C.H. and Wu, C.C., 2022. An ensemble model for link prediction based on graph embedding. Decision Support Systems, 157, p.113753.

[43] E. Rossi, B. Paassen, K. Thekumparampil, H. Zhao, A. Anandkumar, and F. Monti, “Temporal graph networks for deep learning on dynamic graphs,” ICML Workshop on Graph Representation Learning, Vienna, Austria, Jul. 2020.

LIST OF PUBLICATIONS & THEIR PROOF

1. A paper titled 'Enhanced Link Prediction Analysis using Integrated Classifier' was presented in 3rd International Conference on Communication, Security and Artificial Intelligence (ICCSAI- 2025) held from 04th -06th April 2025 organized by Galgotia's University, Greater Noida, UP.

GALGOTIAS UNIVERSITY

ICCSAI

IEEE UP SECTION (INDIA)

Certificate of Presentation

This is to certify that

Dr./Mrs./Ms./Mr. RAHUL JAGGI

has presented the paper entitled

Enhanced Link Prediction Analysis Using Integrated Classifier

in 2025 3rd International Conference on Communication, Security and Artificial Intelligence, 04th-06th April 2025, Organised by Galgotias University, Greater Noida, UP.

Dr. Avadhesh Kumar	Dr. Nitin Kumar Gaur	Dr. K. Mallikharjuna Babu
General Chair	Registrar	Patron, ICCSAI-2025
Pro-VC, Galgotias University (U.P)	Galgotias University (U.P.)	VC, Galgotias University (U.P.)

2. A paper titled 'Dynamic Link Prediction Analysis using Temporal Adaptive Graph Convolutional Long Short Term Memory' was presented in 3rd International Conference on Computer, Electronics and Electrical Engineering (IC2E3- 2025) held from 15th - 16th May 2025 organized by National Institute of Technology, Uttarakhand. University, Greater Noida, UP.

IEEE UP SECTION (INDIA)

IAS IEEE INDUSTRY APPLICATIONS SOCIETY

राष्ट्रीय प्रौद्योगिकी संस्थान, उत्तराखण्ड
National Institute of Technology, Uttarakhand

AN
INTERNATIONAL CONFERENCE ON
COMPUTER, ELECTRONICS AND ELECTRICAL ENGINEERING
AND THEIR APPLICATIONS (IC2E3-2025)

(May 15th – 16th, 2025)

Certificate of Participation

This is to certify that Rahul Jaggi *has participated and presented the paper entitled* Dynamic Link Prediction Analysis Using Temporal Adaptive Graph Convolutional Long Short-Term Memory *contributed by* Rahul Jaggi and Sanjay Kumar *in the* ***3rd International Conference on Computer, Electronics and Electrical Engineering and their Applications (IC2E3-2025)****, held from 15th to 16th May, 2025 at* ***National Institute of Technology Uttarakhand****, India.*

Dr. Pankaj Kumar Pal	Dr. Sourav Bose
Organizing Secretary, IC2E3-2025	*Organizing Chair, IC2E3-2025*

NITUK/IC2E3/25/PAC/OFL/PID/707

PLAGIARISM REPORT

CURRICULUM VITAE/BRIEF PROFILE