Report: GNN's as SOTA methods for Recommendation Systems

Ali Jaabous
Faculty of Economics and Business Administration
Humboldt University of Berlin
jaabousa@student.hu-berlin.de

Bora Eguz
Faculty of Economics and Business Administration
Humboldt University of Berlin
bora.eguz@student.hu-berlin.de

Ozan Ayhan
Faculty of Economics and Business Administration
Humboldt University of Berlin
ozan.ayhan@student.hu-berlin.de

1. Introduction

When people want to listen to different music or watch different movies, it is not hidden that they will seek the ones with similar tastes by hoping to like them. In this point, recommendation systems can have a big effect on people's decisions on whether to use the same platform or continue their research on other platforms. Recent years showed that there are fast and big developments in the Graph Learning based Recommender Systems (GLRS) which enables us to explore homogeneous or heterogeneous relations in graphs [5]. The basic idea of GNN is to model complex relationships between entities in a network. In the context of recommendation systems, this means that not only the isolated preferences of a user or the characteristics of a product are taken into account, but also the relationships between users and products in a network [7].

Our goal in the scope of the seminar course is to build recommendation models with different algorithms to compare them. To do this, the Netflix data (which can be found on Kaggle) will be manipulated (also will be downsized for the prototype). This data is from a competition Netflix organized back in 2009 to have the best recommendation system. They provided the data of movie information and user's rating. But as mentioned above, this time our goal is not to find the best model but rather to benchmark GNN models and traditional recommendation system models to see if it is the state-of-the-art model. To do this, we will apply some preprocess steps to data, build and train the model, and lastly evaluate its performance with some common metrics such as MAE or RMSE and also compare their execution times.

In the next section you will find the literature review focusing on the papers that have used graph neural networks in their recommendation systems and also other models that focuses on recommendation systems. After that, there is the methodology part which explains the steps and algorithms that are used. Experimental design, the part 4, demonstrates all the data preprocessing and feature engineering steps. And then explains the data splitting strategy and fine tuning. In the same section model architecture and evaluation steps are being explained. The results section shows all the metrics for each model and explains it verbally. At last, the conclusion section is the part where the outcomes of this paper is being discussed. All necessary figures, tables, and matrixes are also in the appendix.

2. Literature Review

Recommendation systems and different approaches to this task have been studied many times in the literature. In 2019, Wang and his colleagues conducted a study to propose a better recommendation system than factorization machines do by taking the interactions between an item to different nodes into account [7]. While factorization machines take this task as a supervised problem and try to evaluate each interaction as sole, they argue that interacting with different nodes and the relation between those nodes make a difference. Therefore, they introduced the knowledge graph attention network which recurrently takes information from a node's neighbors to define node embeddings. Based on the empirical studies, KGAT outperforms such factorization machine solutions.

Recently, graph neural networks have increased their popularity especially when it comes to recommendation systems as the method has the ability to investigate high-order information (Gao, C., et al., 2022). In 2019, Fan et al. conducted a study that focused on GNNs in the field of social recommendation [2]. To improve some quality metrics such as accuracy and quality, the way the model captures interactions has been explored while also proposing another framework. In the end, to evaluate the performance different models such as traditional recommender systems have been compared with the GNN model, and the result has shown that GNN has better scores in these metrics. Following that, Cui et al. conducted a study that focused on outfit compatibility by learning latent patterns [1]. Graph neural networks have been used to complete this task which is deciding how much the outfit is compatible. The paper also discusses the way GNN takes the interactions between different outfit items and people and compares the proposed model with different baseline models which result in the proposed model's success. This study improves the field of fashion advice by emphasizing the holistic knowledge of dressing and by employing graph neural networks to capture complex relationships between clothing components for better outfit compatibility predictions.

Yin et al. in their study in 2019, proposed a Graph Neural Network-based Collaborative Filtering (GCF) framework to solve the data sparsity problem in recommender systems [8]. Their neural network method leveraged by attention-based message passing showed significantly higher accuracy compared to existing approaches in recommendation systems. Using real-world datasets, the GCF technique demonstrated significant enhancements in recommendation performance over state-of-the-art models, especially regarding the measures of HR@k and NDCG@k.

Then in 2021, Wang et al. published a paper that reviews recommender systems that are leveraged by graph learning. In the paper, different approaches such as GNNs, collaborative filtering, and matrix factorization have been discussed and reviewed [5]. The authors scrutinized the strengths and weaknesses of various graph learning methods without presenting explicit experimental outcomes, however, the outcome of the paper presents a systematic review of the recent improvements in the field. The study by Redwane Nesmaoui et al. which was published in 2023, has researched the performance of the LightGCN method within Graph Neural Networks in recommendation systems, specifically in movies [4]. The study showed that the LightGCN gives better results compared to conventional methods. Unlike other models, in LightGCN's approach, the bipartite graph approach was used where movies are represented as nodes and their ratings are used as edges. The results showed how the system can suggest new items to its users more efficiently based on this relational graph structure.

The study by Redwane Nesmaoui et al. which was published in 2023, has researched the performance of the LightGCN method within Graph Neural Networks in recommendation systems, specifically in movies [4]. The study showed that the LightGCN gives better results compared to conventional methods. Unlike other models, in LightGCN's approach, the bipartite graph approach was used where movies are represented as nodes and their ratings are used as edges. The results showed how the system can suggest new items to its users more efficiently based on this relational graph structure. And another study by Wang et al. that is published in 2023 has researched the application of data science methods in recommendation systems [6]. The paper proposes some preprocess steps for recommendation tasks and showcases different ML models such as CNN, MLP, RNN, and graph-based neural networks. However, the paper does not compare these different approaches but rather explains each on its own.

And the study conducted by G. Kaur et al. in 2023 aimed to address data sparsity in recommender systems [3]. To overcome this, they introduced the Knowledge Graph Convolution Network (KGCN) as a solution. KGCN was used to incorporate user demographics and item attributes into the user/item interaction matrix. It outperformed Graph Convolution Networks (GCNs) and other state-of-the-art recommender methods, showcasing superior performance in experimental comparisons.

Another study by Wang et al. that was published in 2023 has researched the application of data science methods in recommendation systems [6]. The paper proposes some preprocess steps for recommendation tasks and showcases different ML models such as CNN, MLP, RNN, and graph-based neural networks. However, the paper does not compare these different approaches but rather explains each on its own. And the study conducted by G. Kaur et al. in 2023 aimed to address data sparsity in recommender systems [3]. To overcome this, they introduced the Knowledge Graph Convolution Network (KGCN) as a solution. KGCN was used to incorporate user demographics and item attributes into the user/item interaction matrix. It outperformed Graph Convolution Networks (GCNs) and other state-of-the-art recommender methods, showcasing superior performance in experimental comparisons.

In 2023, Sun et al conducted a study of personalized recommendation algorithms for learning resources based on differential evolution and graph neural networks[20]. In this study a multi-head attention mechanism integrated with expressing learners and learning resources as graph data and the results showed an improved recommendation performance in terms of accuracy, recall, $\Gamma 1$ score, and RMSE value.

In 2018, Ying et al conducted a study that discussed recent improvements in neural networks for graph-structured data[21]. The study presents the highly-scalable GCN framework PinSage and its state of art performance in recommendation tasks at Pinterest. The study proposes possible expansions for large-scale graph representation learning and emphasizes the role of graph convolutional approaches in production recommender systems. The study found out the performance of recommender systems has been greatly enhanced by deep neural networks for graph-structured data and high-quality suggestions can be produced in real-world scenarios using GCN embeddings.

In 2020 Wu et al conducted a study, which presents a novel technique to improve the performance of graph convolutional networks (GCNs) in recommendation systems[22]. This technique is called self-supervised graph learning (SGL). In addition to typical supervised tasks, SGL uses self-supervised learning to increase the robustness and accuracy of recommendations, which benefits long-tail items in particular and provides resistance against interaction noises. The findings of the study based on three benchmark datasets showed how well SGL works to increase recommendation resilience and accuracy in the face of interaction noise.

In 2020, Chen et al. extended the use of Graph Convolutional Networks (GCNs) as state of the art model for learning representations in graphs[23]. The model particularly focuses on collaborative filtering within user-item interaction. In the study, they presented a novel approach, the LR-GCCF model, which merges linear embedding propagation with residual preference learning. By this approach it is aimed to boost the performance of collaborative filtering. Furthermore, the research identifies and addresses one of the main challenges in GCN based collaborative filtering models. This issue is called the over-smoothing problem that arises from graph convolution aggregation with sparse user-item data. To overcome this issue, they proposed a specialized residual network structure. This structure not only mitigates the over-smoothing problem but also enhances overall recommendation performance. The results showed that removing non-linearities results with the improvement in recommendation accuray. In summary, the outcome of Chen et al.'s study showed a significant improvement in collaborative filtering, marking GCN-based recommender models as notably superior to traditional methods, especially in terms of efficiency and effectiveness

In their 2022 study, Yu et al. address the inaccurate movie recommendations problem in traditional systems[19], These systems often overlook specific user and movie aspects in favor of user ratings.To address this Yu et al. provide an integrated approach called the XGBoost-DNN model and compared the success of this model with some traditional models like RandomForest, LightGBM, SVR, and KNN. In this new approach they used 2 models in which XGBoost selects important characteristics for prediction and DNN utilizes these features as inputs to train. The results showed that the Mean Squared Error (MSE) is significantly lower with this strategy than with more conventional models. In summary, their work significantly increased the accuracy of movie rating predictions and recommendations by integrating feature selection with neural network-based predictions, demonstrating an important advancement in recommendation system methodology.

In 2020, Badiâa Dellal-Hedjazi and Zaia Alimazighi conducted a study to utilize deep learning techniques[24]. They aimed to solve the main problems of conventional recommendation systems like the cold start issue and the independent treatment of user and item features. With this research they hoped to improve recommendation accuracy. In order to do this, they present a new deep learning-based approach which makes use of semantic information from item profiles. This new model employs Bidirectional Encoder Representations from Transformers (BERT) and Convolutional Neural Networks (CNN) with customized cross-convolutional filters. With this method, the model was able to accurately represent the complex, nonlinear relationships between users and items. The study showed that their method outperformed other models like MLPs, RNN, MF, NCF in terms of effectively modeling user-item interactions and reducing overfitting issues. This innovative method stands out for its ability to mitigate common challenges in recommendation systems and improve overall recommendation accuracy.

In 2019 Feng et al. presented an innovative approach called Attention-based Graph Convolutional Network (AGCN)[25]. With this approach, they aimed to improve the efficacy of recommendation systems. The approach combines rating data with extra user and item information into a unified framework to solve the matrix completion problem. What makes The AGCN architecture unique is that it integrates graph convolutional networks and attention layers to produce unified, low-rank dense representations. Furthermore, unlike other models, AGCN has achieved to reduce computing complexity by utilizing Chebyshev polynomial graph filters. Moreover, the attention mechanism enables models to weigh neighbor information. This improves the graph's learning ability also to converge more quickly and steadily. Based on the research results, the AGCN has outperformed the other state-of-the-art models and it gave better results in terms of prediction accuracy and convergence speed. In summary, this study showed a significant improvement in recommendation systems by providing accurate and efficient recommendations.

Throughout the literature research it has been observed that there are numerous ways to deal with recommendation system problems. While graph neural networks are the most popular in advancing the field, GNNs also include many different approaches in it such as link prediction, graph classification, node classification, and graph embeddings. These types of tasks are getting more important for especially the companies like Youtube, Netflix, and TikTok which serves its services to online users while selling the content they have for the user who is willing to get exposed to that content even if the user is not aware of such intent. As these platforms continue to grow and diversify, the development of recommendation systems and the application of GNN and similar technologies in recommendation systems increasing its popularity day by day.

3. Methodology

In creating our comparison between conventional models of the realm of machine learning and graph-based neural network architectures, we created a schema that builds on standardized data science modeling steps and meets the requirements for the GNN-based models we created. These steps are data preparation, which subsumes various steps such as feature engineering and graph construction and includes phases such as model development, hyperparameter optimization and finally evaluation and performance analysis. The figure below is intended to visualize the described phases schematically and is informatively enhanced and clarified by the elaboration provided below.

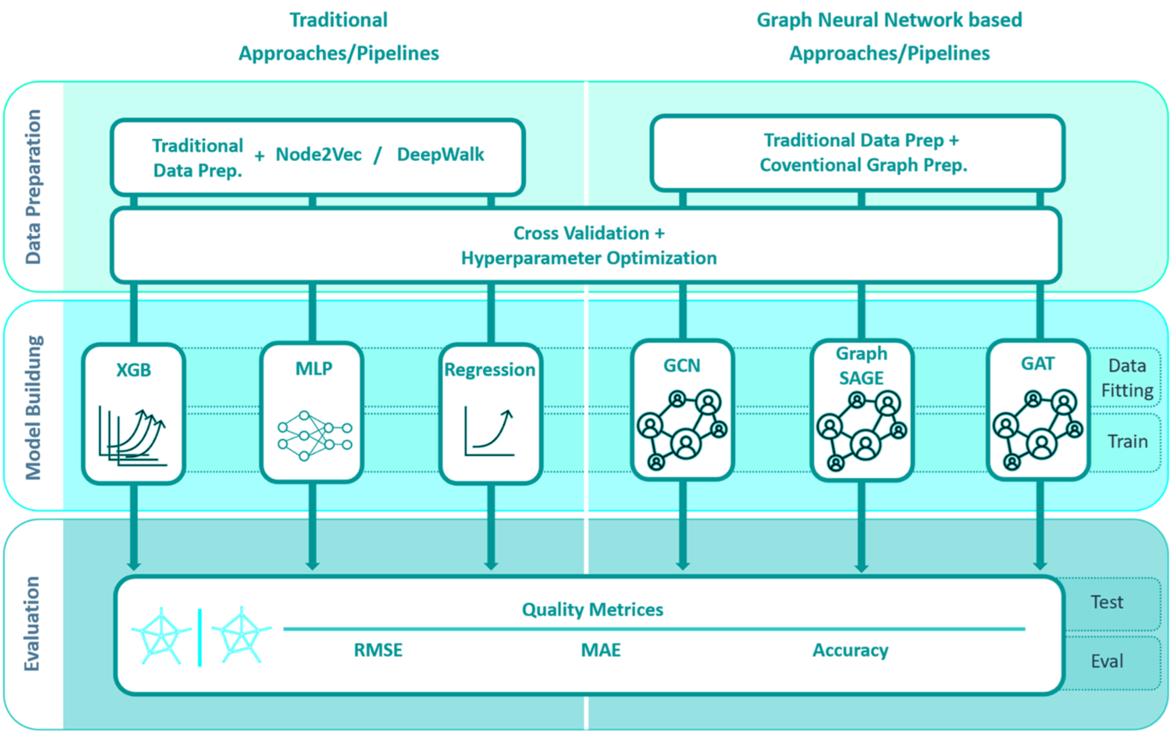

Figure 1: Methodology in schematic form

3.1 Data Preparation

In our analysis, we decided to use the Netflix Prize dataset, a well-known dataset from an older Kaggle competition, for conducting our comparison. This dataset consists of two main subdatasets, which we utilized in the further course of our project. One of these datasets comprises movies released between 1980 and 2005, containing information such as MovieID, YearOfRelease, and Title. The other is a dataset containing MovieID, CustomerID, Rating, and Timestamp, depicting interactions between viewers and movies. In our prototype, for the sake of simplicity and improving usability, we only utilized a portion of the data to avoid computational problems due to insufficient computing power and to provide a quick overview of the methodology and our approach. Initially, we carried out a manual data transformation phase to enhance the quality of the dataset. This involved converting raw text data into CSV format and making simple manual data adjustments. In our prototype, we initially utilized a combination of 175 users and nearly 1,000 movies, resulting in approximately 11,000 ratings. In our final testing phase, we used a total of 2,500 users and over 17,000 movies, leading to more than half a million ratings.

3.2 Feature Engineering and Graph Construction

After this filtering, which allowed us to reduce the dataset to an acceptable yet expressive size in terms of computational runtime and capacity, we conducted several feature engineering activities to increase our feature set and create a stronger data foundation. We employed various approaches for this purpose.

Within traditional data manipulation approaches, we utilized various data operations to generate features regarding both users and movies, enhancing their informative expression. How these features are obtained and how they capture important patterns are explained below:

- rating_count_per_user: This feature represents how many ratings each user has given. It captures patterns because it can indicate the level of user engagement. Users who rate more movies might provide more reliable data for understanding their preferences
- rating_count_per_movie: This feature is the number of times a movie has been rated. This can be a good indicator of how in-demand a film is or how popular it is. The film that received many more screenings may have been considered more widely or widely released.
- avg_rating_per_user: Calculates the average rating each user gave to all movies. This feature is used to measure the user's general tendency to rate movies. This prevents the recommendation system from being manipulated due to different rating scales used by different users.
- avg_rating_per_movie: This represents the average rating received by each movie. It's a direct measure of a movie's overall reception by the users. This feature is crucial as it helps in identifying high-quality or preferred content
- Release_age: We tried to capture aspects of movies over time by calculating their age since 2005. This feature also captures important patterns. Because User preferences may vary across new and old movies. While some users tend to watch new movies, this may be the opposite for some.

Following this approach, we employed an NLP-based feature engineering method through which various clusters regarding the type of movie could be generated based on their names. Two models from Huggingface were utilized for this purpose and a pretrained model "all-MiniLM-L6-v2" is used for this process. The movie titles are treated as sentences and with the mentioned model, embeddings are obtained. These embeddings are used to cluster the titles hoping that they become useful information about the movies. The final feature engineering approach utilized graph-based methods, specifically Node2Vec and DeepWalk, which generated an additional 20 features for each instance based on the interaction between user instances and movie instances.

It is important to note that the generated features were separated and utilized differently for each model. For instance, the graph-based approaches were exclusively used for conventional models such as Ridge Regression, XGB, and MLP. Their use in the graph-based models was avoided as these specifics are already inherent to them. However, due to our technical incapabilities, while training the benchmark models such as ridge regression, XGBoost, and MLP these random walk variables have not been used. Basic features were used within both model streams. Initially, the NLP-based features were intended for use within both model streams; however, due to the low data quality resulting from the utilized model, which was confirmed by the model authors in the documentation, their use within the graph-based models was omitted.

Additionally, a data manipulation step was exclusively realized for the graph-based models. This involved constructing graph structures using tensors, where nodes were realized for user and movie instances, and ratings were represented as their edges. In order to represent the data in the best way, we decided to use an undirected graph, which is frequently preferred in GNN models. Undirected graphs are popular when there is a complex user-item relationship and sparsity problem in data. With the embeddings that are obtained from nodes and edges, the graph representation has been constructed.

