Link List Predection using Graph Neural Network

Final Year Design Project

By
Shifat Kamal Apon
181-15-11191

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the **Degree of Bachelor of Science in Computer Science and Engineering**

Supervised by

Mr. Md. Firoz Hasan Lecturer (Senior Scale)

Department of Computer Science and Engineering Daffodil International University

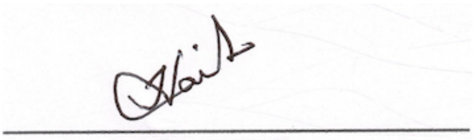
DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

January 13, 2024

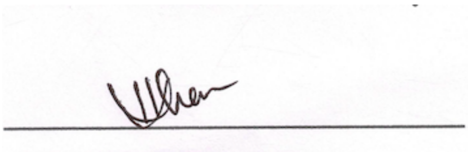
APPROVAL

This project titled "**Link List Prediction using Graph Neural Network**" submitted by **MD. SHIFAT KAMAL APON** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **13-01-2025**.

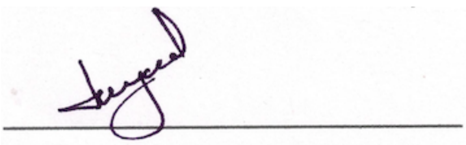
<u>BOARD OF EXAMINERS</u>



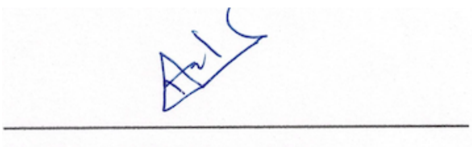
Dr. Sheak Rashed Haider Noori (SRH)
Board Chairman
Professor & Head Chairman, Department of CSE, FSIT
Daffodil International University



Most. Hasan Hena (HH)
Internal Examiner
Assistant Professor, Department of CSE, FSIT
Daffodil International University



Md. Ferdouse Ahmed Foysal (FAF)
Internal Examiner
Lecturer, Department of CSE, FSIT
Daffodil International University



Dr. Md. Arshad Ali (DAA)
External Examiner
Professor, Department of Computer Science and Engineering
Hajee Mohammad Danesh Science and Technology University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Mr. Md. Firoz Hasan**, **Lecturer (Senior Scale)**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

13.1.24

Mr. Md. Firoz Hasan
Lecturer (Senior Scale)
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:

Shah Md. Tanvir Siddiquee
Lecturer (Senior Scale)
Department of Computer Science and Engineering
Daffodil International University

Shifat 13/01/24

MD. SHIFAT KAMAL APON
Student ID: 181-15-11191
Department of Computer Science and Engineering
Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project(FYDP)** successfully.

We are grateful and wish our profound indebtedness to **Mr. Md. Firoz Hasan**, Lecturer (**Senior Scale)** Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Deep Learning** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

In this study, the applications of Graph Neural Networks (GNNs) for link prediction in citation networks have been explored using a dataset called Cora. We can visualize the networks as paper nodes of the graph and the citations as edges of the graph, and we can form a complex graph that can be analyzed using the GNN model. Previously, we used the Stallergraph library to perform some link prediction tasks using some necessary libraries to construct, validate, and train. This is based on the accuracy and the training loss. We achieve a satisfactory result of 0.7771 in predicting the existence of links between nodes. We understand the complex relationship of the graph, and the losses show a consistent and stabilized decline in losses. The loss is 0.9994. We can validate the model by learning the patterns of this complex network. Also, we discuss the challenges, but mainly, we have studied the step processes of how the graph convolution works. In deep learning for having a huge data, it is a must to experience a latency-sensitive structure where previously we used arrey to solve problems of linking and sorting. Link lists present models of the the next generation for handling complex data. Within a graph adgeny we can find different features that previously could not be fetched or not used to validate, but with the help of neural convolution, we can visualize a real-time connection and link within different classes.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

This paper is organized to represent the GNN accuracy over machine learning. Related works are given for an overview of existing methods for graph neural networks.

The background covers the terms and partial knowledge of the field of the deep learning mathods, embedding the data, different algorithm and different model to create a wider knowledge of the field of the GNN. Here we can see the relations within gnn and cnn how its related and how the mathode covulations works through the graph. Moreover the gnn is a multilayer shifting mathode of cnn that works with graph and big size data.

In methodology, we briefly present the binarization and graph labeling of the data and render the data to create a visual representation for the data sorting, and then epochs, validate, loss, and accuracy for different methods, with Adam optimizing and more.

In implementation, the cora dataset is used and sorted for training and testing, and then the sorted data is put into dimension and layer of the convolutional network. The code is to find the accuracy and minimal losses of the data and predict them through Softmax for a better outcome.

Finally, in the result, the model is finalized, and we will review the losses and the data visualization and the data representations the link prediction visualizatin is a big part of the result with the calculating the loss function and accuracy. with in the graph we will be finding out many additional information that previous model cant retrieve with in pre processing mathods, those model who can are not to good with big data like ours.

We will also be finding the implementation and the future work and conclutaion in the last stage and willing to get it to a executable daily uses with the initiative of the future work of this service field like search engine, ai search, mapping , notes, effective keyword relatings etc

1.2 Motivation

In the deep learning field, graphs are very explicit and can handle massive computational power. A new approach for representation learning of graphs can be used in the field of machine learning called Graph Neural Network (GNN) [1], [2]. It was first introduced in 2005 by Marco Gori [3] and 2009 by Franco Scarselli [4]. They introduce the model that GNN used to preprocess the informative graph data and predict the result, respectively, with as few data losses as possible. In the artificial neural network, every node creates a new feature vector by combining the feature vectors of its neighbors; this is called the recursive neighborhood aggregation scheme [5], [6], where previously in machine learning we preprocess the data by squashing the graph structure data into vectors and relating it to the semi-supervised data by a list data processing method. More recently, it has various schemes for different GNN variants. Most likely in link prediction, community detection, graph classification, and node classification, GNN has achieved state-of-the-art performance. It has recently attracted increasing research attention due to its wide range of applications in systems like social science and networks [7], neural science and physical systems [8], [9], interaction networks between proteins [10], knowledge graphs [21], and so on. Also, GNN can be understood by algorithms such as averaging from neighbor classes, iterative classification, random walk method [11], label propagation [12], label spreading [20], and laplacian regularization [13].

We consider link list prediction using machine learning, deep learning, and graph theory. From the labeled and unlabeled classes, the problem of reading data with graph node classification has attracted major focus in recent neural network fields. Furthermore, with the diverse increase in development with graph and neural networks, mass data collected from a large number of datasets, such as Cora, can be analyzed, clustered, and classified with methods like FullBatchNodeGererator and label Binarizer, can be predicted with Softmax, and can be optimized for accuracy with gradient descent [14] .

1.3 Objectives

The main highlight of our work is to find the relations of a group of semi-supervised nodes from their features without data loss and predict the accuracy of the algorithm we went through. We consider the task of classification or annotation of multi-relational linking in multi-label and single-label nodes. For this case, the application is finding the area of nodes (citations) within the nodes (research paper) and classifying them, as well as finding and listing the predictions of their relation. We propose a new method of graph neural networks for binarizing them and predicting their accuracy. This model is not restricted to our applications and can be applied to any related dataset.

1.4 Methodology

GCN with epochs was applied to the CORA dataset using Stellar Graph in order to classify the papers based on the seven classifications. Graph analytics software for neural network machine learning is offered by Stellar Graph. We are comparing the contents of 2708 papers with 1435 parts to see if a paper is related to another part or not. We may also get a summary of all the articles based on the topics and concepts and how closely they are related to one another. Following training and validation accuracy, we can compare the projected classification with the paper's true classification to determine how well the research performed using PCA and t-SNE.

1.5 Project Outcome

This paper is organized to represent the GNN accuracy over machine learning. Related works are given for an overview of existing methods for graph neural networks.

In methodology, we briefly present the binarization and graph labeling of the data and render the data to create a visual representation for the data sorting, and then epochs, validate, loss, and accuracy for different methods, with Adam optimizing and more.

In implementation, the cora dataset is used and sorted for training and testing, and then the sorted data is put into dimension and layer of the convolutional network. The code is to find the accuracy and minimal losses of the data and predict them through Softmax for a better outcome.

Finally, in the result we show the model is finalized, and we will review the outcome.

