MLGA MINI PROJECT

TITLE: FAMILIAR STRANGERS

UE21CS343BB4 – MLGA MINI PROJECT REPORT

Submitted by:

Name: Amulya Pothumarthi PES2UG21CS062
Name: Ananya Singla PES2UG21CS066
Name: C H Sri Charan PES2UG21CS129

Under the guidance of
Dr. Arti Arya
Professor
PES University

January - May 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

Table of Contents

ABSTRACT

In a university setting, students often encounter familiar faces regularly throughout their academic journey, yet they may not be aware of each other's diverse skills and interests. Imagine a university library where a diligent student, Alex, frequents the same section to study for hours. Daily, another student, Taylor, also visits the library and sits nearby. Both share a passion for literature and spend countless hours immersed in their studies, yet they have never exchanged a single word. They remain familiar strangers, connected by their common interests but unaware of each other's presence.This lack of awareness hinders opportunities for collaboration, knowledge sharing, and interdisciplinary learning among students who could benefit from each other's expertise. The goal of this project is to leverage social network analysis techniques to uncover hidden patterns of shared skills and interests among students who are familiar strangers, ultimately facilitating meaningful connections and collaboration within the university community.

Summary of Literature Survey

Paper Details	Objective of the paper, techniques/methods	Advantages	Limitations
Knowledge Sharing Through Academic Social Networking: The Impact of Personal and Social Outcome Expectations	The paper investigates the use of **academic social networks** as tools for knowledge sharing among students. The participants consisted of 245 students, male, and female of different age groups, from different colleges and different academic years, studying for bachelor's degrees, diploma degrees, and Master's degrees at Al Ain University, UAE. Finds out the impact of **personal outcome expectations (POEs)** and **social outcome expectations (SOEs)** on knowledge sharing among students. The empirical evidence showed that POEs and SOEs can be the major motivators in motivating students to share knowledge through academic social networks. The Social Cognitive Theory suggests that **social expectations and social networks** influence the behavior of individuals	Knowledge sharing can increase productivity, contribute to problem-solving, and help students to improve academic performance. It plays a vital role in collaborative learning. It is perceived to impact the community's well-being. The findings of the study revealed that POEs had the strongest impact on knowledge-sharing willingness and behavior than all other investigated factors. The results of the study show that knowledge sharing gives them a sense of accomplishment, helps them build a good reputation, and strengthens ties with other members.	Increasing or decreasing the relationship between the independent variables POE and SOE and the dependent variable (KS)
Similarity-based link prediction in social networks using latent relationships between the users	Direct-indirect common neighbours Impact of common second order neighbours-latent relationships Focuses on structural similarity Method:Create neighbourhood vector for each node,find pearson correlation coeff between the nodes;the final similarity score would be (1+CNij)*(1+Corrij)	Helps identify similarity between nodes when no common neighbours are present. Better accuracy than Common neighbours, Jaccard coefficient,Preferential Attachment Index (di*dj),hub promoted index	DICN does not have high accuracy for small networks **Temporal content based similarity** not considered ie nodes with higher similarity with respect to topics of **interest over time**. As such, those users who share not only similar topical interests but also share similar

			temporal behavior are considered to be like-minded and hence members of the same community.

Temporal Graph Neural Networks for Social Recommendation	Proposes a propose a novel Temporal Enhanced Graph Model for Social Recommendation - TGRec Takes into account three factors: (1) a user's basic preference of items, (2) the collaborative influence of peers, (3) the temporal impact of previous items bought by the user. Constructs a temporal graph with three heterogeneous relations: a user's basic preference of items (*i.e.,* user-item relation), the collaborative influence of peers (*i.e.,* user-user relation) and the temporal dependence relation of items (*i.e.,* item-item relation).	Captures users' social influence and interaction information with hierarchical attention mechanism, and utilizes the temporal interval information between items in the purchase sequence. Higher capability to handle the data sparsity problem. Performs the best compared with all the baseline methods	Does not utilise user opinion like ratings,which limits the power of model
A Graph-Neural-Network-Based Social Network Recommendation Algorithm Using High-Order Neighbor Information Yonghong Yu 1,* , Weiwen Qian 1 , Li Zhang 2 and Rong Gao 3 (2022)	The objective of the paper is to propose a novel Graph Neural Network (GNN)-based social recommendation model that aims to improve recommendation performance by capturing high-order collaborative signals in the process of learning the latent representations of users and items. Other methods such as SR-GNN, NGCF, LightGCN, GraphRec, and DiffNet, were utilized with Graph Neural Networks to capture high-order collaborative signals and improve recommendation performance.	It proposed model outperforms traditional algorithms and is sensitive to the number of propagation layers and the integration of social network information. The lightweight GNN framework eases the training process of the proposed GNN-based social recommendation model and alleviates the problem of overfitting.	The impact of the number of propagation layers and the embedding dimension on the recommendation quality indicates that the model's performance is sensitive to these parameters. The proposed method only considers the local structure of the social network and ignores the global structure when assigning weights to trusted users.