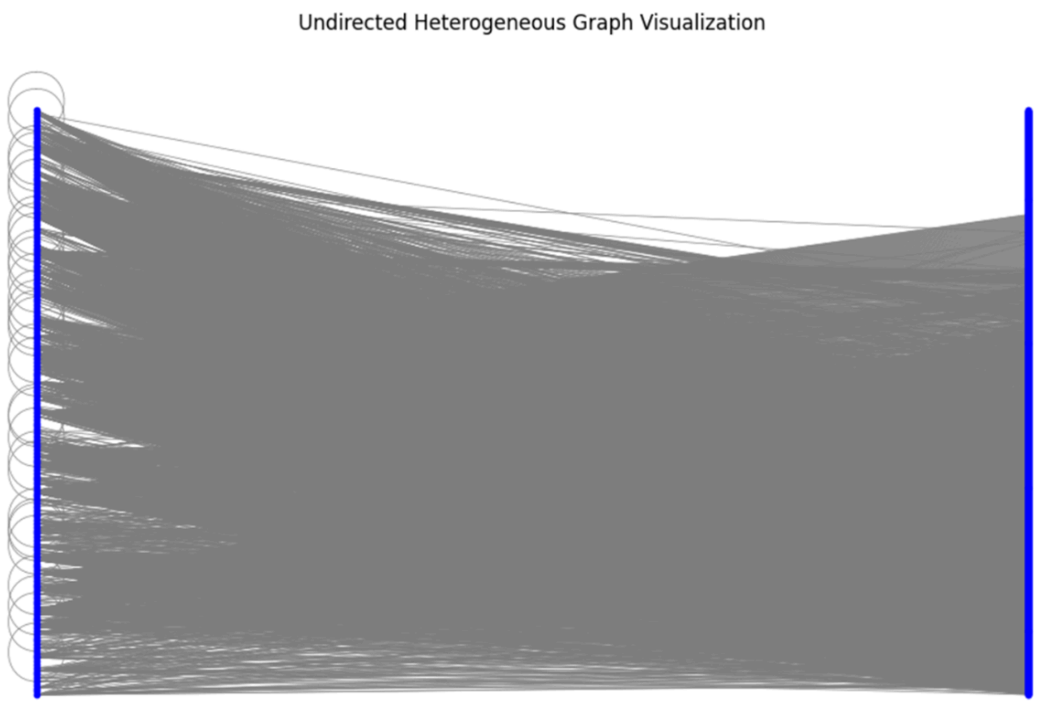

Figure 2: Undirected Heterogeneous Graph

3.3 Data Splitting Strategy

As part of the data partitioning process, we opted for a cross-validation method, as initial tests on the graph-based models revealed a significant variance in performance (MAE value) depending on the chosen data split. Another advantage of our chosen strategy is that it allows for the training data and validation data to be utilized in each other's roles, making data usage more efficient. It's important to note that initially, a standard data split was implemented. Regarding the graph-based test streams, two additional classes were added to reconstruct and experiment with this data partitioning approach if desired. Within the accompanying notebook, for simplicity, only the cross-validation approach was used. It's also worth noting that due to limited data transformation options in the PyTorch Geometric library, the cross-validation approach had to be implemented from scratch, and it could only be realized as a Two-Fold Cross Validation. This was because, during this approach, various complex tensor architectures had to be manually implemented, which became even more complex with a fold size greater than 3, as a single graph had to be divided into multiple subgraphs, resulting in tedious and error-prone validation and training data allocation, as evidenced by data misallocations in the realized 2-fold split, for example. Finally, it's worth mentioning that the chosen data partitioning strategy was embedded within an Optuna hyperparameter optimization approach to determine the best parameters. Additionally, in a final step, the model trained on the best data using both training and validation datasets is evaluated using the intentionally omitted test dataset.

3.4 Model Selection and Model Development

After the explained phases, the core of the project can be examined more closely: the model architecture and model selection. The models we planned to use are Graph Convolutional Network (GCN), Graph Sage, and Graph Attention Network (GAT). These models have pros and cons regarding the nature of the task, graph type, complexity, and computational efficiency. While comparing the output of GNN-based models among themselves, we planned to train the same data with some traditional models in order to benchmark the success of the general GNN output. These base models are XGB, MLP, and Ridge Regression. Below, the theoretical foundations of the mentioned models are further explained:

3.4.1 Graph Convolutional Neural Networks (GCNs)

Graph Convolutional Neural Networks (GCNs) form the fundament in the realm of graph neural networks which are designed to handle data represented in graphical structure. These networks have gained attention due to their effectiveness in various tasks involving graph-structured data, such as social network analysis, drug discovery and recommendation systems, which in turn is our central object of observation [2, 5].

To get a better understanding of the GNNs, the underlying theory will be unraveled and described through mathematical formalisms. The subject of processing, as the name suggests, are the graph structures. These are defined as follows:

- A graph $G=(V,E)$ consists of a set of nodes V and a set of edges E connecting these nodes.
- The graph can be represented using an adjacency matrix A, where A_{ij} is 1 if there is an edge between nodes i and j, and 0 otherwise. Additionally, often the graph structure is represented with a node feature matrix H, where each row corresponds to a node and each column corresponds to a feature.

Now that the formal definition of graphical data structures is given, the methodical applications of the architecture can be explained.

- Unlike traditional convolutional neural networks that are applied on images, GCNs apply a different kind of convolution to graph-structured data as implied earlier.
- The convolution operation in GCNs involves the aggregation of information from a node's neighbors and updating the node's representation based on this aggregated information. The mathematical formulation of the operation looks as following:

$$H^{(l+1)}=\sigma(\check{D}^{-\frac{1}{2}}\tilde{A}\check{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$$

Where:

- $H^{(l)}$ is the node representation matrix at layer
- $W^{(l)}$ is the node representation matrix at layer
- $\tilde{A} = A+I$ is the adjacency matrix with added self-connections (where I is the identity matrix), which allows incorporating node's own features.
- $\check{D}$ is the degree matrix of $\tilde{A}$, where $\check{D}ii = \sum j\tilde{A}ij$.
- σ represents an activation function.

The convolutional operation of this Network family serves as the core element of Graph Convolutional Neural Networks (GCNs) and enables effective treatment and analysis of graph-structured data. Within this framework, the information of the neighboring nodes is aggregated and the node representations are updated based on this information. The explained mechanism is a fundamental concept in GCNs and is known as the neighborhood aggregation, the goal of this capability is to enable the network to capture the complex relationships within the graph [4, 13].

Mathematically, neighborhood aggregation can be represented as follows:

Given a graph $G=(V,E)$ with node features X and adjacency matrix A, the representation of a node i after aggregation can be calculated as:

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in N(i)} \frac{1}{\sqrt{|N(i)|\cdot|N(j)|}} h_j^{(l)} W^{(l)}\right)$$

Where:

- $h_i^{(l)}$ represents the node i at layer l.
- $N(i)$ denotes the set of neighboring nodes of node i.
- $W^{(l)}$ is the weight matrix for layer l.
- σ represents an activation function.

Another fundamental aspect of GCNs, is the concept of parameter sharing which helps in generalizing the learned features across different parts of the graph. In traditional convolutional neural networks (CNNs), the same set of weights is applied across different spatial locations of the input data. Similarly, in GCNs, the same set of weights is shared across different nodes in the graph [4, 13].

Mathematically, parameter sharing can be represented by the weight matrix W in the convolution operation which was explained above.

The last concept is the concept of graph pooling. Graph pooling is a technique for reducing the size of the graph while retaining its important structural information. Similar to the pooling of layers in CNNs, graph pooling aggregates information from groups of nodes, thereby reducing the size of the graph [4].

In summary, Graph Convolutional Neural Networks extend the concept of convolutional neural networks to graph-structured data. By aggregating information from neighboring nodes, GCNs effectively capture the complex relationships within graphs. For regression tasks, adjustments to the output layer are made to predict continuous values rather than discrete labels [13].

3.4.2 Graph Sample and Aggregation (GraphSAGE)

GraphSAGE is a framework that is designed to get node embeddings from graph-structured data. This framework is in fact build up upon the foundations of Graph Convolutional Networks (GCNs) and its goal is to create embeddings from the graph represented data. This method enables the learning from the graph representation in an inductive setting which ensures as long as a new node added, the model adapts and is being trained on that. The unique feature is the thing what differs GraphSAGE from GCNs which usually perform on a transductive setting. This transductive setting treats graph represented data as the final version of it and assumes there will not be any new node and all the possible graph representation is present at the time being training (Hamilton, W. et al., 2017).When it comes to the deep diving into it, it can be splitted into three parts as follows: Theoretical foundations, information aggregation process and the parameter sharing.

The graph representation is quite similar to other graph representations that other graph neural network models such as GCNs or GATs use. It consists of a set of nodes, which are represented as V, and edges that connect these nodes, which are represented as E (Xiao, L. et al., 2019). These types of graphs most of the time are represented by an adjacency matrix A and nodes' features are stored in a node feature matrix H.

The GraphSAGE usually samples k number of neighbor nodes and aggregate their node information

$$h_N^k = AGGREGATE_K\left(\left\{h_u^{k-1},\ \forall u \in N(v)\right\}\right)$$

And there are three kinds of formulas to aggregate the information. One is Mean, second one is Pool and the last one is LSTM.

$$Mean\colon AGG = \sum_{u \in N(v)} \frac{h_u^{k-1}}{|N(v)|}$$

$$Pool\colon AGG = \gamma\left(\left\{Qh_u^{k-1},\ \forall u \in N(v)\right\}\right)$$

$$LSTM\colon AGG = LSTM\left(\left[h_u^{k-1},\ \forall u \in \pi(N(v))\right]\right)$$

This aggregation process is what makes the model unique as it does not utilize all the graph at once but rather prefers to use a fixed size of neighborhood around the node to generate the node information embeddings.

Where:

h_u^{k-1} is a sample of neighborhood nodes,
Q is an activation function.

After that the neighborhood information is combined with the node as follows:

$$CONCAT\left(h_v^{k-1},\ h_{N(v)}^k\right)$$

And then with using the formula below, the embeddings are being created.

$$h_v^k = \sigma\left(W^k \cdot CONCAT\left(h_v^{k-1},\ h_{N(v)}^k\right)\right)$$

Where:

σ is the non-linear activation function,
W^k is the weight matrix.

After the calculations, each node embeddings need to be standardized with the following formula

$$h_v^k = h_v^k / \left|\left|h_v^k\right|\right|2 \cdot \forall v \in V$$

Similar to GCNs, GraphSAGE also utilize the concept of parameter sharing to make features general along the graph with using the same set of weight matrix. GraphSAGE make sure that the generated embeddings are invariant to the node locatıon in the graph representation which improves efficiency and the generalization of the model (Xu, K. et al., 2018).

The method focuses on the differentiation in the way of node-level embedding aggregation, this can also be used and extend to the graph pooling and effective subgraph representations.

In essence, the method GraphSAGE introduces a unique method for representation of nodes for large graphs without using all the nodes in the graph but instead using a fixed size of it which is not only important for graph neural network models but also all types of graph-based learning problems.

3.4.3 Graph Attention Networks (GATs)

Graph Attention Networks(GATs) are a type of Graph Neural Networks that generates embeddings from graph structured data. It was first introduced by Petar Veličković and colleagues in 2018. What makes it different from other types of GNNs is its unique feature aggregation process by assigning different importance values to different nodes. This process is called the attention mechanism. This method gives flexibility to the model to dynamically update the impact of each node rather than giving them equal weights as in GCN and GraphSage. With this flexibility GATs are able to focus on the most important information from the neighborhood of each node and make it possible to capture more crucial information and represent the complex graphs more effectively[15].

GATs use the same graph structure G=(V,E) similar to GCNs and GraphSAGE. The nodes are represented as V and the edges as E. In this type of data, while the relationship of the edges between nodes are stored with a squared matrix called Adcacenjy Matrix A, the relationship between the nodes and their features are represented with Node Feature Matrix H.

The further steps of the GAT can be broken down respectively as computation of attention coefficients and future transformation, normalization, feature aggregation and multi-head attention.

As an initial step of the GAT process, node features are transformed into a higher dimensional representation space. This transformation is achieved by multiplication of Node Feature Matrix h and a shared weight matrix W. The formulation is described below:

$$h_i' = W * h_i$$

Where:

h_i' The transformed feature vector of node i after applying the weight matrix W

Secondly, GAT computes raw attention coefficients to assign the impact of the features of node j to node i. These coefficients are computed by using an attention mechanism a, which consists of a dot-product operation followed by a non-linear activation function, like LeakyReLU, or a tiny neural network[15].

$$e_{ij} = a(W * h_i, W * h_j)$$

Where:

e_{ij} is raw attention scores
a is attention mechanism

After getting the raw attention scores e_{ij} for all nodes j(neighbors i), the sum of the scores of each node j are normalized to 1. In order to do this, the computation uses the softmax function. This step basically transforms the raw scores into attention coefficients a_{ij} to be used in aggregation of neighbor features.

$$a_{ij} = softmax_j(e_{ij}) = \left(\frac{exp(e_{ij})}{\sum_{k \in N(i)} exp(e_{ik})} \right)$$

Where:

$\sum_{k \in N(i)}$ indicates the summation is over all nodes k that are neighbors of node i

$exp(e_{ij})$is the exponential function applied to the raw attention score. It ensures that all values are positive and to amplify differences between the scores.

Furthermore, GAT aggregates the information of each node's neighbors by computing a weighted sum of the transformed features of its neighbors. This aggregation method allows the model to prioritize the most important embeddings while reducing the impact of less important ones[15,16]. In this process normalized attention coefficients are used as weights within the following function:

$$h_i' = \sigma\left(\sum_{j \in N(i)} a_{ij} W * h_j \right)$$

Where:

h_i' represents the updated feature vector for node i after the aggregation process.
σ is the non-linear activation function

Afterwards, GAT uses a multi-head attention method to improve the quality of representation and its learning ability. In this method, several independent attention mechanisms are carried out simultaneously. Then they are combined through concatenation or by averaging.

Concatenation(for intermediate layers):

$$h_i^{'} = ||_{k=1}^{K} \sigma\left(\sum_{j \in N(i)} a_{ij}^{k} W^{k} * h_j\right)$$

Where:

$||_{k=1}^{K}$ denotes the concatenation of the outputs from all K attention heads

K is the number of attention heads

W^k is the weight matrix associated with the k^{th} attention head

Averaging(for the final layer):

$$h_i^{'} = \sigma\left(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in N(i)} a_{ij}^{k} W^{k} * h_j\right)$$

In summary, Graph Attention Networks (GATs) use its unique attention mechanisms to enhance the ability of representation of graph structured data. This mechanism allows the model to prioritize the information aggregating from neighbors and improve the model's ability where complex graph relationships exist. GATs adeptly handle both classification and regression tasks by tailoring aggregation and output processes to the specific demands of the task at hand.

3.4.4 Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting(XGBoost) is an improved version of the Gradient Boosting Algorithm. It was first introduced by Tianqi Chen and Carlos Guestrin in their seminal 2016 paper. XGB with its tree based structure aims to handle a wide range of data science problems focusing on accuracy and speed. Compared to the gradient boosting machine(GBM) algorithm, XGB has shown remarkable success to prevent over-fitting, which makes it a cornerstone algorithm in both regression and classification tasks in the machine learning field.

XGBoost has a tree-based structure. As its core, it ensembles the trees in a sequential manner and tries to decrease the prediction error made by its predecessors. This iterative process is called Gradient Boosting Mechanism, which enhances predictive accuracy over time. Specifically, the algorithm aims to find the best step size and direction to minimize L, using below formula:

$$L(\phi+\epsilon g)\approx L(\phi)+\epsilon g^T\nabla_{\phi}L+21\epsilon^2 g^T Hg$$

Where:

g represents the gradient.

H denotes the Hessian matrix of second-order partial derivatives of the loss function.

Another advantage of XGB is its ability to utilize both L1 (Lasso Regression) and L2 (Ridge Regression) regularization terms into the loss function. By using these techniques it prevents overfitting by penalizing complex models. The regularization terms add to the optimization objective as:

$$Obj=L(\theta)+a|\theta|+\theta\lambda^2$$

Where:

α and λ represent the coefficients for L1 and L2 regularization, respectively.

In summary, XGBoost improves the gradient boosting method by adding regularization terms, and by employing algorithms to optimize tree construction, which makes XGB one of the most popular and preferred machine learning models in the data science world.

3.4.5 Multi Layer Perceptrons(MLPs)

A perceptron is a fundamental element of neural networks. It receives inputs, processes and gives an output. It actually imitates the organization of the neurons in our brains. Perceptrons do the same process as our neurons work together to process signals by passing the information to another. This artificial process is called Multilayer Perceptrons(MLPs). This framework aims to solve non-linear problems where complex relationships exist.

The structure of MLPs consists of 3 main parts, which are the input, hidden, and output layer. These layers are connected to each other with weighted neurons. In this setup, MLPs employ 2 main processes. First one is FeedForward computation, which is the process to make possible predictions. Second one is Backpropagation, which is the method to make model possible to learn. This method updates the network's weights to minimize loss function by computing gradients.

Mathematically, the update rule in MLPs can be expressed as:

$$w_{new}= w_{old} -\eta\left(\frac{\partial L}{\partial W}\right)$$

Where :

w represents the weights, η the learning rate, and $\frac{\partial}{\partial W} \frac{\partial L}{\partial W}$ the gradient of the loss function L with respect to weights.

Multilayer Perceptrons (MLPs) are highly preferred because of its versatility and its ability to customize. It enables this versatility with its wide range of parameters available to tune. These parameters are mainly: number of hidden layers and neurons, the choice of activation functions, learning rate, regularization techniques.

In summary, MLPs are a great advancement in solving complex real life problems from object detection to recommendation systems where complex representations exist. It also gives flexibility and ability to optimize models with its many tunable parameter sets. This makes MLP's very effective for various machine learning tasks.

3.4.6 Ridge Linear Regression

Ridge Linear Regression is a type of linear regression. Its aim is to solve linear problems where multicollinearity exists between the variables of the dataset. This model extends OLS regression by introducing a regularization term to the loss function. This term is called L2. The regularization term is defined as the square of the magnitude of coefficients. The loss function can be formalized as:

$$L(\theta)= || Y - X ||_2^2 + \lambda || \theta ||_2^2$$

Where:

L(θ) is the loss function, Y represents the target variable, X is the matrix of input features, θ is the coefficient vector, and λ is the regularization parameter controlling the magnitude of the penalty added to the coefficients

In summary, Ridge Linear Regression is a simple and effective solution for linear problems especially reducing model complexity and balancing bias variance tradeoff.

