

UNDERGRADUATE PROJECT PROGESS REPORT

|  |  |
| --- | --- |
| **Project Title:** | **Sequential Recommendations with Graph Neural Networks** |
| **Surname:** | **Anju** |
| **First Name:** | **Chen** |
| **Student Number:** | **202118010227** |
| **Supervisor Name:** | **Joojo Walker** |
| **Module Code:** | **CHC 6096** |
| **Module Name:** | **Project** |
| **Date Submitted:** | **2024.12.27** |

**Table of Contents**

[1 Introduction 4](#_Toc7122)

[1.1 Background 4](#_Toc31169)

[1.2 Aim 4](#_Toc32062)

[1.3 Objectives 4](#_Toc14485)

[1.4 Project Overview 5](#_Toc7194)

[1.4.1 Scope 5](#_Toc15393)

[1.4.2 Audience 5](#_Toc6844)

[2 Background Review 7](#_Toc10117)

[2.1 Background Review of Sequential Recommendations 7](#_Toc6471)

[2.2 Background Review of Graph Neural Networks 7](#_Toc1499)

[3 Technical Progress 8](#_Toc7172)

[3.1 Approach 8](#_Toc8591)

[3.2 Technology 8](#_Toc26057)

[3.3 Testing and Evaluation Plan 9](#_Toc21430)

[3.3.1 Data Testing 9](#_Toc28304)

[3.3.2 Model Evaluation 9](#_Toc29391)

[3.3.3 Model Test 10](#_Toc8090)

[3.3.4 Pipeline Testing 11](#_Toc1234)

[3.4 Design and Implementation 11](#_Toc2517)

[3.4.1 Find related projects 11](#_Toc13948)

[3.4.2 Improvement of design direction 14](#_Toc12390)

[3.4.3 Determine dataset(https://gitcode.com/gh\_mirrors/se/Sequential-Recommendation-Datasets/overview) 15](#_Toc27193)

[4 Project Management 17](#_Toc4224)

[4.1 Activities 17](#_Toc794)

[4.2 Schedule 18](#_Toc6382)

[4.3 Project Version Management 18](#_Toc29879)

[4.4 Project Data Management 19](#_Toc5887)

[4.5 Project Deliverables 19](#_Toc9969)

[5 Professional Issues and Risk: 20](#_Toc2990)

[5.1 Risk Analysis 20](#_Toc29932)

[5.2 Professional Issues 21](#_Toc13123)

[5.2.1 Legal Issues 21](#_Toc2237)

[5.2.2 Social Issues 22](#_Toc6845)

[5.2.3 Ethical Issues 22](#_Toc8607)

[5.2.4 Environmental Issues 23](#_Toc14788)

[6 References 24](#_Toc167)

# Introduction

## Background

People are exposed to an ever-increasing amount of information as digital media and e-commerce become more popular, which makes personalized recommendation systems especially important. Recommender systems let users go through vast amounts of information to find interesting content while also allowing content providers to improve their offerings, increasing user satisfaction and business value. As an emerging technology, graph neural networks (GNNs) show great promise for analyzing complex relational data, especially when it comes to user behavior sequence data.

## Aim

This project aims to explore how graph neural networks can be used to build efficient and accurate sequential recommendation systems. The goal is to develop an effective way to capture user behavior patterns and make accurate recommendations based on them.

## Objectives

The objectives of this project encompass, but are not restricted to:

1.Analysing the existing literature, databases, and technical solutions pertaining to sequence suggestion  
2.Design and execute a sequential recommendation system utilising graph neural networks.   
3.Assess the efficacy of the suggested algorithm and enhance its prediction capability;

4.Validate the algorithm's practical usefulness and gather feedback from end users.

## Project Overview

This project focuses on using graph neural network techniques to improve recommender system performance. Recommender systems have the potential to provide more individualized services by accurately identifying the interests and preferences of users. For service companies who rely on user engagement and retention, this is a major breakthrough. The project's history, methodology, management, and execution strategy will all be covered in this proposal.

### Scope

This research aims to use graph neural networks to build efficient sequential recommendation systems to improve recommendation accuracy and user experience. The importance of this research is to reduce the time for users to find interested content, attract more users, and promote the wide application of recommendation systems in many fields.

### Audience

Beneficiary Groups:

In order to increase suggestion accuracy and user happiness, this project attempts to create an effective sequential recommendation system employing graph neural networks (GNNs). The outcomes will have a direct impact on content providers and service platforms, including social media platforms, e-commerce websites, and video-on-demand services. These platforms will be able to use more precise recommendation algorithms to draw in and keep users, which will improve user engagement and boost profits. Additionally, marketers and advertisers gain from being able to more precisely target their customers and implement marketing tactics.

Target Audience:

Business decision makers, researchers, and technology developers make up the majority of the target audience. The techniques and resources offered in this project will assist technology developers in incorporating sophisticated recommendation features into current systems. The implementation of graph neural networks in recommender systems will be better understood and developed by the academic community as a result of this project. Lastly, by using the project's findings, company decision makers would be able to create more competitive business plans and boost market share. In conclusion, the results of this project will be helpful to any company or individual that depends on user engagement and the improvement of a tailored experience.

# Background Review

## Background Review of Sequential Recommendations

Sequential recommendation is a critical task aimed at predicting the next item a user might interact with based on their historical interaction sequences[11]. Another approach involves integrating group behaviors to enhance the modeling of user preferences through collective wisdom[5].Furthermore, sequential recommendation models have evolved to consider multi-dimensional transformations between items[3] and to alleviate the negative impact of data sparsity[2]. Sequential recommendation models have also benefited from the integration of heterogeneous information such as category and time information[3], and the exploration of deep user interests [1].