1.6 Organization of the Report

We briefly describe the data's binarization and graph labeling in the methodology section. We then render the data to produce a visual picture of the data sorting process. We also discuss epochs, validation, loss, and accuracy for several methods, including Adam optimization.
During implementation, the cora dataset is used and sorted for testing and training. The sorted data is then added to the convolutional network's layers and dimensions. For a better result, the code determines the data's correctness and minimal losses and uses Softmax to anticipate them.
Ultimately, the model is completed, and we will examine the losses as well as the data formats and visualizations. The prediction of the link Visualization has a significant role in society

Chapter 2

Background

2.1 Introduction

In this paper, we introduce the link listing module of graph neural networks from a broad perspective. We will first start with the graph.

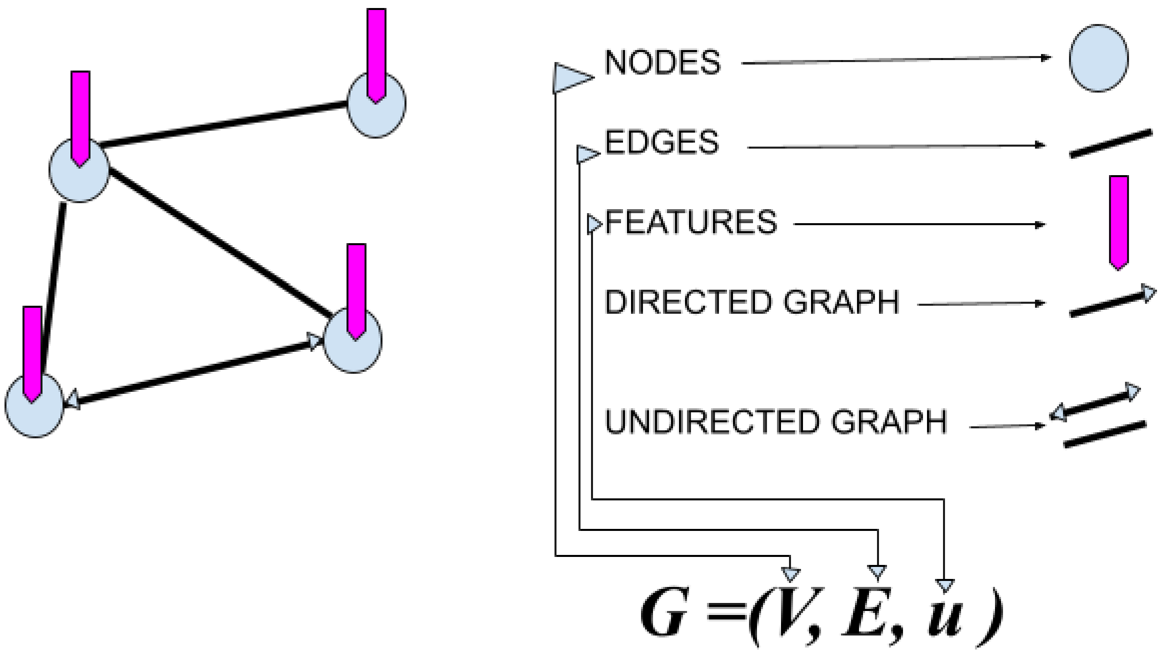


Figure 2.1: Graph Definition

In Figure 2.1, we can see that the nodes are the individual entities, the edges connect them, and the features are the nodes carrying information. For homogeneous, the mathematical representation of the graph is G; the nodes are V; one type of node defines homogeneous, whereas many types of nodes define heterogeneous. The edges can be defined by the adjacency matrix or weight matrix (adjacency, weight) = $(A,\ W)$; and the feature vector is u, which carries the information of the nodes.

A directed graph has edges that point in one direction only; therefore, it is an asymmetric matrix. An undirected graph has edges that point in both directions or neither; therefore, it is a symmetric matrix.

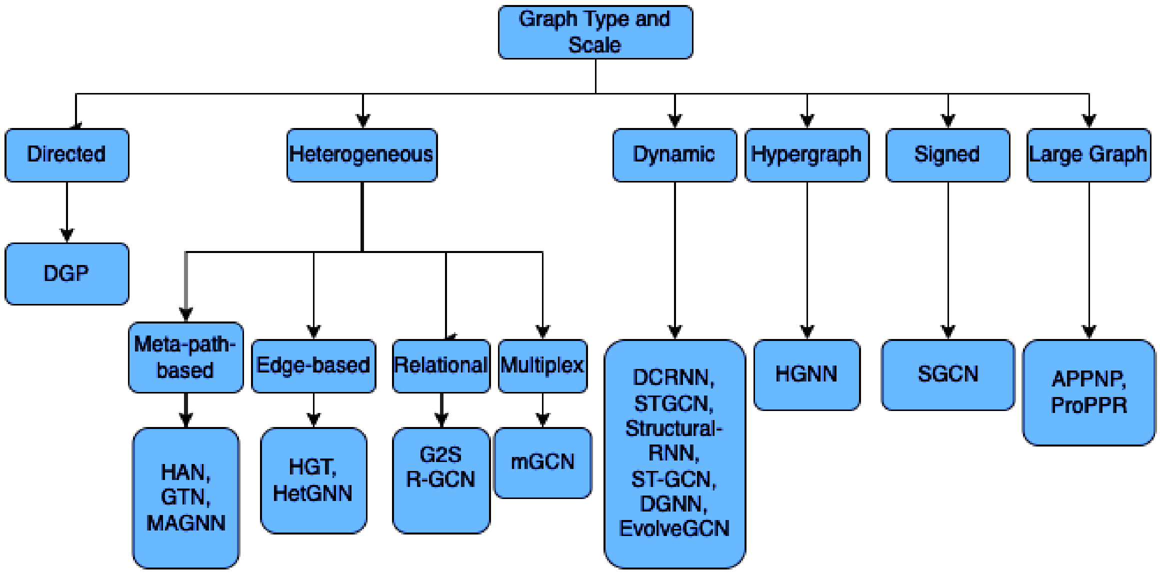


Figure 2.2: Graph type and scale

For the representation of the nodes, we can use a matrix with an edge list, where the first is the source node and the second is the target node. A furthermore graph can have various types and scales, as shown in Figure 2.2 for a visualization of the field of the graph.

2.1.1 Graph Representation

Another way for data to be stored is to use a binary method like an adjacency matrix. With a weight matrix, we can see how powerful the connections are in a float number instead of a binary value. The graph's laplacian is the difference between the degree matrix and the adjacency or weight matrix, whereas the degree indicates the node's influence. $(L = D - A)$ $(L = D - W)$. A laplacian can hold value of a graph of diagonally the connection of each node and non-diagonally the connection of others nodes in form of negative value so we can find more information from it.

$$L_{ij}(G) = \begin{cases} d_v & \text{if } i = j \\ -1 & \text{if } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The diagonal elements l_{ij} of L are therefore equal to the degree of the vertex, v_i and off-diagonal elements l_{ij} are -1 if the vertex v_i is adjacent to v_j or 0 otherwise [15]. Simpler version of the laplacian matrix L [16]

$$L_{ij}(G) = \begin{cases} 1 & \text{if } i = j \text{ and } d_j \neq 0 \\ -\frac{1}{\sqrt{d_i d_j}} & \text{if } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

If the element is on the diagonal of the matrix and the node j has at least one edge, we can consider the graph to be 1, where there is an edge between, i and j then the graph will be $-\frac{1}{\sqrt{d_i d_j}}$ or 0 otherwise [17].

Table 2.1: Basic Notations of Laplacian Matrix.

Notations (s)	**Descriptions**
L	Laplacian
A, W	Adjacency
$\bar{L}, \bar{A}, \bar{W}$	Normalized graph
D	Degree of graph
d_i	Degree of the vertex i
i , j	vector features

2.1.2 Graph Model

Graph data has some model that takes the input and puts it into some learning algorithm to find the specific output. The graph representation is like a data representation, and like many learning models, graph data has some model that describes some sort of output.

For new node connections, we have an application called node prediction that can predict nodes. One of the models can predict the links between the nodes, and this is called link prediction. Another model is the graph representation, where we use a graph to construct a model of another graph with a different representation. In this paper, we will be using link prediction and creating a new model using a pre-trained state-of-the-art algorithm.

In modern times, there are several algorithmic frameworks, among them DeepWalk, node2vec, which are random-walk mathods, GraRep, HOPE, Laplacian Eigenmaps, and Graph Factorization, which are matrix factorization [18]. Also, we have Line, GGCN, GCN, GST, Metapath2Vec, Struc2Vec, GraphSage, G2G, GIN, HGAT, DGI, HGNN, GCNII, GT, and EGT, which are the most popular ones. Each of these variants uses a different functions for aggregation and transforming features mathods like DIFFPOOL [19], Graph Capsule [20], WReN [21], RN [22] MPNN [23], GraphSAGE [24], IN [8], GAT [25].