3.5 Evaluation and Performance Analysis

To be able to evaluate the performance of the models to be trained, we have used regression-based metrics, which are Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) because the ratings we have in the data were continuous values instead of discrete numbers. In addition to these quality metrics, we adjusted our numerous results to enable evaluation using classification-based metrics. For this purpose, we utilized criteria such as precision, recall, the f1-score, as well as the support value. These are complemented in the classification evaluation by a confusion matrix, which provides a straightforward overview, showing in a visual representation at a glance which values were predicted correctly and which ones were predicted incorrectly.

- Mean Absolute Error (MAE): This is the average absolute error between the actual and predicted values. A lower MAE indicates that the model predicts the data more accurately.
- Root Mean Squared Error (RMSE): This is the square root of the average squared error between the actual and predicted values. RMSE indicates how much the predictions deviate, on average, from the actual values. A lower RMSE indicates better prediction accuracy of the model.
- Precision: This is the ratio of correctly predicted positive instances to the total number of positive predicted instances. It measures the accuracy of the model's positive predictions.
- Recall: This is the ratio of correctly predicted positive instances to the total number of actually positive instances. Recall indicates how many of the actual positive instances were identified by the model.
- F1-Score: This is the harmonic mean between Precision and Recall. It provides a balanced assessment of the model's prediction accuracy for positive and negative instances.
- Support: This is the number of actual instances in each class. It provides insight into the distribution of classes in the data.

4. Experimental Design

Within this section, we delve into the chosen parameter settings regarding the methodology described in the previous section.

4.1 Data Preprocessing & Feature Engineering

To transform our data into an appropriate graph-based format, we utilized a HeteroData object from the PyTorch Geometric library. This served as a container for the graph architecture and was filled with nodes and edges in the form of tensors. To then transform these into a bipartite graph, we utilized the to Undirect function from the PyTorch Geometric library. For the split within the standard splits, we used the Random-Link Split function from the existing library. However, since we aimed for a cross-validation approach for our further endeavors, we had to forgo the last two methods and apply a methodology from scratch. This was elucidated in the methodological section and will be further elaborated on in the section.
As mentioned in the methodology part, some features such as rating_count_per_user, rating_count_per_movie, Release_age, avg_rating_per_movie, avg_rating_per_user are created by calculating the averages and frequencies. After that, by using the movie titles, it is aimed to extract valuable information out of it by applying NLP on it. The NLP resulted in embeddings and these embeddings are used to group each movie title in a cluster and then dummy cluster variables are created to be used as additional features in the models. However,

none of the model's performance is increased. Thus, these dummy variables are omitted in the model that predicts the test set. Furthermore, in order to obtain the relational information between the UserIDs and MovieIDs, Node2Vec and Deepwalk methods are used but due to the technical incapabilities, these variables could not be realized.

4.2 Data Splitting Strategy & Fine Tuning

As described in the previous methodological section, due to the lack of support within the PyTorch Geometric library, the application of the KFold algorithm in its original form was not possible for the graph-based model approach. To provide an alternative, a 2-fold algorithm was designed from scratch and adapted within the benchmark approach using the common KFold library. Within this cross-validation, we utilized 90% of all data, resulting in 45% for each fold. This cross-validation was embedded in a hyperparameter optimization. The aim of this study was to determine the optimal parameters of the neural network architecture in the graph-based model approach. Therefore, our fine-tuning can be interpreted as a form of neural architecture search. For this purpose, the Optuna library was used, which follows a Bayesian approach. A total of 50 trials were conducted within both the benchmark and graph-based approaches. Within every trial the GNN's were trained with the ADAM optimizer for 30 epochs and a Learning Rate of 0.0001. The model with the best parameters was then re-trained with all data and evaluated in the final step with the remaining 10%.

4.3 Model architecture

Within our model architecture regarding the three different GNNs, we have opted for a unified encoder-predictor structure, which standardizes the form of all GNNs and allows for easy entry and modification for further experiments. This results in the following advantages of each component.

The encoder acts as the centerpiece and brain of the GNN model within this architecture. Its central task is to transform the input data into an appropriate format. This is achieved through the division and utilization of the respective GNN convolutional layers, which have already been optimized in a reliable and efficient manner. All GNN convolutional layers were sourced from the PyTorch Geometric library during the project. Furthermore, these layers form the sole differentiating feature within the encoder. In addition to the graph-based convolutional layer, GELU activation functions and dropout layers were integrated into the encoder for conventional reasons. The activation function enables the handling of nonlinear problems, while dropout layers bring a certain degree of regularization to the encoder architecture, as instability of the GNNs was detected in initial experiments.

Following the encoder, we designed a predictor architecture, which, like the encoder architecture, integrates activation functions and dropout layers for the same reasons and aims to channel the representations learned from the encoder architecture and output them in the form of ratings. This was pursued through the integration of fully connected layers.

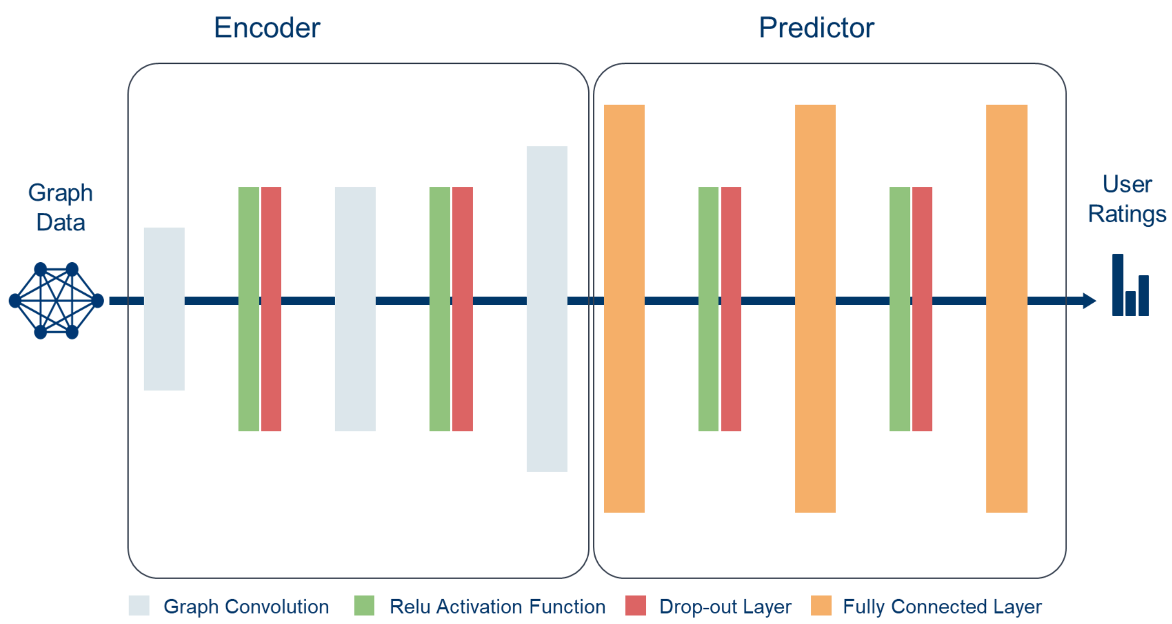

Figure 3: schematic GNN Architecture

The exact arrangement and division of the architectures, as well as all respective units, can be found in Figure 2. The predictor architecture was designed to be applicable to all GNN encoders, which is also reflected in the code, making it leaner. Apart from the graph-based convolutional layer, the remaining layers were sourced from the conventional PyTorch library.

By combining encoder and predictor in a GNN architecture, complex relationships between elements in the graph can be captured and efficiently processed, improving scalability. Furthermore, the encoder-predictor GNN architecture often allows for the interpretation of the model's decisions, as the relationships between the inputs and predictions can be clearly depicted in a sharply separable manner on the graph model.

4.4 Evaluation

To implement the quality metrics presented in the methodological section within our experiments, we utilized the MAE and RMSE values from the torch.nn.functional library, as well as the classification_report and confusion_matrix from the sklearn.metrics library. These tools enabled us to conduct a uniform evaluation across all model streams.

5. Results

During the determination of the optimal hyperparameters concerning the architecture, significant differences have become apparent. Among the graph neural networks, the fastest model proved to be the GraphSAGE architecture at 24:12 minutes. Following closely behind is the classical GCN architecture at 27:48 minutes, and with a considerable margin, the GAT architecture at 1:07:36 hours. We attribute the difference between the GraphSAGE and GCN architectures to the two core improvements of the GraphSAGE architecture. In comparison to the GCN architecture, both sampling and aggregation capabilities were deliberately enhanced. We exclude other factors since the experimental environment and architecture were largely the same. The significant difference between these two architectures and the GAT architecture is attributed, on the one hand, to the resource-intensive computation of the attention mechanism and, on the other hand, to the broad convolutional layers, which were identified within the code by factors within the encoder. The possibility of extending these features to the other architectures was not available in the current version of the PyTorch Geometric package, hence it was not realized. The best architectures are presented in the table below.

	nth Trial	Parameter 1	Parameter 2	Parameter 3	Parameter 4	MAE
GCN	45	105	234	43	0.861	3.551
GraphSAGE	13	158	62	97	0.003	0.974
GAT	13	34	221	146	0.175	**0.940**

Table 1: Best Hyperparameter Values (GNN)

All outputs of the hyperparameter optimization have been appended within the appendix. All testing attempts were conducted on the computer science server of Humboldt university. A notable difference that had a significant impact on the results and became apparent during hyperparameter optimization was the instability of the GCN architecture. This instability was even more pronounced in previous experiments and was also observed in the GraphSAGE and GAT architectures. To counteract this phenomenon, we integrated multiple dropout layers, which proved effective in the case of the GAT and GraphSAGE architectures but had an inhibitory effect in the case of the GCN architecture. The instability is evident within the range of MAE values during hyperparameter optimization. While the GAT and GraphSAGE architectures never exceed the value of 5 except for one instance, the GCN architecture exhibits this behavior very frequently and reaches values in the millions, which are absurd in our setup. Therefore, we propose that the correct choice of architecture plays a crucial role in this class of GNNs, and performance strongly depends on it.

In terms of performance regarding regression and rating prediction, the GAT architecture demonstrated the best performance. Following closely behind is the GraphSAGE architecture, while with a larger margin, the GCN architecture exhibits the poorest performance. All values can be viewed in the table below. We attribute the comparatively poor performance of the GCN architecture to its instability, which is also evident within the classification approach.

	GCN	GraphSAGE	GAT
Test MAE error	3.53	1.69	**1.29**
Test RMSE error	3.72	1.96	**1.65**
Execution Time *(HH:MM:SS)*	00:27:48	**00:24:12**	01:07:36

Table 2: Error scores and Execution Times for Regression task (GNN)

We also include the underlying reports and metrics in the appendix. While the performances of the GAT and GraphSAGE architectures regarding the classification problem can be described as moderate, with the GAT architecture outperforming slightly, it is evident that the GCN architecture serves as a naive classifier. In the case of the GAT architecture, a clear trend towards higher ratings, especially with the value 5, becomes apparent when observing the confusion matrix. This inevitably leads to a high recall value of up to 72%, but it compromises the prediction accuracy for other ratings, particularly noticeable in the ratings of 1 to 3. On the other hand, with the GraphSAGE architecture, fewer rating values are neglected (only 4 and 5); however, this architecture lags behind across all metrics.
Moreover, traditional -benchmark- models' hyperparameters are also tuned. This reflects the alpha parameter for ridge regression; colsample_bytree, learning_rate, max_depth, alpha, n_estimators parameters for XGBoost; hidden_layer_sizes_option, activation, relu, solver, sgd, alpha, learning_rate_init parameters for MLP. The selected parameters are shown in the study outputs of respective models in appendix.
The trained models are used to predict the test set as it is being done for GNN models, therefore following metrics are obtained for these benchmark models.

	Ridge Regression	XGBoost	MLP
Test MAE error	0.72	0.72	**0.71**
Test RMSE error	**0.92**	**0.92**	**0.92**
Execution Time *(HH:MM:SS)*	**00:00:04**	00:02:02	04:47:14

Table 3: Error scores and Execution Times for Regression task (Benchmark Models)

The ridge regression is superior than all the models in terms of the execution time with just four seconds and then XGBoost follows it with around two minutes. However, MLP took the most time among all the models with more than four hours.

Even all these three different benchmark models differ in execution time, all are pretty similar in error scores. All three has 0.92 for test RMSE error score and when it comes to MAE, MLP is the best one but the difference is only 0.01.
With the study outputs of the mentioned benchmark models, also the classification reports are created for such models. The reports for the Ridge Regression, XGBoost and MLP models show the performance of the models to classify the rating that each UserID gave to a specific MovieID.
While Ridge Regression shows moderate precision for scores 1.0 and 5.0, it has difficulty with recall, suggesting that it fails to capture all instances of these scores. XGBoost shows similar patterns, even though it achieves high precision, it still faces difficulties with recall across a range of score points. Similarly, while MLP achieves high precision for certain scores, it also faces recall issues, especially for extreme scores.
Overall, while these models do well in precision for certain score categories, they face recall issues across all scores. These classification reports for all respective model can be found in appendix section.
In the end, we suspect that the suboptimal performance of the GNNs within the classification problem stems from the initial design of the use case as a regression problem and its subsequent extension to a classification problem. The results were originally not intended for the latter, despite rounding to exclude continuous values and enable a multi-class classification problem.

6. Conclusion

In conclusion, throughout our study, it is being investigated whether the graph neural networks (GNNs) are state-of-the-art in recommendation systems or not. To achieve this goal a prediction task has been selected. This task was to predict the rating of a user given to a movie. The data from the Netflix Kaggle prize has been used in this context. So in graph neural networks, three different algorithms have been chosen. These were GCN, GraphSAGE, and GAT. Then three different benchmark models were chosen. These were Ridge regression, XGBoost, and MLP. The same prediction task is aimed for all six models and then their error metrics and execution times are compared.
Before deep diving into the comparison, data preprocessing was the first topic to focus on. After the data was obtained, some explorations were done and at that point, it was clear to see there was not much information regarding neither users nor movies. Therefore, the feature engineering step has started and many different combinations are tried. For example, different averages and frequencies are used. Then the dummy clusters, which are the outcome of the NLP function, are used. Also, the random walk features by using either Node2Vec or Deepwalk have been tried to be added to the data, however as mentioned before due to technical incapabilities and time constraints, this step has been skipped.
After that, the models are trained to find the best hyperparameters. Then the models predicted the ratings of the users given to the movies in the test set with these best parameters. The

error metrics and execution times are obtained for all six models. And at last, all the codes for the mentioned steps above are structured in classes so that the notebook is clear and easy to follow.

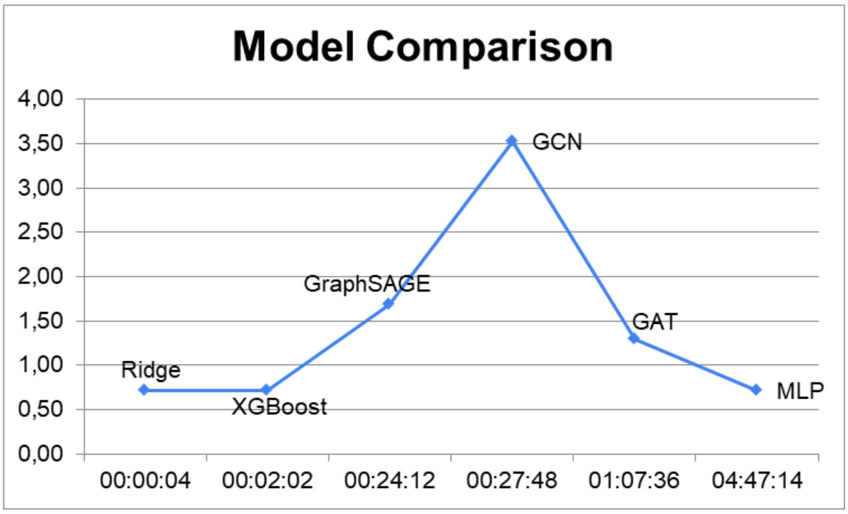

Figure 4: Model Comparison

From the figure above it can be seen that ridge regression and XGBoost are superior to all graph neural network models both in error scores and execution times. Even though MLP has the same level of error scores with ridge regression and XGBoost, it takes the longest time to run the model. In the literature, most of paper that uses GNNs states its success but this is not the case in our study. There might be several reasons for that such as data quality, code efficiency, and feature engineering. In our data, there were not informative variables so it was not easy for us to create meaningful features and feed them to the model.

In future studies, to fully utilize the power of graph neural networks, more information about the users and movies should be checked beforehand such as users' behaviors and movies' genres, so that different combinations and meaningful features can be obtained.

7. Acknowledgments

Finally, we would like to express our gratitude to all the organizers of the seminar who enabled us to explore new innovative topics in the realm of machine learning and presented it in an interesting and engaging manner. Special thanks go to our supervisor, Vincent Gurgul, who has supported us throughout the entire semester, guiding us in the right direction at all times.