## 2.2 Background Review of Graph Neural Networks

Graph Neural Networks (GNNs) have been pivotal in improving recommendation systems by capturing the complex relationships within user-item interactions[7]. In session-based recommendation, GNNs have been utilized to capture complex transitions of items [8]. The Graph Contextualized Self-Attention Network (GC-SAN)[9] further enhances this by dynamically constructing a graph structure for session sequences and capturing rich local dependencies via a graph neural network.These advancements highlight the versatility and effectiveness of GNNs in recommendation systems, from simplifying architectures[4] to integrating complex multi-dimensional transformations[3] and leveraging heterogeneous information[10]. The continued development of GNN-based methods promises to further enhance the accuracy and relevance of recommendations in various domains.

# Technical Progress

## Approach

Sequential recommendation with Graph Neural Networks (GNNs) is grounded in representing user-item interactions as graph structures. Nodes represent users and items, and edges signify interactions. Mathematically, the embeddings of items are updated through layers of graph convolutions, which propagate embeddings based on the graph structure:

Here,E (ℓ) is the embedding matrix at layer ℓ, W (ℓ)is a trainable weight matrix, and σ is a non-linear activation function. This iterative process captures the influence of neighboring nodes on each item.

To prevent overfitting, dropout techniques are applied, typically setting a dropout rate of around 0.2. Attention mechanisms, such as self-attention, prioritize the most relevant items in a user's interaction history. Residual connections are used to help gradients flow through deeper layers, aiding in the training of more complex models.

The assessment of GNN-based sequential recommendation models is performed using datasets such as MovieLens, Steam, UserBehavior, and others. These datasets document user-item interactions together with their corresponding timestamps.

## Technology

The technologies used to implement the projects are:

|  |  |  |
| --- | --- | --- |
| **Type** | **Working area** | **Actions** |
| Software | Development environment and core libraries | Python(3.12), PyTorch(1.7) |
| Data processing and analysis | NumPy(1.16), Pandas(0.24) |
| Scientific Computing and Machine Learning | Scikit-Learn(0.2), SciPy(1.1) |
| Visualization tools | Matplotlib(3.0), TensorBoard(2.0) |
| Hardware | CPU | AMD Ryzen 5 5800H with Radeon Graphics |
| GPU | NVIDIA GeForce RTX 3060 Laptop GPU 4G |
| Memory | Kingston DDR4 3200MHz 16G |

Table 1: Technologies used for product

## Testing and Evaluation Plan

### Data Testing

Comprehensive data testing will be conducted to ensure the integrity and quality of datasets used for training, validation, and testing. Firstly, data consistency will be checked. This involves verifying that there are no missing values, incorrect data types, or inconsistent formatting across different features. For instance, in user-item interaction data, it is necessary to confirm that the timestamps are in the correct chronological order and that the user and item identifiers are unique and properly referenced.

Secondly, a statistical analysis of the data distribution will be performed. This includes examining the frequency of user actions, the popularity distribution of items, and the sparsity patterns in the interaction matrix. Outliers will be identified and further investigated to determine if they are legitimate data points or errors. For example, if a particular user has an abnormally high number of interactions compared to the average, it should be explored whether it is due to a bot or a legitimate power user. By understanding the data distribution, appropriate preprocessing steps can be better designed and the performance of the model in different scenarios can be evaluated.

### Model Evaluation

The model evaluation will focus on multiple key metrics relevant to sequential recommendation tasks. The primary metric is the Hit Ratio (HR), which measures the proportion of relevant items that are successfully recommended within the top-k recommended list. A higher HR indicates that the model is more effective at retrieving items that the user is likely to interact with. For example, if in 100 test cases where a user has interacted with a particular item in the future, the model recommends that item in the top 5 for 30 cases, the HR@5 would be 0.3.

Another crucial metric is the Normalized Discounted Cumulative Gain (NDCG). NDCG takes into account the ranking of relevant items in the recommended list, giving higher scores to models that rank relevant items closer to the top. It discounts the gain of items further down the list, reflecting the fact that users are more likely to engage with top-ranked recommendations. This metric is especially important as it evaluates not only the presence of relevant items but also their relative positions. For example, if a relevant item is ranked first, it will contribute more to the NDCG score than if it were ranked fifth.

Mean Reciprocal Rank (MRR) will also be considered. MRR measures the average of the reciprocal of the rank of the first relevant item in the recommended list. MRR emphasizes the importance of quickly retrieving relevant items and is sensitive to the position of the first correct recommendation. In addition to these, metrics such as Precision and Recall at different cut-off values will be monitored to gain a comprehensive understanding of the model's performance across various aspects of the recommendation task.

### Model Test

For model testing, the dataset will be split into training, validation, and test sets in a stratified manner to ensure that each subset contains a representative sample of the overall data distribution. The training set will be used to train the graph neural network-based sequential recommendation model, while the validation set will be used for hyperparameter tuning and early stopping. Techniques like k-fold cross-validation will be used to further validate the model's robustness and generalization ability.

During testing, the test data, which consists of historical user sequences, will be fed into the trained model and the generated recommendations will be collected. The predictions of the model will then be compared against the ground truth future interactions in the test set to compute the evaluation metrics mentioned above. Ablation studies will also be conducted, where specific components or features of the model (such as different graph layers or attention mechanisms) are removed to understand their individual contributions to the overall performance. This helps in identifying the most critical parts of the model and provides insights for potential improvements.