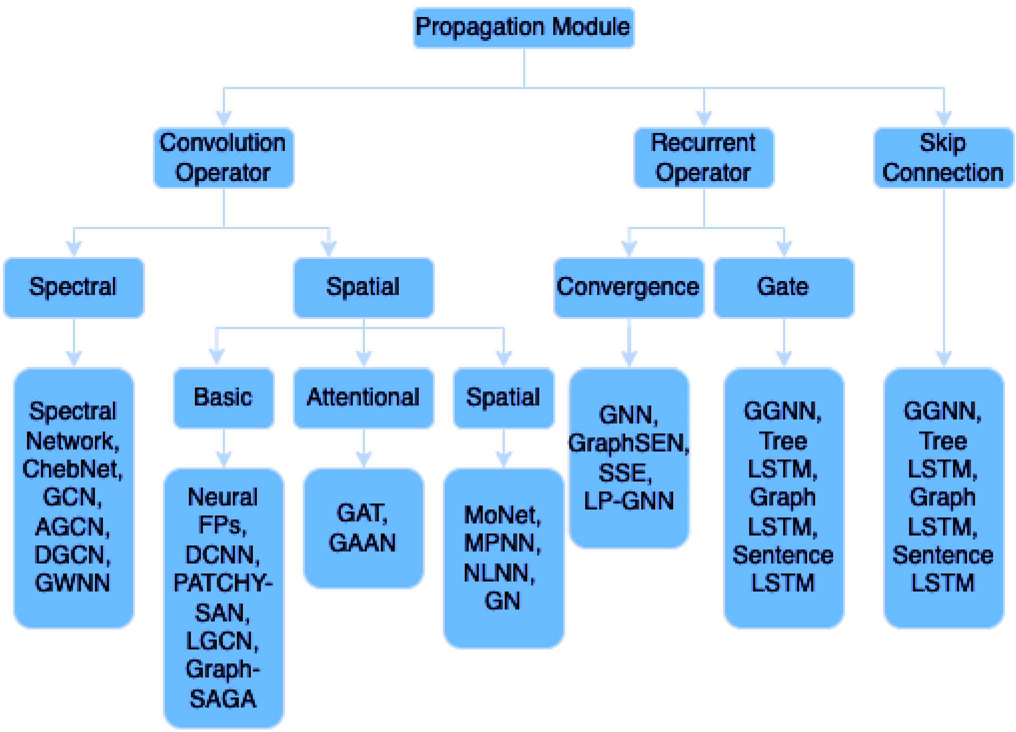


Figure 2.3: Graph Computational Modules

Encoder and decoder functions are used for the embedding space and node to find the similarity. And the main goal is to minimize the loss function from the encoder to the decoder. For that, we introduce many approaches to different encoding methods, and these operations are called models.

2.1.3 Graph Labeling

This technique, known as graph-factorization, uses lower dimensionality matrix factorization methods to divide the user-item interaction matrix into the product of two rectangular matrices. [27].

$$\hat{R} = UV^T \tag{3}$$

Here, U represents the lower dimension space of the user, and V represents items of an utility matrix. If we decompose U , V then equation 3 is the result of it. Where there are hidden characteristics that directly influence the interaction, they have been captured by the latent features, like the different types of features in that matrix.

Table 2.2: Basic Notations of Matrix Factorization.

Notations (s)	**Descriptions**
R	Prediction rating (original item matrix)
U	User matrix (row corresponds to user, column corresponds to latent feature)
V	Item matrix (row corresponds to item, column corresponds to latent feature)

2.2 Literature Review

Over the past decades, many graph representational learning models have been proposed. The graph is a very complex representation that we can include as much as data. But the representation is difficult to analyze or understand. However, it is challenging to analyze or comprehend the representation. The graph embedding space can help with this by making the model simpler with a lower dimension and a less complicated representation. Graph embedding models are mainly divided into five main groups: graph kernels, matrix factorization, shallow, deep neural networks, and non-Eurlidean. The last four years have seen a rise in the number of scholarly publications containing the seven keywords graph representation learning (GRL), graph autoencoder (GAE), graph transformer (GT), graph convolutional networks (GCNs), and non-Euclidean graph embedding (NEGE) [26]. (Figure 2.4) has a structural model of graphs and presents the representation of embedding models that work for various purposes.

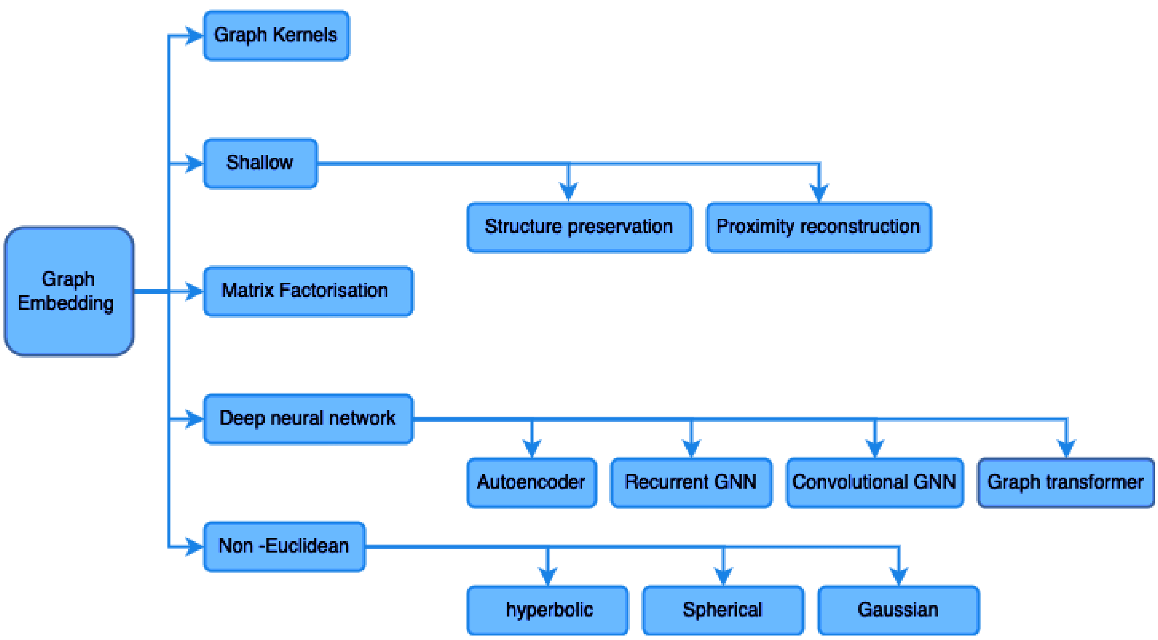


Figure 2.4:: Graph representation learning model

The main concept is to find out the euclidean distance where we can later find similarity between the embedded space and the decoded space and minimize the loss functions.

2.2.1 Convulational Neural Network

Convolution is a unique way of detecting probability from different sets. We can see convolutional in many aspects of math, like multiplying two polonomials together; in image processing, they are used in solving differential equations; and they also play a vital role in probability. The core equation of the convolutional network can be defined by creating a new probability set from two normal sets of information. This new sequence merges the features and can filter the information from the following sequence into a new sequence.

$$(a_i * b_i)_n = \sum_{\substack{i,j \\ i+j=n}} a_i \cdot b_j \tag{4}$$

With a set features, if we name two sets of lists of mixing value of probability to a_i and b_j. We can call the new asterisk * which is called karnal is the new probability set, and we can define the convolutional network of a_i and, b_j which looks like a sum that goes through all the pairs of the different indexes one by one until the sum of these different indexes is equal to n . For example, in image processing, we blur an image by taking a grid of the image, taking the average of the pixels, and creating a sum for the new pixel. With a gaussian distribution, we can get a more accurate value for the convolution[28]. Another example for two long list of numbers we can create a fast method of convolution by fast Fourier transform then multiply them and last an inverse fast Fourier transform. Another example, in signal processing we can use the signal as nodes and the time delay of the signal as the edges and for the dimensions of the signal can have many additional informations which we can label as features. For the convolution the additional information of data positioning and the spacing of the data are both important, which is why it is hard to represent a graph network in a convolutional network.