Appendix

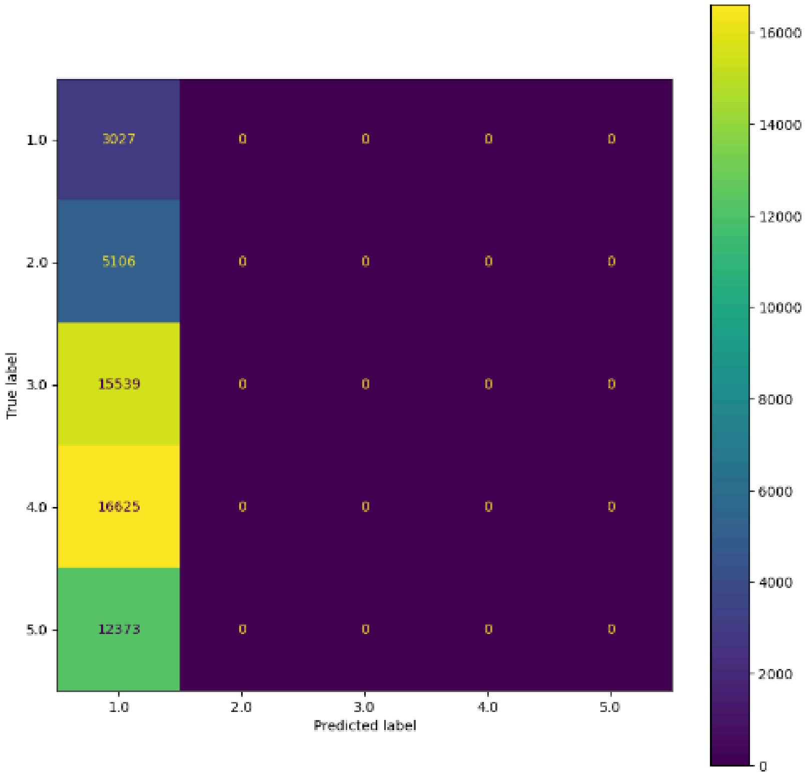

Figure 5: confusion matrix GCN

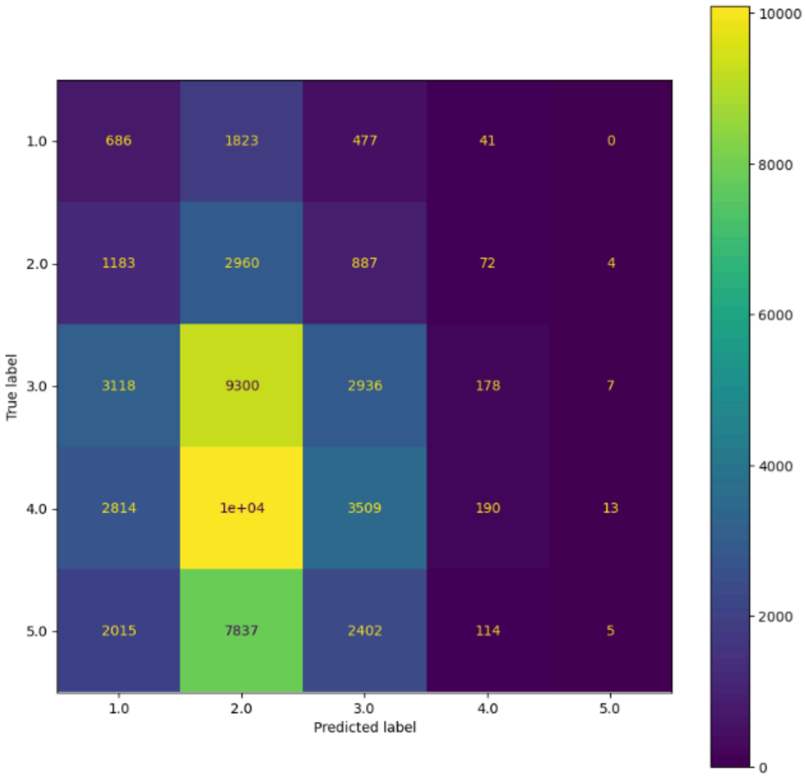

Figure 6: confusion matrix GraphSAGE

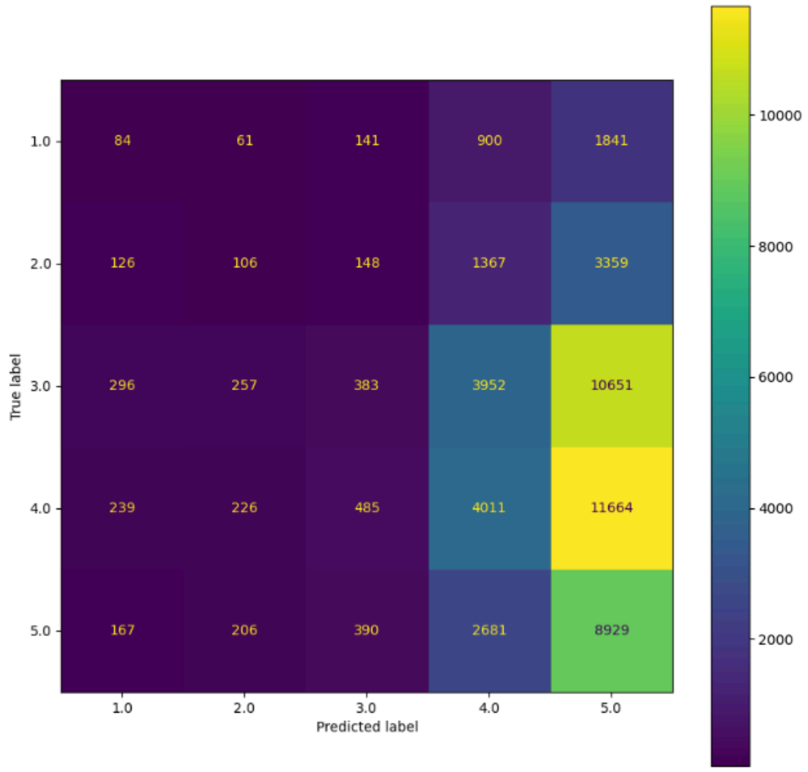

Figure 7: confusion matrix GAT

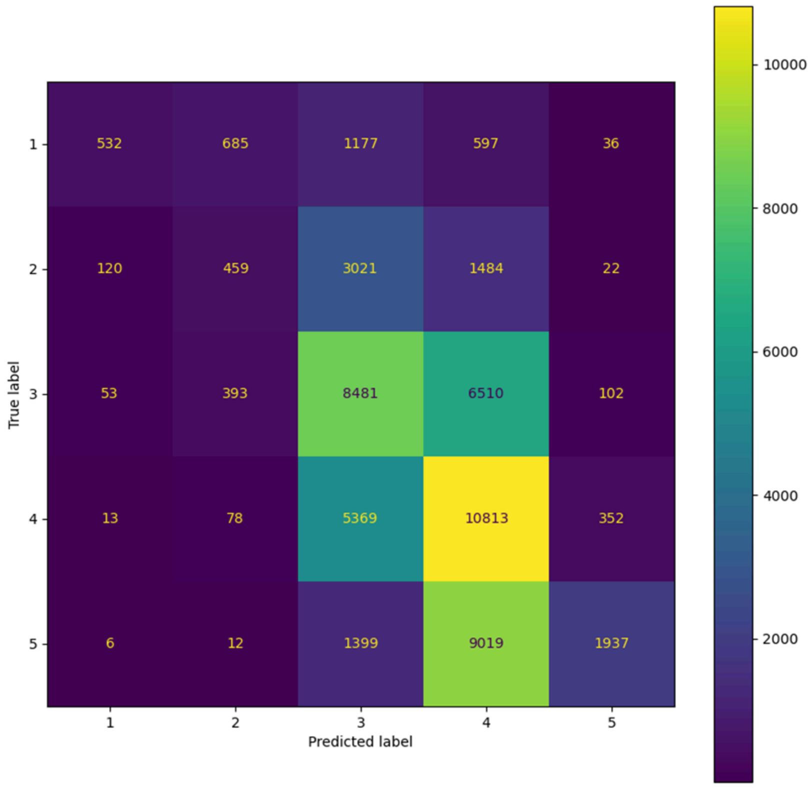

Figure 8: confusion matrix Ridge Regression

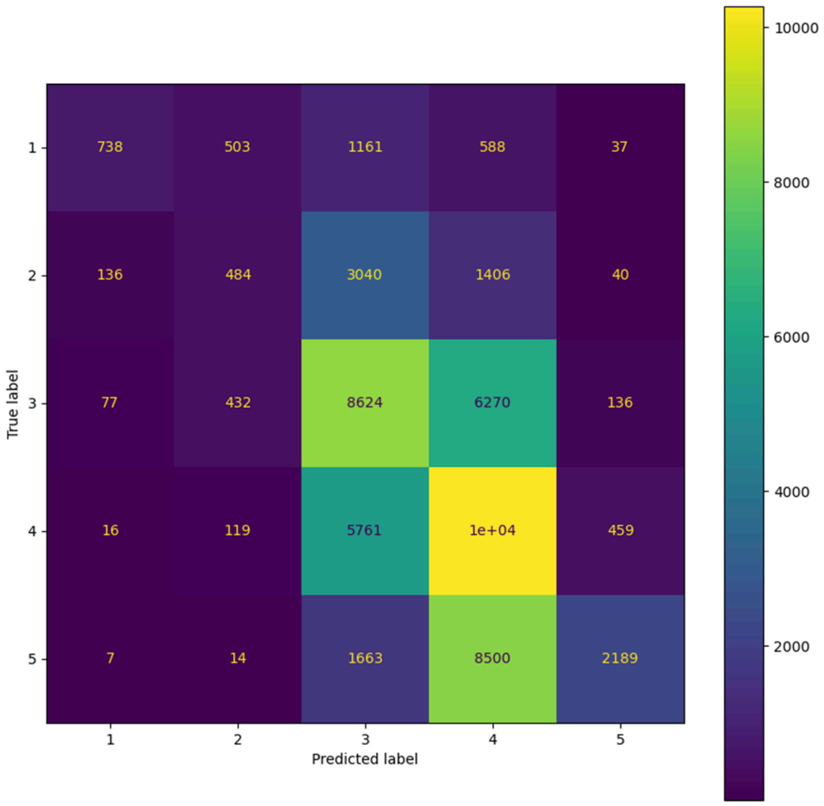

Figure 9: confusion matrix XGBoost

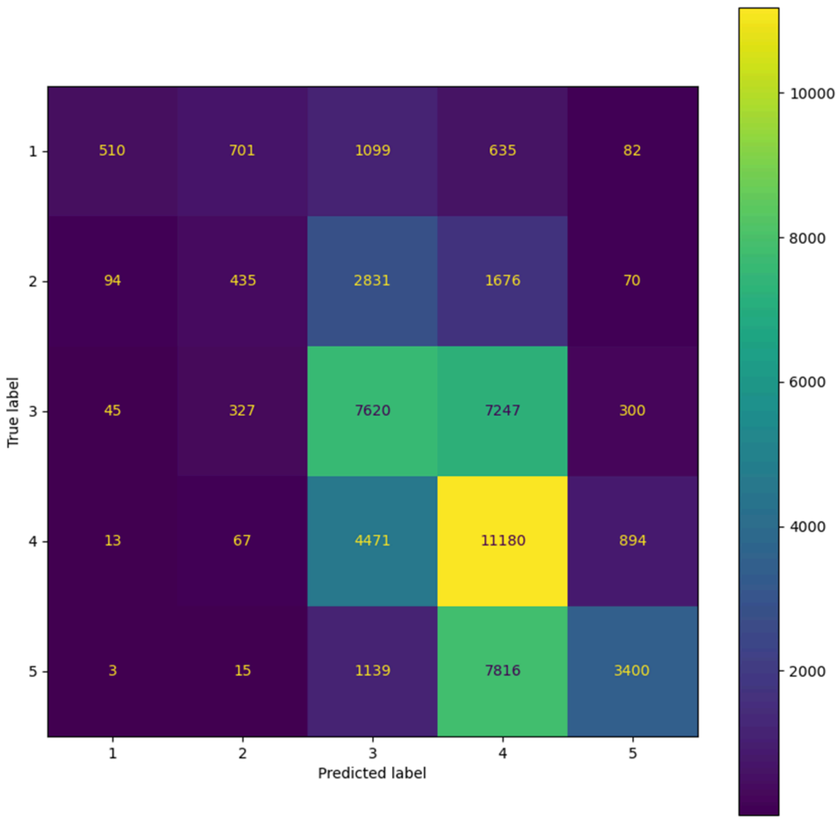

Figure 10: confusion matrix MLP

Rating Score	precision	recall	f1-score	support
1.0	0.06	1.00	0.11	3027
2.0	0.00	0.00	0.00	5106
3.0	0.00	0.00	0.00	15539
4.0	0.00	0.00	0.00	16625
5.0	0.00	0.00	0.00	12373
accuracy			0.06	52670
macro avg	0.01	0.20	0.02	52670
weighted avg	0.00	0.06	0.01	52670

Table 4: classification report GCN

Rating Score	precision	recall	f1-score	support
1.0	0.07	0.23	0.11	3027
2.0	0.09	0.58	0.16	5106
3.0	0.29	0.19	0.23	15539
4.0	0.32	0.01	0.02	16625
5.0	0.17	0.00	0.00	12373
accuracy			0.13	52670
macro avg	0.01	0.20	0.10	52670
weighted avg	0.00	0.06	0.10	52670

Table 5: classification report GraphSAGE

Rating Score	precision	recall	f1-score	support
1.0	0.09	0.03	0.04	3027
2.0	0.12	0.02	0.04	5106
3.0	0.25	0.02	0.04	15539
4.0	0.31	0.24	0.27	16625
5.0	0.25	0.72	0.37	12373
accuracy			0.26	52670
macro avg	0.01	0.20	0.15	52670
weighted avg	0.00	0.06	0.19	52670

Table 6: classification report GAT

Rating Score	precision	recall	f1-score	support
1.0	0.73	0.18	0.28	3027
2.0	0.28	0.09	0.14	5106
3.0	0.44	0.55	0.48	15539
4.0	0.38	0.65	0.48	16625
5.0	0.79	0.16	0.26	12373
accuracy			0.42	52670
macro avg	0.52	0.32	0.33	52670
weighted avg	0.50	0.42	0.39	52670

Table 7: classification report Ridge Regression

Rating Score	precision	recall	f1-score	support
1.0	0.76	0.24	0.37	3027
2.0	0.31	0.09	0.15	5106
3.0	0.43	0.55	0.48	15539
4.0	0.38	0.62	0.47	16625
5.0	0.77	0.18	0.29	12373
accuracy			0.42	52670
macro avg	0.53	0.34	0.35	52670
weighted avg	0.50	0.42	0.39	52670

Table 8: classification report XGBoost

Rating Score	precision	recall	f1-score	support
1.0	0.77	0.17	0.28	3027
2.0	0.28	0.09	0.13	5106
3.0	0.44	0.49	0.47	15539
4.0	0.39	0.67	0.49	16625
5.0	0.72	0.27	0.40	12373
accuracy			0.44	52670
macro avg	0.52	0.34	0.35	52670
weighted avg	0.49	0.44	0.42	52670

Table 9: classification report MLP

Study outcomes for Ridge Regression

[I 2024-03-23 10:52:55,067] A new study created in memory with name: no-name-5ea73bd9-4533-4a48-bca1-54bcf2518caa

[I 2024-03-23 10:52:55,196] Trial 0 finished with value: 0.9079561049871868 and parameters: {'alpha': 0.00010469813645971032}. Best is trial 0 with value: 0.9079561049871868.

[I 2024-03-23 10:52:55,296] Trial 1 finished with value: 0.9079561049831701 and parameters: {'alpha': 0.005125702847462378}. Best is trial 1 with value: 0.9079561049831701.

[I 2024-03-23 10:52:55,403] Trial 2 finished with value: 0.9079561046417357 and parameters: {'alpha': 0.45711469543842115}. Best is trial 2 with value: 0.9079561046417357.

[I 2024-03-23 10:52:55,518] Trial 3 finished with value: 0.9079561039463677 and parameters: {'alpha': 6.594862641053225}. Best is trial 3 with value: 0.9079561039463677.

[I 2024-03-23 10:52:55,646] Trial 4 finished with value: 0.9079561042401438 and parameters: {'alpha': 1.073772938297685}. Best is trial 3 with value: 0.9079561039463677.

[I 2024-03-23 10:52:55,784] Trial 5 finished with value: 0.9079561033461145 and parameters: {'alpha': 4.266384588132632}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:55,910] Trial 6 finished with value: 0.9079561034046647 and parameters: {'alpha': 3.3157577273734016}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:56,024] Trial 7 finished with value: 0.9079561047383788 and parameters: {'alpha': 0.32369433813135}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:56,122] Trial 8 finished with value: 0.9079561049252864 and parameters: {'alpha': 0.07817959370399631}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:56,227] Trial 9 finished with value: 0.9079561048167624 and parameters: {'alpha': 0.21884420965006315}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:56,355] Trial 10 finished with value: 0.9079561049830183 and parameters: {'alpha': 0.005315632780309148}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:56,494] Trial 11 finished with value: 0.9079561050475116 and parameters: {'alpha': 8.290062736526636}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:56,638] Trial 12 finished with value: 0.9079561037278321 and parameters: {'alpha': 2.1213624267942297}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:56,758] Trial 13 finished with value: 0.9079561049718673 and parameters: {'alpha': 0.01928811480994521}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:56,884] Trial 14 finished with value: 0.9079561061631176 and parameters: {'alpha': 9.48811238545385}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:56,989] Trial 15 finished with value: 0.9079561038993649 and parameters: {'alpha': 1.7188656369049424}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:57,099] Trial 16 finished with value: 0.9079561049870659 and parameters: {'alpha': 0.00025558254878367055}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:57,220] Trial 17 finished with value: 0.9079561049231868 and parameters: {'alpha': 0.08085460359555642}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:57,353] Trial 18 finished with value: 0.9079561036082284 and parameters: {'alpha': 2.459197637706113}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:57,493] Trial 19 finished with value: 0.9079561043911459 and parameters: {'alpha': 0.8282659323838323}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:57,600] Trial 20 finished with value: 0.9079561049862825 and parameters: {'alpha': 0.0012343975067105192}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:57,708] Trial 21 finished with value: 0.9079561034252983 and parameters: {'alpha': 3.1916739110234196}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:57,812] Trial 22 finished with value: 0.9079561033744517 and parameters: {'alpha': 3.5448514218656895}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:57,925] Trial 23 finished with value: 0.9079561048898344 and parameters: {'alpha': 0.123584709445235}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:58,078] Trial 24 finished with value: 0.9079561045121203 and parameters: {'alpha': 0.64411824625838}. Best is trial 5 with value: 0.9079561033461145.