Paper Details	Objective of the paper, techniques/methods	Advantages	Limitations
A Deep Graph Neural Network-Based Mechanism for Social Recommendations Zhiwei Guo and Heng Wang (2021)	To develop a deep GNN-SoR that can effectively predict unknown preference ratings in the user-item rating matrix by considering correlations among user preferences, social relationships, and item features. The encoded user and item feature spaces are viewed as two latent factors in the process of matrix factorization. This matrix factorization is formulated to predict unknown preference ratings in the user-item rating matrix	By utilizing the graph neural network method, the paper achieves a better representation of complex relationships within user and item feature spaces. The paper conducts a large number of experiments on three real-world data sets to evaluate the efficiency and stability of the GNN-SoR framework.	The evaluation of the GNN-SoR framework is based on experiments conducted on three specific real-world data sets. As the size and complexity of data increase, the performance and computational efficiency of the framework may be impacted.
Heterogeneous Hypergraph Neural Network for Social Recommendation using Atention Network BILAL KHAN, JIA WU, JIAN YANG, XIAOXIAO MA (2023)	Propose a novel model, HHGSA, that leverages heterogeneous hypergraphs and attention mechanisms to enhance social recommendation systems. The paper constructs heterogeneous hypergraphs that unify user-item bipartite graphs and social networks to accurately represent higher-order relationships between users and items. The paper utilizes GNNs as a backbone for social recommendation to aggregate user embeddings, including information about friends, strangers, and item embeddings.	HHGSA captures higher-order relationships among users, items, friends, and strangers. This allows the model to represent complex interactions more effectively, leading to better recommendation accuracy	The use of hypergraphs and attention mechanisms in the HHGSA model may introduce additional complexity to the recommendation system, impacting scalability and computational efficiency. Fine-tuning these hyperparameters for optimal performance can be a challenging and time-consuming task
A Survey of GNN-based graph similarity learning	Review GNN-based graph similarity learning methods and categorize them. Analyze GNN-CNN hybrid models, Siamese GNN, and GNN-based matching networks.	Graph similarity learning enhances classification, clustering, and subgraph matching tasks. GNN models improve graph representation vectors for comparing graph similarities.	Challenges include over-smoothing, scalability issues, and interpretability concerns.

Paper Details	Objective of the paper, techniques/methods	Advantages	Limitations
Characterizing the communication requirements of GNN accelerators	Characterizing data movement in GNN accelerators using analytical models. Analyzing impact of dataflows on communication requirements in GNN accelerators.	Analytical models for GNN accelerators help compare different architectures. Models capture dataflows and hardware setups, exposing scalability characteristics. Comparative analysis of various GNN accelerators is facilitated.	Aggregation accounts for a large fraction of data movement. Two architectures scale differently. Memory limitations increase total iterations, affecting latency.
Network modeling based on GNN and network behaviors	Objective: Propose Link Delay Model (LDM) based on GNN. Method: Use improved GNN to learn relationship between network behaviors.	Accurate prediction of end-to-end delay with LDM. Improved generalization ability compared to Queuing model and RouteNet. Reduced overhead by 78% with partial common links in LDM.	Existing methods lack generalization and are unsuitable for actual networks.Limited generalization ability under unknown flow scheduling strategy.

PES UNIVERSITY

Proposed Approach

1. Data Preprocessing:

- Load the student dataset from a CSV file containing attributes such as StudentID, Name, Skills, etc.
- Preprocess the data as necessary, including handling missing values, encoding categorical variables, and splitting the dataset into training and testing sets.