The convolutional neural network is a combination of the convolutional filter bank represented by G and the pointwise non-linearities represented by x^{l+1} . When we apply a learnable kernel to a set of matrices or pixels, we apply a convolutional filter bank. And from the result of it we apply point linearity to the summation, and repeat the process as we shift over the data.

Let's assume a series of a, b, c, d by order and represent it in a matrix name x. If we perform a time shift and name it x' and the sequence will be d, a, b, c. So we can call show this time shift mathematically by using a matrix with the following format, which we can multiply S by x to get the same result as the x'.

$$a \to b \to c \to d \qquad\qquad d \to a \to b \to c$$

$$x = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \qquad\qquad x' = \begin{bmatrix} d \\ a \\ b \\ c \end{bmatrix}$$

Equation (5) : time shift

If we repeat the time shift and define it by x'' we can get the result by multiplying the exact matrix again, which will look like $S(Sx)$.

$$x' = \begin{bmatrix} d \\ a \\ b \\ c \end{bmatrix} \quad S = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad\qquad S(Sx) = x'' = \begin{bmatrix} d \\ a \\ b \\ c \end{bmatrix}$$

Equation (6): time shift, second time

Therefore, for the time shift, if we shift the matrix n times, we get something like $S^n x$. If we look at it from the graph point of view, this S matrix can also be viewed as an adjacency matrix of graphs and the series as nodes. In the above time shift of second time the first row of S represent the d, the second represents a, and so on, where x' the nodes are and S is the adjacency matrix of that node. We can use this shift matrix S as a convolution to transform it to graph convolution. So we can see this is a weighted shift that works with the kernels. In the graph, we can say the convolution is a multiplying the weighted shift with S. We can represent it by the same S but for clear view we name it W and for the k number of shifts this the equation of graph convolutional filter bank.

$$G = \sum_{k=0}^{K} w_k S^k$$

Equation (7): graph convolutional filter bank

So by adding this graph, convolutional filter bank and pointwise nonlinearities, we can get convolutional neural network. This can be done in complex graphs, like bidirectional graphs, too. The more the shift, the more CNN will represent locally from global. Which is why the CNN initially shifts less and has less global effect because the first few shifts just affect the nearest nodes and edges and reconstructing them, and they will not interchange within each other. However, the more operations are added, the more shifts happen, and the neighbors will influence each other and convolution will take effect globally. In the above example, we first shift the x then, $S(Sx)$ which we call x' then, x'' and so on. Every shift is operating on the previous shift on the kernel and creating more layers than before.

$$x^{l+1} = \sigma(Gx^{l} + b)$$

Equation (7): pointwise non linearities

We can understand more clearly with the besic notations of above equations with some clear descriptions.

Table 2.3: Basic Notations of Graph Convulational Network.

Notations (s)	Descriptions
x^{l+1}	Output of the next layer
x^{l}	Input to the current layer
G	Graph filter bank that transforms x^{l}
b	Bias term that adjusts the transformation
σ	Nonlinear function

If we assume a layer and call it, l then the original graph will be, x^{l} and we apply the graph convolution filter bank to it, and the result of it will go through the nonlinear function. For CNN this repetition depends on the hidden layer it has [29].

According to recent research, comparing graphs on a larger scale can enhance how well each graph is represented using a latent variable model and then explicitly integrated into feature spaces, much like graphical model inference.

2.3 Gap Analysis

Convolution is using shearing parameter and its translation invariant, which is a challenge for the graph data. In generalizing convolution for graphs, the neighborhood nodes changes, the distance between nodes are important in the graph too, as well as the additional features of the edges which change as well, The heterogeneous graph comes with different type of information, graph can also have Homorphism problem with changes in the ordering of the node. For the convolutional neural network we have the challenge of convolutional filter bank, The regular graph $x^{l+1} = \sigma(Gx^{l} + b)$ data can be pointwise non-linear too because it just introduces the nonlinearity to a function which we put the result in. The convolutional filter bank is not suitable for irregular structure graph data, arbitrary size graph data, non euclidean spacing, and doesn't have a fixed order.

Graph-based semi-supervised learning defines the loss function as a weighted sum of the supervised loss over labeled instances and a graph Laplacian regularization term [30], [31], [32], [33]. For the encoding and decoding methods, simplification is the goal, but there is no parramatta's shearing; it's expensive and exaggerated because of how big the data can be in a graph representation. In the graph structure, the previous module always faced challenges for a lack of semantic information, intrigating featurs in the node has always been difficult for the incoder. Furthermore, for being non inductive data, predicting is hard because the data is not readable.

Previously, many recommendation systems have been applied with matrix factorization. The main disadvantage is that, although the accuracy of the training errors is high, the system's performance gradually deteriorates as the RMSE (root mean square error) rate rises [34].

Lastly, GNN architectures have also been previously constructed to mimic the individual steps of specific types of graph algorithms, but with a different context and motive than our work [35].

Table 2.4: Basic Algorithm and their accuracy

Algorithm	**papar**	**value**
link list	Link List with graph Neural Network	0.771
CBCC	Case-Based Collective Classification	0.754
RDN	Relational Ensemble Classification	0.75
RPT	Learning Relational Probability Trees	0.72

2.4 Summary

The shearing parameter and associated translation invariant are used in convolution, which presents a problem for graph data. The neighborhood nodes change when generalizing convolution for graphs, and the distance between nodes is significant as well. Other properties of the edges also change. In addition to having a variety of information types, a heterogeneous graph may also have a homomorphism issue when the node ordering changes. The convolutional filter bank is a problem for the convolutional neural network. Because it only adds nonlinearity to the function that we enter the result in, ordinary graph data can also be pointwise non-linear. The convolutional filter bank is not appropriate for graph data with irregular structures.

Chapter 3

Research Methodology

3.1 Methodology Analysis

This section will outline the tools utilized, the process for gathering and analyzing data, as well as the suggested model. We use the State of the art scheme method of Graph Convolutional Network for our proposed work. We applied Graph Neural Network (GNN) for this classification. This proposed model classifies a beautiful graph called cora with seven different classes on 2708 scientific publications.Anyone can find the dataset in called cora in the website named relational-data where in the dataset thay have cora visualisation.

3.1.1 Overview

We can find the dataset overview on the paper represent by sen. Prithvirj of this article of collective classification. And we chose the tensorflow library to execute the GNN of link listing on this dataset.

3.1.2 Loss Function

We must forecast both the positive and negative edges. For that we need to use binary classification of teh loss which is a cross entropy. So when the edges will give positive samples and negative samples where one will have edges and one will not. With this training mathod we can find the database classifications accuracy and calculate the big data edges link data success rate.

3.1.3 Data Collections

There are 2708 scientific publications in this collection, which are divided into seven types. The citation network consists of 5429 links. A 0/1-valued word vector that indicates the existence or absence of the appropriate dictionary word characterizes each publication in the dataset. The dictionary consists of 1433 unique words.

3.1.4 Functional and Nonfunctional Requirements

The functional requirements are the library, dataset, modules with the algorithm of the mathematics kernel that we call to execute and the nonfunctional requirements are the documantations of the modules and the libraries working mathods.

3.1.5 UI Design

In this project we focusd on the data representation so we choses google colab user interface and represent the data in a simple superuser way. In future aspact we can create a gnn dictionary or searching mathod user interface integrating api and front-end applications with a mvc framework to create it to a public services. We can also use it as a scss program for daily internet uses and also can use graph ML to visualise the dataset.