[I 2024-03-23 10:52:58,221] Trial 25 finished with value: 0.9079561033450176 and parameters: {'alpha': 4.225850913666109}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:52:58,359] Trial 26 finished with value: 0.9079561049619315 and parameters: {'alpha': 0.03177857734994242}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:52:58,465] Trial 27 finished with value: 0.9079561034032819 and parameters: {'alpha': 4.8896607865482675}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:52:58,579] Trial 28 finished with value: 0.9079561042194971 and parameters: {'alpha': 1.1089035027894487}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:52:58,689] Trial 29 finished with value: 0.907956104838146 and parameters: {'alpha': 0.19072769630003578}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:52:58,813] Trial 30 finished with value: 0.9079561040270243 and parameters: {'alpha': 1.458732185556693}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:52:58,955] Trial 31 finished with value: 0.9079561033559131 and parameters: {'alpha': 4.462072709816713}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:52:59,097] Trial 32 finished with value: 0.9079561033613299 and parameters: {'alpha': 4.533278766556004}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:52:59,214] Trial 33 finished with value: 0.9079561046444912 and parameters: {'alpha': 0.45324300835815345}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:52:59,336] Trial 34 finished with value: 0.9079561051528429 and parameters: {'alpha': 8.417431289127876}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:52:59,440] Trial 35 finished with value: 0.9079561035324266 and parameters: {'alpha': 5.49938446013441}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:52:59,552] Trial 36 finished with value: 0.9079561044559443 and parameters: {'alpha': 0.7283808301049741}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:52:59,683] Trial 37 finished with value: 0.9079561033522867 and parameters: {'alpha': 4.405032800666983}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:52:59,823] Trial 38 finished with value: 0.9079561038527041 and parameters: {'alpha': 1.821324088239677}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:52:59,964] Trial 39 finished with value: 0.9079561047376836 and parameters: {'alpha': 0.324637245615851}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:53:00,071] Trial 40 finished with value: 0.9079561041541543 and parameters: {'alpha': 1.222904866350428}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:53:00,189] Trial 41 finished with value: 0.9079561033568729 and parameters: {'alpha': 4.475692540001629}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:53:00,292] Trial 42 finished with value: 0.9079561033827397 and parameters: {'alpha': 4.740719421989371}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:53:00,404] Trial 43 finished with value: 0.9079561035284331 and parameters: {'alpha': 2.729937288675669}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:53:00,534] Trial 44 finished with value: 0.9079561062193429 and parameters: {'alpha': 9.541508889541944}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:53:00,675] Trial 45 finished with value: 0.9079561036025514 and parameters: {'alpha': 5.737591330452295}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:53:00,819] Trial 46 finished with value: 0.9079561036405424 and parameters: {'alpha': 2.361463981405922}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:53:00,942] Trial 47 finished with value: 0.907956104588157 and parameters: {'alpha': 0.5332287105500165}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:53:01,061] Trial 48 finished with value: 0.9079561049837084 and parameters: {'alpha': 0.004452642721844145}. Best is trial 25 with value: 0.9079561033450176.

[I 2024-03-23 10:53:01,166] Trial 49 finished with value: 0.9079561041666868 and parameters: {'alpha': 1.200693251860141}. Best is trial 25 with value: 0.9079561033450176.

Study outcomes for XGBoost

[I 2024-03-23 14:32:51,520] A new study created in memory with name: no-name-95ef750e-6fff-4a48-9e0b-a317b9beeab4

[I 2024-03-23 14:32:53,133] Trial 0 finished with value: 0.9080809507232026 and parameters: {'colsample_bytree': 0.31513373347738055, 'learning_rate': 0.26605021347908336, 'max_depth': 17, 'alpha': 72.3409924807968, 'n_estimators': 275}. Best is trial 0 with value: 0.9080809507232026.

[I 2024-03-23 14:32:53,581] Trial 1 finished with value: 0.9211080024842597 and parameters: {'colsample_bytree': 0.32847249760775965, 'learning_rate': 0.12449678101489373, 'max_depth': 8, 'alpha': 58.27277032300494, 'n_estimators': 53}. Best is trial 0 with value: 0.9080809507232026.

[I 2024-03-23 14:32:55,970] Trial 2 finished with value: 0.8998732763533168 and parameters: {'colsample_bytree': 0.5096937362922369, 'learning_rate': 0.14413766370013192, 'max_depth': 20, 'alpha': 57.01155412099106, 'n_estimators': 283}. Best is trial 2 with value: 0.8998732763533168.

[I 2024-03-23 14:32:58,595] Trial 3 finished with value: 0.8958819332074539 and parameters: {'colsample_bytree': 0.7642840924200827, 'learning_rate': 0.21977603024704614, 'max_depth': 10, 'alpha': 24.283494433565945, 'n_estimators': 266}. Best is trial 3 with value: 0.8958819332074539.

[I 2024-03-23 14:32:59,489] Trial 4 finished with value: 0.9037460335959512 and parameters: {'colsample_bytree': 0.42325438695558415, 'learning_rate': 0.08710022343362485, 'max_depth': 12, 'alpha': 95.10749625279277, 'n_estimators': 110}. Best is trial 3 with value: 0.8958819332074539.

[I 2024-03-23 14:33:03,387] Trial 5 finished with value: 0.8950412323560732 and parameters: {'colsample_bytree': 0.8392652749033128, 'learning_rate': 0.11933482693214745, 'max_depth': 18, 'alpha': 28.97308923319617, 'n_estimators': 240}. Best is trial 5 with value: 0.8950412323560732.

[I 2024-03-23 14:33:04,874] Trial 6 finished with value: 0.894825400371771 and parameters: {'colsample_bytree': 0.879554239162179, 'learning_rate': 0.1534370486753339, 'max_depth': 16, 'alpha': 17.14518900911758, 'n_estimators': 83}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:05,506] Trial 7 finished with value: 0.9095175952023833 and parameters: {'colsample_bytree': 0.20983474723757164, 'learning_rate': 0.1574595849319752, 'max_depth': 20, 'alpha': 49.04745412337137, 'n_estimators': 110}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:06,158] Trial 8 finished with value: 0.9079509132610493 and parameters: {'colsample_bytree': 0.22178780069241627, 'learning_rate': 0.2866026765295448, 'max_depth': 9, 'alpha': 46.85051927571856, 'n_estimators': 135}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:06,933] Trial 9 finished with value: 0.8988508957372872 and parameters: {'colsample_bytree': 0.7318621229855081, 'learning_rate': 0.0970990894263437, 'max_depth': 8, 'alpha': 41.71230283552498, 'n_estimators': 121}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:07,542] Trial 10 finished with value: 0.9041322484757764 and parameters: {'colsample_bytree': 0.9746665661218687, 'learning_rate': 0.04287432179591737, 'max_depth': 3, 'alpha': 0.97107827344486, 'n_estimators': 197}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:10,633] Trial 11 finished with value: 0.8991808309236624 and parameters: {'colsample_bytree': 0.9769533339774262, 'learning_rate': 0.1906723094516183, 'max_depth': 16, 'alpha': 20.272267149611057, 'n_estimators': 213}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:13,793] Trial 12 finished with value: 0.9036141835276972 and parameters: {'colsample_bytree': 0.7602332448553858, 'learning_rate': 0.014812500470458373, 'max_depth': 15, 'alpha': 20.831132140120836, 'n_estimators': 230}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:15,839] Trial 13 finished with value: 0.9101868813835725 and parameters: {'colsample_bytree': 0.8619369761688102, 'learning_rate': 0.19798426141309883, 'max_depth': 13, 'alpha': 0.20919171291927796, 'n_estimators': 55}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:18,147] Trial 14 finished with value: 0.8958712222464789 and parameters: {'colsample_bytree': 0.6247557154846213, 'learning_rate': 0.07263743013734283, 'max_depth': 18, 'alpha': 30.75206009326747, 'n_estimators': 172}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:22,045] Trial 15 finished with value: 0.9044387946937613 and parameters: {'colsample_bytree': 0.8783851791144756, 'learning_rate': 0.16359800566732802, 'max_depth': 14, 'alpha': 10.541067212942249, 'n_estimators': 241}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:23,878] Trial 16 finished with value: 0.8963568741038743 and parameters: {'colsample_bytree': 0.6463429621936654, 'learning_rate': 0.23918407994215224, 'max_depth': 17, 'alpha': 36.96626386051735, 'n_estimators': 165}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:25,812] Trial 17 finished with value: 0.8958233018830012 and parameters: {'colsample_bytree': 0.6246656636681586, 'learning_rate': 0.10427438667182522, 'max_depth': 19, 'alpha': 12.251087102725833, 'n_estimators': 90}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:26,505] Trial 18 finished with value: 0.900653062766924 and parameters: {'colsample_bytree': 0.8672221695661724, 'learning_rate': 0.131543763087432, 'max_depth': 5, 'alpha': 70.1707900974398, 'n_estimators': 156}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:28,075] Trial 19 finished with value: 0.9006703278765318 and parameters: {'colsample_bytree': 0.5232802277667126, 'learning_rate': 0.05718647286058515, 'max_depth': 15, 'alpha': 34.957442085739814, 'n_estimators': 197}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:30,628] Trial 20 finished with value: 0.8989007539294626 and parameters: {'colsample_bytree': 0.9020100257877242, 'learning_rate': 0.17457525307790314, 'max_depth': 11, 'alpha': 10.260506724727414, 'n_estimators': 251}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:32,385] Trial 21 finished with value: 0.8960912873637595 and parameters: {'colsample_bytree': 0.6775765987319918, 'learning_rate': 0.11814500400596141, 'max_depth': 19, 'alpha': 12.749563408218417, 'n_estimators': 88}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:33,428] Trial 22 finished with value: 0.8974781141335872 and parameters: {'colsample_bytree': 0.7941822952744271, 'learning_rate': 0.09849237979988146, 'max_depth': 18, 'alpha': 27.59929673721182, 'n_estimators': 80}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:34,356] Trial 23 finished with value: 0.9005373802755183 and parameters: {'colsample_bytree': 0.5894473814864363, 'learning_rate': 0.12996783576139406, 'max_depth': 18, 'alpha': 16.01164548269088, 'n_estimators': 75}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:37,596] Trial 24 finished with value: 0.899501843638145 and parameters: {'colsample_bytree': 0.710818113321046, 'learning_rate': 0.09909174169639799, 'max_depth': 16, 'alpha': 6.0386606826839655, 'n_estimators': 145}. Best is trial 6 with value: 0.894825400371771.

[I 2024-03-23 14:33:42,150] Trial 25 finished with value: 0.8941750891557518 and parameters: {'colsample_bytree': 0.8083553970795299, 'learning_rate': 0.04385439812653531, 'max_depth': 20, 'alpha': 27.645552139138914, 'n_estimators': 296}. Best is trial 25 with value: 0.8941750891557518.

[I 2024-03-23 14:33:45,457] Trial 26 finished with value: 0.8992386010649667 and parameters: {'colsample_bytree': 0.8297005882792018, 'learning_rate': 0.01375154331105596, 'max_depth': 14, 'alpha': 29.074388674405533, 'n_estimators': 295}. Best is trial 25 with value: 0.8941750891557518.

[I 2024-03-23 14:33:48,689] Trial 27 finished with value: 0.895451105132435 and parameters: {'colsample_bytree': 0.9409139840096171, 'learning_rate': 0.044594475223483754, 'max_depth': 20, 'alpha': 39.843920997117024, 'n_estimators': 260}. Best is trial 25 with value: 0.8941750891557518.

[I 2024-03-23 14:33:53,071] Trial 28 finished with value: 0.8949286971772523 and parameters: {'colsample_bytree': 0.8133020551749609, 'learning_rate': 0.07530075464866096, 'max_depth': 17, 'alpha': 20.726968453301417, 'n_estimators': 298}. Best is trial 25 with value: 0.8941750891557518.

[I 2024-03-23 14:33:55,588] Trial 29 finished with value: 0.8984332716046906 and parameters: {'colsample_bytree': 0.9254142039658466, 'learning_rate': 0.06817900293465601, 'max_depth': 17, 'alpha': 89.7300610148256, 'n_estimators': 298}. Best is trial 25 with value: 0.8941750891557518.

[I 2024-03-23 14:33:59,221] Trial 30 finished with value: 0.8946557124019854 and parameters: {'colsample_bytree': 0.797992152739072, 'learning_rate': 0.04879341716866286, 'max_depth': 15, 'alpha': 18.082528168634852, 'n_estimators': 279}. Best is trial 25 with value: 0.8941750891557518.

[I 2024-03-23 14:34:03,746] Trial 31 finished with value: 0.8939675358902199 and parameters: {'colsample_bytree': 0.8042476445515151, 'learning_rate': 0.03413183673289817, 'max_depth': 16, 'alpha': 18.834533332168487, 'n_estimators': 273}. Best is trial 31 with value: 0.8939675358902199.

[I 2024-03-23 14:34:07,371] Trial 32 finished with value: 0.8939966377547777 and parameters: {'colsample_bytree': 0.9974071058632458, 'learning_rate': 0.03455313093426715, 'max_depth': 13, 'alpha': 17.263247094327635, 'n_estimators': 277}. Best is trial 31 with value: 0.8939675358902199.

[I 2024-03-23 14:34:09,270] Trial 33 finished with value: 0.9025770776869244 and parameters: {'colsample_bytree': 0.43006414403163395, 'learning_rate': 0.033826478150956164, 'max_depth': 12, 'alpha': 56.89437927494605, 'n_estimators': 278}. Best is trial 31 with value: 0.8939675358902199.

[I 2024-03-23 14:34:14,238] Trial 34 finished with value: 0.8946291970026876 and parameters: {'colsample_bytree': 0.9977037570495122, 'learning_rate': 0.029862742744793162, 'max_depth': 13, 'alpha': 5.649846204031697, 'n_estimators': 274}. Best is trial 31 with value: 0.8939675358902199.

[I 2024-03-23 14:34:19,853] Trial 35 finished with value: 0.8945929473980989 and parameters: {'colsample_bytree': 0.9842409546589967, 'learning_rate': 0.02592094308911674, 'max_depth': 13, 'alpha': 5.776656144643492, 'n_estimators': 265}. Best is trial 31 with value: 0.8939675358902199.

[I 2024-03-23 14:34:23,400] Trial 36 finished with value: 0.8942681863207043 and parameters: {'colsample_bytree': 0.9393181072356842, 'learning_rate': 0.02575234823785392, 'max_depth': 11, 'alpha': 5.112954694177763, 'n_estimators': 260}. Best is trial 31 with value: 0.8939675358902199.

[I 2024-03-23 14:34:25,725] Trial 37 finished with value: 0.9056647764247094 and parameters: {'colsample_bytree': 0.9343445589313398, 'learning_rate': 0.010516873495821115, 'max_depth': 11, 'alpha': 24.021292426361637, 'n_estimators': 230}. Best is trial 31 with value: 0.8939675358902199.

[I 2024-03-23 14:34:26,922] Trial 38 finished with value: 0.9095012549797732 and parameters: {'colsample_bytree': 0.14010310374326995, 'learning_rate': 0.05984702247974886, 'max_depth': 7, 'alpha': 34.39908743586417, 'n_estimators': 254}. Best is trial 31 with value: 0.8939675358902199.

[I 2024-03-23 14:34:29,274] Trial 39 finished with value: 0.896877088772638 and parameters: {'colsample_bytree': 0.7607775027488491, 'learning_rate': 0.031952422181715005, 'max_depth': 10, 'alpha': 23.854830205681253, 'n_estimators': 289}. Best is trial 31 with value: 0.8939675358902199.

[I 2024-03-23 14:34:30,622] Trial 40 finished with value: 0.9016325821928634 and parameters: {'colsample_bytree': 0.4467101475560775, 'learning_rate': 0.0772005770885042, 'max_depth': 9, 'alpha': 70.91830009208223, 'n_estimators': 218}. Best is trial 31 with value: 0.8939675358902199.

[I 2024-03-23 14:34:36,036] Trial 41 finished with value: 0.8950350596250923 and parameters: {'colsample_bytree': 0.9976772590839849, 'learning_rate': 0.024992557714369767, 'max_depth': 13, 'alpha': 4.492220820810811, 'n_estimators': 260}. Best is trial 31 with value: 0.8939675358902199.

[I 2024-03-23 14:34:39,252] Trial 42 finished with value: 0.893774858323338 and parameters: {'colsample_bytree': 0.9410171837341702, 'learning_rate': 0.057184343807326135, 'max_depth': 12, 'alpha': 9.139772177346375, 'n_estimators': 270}. Best is trial 42 with value: 0.893774858323338.

[I 2024-03-23 14:34:41,584] Trial 43 finished with value: 0.8941962407945241 and parameters: {'colsample_bytree': 0.9171861830258665, 'learning_rate': 0.05352676353040474, 'max_depth': 10, 'alpha': 15.585949935754645, 'n_estimators': 274}. Best is trial 42 with value: 0.893774858323338.

[I 2024-03-23 14:34:43,949] Trial 44 finished with value: 0.8943006843271704 and parameters: {'colsample_bytree': 0.8484863218948275, 'learning_rate': 0.048551328595252005, 'max_depth': 10, 'alpha': 13.619702821515304, 'n_estimators': 282}. Best is trial 42 with value: 0.893774858323338.

[I 2024-03-23 14:34:45,311] Trial 45 finished with value: 0.8961049541702399 and parameters: {'colsample_bytree': 0.8936438064624262, 'learning_rate': 0.08499016161183895, 'max_depth': 7, 'alpha': 25.699474215718354, 'n_estimators': 270}. Best is trial 42 with value: 0.893774858323338.

[I 2024-03-23 14:34:46,486] Trial 46 finished with value: 0.9100909527403169 and parameters: {'colsample_bytree': 0.3609460642977271, 'learning_rate': 0.059465306946716937, 'max_depth': 12, 'alpha': 45.33571906012253, 'n_estimators': 246}. Best is trial 42 with value: 0.893774858323338.

[I 2024-03-23 14:34:48,465] Trial 47 finished with value: 0.896031216112491 and parameters: {'colsample_bytree': 0.7319314565799253, 'learning_rate': 0.04108359994439799, 'max_depth': 9, 'alpha': 15.578611461081195, 'n_estimators': 284}. Best is trial 42 with value: 0.893774858323338.

[I 2024-03-23 14:34:51,008] Trial 48 finished with value: 0.8947646287878243 and parameters: {'colsample_bytree': 0.9534146969980122, 'learning_rate': 0.06527488605440239, 'max_depth': 14, 'alpha': 32.91394855937207, 'n_estimators': 234}. Best is trial 42 with value: 0.893774858323338.

[I 2024-03-23 14:34:53,268] Trial 49 finished with value: 0.8942260922139429 and parameters: {'colsample_bytree': 0.902221945412649, 'learning_rate': 0.11172078026633771, 'max_depth': 10, 'alpha': 9.278698182915123, 'n_estimators': 271}. Best is trial 42 with value: 0.893774858323338.

<u>Study outcomes for MLP</u>

[I 2024-03-23 15:04:33,783] A new study created in memory with name: no-name-fdcddb81-be17-4a51-97e3-36f06c036d3a

[I 2024-03-23 15:08:21,170] Trial 0 finished with value: 0.8302019872665609 and parameters: {'hidden_layer_sizes_option': '100', 'activation': 'relu', 'solver': 'adam', 'alpha': 0.0014012986240809562, 'learning_rate_init': 1.19989899232219e-06}. Best is trial 0 with value: 0.8302019872665609.

[I 2024-03-23 15:08:33,858] Trial 1 finished with value: 0.8185338097607722 and parameters: {'hidden_layer_sizes_option': '50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 0.0001899999408013173, 'learning_rate_init': 0.0015214515742505482}. Best is trial 1 with value: 0.8185338097607722.