2. Graph Construction:

- Construct a graph representation of the student collaboration network based on shared skills.
- Each student is represented as a node, and edges between nodes represent shared skills between students.

```python
# Function to add edges based on shared skills
def add_edges_based_on_skills(G):
    for i, student1 in student_data.iterrows():
        G.add_node(student1['StudentID'], skills=student1['Skills'])
        for j, student2 in student_data.iterrows():
            if i != j:
                skills1 = set(student1['Skills'].split(', '))
                skills2 = set(student2['Skills'].split(', '))
                #print(skills1.intersection(skills2))
                common_skills = len(skills1.intersection(skills2))
                if common_skills > 0:
                    G.add_edge(i,j,weight=common_skills)
```

3. Graph Neural Network (GNN) Model:

- Define a Graph Neural Network (GNN) model using PyTorch Geometric.
- The model architecture consists of multiple graph convolutional layers followed by activation functions such as ReLU.
- Train the GNN model using the constructed graph and student features to learn node representations capturing collaboration patterns based on skills.

```
# Convert graph to PyTorch Geometric Data object
edge_index = torch.tensor(list(G.edges()), dtype=torch.long).t().contiguous()
x = torch.tensor([[node[0]] for node in G.nodes(data=True)], dtype=torch.float)
data = Data(x=x, edge_index=edge_index)
# Define the Graph Neural Network model
class GNNModel(nn.Module):
    def __init__(self):
        super(GNNModel, self).__init__()
        self.conv1 = GCNConv(1, 16)
        self.conv2 = GCNConv(16, 1)
        self.dropout = nn.Dropout(p=0.5)  # Dropout layer for regularization to p

    def forward(self, data):
        x, edge_index = data.x, data.edge_index
        x = self.conv1(x, edge_index)
        x = torch.relu(x)
        x = self.conv2(x, edge_index)
        return x
```

4. Training and Evaluation:

- Train the GNN model using the training dataset, optimizing for a suitable loss function such as Mean Squared Error (MSE).
- Evaluate the trained model's performance on a separate validation or test dataset using appropriate evaluation metrics such as accuracy, precision, recall, or F1-score.

```
# Train the model
def train_model(model, data, optimizer, criterion, num_epochs=200):
    model.train()
    for epoch in range(num_epochs):
        optimizer.zero_grad() # resets the gradients to zero
        output = model(data) # forward pass of data to model
        loss = criterion(output, torch.ones_like(output))  # Dummy target # MSE Loss
        loss.backward() # computes gradients of the loss using backpropogation
        optimizer.step() # optimizer updates the model parameters based on the gradients
```

5. Collaboration Partner Recommendation:

- Utilize the trained GNN model to predict collaboration partners for a given student based on their skills.
- Set a threshold to filter out predicted collaboration partners with a confidence score above a certain level.
- Generate a list of recommended collaboration partners for each student.

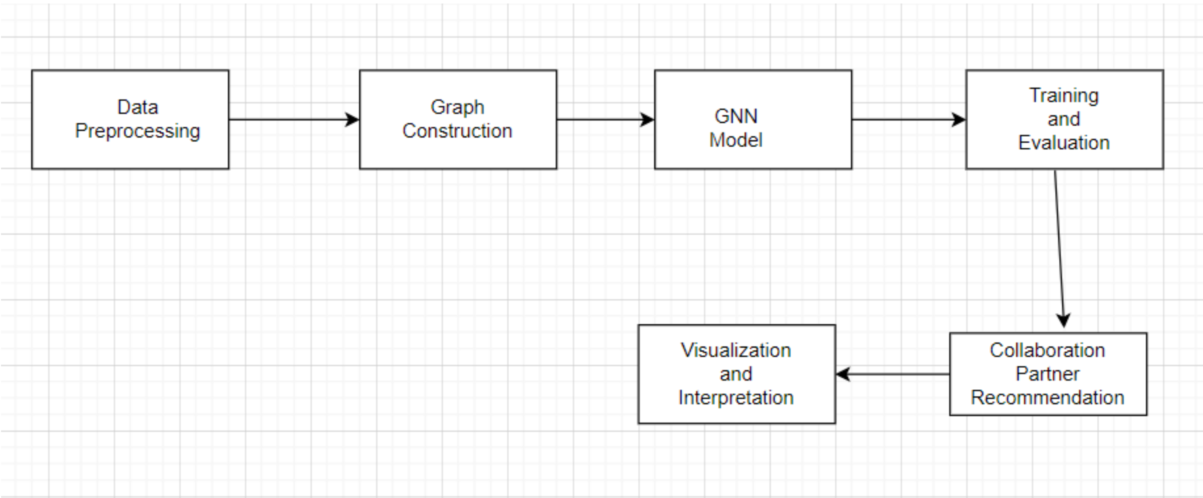
```
# Function to recommend collaboration partners based on skills
def recommend_collaboration_partners_by_skills(student_id, student_data,G ,threshold=0.2):
    model=GNNModel()
    criterion = nn.MSELoss()
    optimizer = optim.Adam(model.parameters(), lr=0.001, weight_decay=1e-4)
    train_model(model, data, optimizer, criterion)
    student_neighbors = list(G.neighbors(student_id))
    collaboration_partners = []
    for neighbor_id in student_neighbors:
        if neighbor_id < len(student_data):
            neighbor_info = student_data.loc[student_data['StudentID'] == neighbor_id]
            neighbor_name = neighbor_info['Name'].iloc[0]
            neighbor_skills = neighbor_info['Skills'].iloc[0]
            prediction = model(data).detach().numpy()[neighbor_id]
            if prediction > threshold:
                print("Prediction:", prediction)
                collaboration_partners.append({'StudentID': neighbor_id, 'Name': neighbor_name, 'Skills': neighbor_skills})
    return collaboration_partners
```