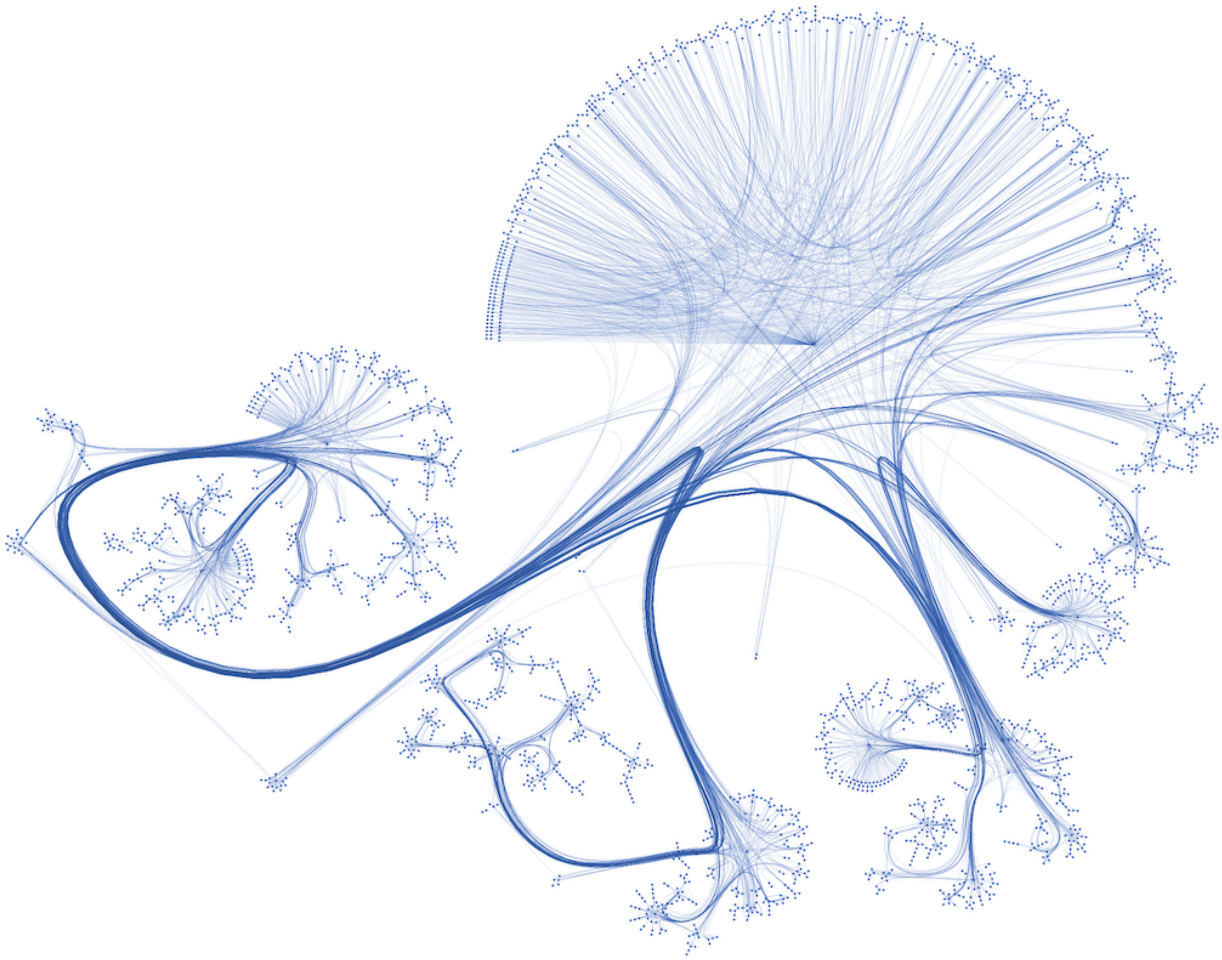


Figure 3.1 :Representation of the cora

3.1.6 Data Flow Diagram Level 1

The data we shown is figure 3.1 combined of all relations from the citations and papers classes links. The longer dots is representing a good relations and the shorts one representing very low relations. Where the outer side is quite densed cause of the second lebeling of data holding for finding out the neares edges in this case not fully connected but related in many way where the middle one shows less dense cause of low connections between them.

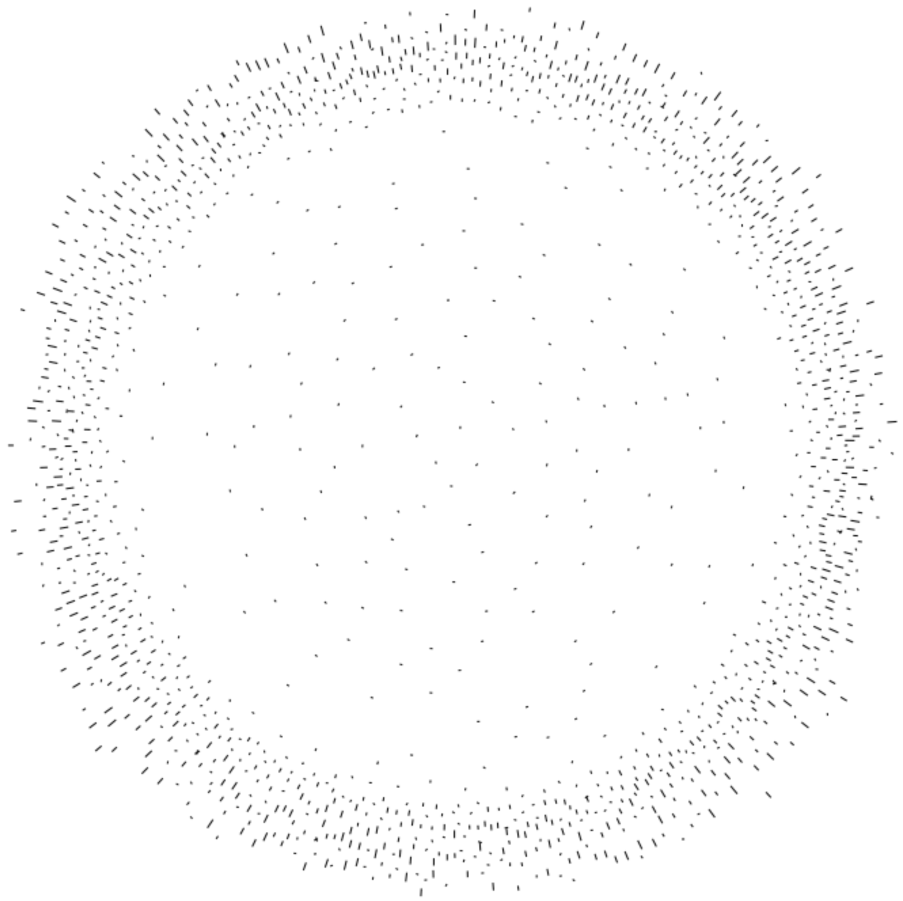


Figure 3.2 :Representation of the dataset

3.2 Detailed Methodology

The 2708 publications in the CORA dataset are arranged into seven classes according to their topic matter. These publications' citations are connected to one another in such a way as to form a network. Consequently, 5429 connections are made by the network of citations. Predicting the paper's class from the seven classes in which it should be included was the rationale behind this study. We also attempted to solve the problem of knowing the entire network tree of a given document with link prediction if it references other papers and those other papers continue to be linked with citations. Graph Neural Networking allowed us to fulfill the purpose of linking one paper with another to get the true network of a paper.It's possible for some people to believe that papers based on theory will probably cite other theories, or that papers based on neural networks would probably cite other neural networks. However, in practice, citation may not operate in this manner. Papers of one kind may also contain classes of other sorts of papers or possible related topics. Thus, by training and testing the datasets using GCN, we could determine the relevance of the paper with the correlated different publications if we could construct a graph based on how the papers are connected to each other.

3.3 Project Plan

This deep learning project is the future of searching optimization. If we want to work with pre-processsed database like cora we need to create a model that present and can handle bigdata with minimum losses. this sellergraph can handle all of this requirement with its pre train stage. Next we can compose the data and catagorized it in our visible state and shift it to the convolution and create a relation out of the binary version data that visualized the relition and showcases its additional information within it.

3.4 Task Allocation

To categorize the papers according to the seven classifications, Stellar Graph was utilized to apply GCN with epochs on the CORA dataset. Stellar Graph provides graph analytics software for machine learning on neural networks. In order to determine if a paper is related to another part or not, we are correlating the contents of 2708 papers with 1435 parts. Based on subjects and ideas, we can also obtain an overview of all the articles based on how closely they relate to each other. After getting the accuracy on training and validation, we can compare the predicted classification with the true classification of the paper to know about the performance of the research along with PCA and t-SNE. To decrease data, PCA makes advantage of the global covariance matrix. With the same outcome, we can obtain the matrix and apply it to a fresh batch of data. This is useful when we need to try to employ a smaller feature list and a matrix made using train data. The main use of t-SNE is the understanding of high-dimensional data and its projection into low-dimensional space, such as 2D or 3D. For working with neural networks, this makes it quite helpful.

3.5 Summary

This study aims to predict a paper's class from the 2708 publications in the CORA dataset, which are arranged into seven classes based on topic matter. The researchers used Graph Neural Networking to construct a graph based on how the papers are connected to each other, allowing them to determine the relevance of the paper with the correlated different publications. This approach helps solve the problem of knowing the entire network tree of a given document.