[I 2024-03-23 15:09:03,446] Trial 2 finished with value: 0.8186705435464332 and parameters: {'hidden_layer_sizes_option': '100', 'activation': 'relu', 'solver': 'adam', 'alpha': 0.0003356664885780762, 'learning_rate_init': 4.600653582164313e-05}. Best is trial 1 with value: 0.8185338097607722.

[I 2024-03-23 15:09:22,745] Trial 3 finished with value: 0.8451874369679585 and parameters: {'hidden_layer_sizes_option': '50', 'activation': 'tanh', 'solver': 'sgd', 'alpha': 0.016172260680902308, 'learning_rate_init': 7.89695582848984e-06}. Best is trial 1 with value: 0.8185338097607722.

[I 2024-03-23 15:14:24,555] Trial 4 finished with value: 0.8222675684177307 and parameters: {'hidden_layer_sizes_option': '100_100', 'activation': 'tanh', 'solver': 'sgd', 'alpha': 0.0023818306317875265, 'learning_rate_init': 0.0001521656064244397}. Best is trial 1 with value: 0.8185338097607722.

[I 2024-03-23 15:30:37,351] Trial 5 finished with value: 0.8309733779612426 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'tanh', 'solver': 'adam', 'alpha': 0.016266157631262813, 'learning_rate_init': 2.2752789265242656e-06}. Best is trial 1 with value: 0.8185338097607722.

[I 2024-03-23 15:45:43,140] Trial 6 finished with value: 0.8375256091181214 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'tanh', 'solver': 'sgd', 'alpha': 4.862941372532062e-05, 'learning_rate_init': 9.445316026868337e-06}. Best is trial 1 with value: 0.8185338097607722.

[I 2024-03-23 15:56:26,768] Trial 7 finished with value: 0.8264688396704624 and parameters: {'hidden_layer_sizes_option': '100_100', 'activation': 'relu', 'solver': 'sgd', 'alpha': 0.005142007243762936, 'learning_rate_init': 8.0150467698532e-06}. Best is trial 1 with value: 0.8185338097607722.

[I 2024-03-23 16:00:23,947] Trial 8 finished with value: 0.8185016271349415 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 1.1170409538258965e-05, 'learning_rate_init': 0.00043458975788141754}. Best is trial 8 with value: 0.8185016271349415.

[I 2024-03-23 16:00:54,904] Trial 9 finished with value: 0.8299709108029725 and parameters: {'hidden_layer_sizes_option': '50', 'activation': 'tanh', 'solver': 'adam', 'alpha': 0.032909295797935986, 'learning_rate_init': 6.061214874048555e-06}. Best is trial 8 with value: 0.8185016271349415.

[I 2024-03-23 16:08:33,021] Trial 10 finished with value: 0.8169259534404072 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 1.7866865955501047e-05, 'learning_rate_init': 0.008725770944763908}. Best is trial 10 with value: 0.8169259534404072.

[I 2024-03-23 16:14:43,127] Trial 11 finished with value: 0.8160160896442521 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 1.1097267207583997e-05, 'learning_rate_init': 0.009858718269782936}. Best is trial 11 with value: 0.8160160896442521.

[I 2024-03-23 16:22:01,559] Trial 12 finished with value: 0.8158937468406888 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 1.013467144798458e-05, 'learning_rate_init': 0.008457148217607123}. Best is trial 12 with value: 0.8158937468406888.

[I 2024-03-23 16:30:39,832] Trial 13 finished with value: 0.8141676058664161 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 6.888714301032338e-05, 'learning_rate_init': 0.00990556611823484}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 16:34:45,414] Trial 14 finished with value: 0.8168689157711168 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 9.06019119047062e-05, 'learning_rate_init': 0.0027668096088442735}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 16:40:01,755] Trial 15 finished with value: 0.8169594144873322 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 5.911684749345385e-05, 'learning_rate_init': 0.0016111144605571196}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 16:44:16,604] Trial 16 finished with value: 0.8188927644407309 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 0.0004162890764446877, 'learning_rate_init': 0.00030436737789919826}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 16:49:22,484] Trial 17 finished with value: 0.8174465385916267 and parameters: {'hidden_layer_sizes_option': '100_100', 'activation': 'relu', 'solver': 'sgd', 'alpha': 3.0363622130511028e-05, 'learning_rate_init': 0.003373407967545083}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 16:49:51,783] Trial 18 finished with value: 0.8174740340467195 and parameters: {'hidden_layer_sizes_option': '100', 'activation': 'relu', 'solver': 'adam', 'alpha': 0.00012997080293300706, 'learning_rate_init': 0.0005890650274760266}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 16:55:22,803] Trial 19 finished with value: 0.82389619943622 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 0.0005890835013956926, 'learning_rate_init': 4.981943645185833e-05}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 17:02:38,568] Trial 20 finished with value: 0.814893462063028 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 3.0035233817814197e-05, 'learning_rate_init': 0.0048026583620825}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 17:08:33,530] Trial 21 finished with value: 0.8163752002513818 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 3.136063982022364e-05, 'learning_rate_init': 0.004716152409737957}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 17:12:29,019] Trial 22 finished with value: 0.8174550615655436 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 2.534718693125634e-05, 'learning_rate_init': 0.0009871796091506593}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 17:18:27,811] Trial 23 finished with value: 0.8162322344953025 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 1.0032610365004105e-05, 'learning_rate_init': 0.005102308064356801}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 17:24:50,496] Trial 24 finished with value: 0.8167915372074404 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 7.235856725065697e-05, 'learning_rate_init': 0.009021750058328443}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 17:28:51,902] Trial 25 finished with value: 0.81782847281894 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 0.00027553331035032894, 'learning_rate_init': 0.001905200977467709}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 17:29:13,253] Trial 26 finished with value: 0.8175910266316809 and parameters: {'hidden_layer_sizes_option': '100', 'activation': 'relu', 'solver': 'sgd', 'alpha': 3.816560363644917e-05, 'learning_rate_init': 0.0008341086277483468}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 17:36:23,839] Trial 27 finished with value: 0.8312642443773981 and parameters: {'hidden_layer_sizes_option': '100_100', 'activation': 'tanh', 'solver': 'adam', 'alpha': 0.00014814825881275795, 'learning_rate_init': 0.005053950013032162}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 17:36:37,649] Trial 28 finished with value: 0.8216432457876843 and parameters: {'hidden_layer_sizes_option': '50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 1.78833457287723e-05, 'learning_rate_init': 0.00020508737185558835}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 17:37:15,368] Trial 29 finished with value: 0.8172531920655577 and parameters: {'hidden_layer_sizes_option': '100', 'activation': 'relu', 'solver': 'adam', 'alpha': 0.001150283282494946, 'learning_rate_init': 0.0021516469209559423}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 17:41:39,147] Trial 30 finished with value: 0.8160639228921686 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 0.00222426915134079, 'learning_rate_init': 0.004955689461832601}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 17:49:25,462] Trial 31 finished with value: 0.8165335296758565 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 1.616165142070838e-05, 'learning_rate_init': 0.009300570860959772}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 17:58:25,831] Trial 32 finished with value: 0.8148604129424066 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 1.0114943876177236e-05, 'learning_rate_init': 0.00992369276394859}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 18:02:53,401] Trial 33 finished with value: 0.8163356229250684 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 9.705575850343893e-05, 'learning_rate_init': 0.0032400789165560246}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 18:03:08,318] Trial 34 finished with value: 0.8189679935201163 and parameters: {'hidden_layer_sizes_option': '50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 2.154660154435542e-05, 'learning_rate_init': 0.0011274482688328673}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 18:07:19,781] Trial 35 finished with value: 0.817053615134941 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 5.070636152964709e-05, 'learning_rate_init': 0.0064664710642487145}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 18:14:17,240] Trial 36 finished with value: 0.8248970919189528 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'tanh', 'solver': 'adam', 'alpha': 0.00023722755027106207, 'learning_rate_init': 0.003215871748009678}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 18:14:47,760] Trial 37 finished with value: 0.8232340780388486 and parameters: {'hidden_layer_sizes_option': '100', 'activation': 'relu', 'solver': 'sgd', 'alpha': 3.670378529378675e-05, 'learning_rate_init': 4.4270251929848295e-05}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 18:19:13,208] Trial 38 finished with value: 0.8177432553423344 and parameters: {'hidden_layer_sizes_option': '100_100', 'activation': 'tanh', 'solver': 'sgd', 'alpha': 1.4662104672562198e-05, 'learning_rate_init': 0.001581703387562406}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 18:56:12,333] Trial 39 finished with value: 0.8479737715531295 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 2.4062039987925288e-05, 'learning_rate_init': 1.2408931084059615e-06}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 18:56:32,154] Trial 40 finished with value: 0.8229426473784627 and parameters: {'hidden_layer_sizes_option': '50', 'activation': 'tanh', 'solver': 'adam', 'alpha': 0.07808358843649515, 'learning_rate_init': 0.006632248816336127}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 19:02:45,245] Trial 41 finished with value: 0.8158184059255045 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 1.085904147120976e-05, 'learning_rate_init': 0.008835137489450104}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 19:08:17,734] Trial 42 finished with value: 0.8154957045603548 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 1.2101193956603372e-05, 'learning_rate_init': 0.004065894486987299}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 19:12:29,824] Trial 43 finished with value: 0.8162804665687726 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 4.610891191255149e-05, 'learning_rate_init': 0.003958668831368689}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 19:19:06,478] Trial 44 finished with value: 0.816238493032178 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 1.5496436860225094e-05, 'learning_rate_init': 0.0023712717276196723}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 19:24:41,650] Trial 45 finished with value: 0.8169940696360596 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 2.3690209337731003e-05, 'learning_rate_init': 0.006798148717228781}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 19:29:36,905] Trial 46 finished with value: 0.8219140962921536 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 0.004342847112173223, 'learning_rate_init': 8.873456286074733e-05}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 19:34:16,749] Trial 47 finished with value: 0.816858975780471 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'relu', 'solver': 'sgd', 'alpha': 6.828492133670785e-05, 'learning_rate_init': 0.001201349380629229}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 19:38:03,839] Trial 48 finished with value: 0.8168634113263751 and parameters: {'hidden_layer_sizes_option': '100_100', 'activation': 'relu', 'solver': 'sgd', 'alpha': 1.1701900036543365e-05, 'learning_rate_init': 0.0005763983320585559}. Best is trial 13 with value: 0.8141676058664161.

[I 2024-03-23 19:51:47,795] Trial 49 finished with value: 0.8319716671896167 and parameters: {'hidden_layer_sizes_option': '50_50', 'activation': 'tanh', 'solver': 'sgd', 'alpha': 0.00010614247830278464, 'learning_rate_init': 1.6287436714720825e-05}. Best is trial 13 with value: 0.8141676058664161.

Study outcomes for GCN

[I 2024-03-21 19:43:50,474] A new study created in memory with name: no-name-00db722e-ec35-498c-b1db-64ec1ab093b6

[I 2024-03-21 19:44:10,334] Trial 0 finished with value: 7414077.405661961 and parameters: {'hidden_channels_encoder': 107, 'latent_space_dim': 45, 'hidden_channels_predictor': 102, 'dropout_rate': 0.7827661712540636}. Best is trial 0 with value: 7414077.405661961.

[I 2024-03-21 19:45:03,756] Trial 1 finished with value: 1026686.2995457497 and parameters: {'hidden_channels_encoder': 236, 'latent_space_dim': 174, 'hidden_channels_predictor': 234, 'dropout_rate': 0.14103689383074053}. Best is trial 1 with value: 1026686.2995457497.

[I 2024-03-21 19:45:40,074] Trial 2 finished with value: 3132180.2158025424 and parameters: {'hidden_channels_encoder': 32, 'latent_space_dim': 158, 'hidden_channels_predictor': 217, 'dropout_rate': 0.07539593184526837}. Best is trial 1 with value: 1026686.2995457497.

[I 2024-03-21 19:46:25,584] Trial 3 finished with value: 3341267.0405901764 and parameters: {'hidden_channels_encoder': 184, 'latent_space_dim': 146, 'hidden_channels_predictor': 223, 'dropout_rate': 0.304915524084419}. Best is trial 1 with value: 1026686.2995457497.

[I 2024-03-21 19:46:57,778] Trial 4 finished with value: 1046689.0523954988 and parameters: {'hidden_channels_encoder': 205, 'latent_space_dim': 113, 'hidden_channels_predictor': 112, 'dropout_rate': 0.17437070447389777}. Best is trial 1 with value: 1026686.2995457497.

[I 2024-03-21 19:47:28,207] Trial 5 finished with value: 1448870.3930802655 and parameters: {'hidden_channels_encoder': 142, 'latent_space_dim': 128, 'hidden_channels_predictor': 130, 'dropout_rate': 0.41115456046233906}. Best is trial 1 with value: 1026686.2995457497.

[I 2024-03-21 19:47:55,363] Trial 6 finished with value: 31227.890987908773 and parameters: {'hidden_channels_encoder': 105, 'latent_space_dim': 177, 'hidden_channels_predictor': 58, 'dropout_rate': 0.7263962365053943}. Best is trial 6 with value: 31227.890987908773.

[I 2024-03-21 19:48:32,072] Trial 7 finished with value: 1905017.790537771 and parameters: {'hidden_channels_encoder': 145, 'latent_space_dim': 79, 'hidden_channels_predictor': 197, 'dropout_rate': 0.28537053935566886}. Best is trial 6 with value: 31227.890987908773.

[I 2024-03-21 19:49:10,391] Trial 8 finished with value: 203720.17404200806 and parameters: {'hidden_channels_encoder': 222, 'latent_space_dim': 122, 'hidden_channels_predictor': 145, 'dropout_rate': 0.7244687324795873}. Best is trial 6 with value: 31227.890987908773.

[I 2024-03-21 19:49:38,350] Trial 9 finished with value: 2401663.4972389154 and parameters: {'hidden_channels_encoder': 181, 'latent_space_dim': 43, 'hidden_channels_predictor': 149, 'dropout_rate': 0.8027694553563935}. Best is trial 6 with value: 31227.890987908773.

[I 2024-03-21 19:50:06,378] Trial 10 finished with value: 186982.31928728646 and parameters: {'hidden_channels_encoder': 70, 'latent_space_dim': 233, 'hidden_channels_predictor': 39, 'dropout_rate': 0.6444394278722367}. Best is trial 6 with value: 31227.890987908773.

[I 2024-03-21 19:50:32,469] Trial 11 finished with value: 28231.090507223955 and parameters: {'hidden_channels_encoder': 64, 'latent_space_dim': 240, 'hidden_channels_predictor': 33, 'dropout_rate': 0.6098445189934013}. Best is trial 11 with value: 28231.090507223955.

[I 2024-03-21 19:51:00,353] Trial 12 finished with value: 18.38125253871321 and parameters: {'hidden_channels_encoder': 80, 'latent_space_dim': 250, 'hidden_channels_predictor': 33, 'dropout_rate': 0.5690963789492898}. Best is trial 12 with value: 18.38125253871321.

[I 2024-03-21 19:51:32,454] Trial 13 finished with value: 231186.7010682809 and parameters: {'hidden_channels_encoder': 44, 'latent_space_dim': 251, 'hidden_channels_predictor': 70, 'dropout_rate': 0.5460911761382139}. Best is trial 12 with value: 18.38125253871321.

[I 2024-03-21 19:51:59,288] Trial 14 finished with value: 212.6828511005322 and parameters: {'hidden_channels_encoder': 83, 'latent_space_dim': 218, 'hidden_channels_predictor': 38, 'dropout_rate': 0.5288321434036292}. Best is trial 12 with value: 18.38125253871321.

[I 2024-03-21 19:52:30,351] Trial 15 finished with value: 430673.67789050064 and parameters: {'hidden_channels_encoder': 92, 'latent_space_dim': 201, 'hidden_channels_predictor': 82, 'dropout_rate': 0.4853555795433907}. Best is trial 12 with value: 18.38125253871321.

[I 2024-03-21 19:53:00,952] Trial 16 finished with value: 27704.20326963104 and parameters: {'hidden_channels_encoder': 135, 'latent_space_dim': 210, 'hidden_channels_predictor': 60, 'dropout_rate': 0.38263173487370417}. Best is trial 12 with value: 18.38125253871321.

[I 2024-03-21 19:53:42,776] Trial 17 finished with value: 679897.2274949648 and parameters: {'hidden_channels_encoder': 75, 'latent_space_dim': 213, 'hidden_channels_predictor': 188, 'dropout_rate': 0.5237963504536081}. Best is trial 12 with value: 18.38125253871321.

[I 2024-03-21 19:54:18,263] Trial 18 finished with value: 3.6009739929590223 and parameters: {'hidden_channels_encoder': 122, 'latent_space_dim': 255, 'hidden_channels_predictor': 97, 'dropout_rate': 0.8970890334296011}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 19:54:54,024] Trial 19 finished with value: 831.1349642861794 and parameters: {'hidden_channels_encoder': 122, 'latent_space_dim': 249, 'hidden_channels_predictor': 97, 'dropout_rate': 0.8845415139287217}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 19:55:35,884] Trial 20 finished with value: 24983.32875374701 and parameters: {'hidden_channels_encoder': 163, 'latent_space_dim': 191, 'hidden_channels_predictor': 164, 'dropout_rate': 0.8980428480071303}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 19:56:04,921] Trial 21 finished with value: 570.0904457422214 and parameters: {'hidden_channels_encoder': 90, 'latent_space_dim': 225, 'hidden_channels_predictor': 48, 'dropout_rate': 0.6033841269276551}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 19:56:39,748] Trial 22 finished with value: 488719.13915626495 and parameters: {'hidden_channels_encoder': 112, 'latent_space_dim': 256, 'hidden_channels_predictor': 80, 'dropout_rate': 0.333889541069459}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 19:57:06,275] Trial 23 finished with value: 917.0308528859898 and parameters: {'hidden_channels_encoder': 62, 'latent_space_dim': 225, 'hidden_channels_predictor': 34, 'dropout_rate': 0.6474328115401038}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 19:58:06,556] Trial 24 finished with value: 1754005.2800012152 and parameters: {'hidden_channels_encoder': 84, 'latent_space_dim': 227, 'hidden_channels_predictor': 255, 'dropout_rate': 0.45611925295447114}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 19:58:34,992] Trial 25 finished with value: 2739993.2784682075 and parameters: {'hidden_channels_encoder': 51, 'latent_space_dim': 190, 'hidden_channels_predictor': 83, 'dropout_rate': 0.2243525668986544}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 19:59:15,826] Trial 26 finished with value: 1366.6004466945062 and parameters: {'hidden_channels_encoder': 123, 'latent_space_dim': 240, 'hidden_channels_predictor': 122, 'dropout_rate': 0.7229621080053186}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 19:59:48,821] Trial 27 finished with value: 3.648540861119038 and parameters: {'hidden_channels_encoder': 159, 'latent_space_dim': 211, 'hidden_channels_predictor': 49, 'dropout_rate': 0.543946352962701}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 20:00:23,519] Trial 28 finished with value: 3.673519921872152 and parameters: {'hidden_channels_encoder': 152, 'latent_space_dim': 255, 'hidden_channels_predictor': 61, 'dropout_rate': 0.8383266064504825}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 20:00:45,881] Trial 29 finished with value: 1865036.8642176657 and parameters: {'hidden_channels_encoder': 156, 'latent_space_dim': 74, 'hidden_channels_predictor': 98, 'dropout_rate': 0.8196106525049548}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 20:01:17,812] Trial 30 finished with value: 169.55053819337306 and parameters: {'hidden_channels_encoder': 181, 'latent_space_dim': 198, 'hidden_channels_predictor': 70, 'dropout_rate': 0.8251612141002905}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 20:01:51,135] Trial 31 finished with value: 3.6105202492815773 and parameters: {'hidden_channels_encoder': 154, 'latent_space_dim': 254, 'hidden_channels_predictor': 55, 'dropout_rate': 0.6819903133304093}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 20:02:26,443] Trial 32 finished with value: 3.6915065621061975 and parameters: {'hidden_channels_encoder': 163, 'latent_space_dim': 256, 'hidden_channels_predictor': 66, 'dropout_rate': 0.7519057247482911}. Best is trial 18 with value: 3.6009739929590223.