### Pipeline Testing

Pipeline testing will focus on ensuring the seamless integration and operation of all components involved in the sequential recommendation system. Starting from the data ingestion process, it is necessary to verify that the raw data is correctly loaded, preprocessed, and transformed into the appropriate format for the model input. This includes checking the compatibility of data formats between different stages, such as ensuring that the output of the data preprocessing step is correctly fed into the graph construction module.

The graph construction process will be tested to ensure that the user-item graphs are built accurately, with the correct edge weights and node features. It is also important to verify that the temporal information is properly incorporated into the graph structure if applicable. The integration of the trained model with the recommendation generation module will be thoroughly examined. This involves testing the generation of top-k recommendations and ensuring that the ranking and scoring mechanisms work as expected. The system's scalability will also be tested by gradually increasing the size of the input data and monitoring the performance and resource utilization. Any bottlenecks or errors in the pipeline will be identified and resolved to guarantee a smooth end-to-end operation of the sequential recommendation system.

## Design and Implementation

### Find related projects

#### ASGNN (<https://github.com/HduDBSI/ASGNN>)

This project presents an Attentive Sequential Model Based on Graph Neural Network for Next POI (Point - of - Interest) Recommendation. It aims to leverage graph neural network techniques to make more accurate sequential recommendations in the context of POI prediction.

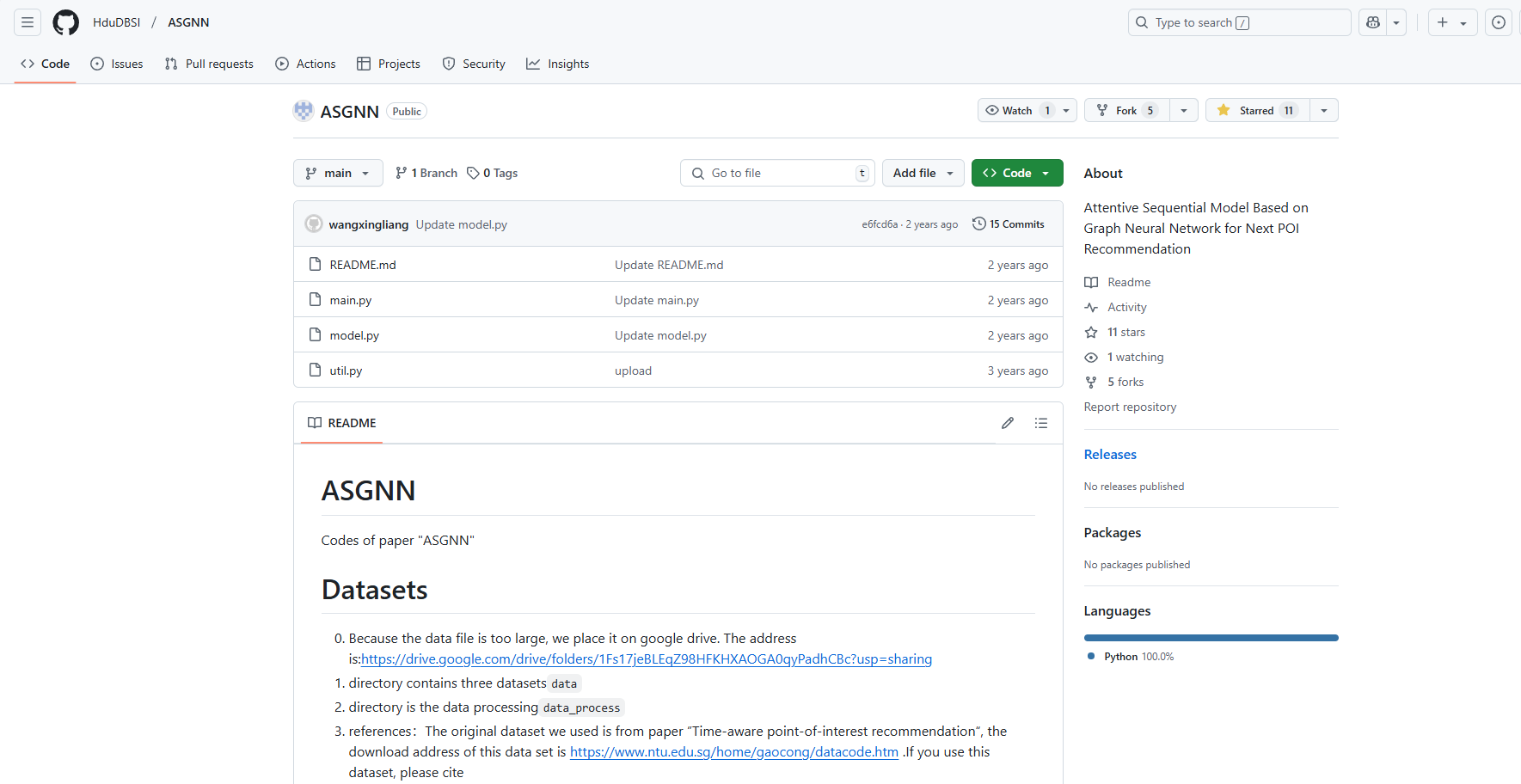


Figure 1: ASGNN

#### RecBole - GNN (<https://github.com/RUCAIBox/RecBole-GNN>)

It is an efficient and extensible GNNs enhanced recommender library based on RecBole. The library covers multiple types of recommendation algorithms, including those for sequential recommendation. It incorporates various state - of - the - art sequential recommendation models like SR - GNN, GC - SAN, etc., making it a comprehensive resource for researchers and practitioners in this field.