6. Visualization and Interpretation:

- Visualize the student collaboration graph and predicted collaboration partners using tools like NetworkX and Matplotlib.
- Interpret the results, analyze the collaboration patterns, and provide insights into potential collaboration opportunities between students.

```
# Visualize the graph
plt.figure(figsize=(10, 8))
pos = nx.spring_layout(G)
nx.draw(G, pos, with_labels=True, node_color='lightblue', node_size=200)
plt.title('Student Collaboration Graph Based on Shared Skills')
plt.show()
```

Architecture Diagram

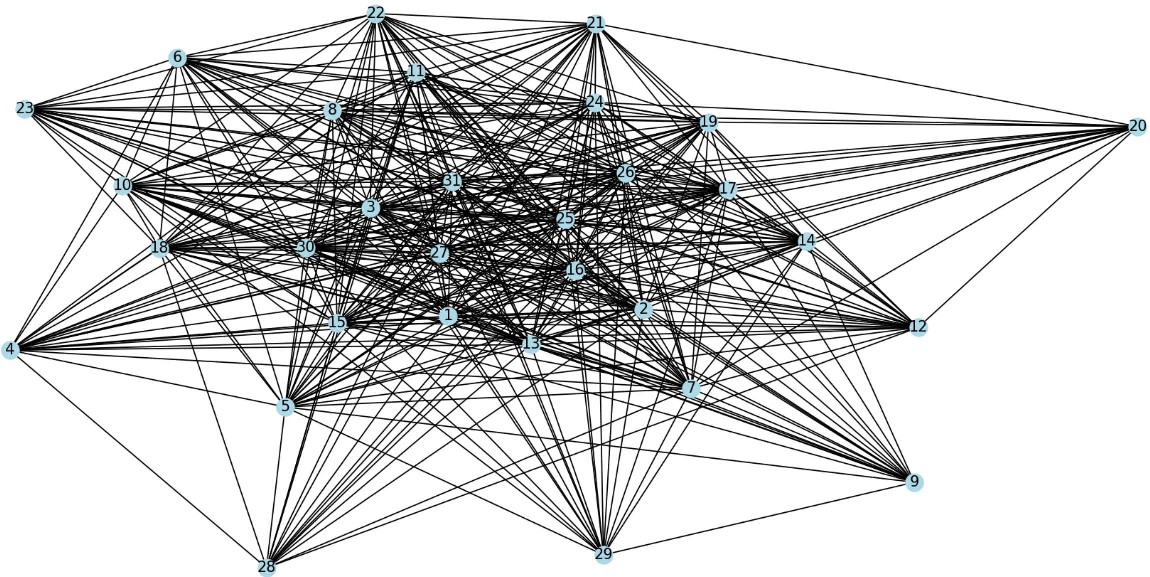


Novelty

The project introduces a novel approach to fostering collaboration among students by leveraging a graph-based representation of their skills and interests. It utilizes Graph Neural Networks (GNNs) to analyze the collaboration graph and recommend personalized collaboration partners based on shared skills. This skill-based recommendation system encourages interdisciplinary collaboration and offers targeted suggestions tailored to each student's unique profile and objectives.