[I 2024-03-21 20:02:57,966] Trial 33 finished with value: 3.5787023738587598 and parameters: {'hidden_channels_encoder': 153, 'latent_space_dim': 237, 'hidden_channels_predictor': 52, 'dropout_rate': 0.6751290012273564}. Best is trial 33 with value: 3.5787023738587598.

[I 2024-03-21 20:03:27,421] Trial 34 finished with value: 3.5610532568895437 and parameters: {'hidden_channels_encoder': 193, 'latent_space_dim': 178, 'hidden_channels_predictor': 49, 'dropout_rate': 0.6643663815540245}. Best is trial 34 with value: 3.5610532568895437.

[I 2024-03-21 20:04:01,421] Trial 35 finished with value: 66197.4406837395 and parameters: {'hidden_channels_encoder': 204, 'latent_space_dim': 158, 'hidden_channels_predictor': 90, 'dropout_rate': 0.6692524883106832}. Best is trial 34 with value: 3.5610532568895437.

[I 2024-03-21 20:04:41,323] Trial 36 finished with value: 209.47845412138855 and parameters: {'hidden_channels_encoder': 252, 'latent_space_dim': 171, 'hidden_channels_predictor': 109, 'dropout_rate': 0.7724183115190787}. Best is trial 34 with value: 3.5610532568895437.

[I 2024-03-21 20:05:15,383] Trial 37 finished with value: 3.717559060015109 and parameters: {'hidden_channels_encoder': 195, 'latent_space_dim': 237, 'hidden_channels_predictor': 50, 'dropout_rate': 0.6603523099262525}. Best is trial 34 with value: 3.5610532568895437.

[I 2024-03-21 20:05:51,361] Trial 38 finished with value: 16145.190800265638 and parameters: {'hidden_channels_encoder': 222, 'latent_space_dim': 156, 'hidden_channels_predictor': 117, 'dropout_rate': 0.7009933884186395}. Best is trial 34 with value: 3.5610532568895437.

[I 2024-03-21 20:06:22,828] Trial 39 finished with value: 1049893.340021848 and parameters: {'hidden_channels_encoder': 172, 'latent_space_dim': 180, 'hidden_channels_predictor': 75, 'dropout_rate': 0.013248928512405}. Best is trial 34 with value: 3.5610532568895437.

[I 2024-03-21 20:06:43,593] Trial 40 finished with value: 775728.1889648845 and parameters: {'hidden_channels_encoder': 133, 'latent_space_dim': 104, 'hidden_channels_predictor': 50, 'dropout_rate': 0.5913017325365948}. Best is trial 34 with value: 3.5610532568895437.

[I 2024-03-21 20:07:15,029] Trial 41 finished with value: 3.6870311917136305 and parameters: {'hidden_channels_encoder': 192, 'latent_space_dim': 210, 'hidden_channels_predictor': 53, 'dropout_rate': 0.492988102506408}. Best is trial 34 with value: 3.5610532568895437.

[I 2024-03-21 20:07:50,191] Trial 42 finished with value: 3.6054239545281566 and parameters: {'hidden_channels_encoder': 171, 'latent_space_dim': 234, 'hidden_channels_predictor': 46, 'dropout_rate': 0.6936128001734929}. Best is trial 34 with value: 3.5610532568895437.

[I 2024-03-21 20:08:24,659] Trial 43 finished with value: 27341.910801102717 and parameters: {'hidden_channels_encoder': 175, 'latent_space_dim': 136, 'hidden_channels_predictor': 130, 'dropout_rate': 0.7702164179044684}. Best is trial 34 with value: 3.5610532568895437.

[I 2024-03-21 20:09:02,482] Trial 44 finished with value: 437.00409253903814 and parameters: {'hidden_channels_encoder': 142, 'latent_space_dim': 241, 'hidden_channels_predictor': 88, 'dropout_rate': 0.6904864838420044}. Best is trial 34 with value: 3.5610532568895437.

[I 2024-03-21 20:09:32,796] Trial 45 finished with value: 3.551481635680431 and parameters: {'hidden_channels_encoder': 105, 'latent_space_dim': 234, 'hidden_channels_predictor': 43, 'dropout_rate': 0.8608226548287041}. Best is trial 45 with value: 3.551481635680431.

[I 2024-03-21 20:10:03,165] Trial 46 finished with value: 63.31794922522803 and parameters: {'hidden_channels_encoder': 109, 'latent_space_dim': 225, 'hidden_channels_predictor': 45, 'dropout_rate': 0.8750683687149916}. Best is trial 45 with value: 3.551481635680431.

[I 2024-03-21 20:10:35,382] Trial 47 finished with value: 3.624117460376464 and parameters: {'hidden_channels_encoder': 129, 'latent_space_dim': 235, 'hidden_channels_predictor': 65, 'dropout_rate': 0.8403279846871703}. Best is trial 45 with value: 3.551481635680431.

[I 2024-03-21 20:11:04,327] Trial 48 finished with value: 37390.80264211539 and parameters: {'hidden_channels_encoder': 100, 'latent_space_dim': 201, 'hidden_channels_predictor': 44, 'dropout_rate': 0.7813266708973178}. Best is trial 45 with value: 3.551481635680431.

[I 2024-03-21 20:11:38,283] Trial 49 finished with value: 3.6881479540267392 and parameters: {'hidden_channels_encoder': 222, 'latent_space_dim': 168, 'hidden_channels_predictor': 73, 'dropout_rate': 0.8614436295259795}. Best is trial 45 with value: 3.551481635680431.

Study outcomes for GraphSAGE:

[I 2024-03-22 00:53:26,410] A new study created in memory with name: no-name-2b2e0e5d-2fa8-4ac3-9385-39bdbe5a2c69

[I 2024-03-22 00:53:46,735] Trial 0 finished with value: 3.1243938037185583 and parameters: {'hidden_channels_encoder': 61, 'latent_space_dim': 40, 'hidden_channels_predictor': 140, 'dropout_rate': 0.5812858307278074}. Best is trial 0 with value: 3.1243938037185583.

[I 2024-03-22 00:54:13,595] Trial 1 finished with value: 2.59649988706209 and parameters: {'hidden_channels_encoder': 102, 'latent_space_dim': 195, 'hidden_channels_predictor': 36, 'dropout_rate': 0.3696564835152454}. Best is trial 1 with value: 2.59649988706209.

[I 2024-03-22 00:54:50,924] Trial 2 finished with value: 3.2765836252142124 and parameters: {'hidden_channels_encoder': 203, 'latent_space_dim': 143, 'hidden_channels_predictor': 168, 'dropout_rate': 0.4760228952435153}. Best is trial 1 with value: 2.59649988706209.

[I 2024-03-22 00:55:32,357] Trial 3 finished with value: 3.3899725741736395 and parameters: {'hidden_channels_encoder': 159, 'latent_space_dim': 224, 'hidden_channels_predictor': 146, 'dropout_rate': 0.5894601303076261}. Best is trial 1 with value: 2.59649988706209.

[I 2024-03-22 00:56:06,735] Trial 4 finished with value: 1.7118080167059166 and parameters: {'hidden_channels_encoder': 67, 'latent_space_dim': 173, 'hidden_channels_predictor': 162, 'dropout_rate': 0.4944257374499566}. Best is trial 4 with value: 1.7118080167059166.

[I 2024-03-22 00:56:50,850] Trial 5 finished with value: 3.2209121980895192 and parameters: {'hidden_channels_encoder': 214, 'latent_space_dim': 115, 'hidden_channels_predictor': 222, 'dropout_rate': 0.8782302175257847}. Best is trial 4 with value: 1.7118080167059166.

[I 2024-03-22 00:57:33,809] Trial 6 finished with value: 3.3652691828010557 and parameters: {'hidden_channels_encoder': 250, 'latent_space_dim': 210, 'hidden_channels_predictor': 129, 'dropout_rate': 0.537119361006366}. Best is trial 4 with value: 1.7118080167059166.

[I 2024-03-22 00:57:57,360] Trial 7 finished with value: 3.5287785178965207 and parameters: {'hidden_channels_encoder': 83, 'latent_space_dim': 175, 'hidden_channels_predictor': 45, 'dropout_rate': 0.6387522414841112}. Best is trial 4 with value: 1.7118080167059166.

[I 2024-03-22 00:58:55,945] Trial 8 finished with value: 3.5831291782434462 and parameters: {'hidden_channels_encoder': 246, 'latent_space_dim': 200, 'hidden_channels_predictor': 254, 'dropout_rate': 0.7341976130567965}. Best is trial 4 with value: 1.7118080167059166.

[I 2024-03-22 00:59:51,535] Trial 9 finished with value: 1.9752439750109114 and parameters: {'hidden_channels_encoder': 54, 'latent_space_dim': 244, 'hidden_channels_predictor': 232, 'dropout_rate': 0.4129852081053004}. Best is trial 4 with value: 1.7118080167059166.

[I 2024-03-22 01:00:18,294] Trial 10 finished with value: 1.0167287811175982 and parameters: {'hidden_channels_encoder': 127, 'latent_space_dim': 84, 'hidden_channels_predictor': 99, 'dropout_rate': 0.10695132044849653}. Best is trial 10 with value: 1.0167287811175982.

[I 2024-03-22 01:00:40,118] Trial 11 finished with value: 1.1666471928485334 and parameters: {'hidden_channels_encoder': 124, 'latent_space_dim': 84, 'hidden_channels_predictor': 68, 'dropout_rate': 0.0983634564157786}. Best is trial 10 with value: 1.0167287811175982.

[I 2024-03-22 01:01:03,702] Trial 12 finished with value: 1.1296760689694714 and parameters: {'hidden_channels_encoder': 132, 'latent_space_dim': 74, 'hidden_channels_predictor': 86, 'dropout_rate': 0.07542056159805752}. Best is trial 10 with value: 1.0167287811175982.

[I 2024-03-22 01:01:27,932] Trial 13 finished with value: 0.9735124451808328 and parameters: {'hidden_channels_encoder': 158, 'latent_space_dim': 62, 'hidden_channels_predictor': 97, 'dropout_rate': 0.0031889426798753506}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:01:51,554] Trial 14 finished with value: 1.661589533310149 and parameters: {'hidden_channels_encoder': 158, 'latent_space_dim': 56, 'hidden_channels_predictor': 100, 'dropout_rate': 0.2360049185723723}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:02:21,401] Trial 15 finished with value: 1.5519462621006697 and parameters: {'hidden_channels_encoder': 175, 'latent_space_dim': 103, 'hidden_channels_predictor': 115, 'dropout_rate': 0.20993405353566935}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:02:45,537] Trial 16 finished with value: 0.977350966380368 and parameters: {'hidden_channels_encoder': 109, 'latent_space_dim': 129, 'hidden_channels_predictor': 70, 'dropout_rate': 0.008507896628242098}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:03:07,908] Trial 17 finished with value: 1.0175502851310372 and parameters: {'hidden_channels_encoder': 99, 'latent_space_dim': 132, 'hidden_channels_predictor': 64, 'dropout_rate': 0.028979884367139434}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:03:32,048] Trial 18 finished with value: 1.2219757434175302 and parameters: {'hidden_channels_encoder': 39, 'latent_space_dim': 36, 'hidden_channels_predictor': 191, 'dropout_rate': 0.2544598061014289}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:04:02,838] Trial 19 finished with value: 2.092564839145028 and parameters: {'hidden_channels_encoder': 190, 'latent_space_dim': 162, 'hidden_channels_predictor': 73, 'dropout_rate': 0.3232719496050474}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:04:24,406] Trial 20 finished with value: 0.9901205430907739 and parameters: {'hidden_channels_encoder': 106, 'latent_space_dim': 116, 'hidden_channels_predictor': 51, 'dropout_rate': 0.00675725259529632}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:04:45,091] Trial 21 finished with value: 1.0173779733375548 and parameters: {'hidden_channels_encoder': 103, 'latent_space_dim': 109, 'hidden_channels_predictor': 52, 'dropout_rate': 0.01696700654882942}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:05:07,273] Trial 22 finished with value: 1.2603829748301651 and parameters: {'hidden_channels_encoder': 140, 'latent_space_dim': 129, 'hidden_channels_predictor': 32, 'dropout_rate': 0.15628072437978174}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:05:28,699] Trial 23 finished with value: 1.3845864963567343 and parameters: {'hidden_channels_encoder': 107, 'latent_space_dim': 67, 'hidden_channels_predictor': 92, 'dropout_rate': 0.17412374779719259}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:05:56,295] Trial 24 finished with value: 1.0145583603047148 and parameters: {'hidden_channels_encoder': 152, 'latent_space_dim': 101, 'hidden_channels_predictor': 111, 'dropout_rate': 0.013660930261795823}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:06:21,878] Trial 25 finished with value: 1.9190999664276087 and parameters: {'hidden_channels_encoder': 78, 'latent_space_dim': 151, 'hidden_channels_predictor': 81, 'dropout_rate': 0.31328874317689087}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:06:47,701] Trial 26 finished with value: 1.244789556717415 and parameters: {'hidden_channels_encoder': 179, 'latent_space_dim': 123, 'hidden_channels_predictor': 56, 'dropout_rate': 0.12543362040795747}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:07:14,785] Trial 27 finished with value: 1.0108445021821812 and parameters: {'hidden_channels_encoder': 117, 'latent_space_dim': 94, 'hidden_channels_predictor': 123, 'dropout_rate': 0.00601912279467696}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:07:34,737] Trial 28 finished with value: 1.0243806801527944 and parameters: {'hidden_channels_encoder': 145, 'latent_space_dim': 63, 'hidden_channels_predictor': 66, 'dropout_rate': 0.07435141777864906}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:07:52,197] Trial 29 finished with value: 1.8450131260511364 and parameters: {'hidden_channels_encoder': 89, 'latent_space_dim': 49, 'hidden_channels_predictor': 82, 'dropout_rate': 0.2801803405996344}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:08:28,077] Trial 30 finished with value: 1.1420437700782147 and parameters: {'hidden_channels_encoder': 222, 'latent_space_dim': 142, 'hidden_channels_predictor': 136, 'dropout_rate': 0.17635169177241478}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:08:53,863] Trial 31 finished with value: 1.0187687982947229 and parameters: {'hidden_channels_encoder': 117, 'latent_space_dim': 89, 'hidden_channels_predictor': 123, 'dropout_rate': 0.01119180390199332}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:09:18,052] Trial 32 finished with value: 0.990862302737723 and parameters: {'hidden_channels_encoder': 115, 'latent_space_dim': 97, 'hidden_channels_predictor': 104, 'dropout_rate': 0.06757500322183164}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:09:46,525] Trial 33 finished with value: 1.011130659957328 and parameters: {'hidden_channels_encoder': 169, 'latent_space_dim': 114, 'hidden_channels_predictor': 103, 'dropout_rate': 0.06771295091365524}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:10:03,487] Trial 34 finished with value: 1.2206057609885252 and parameters: {'hidden_channels_encoder': 93, 'latent_space_dim': 77, 'hidden_channels_predictor': 43, 'dropout_rate': 0.12441384231111335}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:10:27,004] Trial 35 finished with value: 1.0774818976732352 and parameters: {'hidden_channels_encoder': 112, 'latent_space_dim': 154, 'hidden_channels_predictor': 56, 'dropout_rate': 0.05779376827303484}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:10:48,438] Trial 36 finished with value: 1.6169492793950586 and parameters: {'hidden_channels_encoder': 72, 'latent_space_dim': 49, 'hidden_channels_predictor': 148, 'dropout_rate': 0.1884043020435745}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:11:15,157] Trial 37 finished with value: 1.242157487767439 and parameters: {'hidden_channels_encoder': 137, 'latent_space_dim': 138, 'hidden_channels_predictor': 75, 'dropout_rate': 0.1409212091745812}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:11:50,445] Trial 38 finished with value: 1.0208442503497372 and parameters: {'hidden_channels_encoder': 164, 'latent_space_dim': 121, 'hidden_channels_predictor': 163, 'dropout_rate': 0.059521535174841954}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:12:16,159] Trial 39 finished with value: 2.685279505135284 and parameters: {'hidden_channels_encoder': 48, 'latent_space_dim': 178, 'hidden_channels_predictor': 89, 'dropout_rate': 0.40311560444139716}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:12:33,061] Trial 40 finished with value: 3.7367285668340875 and parameters: {'hidden_channels_encoder': 64, 'latent_space_dim': 102, 'hidden_channels_predictor': 45, 'dropout_rate': 0.8433271680785788}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:12:58,703] Trial 41 finished with value: 1.0614538852319213 and parameters: {'hidden_channels_encoder': 121, 'latent_space_dim': 92, 'hidden_channels_predictor': 117, 'dropout_rate': 8.390504661524856e-05}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:13:26,459] Trial 42 finished with value: 0.9946126471370098 and parameters: {'hidden_channels_encoder': 148, 'latent_space_dim': 95, 'hidden_channels_predictor': 132, 'dropout_rate': 0.042089186954199775}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:13:57,632] Trial 43 finished with value: 1.639778142274762 and parameters: {'hidden_channels_encoder': 148, 'latent_space_dim': 71, 'hidden_channels_predictor': 180, 'dropout_rate': 0.10201504999957364}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:14:32,943] Trial 44 finished with value: 0.9823263334947161 and parameters: {'hidden_channels_encoder': 186, 'latent_space_dim': 115, 'hidden_channels_predictor': 151, 'dropout_rate': 0.05534429766563637}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:15:09,323] Trial 45 finished with value: 1.1689058268910573 and parameters: {'hidden_channels_encoder': 191, 'latent_space_dim': 121, 'hidden_channels_predictor': 156, 'dropout_rate': 0.0963258726580571}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:15:50,687] Trial 46 finished with value: 2.6644146178944093 and parameters: {'hidden_channels_encoder': 210, 'latent_space_dim': 110, 'hidden_channels_predictor': 187, 'dropout_rate': 0.6080612768532256}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:16:28,886] Trial 47 finished with value: 3.025639132475488 and parameters: {'hidden_channels_encoder': 189, 'latent_space_dim': 162, 'hidden_channels_predictor': 145, 'dropout_rate': 0.501290824990308}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:17:03,314] Trial 48 finished with value: 1.0436171233181144 and parameters: {'hidden_channels_encoder': 130, 'latent_space_dim': 81, 'hidden_channels_predictor': 206, 'dropout_rate': 0.049292184980416494}. Best is trial 13 with value: 0.9735124451808328.