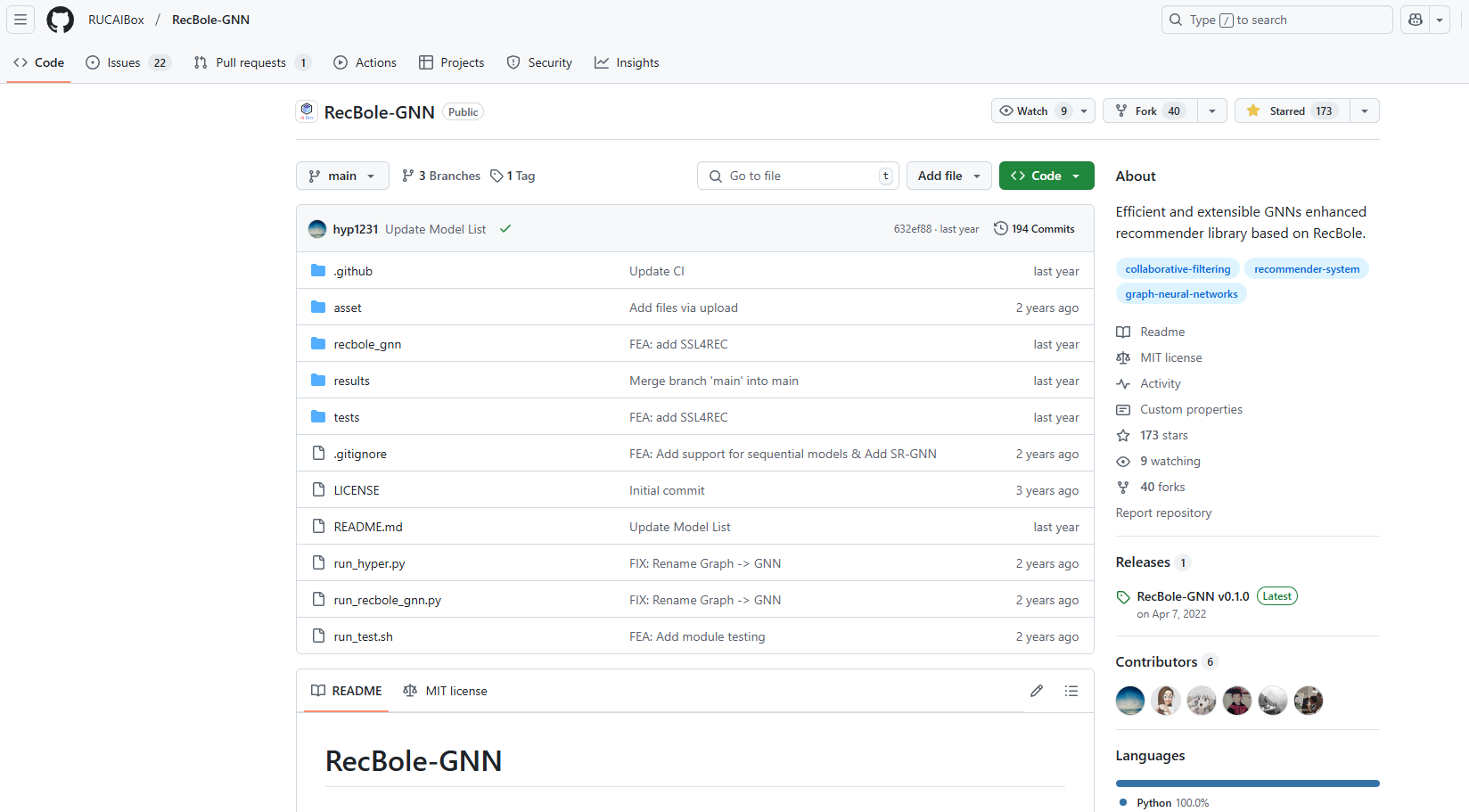


Figure 2: RecBole - GNN

#### SelfGNN (<https://github.com/zekarias-tilahun/SelfGNN>)

It is a PyTorch implementation of the "SelfGNN: Self - supervised Graph Neural Networks without explicit negative sampling" paper. Although its main focus is on self - supervised graph neural networks without explicit negative sampling, the techniques and architectures used may have implications for sequential recommendation tasks, especially in terms of learning useful graph representations that can be applied to sequential data.