Results

Search based on Club_Memberships



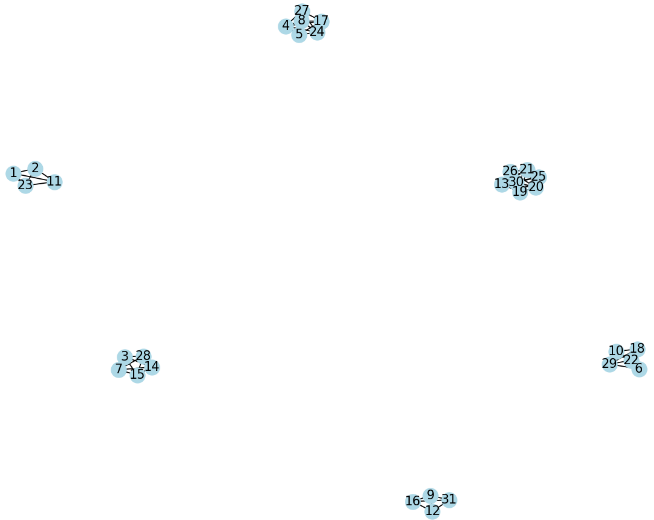
Choose the option

```
Choose option
1) Find based on skills
2) Find based on research
3) Find based on major
4) Find based on extracurricular_Activities
5) Find based on Club Memebership
6) Find based on academic_interests
7). Exit
Enter your choice (1/2/3/4/5/6/7): 5
Enter your Student ID: 6
```

Recommended students

```
Recommended collaboration partners based on your ClubMemberships:
{'StudentID': 10, 'Name': 'Student 10', 'ClubMemberships': 'Coding Club, Volunteer Group, Music Club'}
{'StudentID': 11, 'Name': 'Student 11', 'ClubMemberships': 'Coding Club, Sports Team, Art Club'}
{'StudentID': 13, 'Name': 'Student 13', 'ClubMemberships': 'Debate Club, Volunteer Group, Music Club, Art Club, Sports Team'}
{'StudentID': 14, 'Name': 'Student 14', 'ClubMemberships': 'Art Club, Debate Club, Music Club'}
{'StudentID': 15, 'Name': 'Student 15', 'ClubMemberships': 'Sports Team, Coding Club, Music Club, Debate Club, Volunteer Group'}
{'StudentID': 16, 'Name': 'Student 16', 'ClubMemberships': 'Coding Club, Debate Club, Music Club, Sports Team, Art Club'}
{'StudentID': 17, 'Name': 'Student 17', 'ClubMemberships': 'Volunteer Group, Debate Club, Art Club, Music Club'}
{'StudentID': 18, 'Name': 'Student 18', 'ClubMemberships': 'Volunteer Group, Sports Team, Art Club, Coding Club'}
{'StudentID': 19, 'Name': 'Student 19', 'ClubMemberships': 'Debate Club, Sports Team, Art Club, Coding Club'}
{'StudentID': 21, 'Name': 'Student 21', 'ClubMemberships': 'Art Club, Volunteer Group, Debate Club'}
{'StudentID': 22, 'Name': 'Student 22', 'ClubMemberships': 'Art Club, Music Club'}
{'StudentID': 23, 'Name': 'Student 23', 'ClubMemberships': 'Coding Club, Art Club'}
{'StudentID': 24, 'Name': 'Student 24', 'ClubMemberships': 'Debate Club, Art Club, Coding Club, Music Club'}
{'StudentID': 25, 'Name': 'Student 25', 'ClubMemberships': 'Debate Club, Sports Team, Music Club, Coding Club, Art Club'}
{'StudentID': 26, 'Name': 'Student 26', 'ClubMemberships': 'Sports Team, Art Club, Music Club, Debate Club'}
{'StudentID': 27, 'Name': 'Student 27', 'ClubMemberships': 'Sports Team, Volunteer Group, Debate Club, Art Club, Coding Club, Music Club'}
```