[I 2024-03-22 01:17:38,923] Trial 49 finished with value: 3.4749615662896014 and parameters: {'hidden_channels_encoder': 233, 'latent_space_dim': 133, 'hidden_channels_predictor': 108, 'dropout_rate': 0.6863471795258129}. Best is trial 13 with value: 0.9735124451808328.

Study outcomes for GAT:

[I 2024-03-22 01:55:45,653] A new study created in memory with name: no-name-31de3147-399b-4147-9db9-d1ab8566fd3f

[I 2024-03-22 01:57:25,789] Trial 0 finished with value: 2.7157098436307265 and parameters: {'hidden_channels_encoder': 140, 'latent_space_dim': 120, 'hidden_channels_predictor': 242, 'dropout_rate': 0.41769294128147877}. Best is trial 0 with value: 2.7157098436307265.

[I 2024-03-22 01:58:45,799] Trial 1 finished with value: 3.0637301403215065 and parameters: {'hidden_channels_encoder': 154, 'latent_space_dim': 80, 'hidden_channels_predictor': 37, 'dropout_rate': 0.6275914414048435}. Best is trial 0 with value: 2.7157098436307265.

[I 2024-03-22 02:00:06,842] Trial 2 finished with value: 4.662941441847871 and parameters: {'hidden_channels_encoder': 82, 'latent_space_dim': 92, 'hidden_channels_predictor': 225, 'dropout_rate': 0.7029481386904642}. Best is trial 0 with value: 2.7157098436307265.

[I 2024-03-22 02:01:31,926] Trial 3 finished with value: 2.023740329429225 and parameters: {'hidden_channels_encoder': 132, 'latent_space_dim': 184, 'hidden_channels_predictor': 101, 'dropout_rate': 0.089868451174226}. Best is trial 3 with value: 2.023740329429225.

[I 2024-03-22 02:03:21,189] Trial 4 finished with value: 5.692390129018076 and parameters: {'hidden_channels_encoder': 218, 'latent_space_dim': 149, 'hidden_channels_predictor': 106, 'dropout_rate': 0.89757112712996}. Best is trial 3 with value: 2.023740329429225.

[I 2024-03-22 02:04:50,171] Trial 5 finished with value: 2.050278058828334 and parameters: {'hidden_channels_encoder': 191, 'latent_space_dim': 77, 'hidden_channels_predictor': 56, 'dropout_rate': 0.3657807852661859}. Best is trial 3 with value: 2.023740329429225.

[I 2024-03-22 02:07:13,299] Trial 6 finished with value: 2.525200830241767 and parameters: {'hidden_channels_encoder': 234, 'latent_space_dim': 231, 'hidden_channels_predictor': 243, 'dropout_rate': 0.5259174148138746}. Best is trial 3 with value: 2.023740329429225.

[I 2024-03-22 02:09:18,839] Trial 7 finished with value: 2.924588320190618 and parameters: {'hidden_channels_encoder': 255, 'latent_space_dim': 169, 'hidden_channels_predictor': 71, 'dropout_rate': 0.7257652052896044}. Best is trial 3 with value: 2.023740329429225.

[I 2024-03-22 02:11:11,452] Trial 8 finished with value: 1.9909636763542218 and parameters: {'hidden_channels_encoder': 231, 'latent_space_dim': 186, 'hidden_channels_predictor': 85, 'dropout_rate': 0.28402558336109185}. Best is trial 8 with value: 1.9909636763542218.

[I 2024-03-22 02:12:58,808] Trial 9 finished with value: 2.4283607689742945 and parameters: {'hidden_channels_encoder': 195, 'latent_space_dim': 117, 'hidden_channels_predictor': 197, 'dropout_rate': 0.4103156042979175}. Best is trial 8 with value: 1.9909636763542218.

[I 2024-03-22 02:14:13,524] Trial 10 finished with value: 1.7726216046746548 and parameters: {'hidden_channels_encoder': 43, 'latent_space_dim': 249, 'hidden_channels_predictor': 166, 'dropout_rate': 0.06503130401942969}. Best is trial 10 with value: 1.7726216046746548.

[I 2024-03-22 02:15:32,674] Trial 11 finished with value: 1.3159755477230977 and parameters: {'hidden_channels_encoder': 36, 'latent_space_dim': 246, 'hidden_channels_predictor': 183, 'dropout_rate': 0.06796445063047829}. Best is trial 11 with value: 1.3159755477230977.

[I 2024-03-22 02:16:44,408] Trial 12 finished with value: 4.5927163384668255 and parameters: {'hidden_channels_encoder': 33, 'latent_space_dim': 246, 'hidden_channels_predictor': 170, 'dropout_rate': 0.007512975092654882}. Best is trial 11 with value: 1.3159755477230977.

[I 2024-03-22 02:17:45,815] Trial 13 finished with value: 0.939781204976028 and parameters: {'hidden_channels_encoder': 34, 'latent_space_dim': 221, 'hidden_channels_predictor': 146, 'dropout_rate': 0.17511791824226366}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:18:59,135] Trial 14 finished with value: 2.665061158074166 and parameters: {'hidden_channels_encoder': 92, 'latent_space_dim': 213, 'hidden_channels_predictor': 136, 'dropout_rate': 0.18799638569999305}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:19:51,954] Trial 15 finished with value: 4.442192196873667 and parameters: {'hidden_channels_encoder': 70, 'latent_space_dim': 44, 'hidden_channels_predictor': 154, 'dropout_rate': 0.20112731518944038}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:21:09,097] Trial 16 finished with value: 1.083028144193717 and parameters: {'hidden_channels_encoder': 59, 'latent_space_dim': 210, 'hidden_channels_predictor': 200, 'dropout_rate': 0.2226137928057827}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:22:48,506] Trial 17 finished with value: 1.5569548942608968 and parameters: {'hidden_channels_encoder': 107, 'latent_space_dim': 209, 'hidden_channels_predictor': 202, 'dropout_rate': 0.26010810607885143}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:23:57,850] Trial 18 finished with value: 1.1322888951048198 and parameters: {'hidden_channels_encoder': 60, 'latent_space_dim': 206, 'hidden_channels_predictor': 128, 'dropout_rate': 0.16858251733478988}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:25:16,515] Trial 19 finished with value: 1.3960091934852827 and parameters: {'hidden_channels_encoder': 101, 'latent_space_dim': 149, 'hidden_channels_predictor': 214, 'dropout_rate': 0.3157082272989783}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:26:22,496] Trial 20 finished with value: 2.156955815844346 and parameters: {'hidden_channels_encoder': 61, 'latent_space_dim': 223, 'hidden_channels_predictor': 127, 'dropout_rate': 0.5097565631630012}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:27:22,908] Trial 21 finished with value: 1.3399613459946411 and parameters: {'hidden_channels_encoder': 59, 'latent_space_dim': 193, 'hidden_channels_predictor': 121, 'dropout_rate': 0.15926577605580627}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:28:54,551] Trial 22 finished with value: 0.9516401630303761 and parameters: {'hidden_channels_encoder': 121, 'latent_space_dim': 201, 'hidden_channels_predictor': 148, 'dropout_rate': 0.1513911146261644}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:30:11,974] Trial 23 finished with value: 2.488730553105606 and parameters: {'hidden_channels_encoder': 115, 'latent_space_dim': 172, 'hidden_channels_predictor': 149, 'dropout_rate': 0.2457001581849526}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:32:00,294] Trial 24 finished with value: 2.425158400227942 and parameters: {'hidden_channels_encoder': 168, 'latent_space_dim': 223, 'hidden_channels_predictor': 182, 'dropout_rate': 0.35940465945120653}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:33:29,140] Trial 25 finished with value: 1.9930996305359172 and parameters: {'hidden_channels_encoder': 123, 'latent_space_dim': 196, 'hidden_channels_predictor': 155, 'dropout_rate': 0.1118287625693689}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:34:44,501] Trial 26 finished with value: 1.2331668057417282 and parameters: {'hidden_channels_encoder': 78, 'latent_space_dim': 165, 'hidden_channels_predictor': 185, 'dropout_rate': 0.009395772876858083}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:35:47,429] Trial 27 finished with value: 2.8401697812048083 and parameters: {'hidden_channels_encoder': 48, 'latent_space_dim': 233, 'hidden_channels_predictor': 106, 'dropout_rate': 0.22730429899049426}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:37:01,010] Trial 28 finished with value: 2.616518989792935 and parameters: {'hidden_channels_encoder': 88, 'latent_space_dim': 133, 'hidden_channels_predictor': 222, 'dropout_rate': 0.13247199832197673}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:38:53,147] Trial 29 finished with value: 2.77461632905385 and parameters: {'hidden_channels_encoder': 147, 'latent_space_dim': 203, 'hidden_channels_predictor': 250, 'dropout_rate': 0.4389426084172781}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:40:40,113] Trial 30 finished with value: 2.1589068963743783 and parameters: {'hidden_channels_encoder': 169, 'latent_space_dim': 256, 'hidden_channels_predictor': 168, 'dropout_rate': 0.3396003073638427}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:41:43,880] Trial 31 finished with value: 1.1910863994248215 and parameters: {'hidden_channels_encoder': 55, 'latent_space_dim': 213, 'hidden_channels_predictor': 133, 'dropout_rate': 0.16558134370640534}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:42:49,292] Trial 32 finished with value: 1.7328390402911684 and parameters: {'hidden_channels_encoder': 72, 'latent_space_dim': 182, 'hidden_channels_predictor': 115, 'dropout_rate': 0.2213819221682157}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:44:08,313] Trial 33 finished with value: 1.1938987262511755 and parameters: {'hidden_channels_encoder': 63, 'latent_space_dim': 230, 'hidden_channels_predictor': 140, 'dropout_rate': 0.2905714716546572}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:45:13,818] Trial 34 finished with value: 1.1050831509792665 and parameters: {'hidden_channels_encoder': 48, 'latent_space_dim': 201, 'hidden_channels_predictor': 96, 'dropout_rate': 0.12312963990187722}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:45:56,566] Trial 35 finished with value: 2.3432790031186124 and parameters: {'hidden_channels_encoder': 39, 'latent_space_dim': 161, 'hidden_channels_predictor': 44, 'dropout_rate': 0.06504860161865986}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:47:20,786] Trial 36 finished with value: 1.1751688394278812 and parameters: {'hidden_channels_encoder': 133, 'latent_space_dim': 180, 'hidden_channels_predictor': 93, 'dropout_rate': 0.11590461531207723}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:48:24,651] Trial 37 finished with value: 1.2545703540924866 and parameters: {'hidden_channels_encoder': 49, 'latent_space_dim': 196, 'hidden_channels_predictor': 65, 'dropout_rate': 0.018540441785574235}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:49:38,860] Trial 38 finished with value: 4.675783987775218 and parameters: {'hidden_channels_encoder': 96, 'latent_space_dim': 220, 'hidden_channels_predictor': 82, 'dropout_rate': 0.5828457826354954}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:51:14,491] Trial 39 finished with value: 1.8160065909028846 and parameters: {'hidden_channels_encoder': 81, 'latent_space_dim': 237, 'hidden_channels_predictor': 235, 'dropout_rate': 0.3944488551926529}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:51:58,824] Trial 40 finished with value: 2.542461632447737 and parameters: {'hidden_channels_encoder': 33, 'latent_space_dim': 109, 'hidden_channels_predictor': 111, 'dropout_rate': 0.13576072717425863}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:52:54,414] Trial 41 finished with value: 1.9968898000951802 and parameters: {'hidden_channels_encoder': 51, 'latent_space_dim': 204, 'hidden_channels_predictor': 95, 'dropout_rate': 0.1652627879272549}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:54:00,519] Trial 42 finished with value: 2.6612875788446946 and parameters: {'hidden_channels_encoder': 70, 'latent_space_dim': 194, 'hidden_channels_predictor': 124, 'dropout_rate': 0.8450398877372594}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:55:02,316] Trial 43 finished with value: 1.5604705899039693 and parameters: {'hidden_channels_encoder': 44, 'latent_space_dim': 156, 'hidden_channels_predictor': 144, 'dropout_rate': 0.09624397099622438}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:55:58,957] Trial 44 finished with value: 4.436385881490325 and parameters: {'hidden_channels_encoder': 63, 'latent_space_dim': 175, 'hidden_channels_predictor': 77, 'dropout_rate': 0.19202681587136766}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:57:28,571] Trial 45 finished with value: 1.8246889603489773 and parameters: {'hidden_channels_encoder': 111, 'latent_space_dim': 238, 'hidden_channels_predictor': 98, 'dropout_rate': 0.2760663874642551}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 02:59:16,832] Trial 46 finished with value: 1.7466005740481738 and parameters: {'hidden_channels_encoder': 199, 'latent_space_dim': 136, 'hidden_channels_predictor': 163, 'dropout_rate': 0.04886858533784451}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 03:00:38,244] Trial 47 finished with value: 1.8950627442257428 and parameters: {'hidden_channels_encoder': 84, 'latent_space_dim': 218, 'hidden_channels_predictor': 196, 'dropout_rate': 0.22948566276448087}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 03:02:20,256] Trial 48 finished with value: 2.378505019704437 and parameters: {'hidden_channels_encoder': 160, 'latent_space_dim': 187, 'hidden_channels_predictor': 175, 'dropout_rate': 0.4760704704501475}. Best is trial 13 with value: 0.939781204976028.

[I 2024-03-22 03:03:21,520] Trial 49 finished with value: 1.5832520315953258 and parameters: {'hidden_channels_encoder': 32, 'latent_space_dim': 202, 'hidden_channels_predictor': 133, 'dropout_rate': 0.30636503301666074}. Best is trial 13 with value: 0.939781204976028

References

1. Cui, Z., Li, Z., Wu, S., Zhang, X. Y., & Wang, L. (2019). Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks. In The world wide web conference (pp. 307-317).
2. Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., & Yin, D. (2019). Graph neural networks for social recommendation. In The world wide web conference (pp. 417-426).
3. Kaur, G., Liu, F., & Chen, Y. P. P. (2023). A deep learning knowledge graph neural network for recommender systems. Machine Learning with Applications, 100507.
4. Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations (ICLR).
5. Nesmaoui, R., Louhichi, M., & Lazaar, M. (2023). A Collaborative Filtering Movies Recommendation System based on Graph Neural Network. Procedia Computer Science, 220, 456-461.
6. Wang, S., Hu, L., Wang, Y., He, X., Sheng, Q. Z., Orgun, M. A., ... & Yu, P. S. (2021). Graph learning based recommender systems: A review. arXiv preprint arXiv:2105.06339
7. Wang, S., Wang, Y., Sivrikaya, F., Albayrak, S., & Anelli, V. W. (2023). Data science for next-generation recommender systems. International Journal of Data Science and Analytics, 16(2), 135-145.
8. Wang, X., He, X., Cao, Y., Liu, M., & Chua, T. S. (2019). Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 950-958).
9. Yin, R., Li, K., Zhang, G., & Lu, J. (2019). A deeper graph neural network for recommender systems. Knowledge-Based Systems, 185, 105020.
10. Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS).
11. Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K., & Jegelka, S. (2018). Representation Learning on Graphs with Jumping Knowledge Networks. In Proceedings of the International Conference on Machine Learning (ICML).
12. Xiao, L., Wu, X., & Wang, G. (2019). Social Network Analysis Based on Graph. 2019 12th International Symposium on Computational Intelligence and Design (ISCID).
13. Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations (ICLR).

14. Zhang, M., & Chen, Y. (2018). Large-scale Graph Representation Learning with Gated Graph Neural Networks. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI).
15. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph Attention Networks. In International Conference on Learning Representations (ICLR).
16. Lee, J., Lee, I., & Kang, J. (2019). Self-Attention Graph Pooling. In Proceedings of the International Conference on Machine Learning (ICML).
17. Gao, C., Wang, X., He, X., & Li, Y. (2022, February). Graph neural networks for recommender system. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (pp. 1623-1625).
18. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). ACM, New York, NY, USA.
19. Yu, S., Qiu, J., Bao, X., Guo, M., Chen, X., & Sun, J. (2022). Movie Rating Prediction Recommendation Algorithm Based on XGBoost-DNN. In Proceedings of the International Conference on Information and Software Technologies.
20. Sun, T., Wen, J., & Gong, J. (2023). Personalized Learning Resource Recommendation Using Differential Evolution-Based Graph Neural Network: A GraphSAGE Approach. In Proceedings of the 2023 4th International Symposium on Computer Engineering and Intelligent Communications (ISCEIC).
21. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph Convolutional Neural Networks for Web-Scale Recommender Systems. Knowledge Discovery and Data Mining.
22. Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J., & Xie, X. (2020). Self-supervised Graph Learning for Recommendation. In Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.
23. Chen, L., Wu, L., Hong, R., Zhang, K., & Wang, M. (2020). Revisiting Graph-Based Collaborative Filtering: A Linear Residual Graph Convolutional Network Approach. In Proceedings of the AAAI Conference on Artificial Intelligence.
24. Dellal-Hedjazi, B., & Alimazighi, Z. (2020). Deep Learning for Recommendation Systems. In Proceedings of the Colloquium in Information Science and Technology.
25. Feng, C., Liu, Z., Lin, S., & Quek, T. Q. S. (2019). Attention-Based Graph Convolutional Network for Recommendation System. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing.
26. Gao, C., Wang, X., He, X., & Li, Y. (2022, February). Graph neural networks for recommender system. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (pp. 1623-1625).