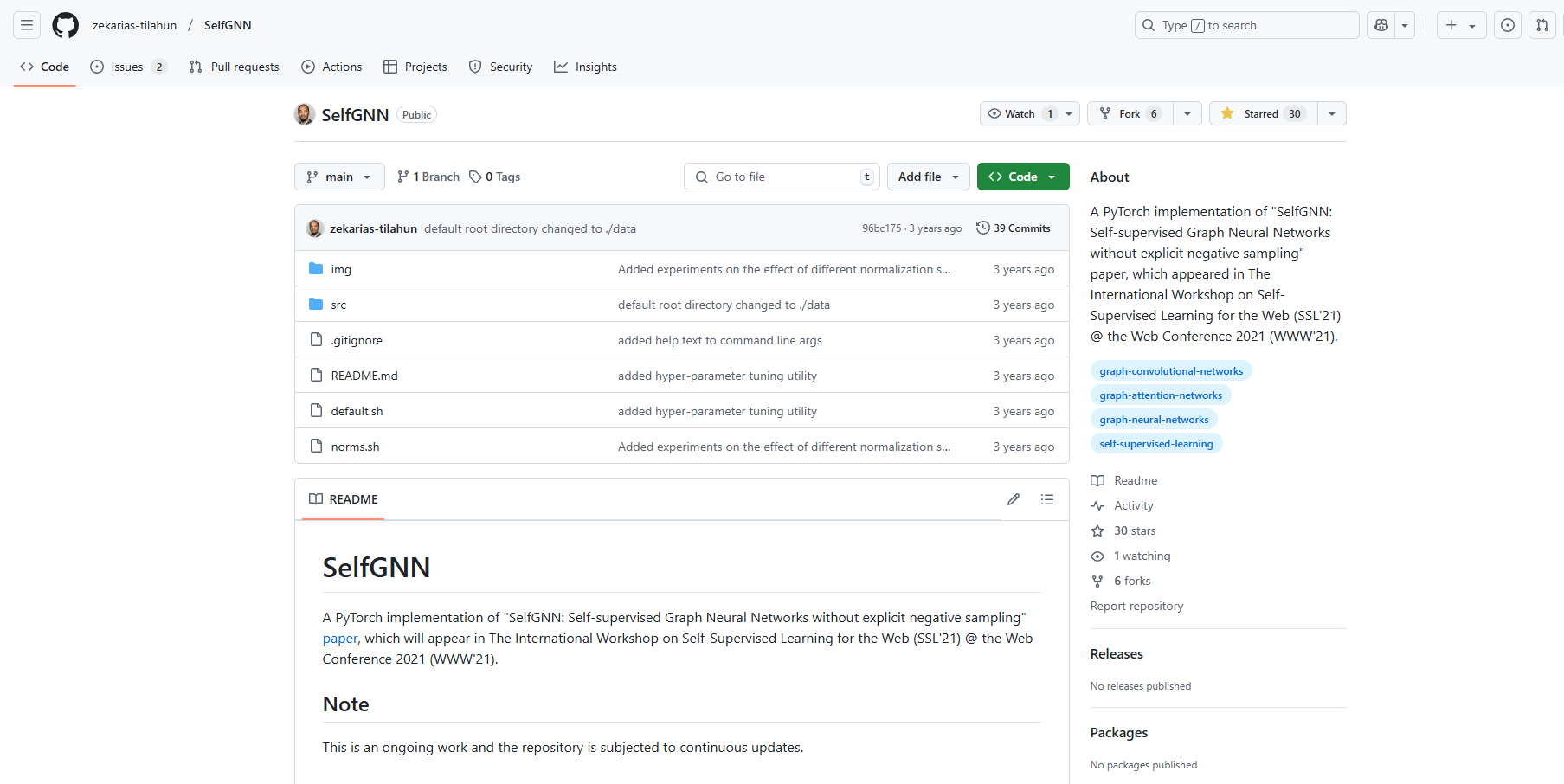


Figure 3: SelfGNN

### Improvement of design direction

#### Data Processing

ASGNN disseminates information regarding the origins and processing methodologies of its datasets through the integration of multi-source data. Information regarding user behavior in practical applications typically originates from multiple sources. To enhance the feature representation of sequence data, we can draw inspiration from its data processing methodologies, incorporate a broader range of data sources, including social network and temporal data, and manage diverse dataset types in a manner akin to RecBole-GNN. Simultaneously, SelfGNN facilitates an array of data augmentation techniques to enhance augmentation strategies. Moreover, additional specialized methodologies for data augmentation can be formulated to correspond with the attributes of the sequential recommendation task. To enhance the model's generalization ability, data augmentation may be performed in alignment with the temporal sequence of the series, or more robust training samples could be generated by amalgamating users' historical behavioral patterns.

#### Training Strategies

Hyperparameter optimization is crucial as SelfGNN offers a mechanism for OPTUNA hyperparameter adjustment, and a comparable automated hyperparameter optimization method can be employed during training. Simultaneously, employing techniques like Bayesian optimization facilitates a more efficient search for the optimal hyperparameter combination, hence minimizing the cost and duration of manual parameter tweaking. Moreover, the self-supervised learning methodology of SelfGNN presents an innovative concept for model training by integrating self-supervised and supervised learning. It enables the integration of traditional supervised learning objectives with self-supervised learning tasks, such as contrastive learning and node reconstruction. To enhance the model's efficacy in sequential recommendation tasks, it may be pretrained on extensive unlabeled sequence data and subsequently fine-tuned on labeled analysis.

### Determine dataset(<https://gitcode.com/gh_mirrors/se/Sequential-Recommendation-Datasets/overview>)

This project involves the selection of Amazon-related datasets. These datasets have significant advantages. Comprehensive sequential data is included, illustrating consumers' consecutive purchasing choices and assisting the model in identifying behavioral trends. They facilitate the creation of graphs depicting product interactions, utilizing graph neural networks to improve suggestion efficacy. The extensive scale and varied product categories facilitate the training of robust models and the assessment of recommendation approaches. Moreover, the repository contains preprocessing tools and documentation that facilitate efficient data preparation. In conclusion, the datasets align with the research objectives and can facilitate the advancement of efficient sequential recommendation models.