Chapter 4

Implementation and Results

4.1 Environment Setup

The Cora dataset is a citation network where the nodes represent paper and the edges represent citations. where each node has some features that represent the content of the paper. The dataset contains 2708 rows of citations and 1433 columns of dictionary words. We first sort and switch the data vertically and horizontally as 2708 columns and 1433 rows for better visibility. So this dataset got many junk values, and the columns got many spaces in between them, so we sorted it as paper ID and summed it by the subjects. Every column word represents the true or null value by 0 and 1; if the dictionary word exists on the paper, then the value of the column is 1; if not, then 0; and lastly, that paper value determines the subject of the paper. So the 2708 citations have an edge list connection of 5429 and are very likely to be connected to each other. We can get value out of this, where we can find the 7 class connections.

Table 4.1: Number of Classes and Citations.

Subject (s)	**Amount**
Neural_Networks	818
Probabilistic_Methods	426
Genetic_Algorithms	418
Theory	351
Case_Based	298

Reinforcement_Learning	217
Rule_Learning	180

Then we copy the data and convert it to an integer value by lambda. We construct the dataset into graph format for processing the data and this conversion is suitable for GNN to process. We use a graph convolution network (GCN) architecture into some layer within which the input processes the vectors of nodes and has some hidden layer that transforms and aggregates the nodes shifting features on their neighbors. In the output, we predict the probability of the edges of the link between node pairs. We use binary cross-entropy to measure the loss function and the difference between the prediction and the actual links. We optimized the model parameter to minimize loss with the Adam optimizer, trained the model with 100 epochs, and refined the model's link prediction. We use accuracy and loss metrics to evaluate and validate the unseen data. Usually we can put it in PCL to reduce the dimensionality, which is used to find out the higher dimension dataset patterns and some relations of the principal component, but in this case, we use TCNE dimensionality reduction on the dataset to plot it in the second dimension and compare its similarity and relations [36].

4.2 Testing and Evaluation

In code, we use a stallergraph train with Graph Neural Network which is used for transfer learning. We use Tensorflow, and within Tensorflow we use Keras. We represent it on Matplotlib. Then we convert the string value to an integer and sort the data in the range of 1433. Then, we preprocess the categorized label data into an array. Stallergraph assembles the node classification of graph convolution networks to predict. We could use Reinstead of Softmax. Lastly, with learning recall at 0.01, we binarized the string and ran epochs to train and test representations with node prediction visuality.

4.3 Results and Discussion

The graph neural network achieved high performance in predicting the link between nodes in the dataset's citation network. As we can say, this was processed by the GCN model, which fetches the graph structure and feature connections from the Cora dataset, in the citation and the binarize nodes. The accuracy of the GNN is 0.7771. This can indicate that the model can effectively identify the link. Also, the loss decreased over 100 epochs, and it eventually balanced at around 0.9994. So the model needs to minimize the difference between its predictions and the link in the paper, which it did very successfully.

For the dataset, we first get the visualization of the paper or node connection to the dictionary words, and the different colors represent the 7 classes of the subjects. Figure 5 has 7 colors, and in the middle, the distances in between them are higher on the edge. It indicates that the dataset papers that connect with their linked papers are close to each other. Another characteristic of the Figuer 5 is its representation of the citation value by its size. We divided the total value and gave 140 to train and 500 to test.

Table 4.2: Number of Classes and train value.

Subject (s)	**Amount**
Neural_Networks	42
Probabilistic_Methods	22
Genetic_Algorithms	22
Theory	18
Case_Based	16
Reinforcement_Learning	11
Rule_Learning	9

We had some challenges over the imbalance with the non-existent links in the cora link of the paper citations, and we can have better results by making some augmentations to the data in the future. Also, if we read the epoch by every 10 layers we can see the validatation accuracy slowly incresing. We also can see the validation losses difference. And we can compaire the true accuray and losses with the validation.

Table 4.3: loss and accuracy

No. (s)	Loss	Accuracy	validate Loss	validate Accuracy
001	1.9725	0.0714	1.9176	0.3440
010	1.3363	0.5143	1.4241	0.4700
020	0.5717	0.8143	0.9042	0.7360
030	0.2274	0.9357	0.8043	0.7740
040	0.1302	0.9643	0.9307	07560
050	0.0642	0.9857	1.0067	0.7780
060	0.0376	0.9857	1.1546	0.7560
070	0.0564	0.9786	1.1829	0.7660
080	0.0511	0.9786	1.1930	0.7560

090	0.0193	0.9929	1.2480	0.7540
100	0.0171	1.000	1.2748	0.7560

We can run more epoch in the future to get even grater value with in this graph. But first let us show the train and test graph difference by a visualization.

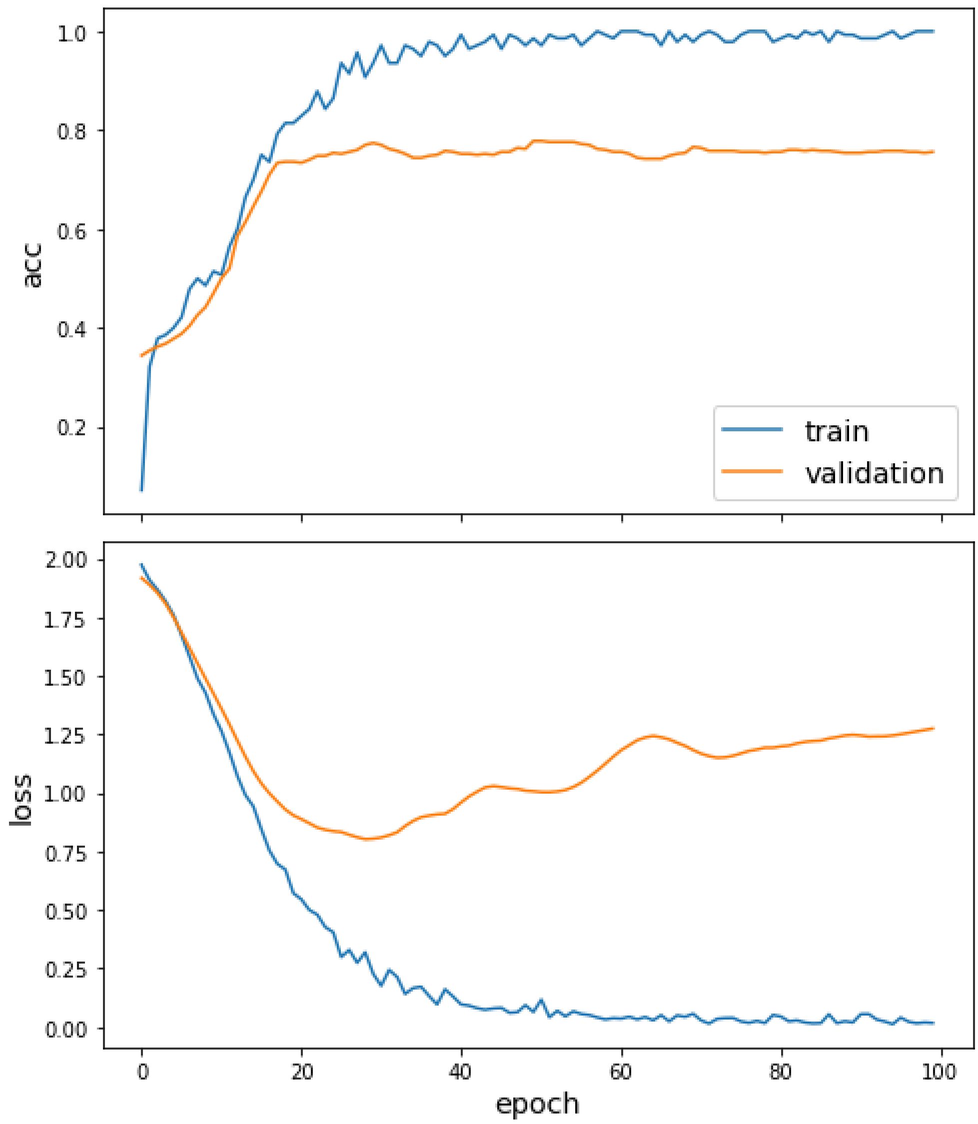


Figure 4.1 :Representation of the train and Validation

From this epoch and true value prediction, we got a visualization like above. We then run it through principal component analysis using the TSNE package of the sklearn library, as shown in Figure 6, and find the relations of the validataions from true value in the second dimension.