Search based on Academic Interests



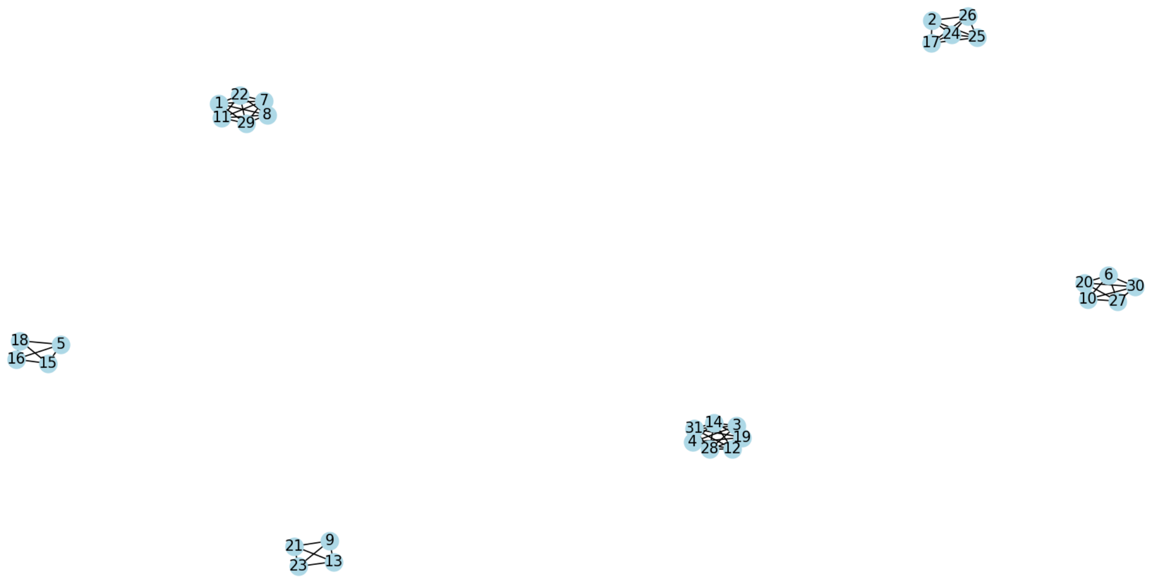
Choose an option

```
Choose option
1) Find based on skills
2) Find based on research
3) Find based on major
4) Find based on extracurricular_Activities
5) Find based on Club Memebership
6) Find based on academic_interests
7). Exit
Enter your choice (1/2/3/4/5/6/7): 6
Enter your Student ID: 7
```

Recommended students

```
Recommended collaboration partners based on your academic interests:
{'StudentID': 3, 'Name': 'Student 3', 'AcademicInterest': 'History'}
{'StudentID': 14, 'Name': 'Student 14', 'AcademicInterest': 'History'}
{'StudentID': 15, 'Name': 'Student 15', 'AcademicInterest': 'History'}
{'StudentID': 28, 'Name': 'Student 28', 'AcademicInterest': 'History'}
```

Search based on Major



Choose the option

```
Choose option
1) Find based on skills
2) Find based on research
3) Find based on major
4) Find based on extracurricular_Activities
5) Find based on Club Memebership
6) Find based on academic_interests
7). Exit
Enter your choice (1/2/3/4/5/6/7): 3
Enter your Student ID: 15
```

Recommended students

```
Recommended collaboration partners based on your Major:
{'StudentID': 5, 'Name': 'Student 5', 'Major': 'Computer Science'}
{'StudentID': 16, 'Name': 'Student 16', 'Major': 'Computer Science'}
{'StudentID': 18, 'Name': 'Student 18', 'Major': 'Computer Science'}
```

Conclusion

In conclusion, this project presents a promising framework for enhancing collaboration among students through an innovative skill-based recommendation system. By leveraging Graph Neural Networks (GNNs) and a graph representation of student skills and interests, the system provides personalized collaboration suggestions that foster interdisciplinary partnerships. The project demonstrates the potential of leveraging machine learning and graph analysis techniques to facilitate meaningful collaborations and knowledge exchange in educational settings. Further research and development in this area could lead to valuable advancements in collaborative learning environments.

References

- Knowledge Sharing Through Academic Social Networking: The Impact of Personal and Social Outcome Expectations
- Similarity-based link prediction in social networks using latent relationships between the users
- Temporal Graph Neural Networks for Social Recommendation
- A Graph-Neural-Network-Based Social Network Recommendation Algorithm Using High Order Neighbor Information Yonghong Yu 1,* , Weiwen Qian 1 , Li Zhang 2 and Rong Gao 3 (2022)
- A Deep Graph Neural Network-Based Mechanism for Social Recommendations Zhiwei Guo and Heng Wang (2021)
- Heterogeneous Hypergraph Neural Network for Social Recommendation using Attention Network BILAL KHAN, JIA WU, JIAN YANG, XIAOXIAO MA (2023)
- A Survey of GNN-based graph similarity learning
- Characterizing the communication requirements of GNN accelerators
- Network modeling based on GNN and network behaviors

Github link

https://github.com/amulyap23/Familiar_Strangers

Team Members and contributions:

Amulya: Model Training and recommendation functions

Ananya: Graph Construction and model creation

Charan: Graph Visualization

Amulya Pothumarthi

(PES2UG21CS062)

Ananya Singla

(PES2UG21CS066)

C H Sri Charan

(PES2UG21CS129)