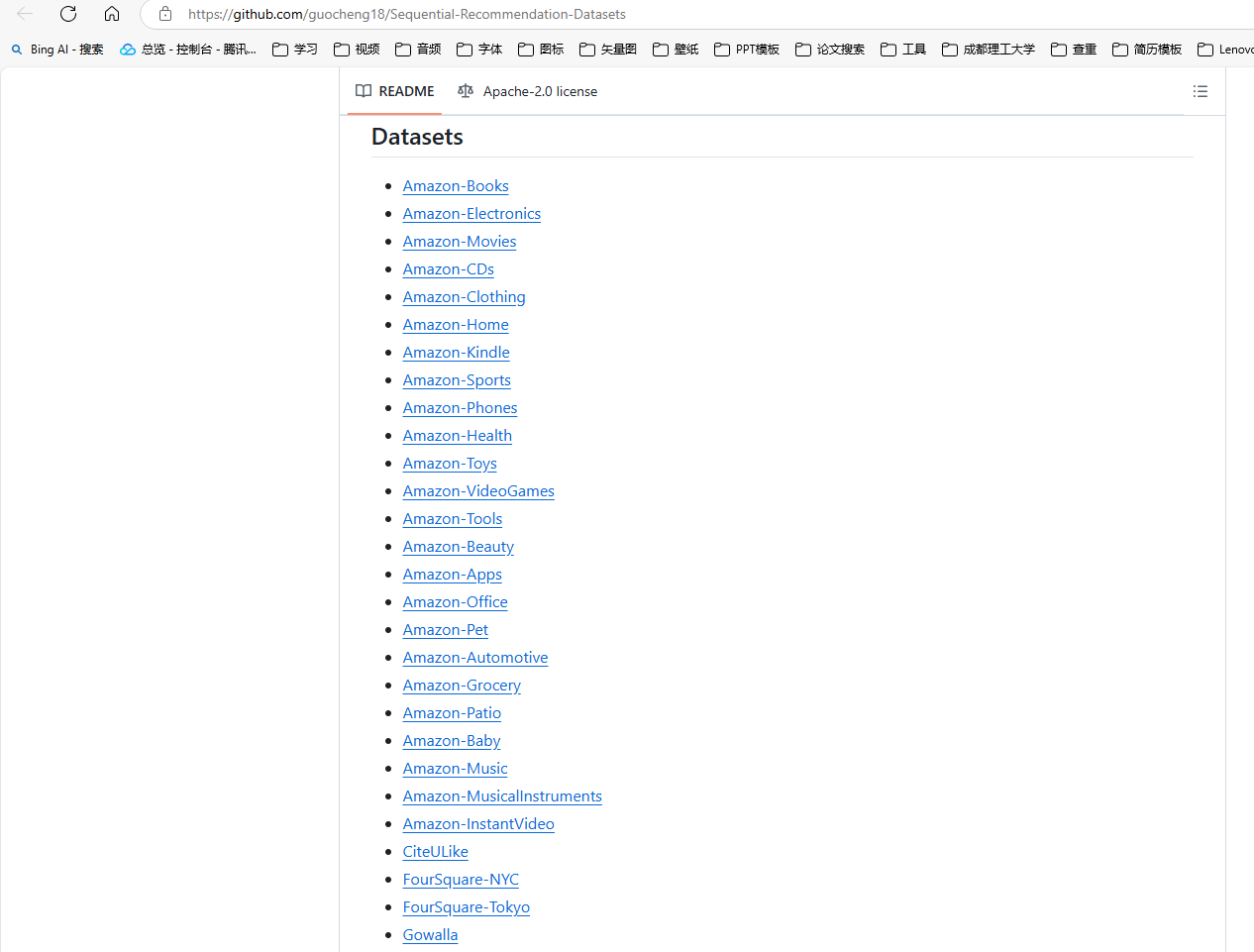


Figure 4: Dataset

# Project Management

## Activities

|  |  |
| --- | --- |
| **Objectives** | **Actions** |
| a. Review the current literature, databases, and technical schemes in the field of sequence recommendation | Collect the latest research papers on sequence recommendation within the last 5 years, especially those using deep learning methods. |
| Analyzes the current mainstream sequential recommendation models, and records their advantages and disadvantages. |
| Explore existing publicly available datasets for training and testing recommender systems. |
| Examine open source frameworks and technology stacks as well as specific graph processing libraries. |
| b. Designs and implements a sequential recommendation algorithm based on graph neural network | Define the core components of the algorithm, including how to build the user-item interaction graph and how to apply GNNS on the graph. |
| To implement the algorithm prototype, start with a simple GNN architecture and gradually increase the complexity. |
| Preliminary models are trained on selected datasets and key parameter adjustments during training are recorded. |
| Use version control to manage code updates and iterations. |
| c. Evaluate the performance of the proposed algorithm and optimize its predictive ability | Standard evaluation metrics are used to measure recommendation accuracy. |
| Cross-validation was performed to determine the best hyperparameter Settings |
| The model is regularized to prevent overfitting and an early stopping strategy is used. |
| Experiment with different combinations of loss functions and optimizers to find the most efficient configuration. |
| d. To verify the practical application effect of the algorithm, and collect the feedback of end users | Deploying recommender systems in the real world. |
| An A/B test is designed to compare the performance difference between the new algorithm and the existing recommender systems. |
| Investigate the end user satisfaction with the recommendation results |
| The algorithm is further adjusted based on user feedback and how these improvements affect the user experience is documented. |

Table 2 : Activities table

## Schedule

Below is the schedule of this project, the schedule starts at October 28th 2024 and end at

March 27th 2025.