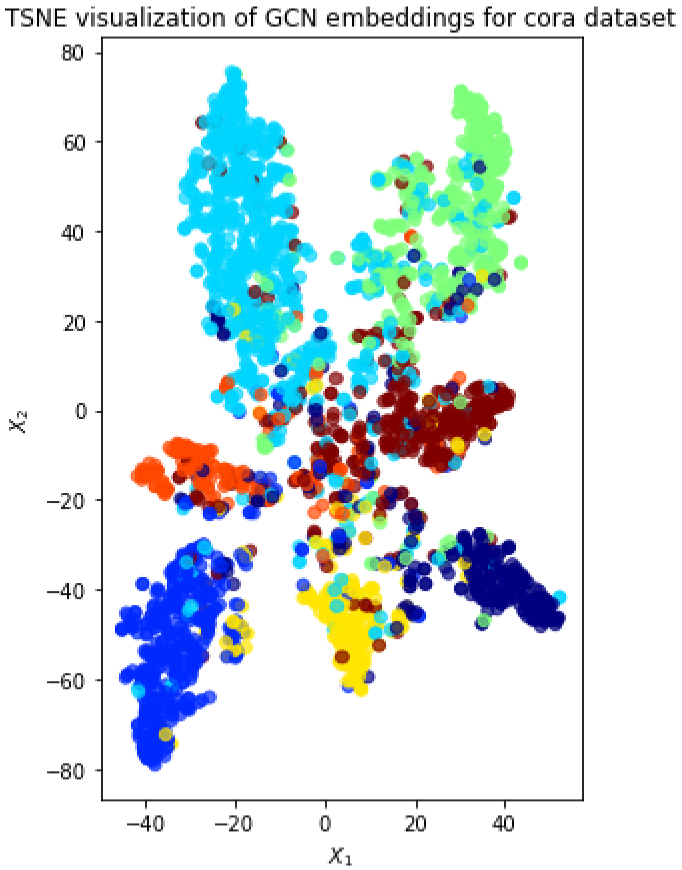


Figure 4.2 :Representation of the predection

We can see that the tsne benefit of visualization and understanding dataset. We had created seven classes of visualization and the seven type of demonstration where ench class had gone through graph embeddings, so the main dataset of cora can have this tested and trained and can be represented with some specific data visualization.

4.4 Summary

Using the dataset, we first obtain a visual representation of the paper or node relationship to the dictionary terms, with the seven subject classes represented by the various colors. The distances between the seven colors in Figure 5 are greater on the edge than in the center. It shows how near together the dataset publications are that link to their connected papers. The Figuer 5's display of the citation value by size is another featu

Chapter 5

Engineering Standards

5.1 Compliance with the Standards

The main compliance problem is the standed regulations of nodes and the relations of edges. We face problem with the node and edge features too like if its a score or metadata it need to be formatted specifically with in a type and its collection gets complicated.

5.1.1 Software Standards

For software we use google coleb the stander way is to install anaconda or matlab to create a offline machine that can have a own eco system to go more effectively.

5.1.2 Hardware Standards

Hardware standards depends of how time consuming the program have to be. the minimum configuration can be a device with a GPU on it, that runs any os and python and compile g++ and gcc.

5.2 Impact on Society, Environment and Sustainability

Socal impact of this model can be wide. If we exiute a problem by the link prediction the commuting power can process random thought. Already image generator and Ai search engine can go through this shifting with the gient company like meta, google, amazon big data and its semi- supervise data can be a big step ahead for the socal and environmental platphorm.

5.2.1 Impact on Life

We dont fully yet can integrated with the commuting system because of the random choice of the system but with this model one can correlated and have a train model implement and visualized big data and solve many problems.

5.2.2 Impact on Society & Environment

As earlier connection in society and environment is the kye aspact of this models this can connect a bride between socal or environmental data to a deep learing stage and its every possibility can be reladed through linking.

5.2.3 Ethical Aspects

According to recent research, ethical challanges is more importent when it comes to ai earand many probabilistic automations planing field. So we mesure a block box that contain model and create it within a environment as so when ever its a priority anyone can shut it down.

5.2.4 Sustainability Plan

The field is open source with many frmwork and one can never know what to explore. So it is reassuring that the deep learning, neural network lisk list contain a vision for commercial uses

5.3 Complex Engineering Problem

5.3.1 Complex Problem Solving

In this section, provide a mapping with problem solving categories.. For P1, you need to put another mapping with Knowledge profile and rational thereof.

5.3.2 Engineering Activities

In this section, provide a mapping with engineering activities. For each mapping add subsections to put rationale (Use Table 5.3).

5.4 Summary

One must need depth of knowledge and analysis to obtain this work process.The Engineering Fundementals are a field for all. For this specific project one must need specialist knowledge on python and one must practice that knowledge to obtain engineering skills.

One must need range of resources to obtain the data and this project will help one to find.

Chapter 6

Conclusion

6.1 Summary

The GNN for link prediction in Cora datasets with complex graph structure data can be effectively studied, and we can predict the relationships. The prediction link with high accuracy and balance losses is a success, and we can say that it is well suited. The domain of it is vast, and we can perform various tasks and analyze or predict. The application itself is for complex structured graph representation and can be performed in many fields if we perform the task in the form of a graph. What, visibly, is a regular process? Mathematically, we can understand it in a diverse and more dynamic way. If we take semi supervised data and refine it to put it in some layer of GNN, we can validate and see the prediction accuracy in a dimentional way. By using the representation of the graph, we can analyze the complex structure of big data and find the link between the agencies of the nodes.

6.2 Limitation

The main limitations in this tech era is the power house. the bit data holds many many layer of convolution the more it will holds gpu the more it will gain its efficiency. The issues are not only quantity its the price of its qualitifull environment too The lebelizer holds the computing power that cost way more for a testing purpose. Moreover the limitations of investment is the main reason this field is cant went it its peak.

For the Convulations the Gnn can not perform in its early stage so small data relations will look for its neibor and the will look for there and so on. its then comes to a point when a veriious count of shifting has occurs then the gnn can take form. the image processing is stil not too advanc canse of the equipment. a gnn or cnn can update the image to its original form but can never occur the system image accuracy to break and not loss a bt, every signal cant identify but most of the model has limitations and will be implemanted and more epoch will perform more acciracy that gnn accuray rises to chances of not having a data loss.

6.3 Future Work

We have found many sophisticated GNN models, like those we discuss in the related works, in models where graph attention networks (GTA), graph isomorphism networks (GIN) can offer a deeper performance and can be more intricate. We will test this model on many datasets, including CiteSeer, WebKB, and many more complex datasets. Also, we need hyperparameter tuning, which is an extensive optimization that can fine tune the model even better, and the accuracy and losses will be more stabilized.
However, the imbalanced dataset with its resources and computational advances is challenging. Also, the GNN interpretability of the model predictions needs to be enhanced. In the future, we can run epoch 200, the more the shift layer, the more we can get accurate predictions.

In the field of search program we can highlight our work and comparing it with other model can create a scope to the new field of gnn performance. For true perphorment we need to run more epoch.

References

[1] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, 'Gated Graph Sequence Neural Networks', Sep. 22, 2017, *arXiv*: arXiv:1511.05493. doi: 10.48550/arXiv.1511.05493.

[2] T. N. Kipf and M. Welling, 'Semi-Supervised Classification with Graph Convolutional Networks', Feb. 22, 2017, *arXiv*: arXiv:1609.02907. doi: 10.48550/arXiv.1609.02907.

[3] M. Gori, G. Monfardini, and F. Scarselli, *A new model for earning in raph domains*, vol. 2. 2005, p. 734 vol. 2. doi: 10.1109/IJCNN.2005.1555942.

[4] F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini, 'The Graph Neural Network Model', *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009, doi: 10.1109/TNN.2008.2005605.

[5] K. Xu, C. Li, Y. Tian, T. Sonobe, K. Kawarabayashi, and S. Jegelka, 'Representation Learning on Graphs with Jumping Knowledge Networks', in *Proceedings of the 35th International Conference on Machine Learning*, PMLR, Jul. 2018, pp. 5453–5462. Accessed: Feb. 08, 2024. [Online]. Available: https://proceedings.mlr.press/v80/xu18c.html

[6] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, 'Neural Message Passing for Quantum Chemistry', in *Proceedings of the 34th International Conference on Machine Learning*, PMLR, Jul. 2017, pp. 1263–1272. Accessed: Feb. 08, 2024. [Online]. Available: https://proceedings.mlr.press/v70/gilmer17a.html

[7] Y. Wu, D. Lian, Y. Xu, L. Wu, and E. Chen, 'Graph Convolutional Networks with Markov Random Field Reasoning for Social Spammer Detection', *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 01, Art. no. 01, Apr. 2020, doi: 10.1609/aaai.v34i01.5455.