The Gantt graph was shown below:

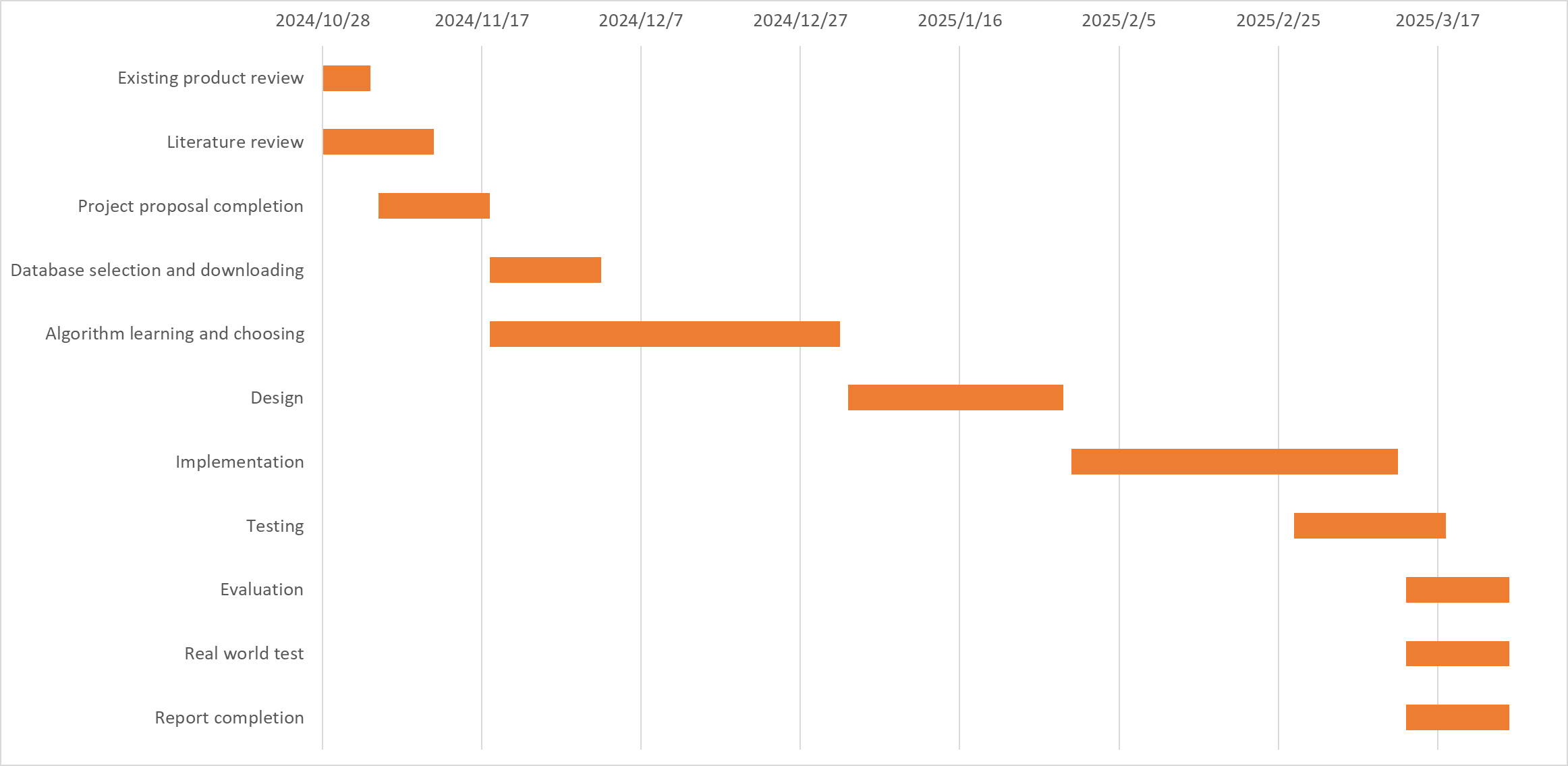


Figure 4: Gantt chart of the schedule

## Project Version Management

Each version will be stored in the Git repository with the help of GitHub.

Git repository link : https://github.com/B-starshine/project

## Project Data Management

All project logs, reports and literature generated by this project will be saved in Baidu online disk, and each version will be numbered and iterated. Baidu Cloud desk link: <https://pan.baidu.com/s/1DMeNeMMzVC4IXmbEU8ETYA?pwd=ckjk>

## Project Deliverables

There are in total 5 deliverables, the project proposal, the progress report, final report, presentation files and the project code.

# Professional Issues and Risk:

## Risk Analysis

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Risk ID** | **Potential Risk** | **Cause ID** | **Potential Causes** | **Severity** | **Likelihood** | **Risk** | **Mitigation ID** | **Mitigation** |
| R1.1 | Missed deadline | C1.1.1 | Illness | 1 | 3 | 3 | M1.1.1 | Register exceptional circumstances if ill. |
| C1.1.2 | Poor time management | 4 | 3 | 12 | M1.1.2 | Strictly follow the Gantt chart |
| R1.2 | Model over - fitting | C1.2.1 | Insufficient training data | 3 | 3 | 9 | M1.2.1 | Use data augmentation techniques |
| C1.2.2 | High model complexity | 4 | 2 | 8 | M1.2.2 | Implement regularization methods |
| C1.2.3 | Improper choice of learning rate or optimization | 5 | 3 | 15 | M1.2.3 | Conduct a hyper - parameter search for optimal learning rate and optimization algorithm |
| R1.3 | Slow convergence | C1.3.1 | Inappropriate learning rate | 2 | 3 | 6 | M1.3.1 | Perform hyper - parameter tuning |
| C1.3.2 | Poorly initialized weights | 3 | 2 | 6 | M1.3.2 | Use advanced initialization techniques |
| R1.4 | Loss of data | C1.4.1 | Ignoring sequential dependencies | 4 | 3 | 12 | M1.4.1 | Incorporate sequential information in the model |
| C1.4.2 | Inadequate graph representation | 3 | 3 | 9 | M1.4.2 | Improve graph construction and feature engineering |