[8] P. W. Battaglia, R. Pascanu, M. Lai, D. Rezende, and K. Kavukcuoglu, 'Interaction Networks for Learning about Objects, Relations and Physics', Dec. 01, 2016, *arXiv*: arXiv:1612.00222. doi: 10.48550/arXiv.1612.00222.

[9] A. Sanchez-Gonzalez *et al.*, 'Graph Networks as Learnable Physics Engines for Inference and Control', in *Proceedings of the 35th International Conference on Machine Learning*, PMLR, Jul. 2018, pp. 4470–4479. Accessed: Feb. 09, 2024. [Online]. Available: https://proceedings.mlr.press/v80/sanchez-gonzalez18a.html

[10] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur, 'Protein Interface Prediction using Graph Convolutional Networks', in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: Feb. 09, 2024. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/f507783927f2ec2737ba40afbd17efb5-Abstract.html

[11] F. Xia, J. Liu, H. Nie, Y. Fu, L. Wan, and X. Kong, 'Random Walks: A Review of Algorithms and Applications', *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 2, pp. 95–107, Apr. 2020, doi: 10.1109/TETCI.2019.2952908.

[12] X. Zhu and Z. Ghahramani, 'Learning from Labeled and Unlabeled Data with Label Propagation', Jul. 2003.

[13] M. Belkin and P. Niyogi, 'Laplacian Eigenmaps for Dimensionality Reduction and Data Representation', *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003, doi: 10.1162/089976603321780317.

[14] W. E, C. Ma, and L. Wu, ‘A Comparative Analysis of the Optimization and Generalization Property of Two-layer Neural Network and Random Feature Models Under Gradient Descent Dynamics’, *Sci. China Math.*, vol. 63, no. 7, pp. 1235–1258, Jul. 2020, doi: 10.1007/s11425-019-1628-5.
[15] E. W. Weisstein, ‘Laplacian Matrix’. Accessed: May 17, 2024. [Online]. Available: https://mathworld.wolfram.com/
[16] F. R. K. Chung, ‘Lectures on Spectral Graph Theory’.
[17] ‘Spectral Graph Theory’. Accessed: May 17, 2024. [Online]. Available: https://bookstore.ams.org/cbms-92
[18] W. L. Hamilton, R. Ying, and J. Leskovec, ‘Representation Learning on Graphs: Methods and Applications’, Apr. 10, 2018, *arXiv*: arXiv:1709.05584. doi: 10.48550/arXiv.1709.05584.
[19] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec, ‘Hierarchical Graph Representation Learning with Differentiable Pooling’, Feb. 20, 2019, *arXiv*: arXiv:1806.08804. doi: 10.48550/arXiv.1806.08804.
[20] S. Verma and Z.-L. Zhang, ‘Graph Capsule Convolutional Neural Networks’, Aug. 25, 2018, *arXiv*: arXiv:1805.08090. doi: 10.48550/arXiv.1805.08090.
[21] D. G. T. Barrett, F. Hill, A. Santoro, A. S. Morcos, and T. Lillicrap, ‘Measuring abstract reasoning in neural networks’, Jul. 11, 2018, *arXiv*: arXiv:1807.04225. doi: 10.48550/arXiv.1807.04225.
[22] A. Santoro *et al.*, ‘A simple neural network module for relational reasoning’, Jun. 05, 2017, *arXiv*: arXiv:1706.01427. doi: 10.48550/arXiv.1706.01427.
[23] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, ‘Molecular graph convolutions: moving beyond fingerprints’, *J. Comput. Aided Mol. Des.*, vol. 30, no. 8, pp. 595–608, Aug. 2016, doi: 10.1007/s10822-016-9938-8.
[24] W. L. Hamilton, R. Ying, and J. Leskovec, ‘Inductive Representation Learning on Large Graphs’, Sep. 10, 2018, *arXiv*: arXiv:1706.02216. doi: 10.48550/arXiv.1706.02216.
[25] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, ‘Graph Attention Networks’, Feb. 04, 2018, *arXiv*: arXiv:1710.10903. doi: 10.48550/arXiv.1710.10903.
[26] F. Chen, Y.-C. Wang, B. Wang, and C.-C. J. Kuo, ‘Graph representation learning: a survey’, *APSIPA Trans. Signal Inf. Process.*, vol. 9, no. 1, May 2020, doi: 10.1017/ATSIP.2020.13.
[27] Y. Koren, R. Bell, and C. Volinsky, ‘Matrix Factorization Techniques for Recommender Systems’, *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009, doi: 10.1109/MC.2009.263.
[28] C. Williams, ‘The Gaussian Distribution’.
[29] Y. LeCun, Y. Bengio, and G. Hinton, ‘Deep learning’, *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
[30] M. Belkin, P. Niyogi, and V. Sindhwani, ‘Manifold regularization: A geometric framework for learning from labeled and unlabeled examples.’, *J. Mach. Learn. Res.*, vol. 7, no. 11, 2006, Accessed: Feb. 03, 2024. [Online]. Available: https://www.jmlr.org/papers/volume7/belkin06a/belkin06a.pdf
[31] J. Weston, F. Ratle, and R. Collobert, ‘Deep learning via semi-supervised embedding’, in *Proceedings of the 25th international conference on Machine learning - ICML ’08*, Helsinki, Finland: ACM Press, 2008, pp. 1168–1175. doi: 10.1145/1390156.1390303.
[32] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Olkopf, ‘Learning with Local and Global Consistency’, *Adv. Neural Inf. Process. Syst. 16*, vol. 16, Mar. 2004.
[33] X. Zhu, Z. Ghahramani, and J. Lafferty, ‘Semi-supervised learning using Gaussian fields and harmonic functions’, in *Proceedings of the Twentieth International Conference*

on International Conference on Machine Learning, in ICML'03. Washington, DC, USA: AAAI Press, Aug. 2003, pp. 912–919.
[34] 'Netflix Update: Try This at Home'. Accessed: Jun. 08, 2024. [Online]. Available: https://sifter.org/~simon/journal/20061211.html
[35] P. Veličković, R. Ying, M. Padovano, R. Hadsell, and C. Blundell, 'Neural Execution of Graph Algorithms', Jan. 15, 2020, *arXiv*: arXiv:1910.10593. doi: 10.48550/arXiv.1910.10593.
[36] 'Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition [Book]'. Accessed: Jul. 08, 2024. [Online]. Available: https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

Link List Prediction using Graph Neural Network

ORIGINALITY REPORT

7%	4%	3%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	2%
2	Submitted to United International University Student Paper	1%
3	stellargraph.readthedocs.io Internet Source	1%
4	www.mdpi.com Internet Source	<1%
5	www.lti.cs.cmu.edu Internet Source	<1%
6	Stéphane Peters, Yann Jacob, Ludovic Denoyer, Patrick Gallinari. "Iterative Multi-label Multi-relational Classification Algorithm for complex social networks", Social Network Analysis and Mining, 2011 Publication	<1%
7	Submitted to University of New Haven Student Paper	<1%
8	dspace.uiu.ac.bd Internet Source	

<1%

9 Meihui Zhong, Mingwei Lin, Chao Zhang, Zeshui Xu. "A Survey on Graph Neural Networks for Intrusion Detection Systems: Methods, Trends and Challenges", Computers & Security, 2024
Publication
<1%

10 www.ros-test.hw.ac.uk
Internet Source
<1%

11 Nam Do-Hoang Le, Thai-Son Tran, Minh-Triet Tran. "Exploring Neighborhood Influence in Text Classification", 2012 Fourth International Conference on Knowledge and Systems Engineering, 2012
Publication
<1%

12 Submitted to The University of Memphis
Student Paper
<1%

13 Abdolrahman Dadvand, Boo Cheong Khoo, Mohammad T. Shervani-Tabar, Saman Khalilpourazary. "Boundary element analysis of the droplet dynamics induced by spark-generated bubble", Engineering Analysis with Boundary Elements, 2012
Publication
<1%

14 ethesis.nitrkl.ac.in
Internet Source
<1%

15	file-thesis.pide.org.pk Internet Source	<1 %
16	Van Thuy Hoang, Hyeon-Ju Jeon, Eun-Soon You, Yoewon Yoon, Sungyeop Jung, O-Joun Lee. "Graph Representation Learning and Its Applications: A Survey", Sensors, 2023 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		