Table 3: Risk Analysis

In the project, a comprehensive risk analysis table has been built to identify, assess and mitigate various risks during execution. Its structure covers columns like Risk ID, Potential Risk Causes, Severity, Likelihood, Risk Priority and Risk Mitigation, enabling the project team to comprehensively and deeply consider and handle risks.

Under the “Missed deadline (R.1.1)” category, there are three potential causes. For “Illness (C1.1.1)”, with a severity of 1, likelihood of 3 and priority of 3, recording exceptional circumstances is the mitigation. For “Poor time management (C1.1.2)” of higher severity 4, likelihood 3 and priority 12, following the Gantt chart is emphasized. And for “Insufficient training data (C1.1.3)” with severity 3, likelihood 3 and priority 9, though no specific mitigation is currently listed, it needs attention as it may delay deadlines.

For “Model over - fitting (R.1.2)”, two potential causes exist. “High model complexity (C1.1.1)” has a severity of 4, likelihood of 2 and priority of 8, and data augmentation techniques are suggested. “Improper choice of learning rate or optimization (C1.1.2)” has a severity of 5, likelihood of 3 and priority as high as 15, so a series of measures including data augmentation, regularization and hyper - parameter searches are proposed.

In “Slow convergence (R.1.3)”, for “Inappropriate learning rate (C1.1.1)” with severity 2, likelihood 3 and priority 6, hyper - parameter tuning is proposed. For “Poorly initialized weights (C1.1.2)” with severity 3, likelihood 2 and priority 6, advanced initialization techniques are recommended.

In “Loss of data (R.1.4)”, two potential causes are there. For “Ignoring sequential dependencies (C1.1.1)” with severity 4, likelihood 3 and priority 12, incorporating sequential information is the mitigation. For “Inadequate graph representation (C1.1.2)” with severity 3, likelihood 3 and priority 9, improving graph construction and feature engineering are believed to be effective.

In summary, the risk analysis table offers an efficient and practical risk management framework for the project team, by detailing potential causes, accurately assessing severities and likelihoods, and proposing targeted mitigations, enabling the project to predict risks in advance and take proactive preventive measures for smooth progress.

## Professional Issues

### Legal Issues

Intellectual Property Rights (IPR) are crucial for the development of graph-based neural network models for sequential recommendations, as they may involve the construction of original components, rendering intellectual property protection essential. Adherence to copyright laws necessitates the provision of appropriate licenses and attribution when employing pre-existing algorithms, code snippets, and open-source libraries. Concurrent adherence to data protection regulations is essential, as this project will utilize user data for modeling purposes. This encompasses obtaining consent for data collecting, guaranteeing secure storage, and anonymizing or pseudonymizing data. Consent must be acquired prior to the collection of interaction data, and the data should be encrypted to avert unauthorized access. Moreover, due to the potential risk of computer misuse, the project's recommendation system must be safeguarded against unauthorized access in compliance with the Computer Misuse Act of 1990 by implementing robust security measures to prevent hackers from compromising its functionality, altering data, or exploiting it for unethical purposes.

### Social Issues

Social responsibility necessitates that the sequential recommendation system employed in this project be structured to foster beneficial social outcomes, such as shaping user behavior (e.g., in e-commerce, where it can affect product purchases and business success) and preventing the endorsement of detrimental products. Furthermore, due to its influence on communities, the implementation of this recommendation system may include specific ramifications, as excessive dependence can foster echo chambers. Consequently, methods must be implemented to foster serendipity and diversity, thereby enhancing user experiences and benefiting society.

### Ethical Issues

The sequential suggestion process must be transparent and comprehensible to users through the application of suitable methodologies. Furthermore, in alignment with the ACM and BCS codes of conduct, unethical practices must be eschewed throughout the project to guarantee that all associated research and development activities are conducted with integrity, and that assertions regarding the performance or capabilities of the sequential recommendation system are founded on precise and verifiable data. Graph neural networks trained on biased historical data may lead to inequitable outcomes; hence, it is essential to employ strategies for bias detection and mitigation.

### Environmental Issues

The progression and improvement of the system may result in hardware obsolescence and electronic waste; nevertheless, the environmental consequences of hardware upgrades and disposal, including the recycling of obsolete components or the use of sustainable hardware, must be taken into account. In accordance with the BCS Code of Conduct, it is essential to evaluate environmental factors for this project by adopting proactive strategies to minimize the environmental footprint and staying informed about industry best practices and emerging standards, such as those proposed by IEEE concerning sustainable computing, to ensure alignment with environmental goals. Energy Utilization: The training and implementation of graph neural network models for sequential recommendations are resource-intensive and energy-consuming; therefore, it is crucial to investigate ways for improving model architecture and training algorithms to alleviate this problem.

# References

[1] Chen, Junyang, Jingcai Guo, Qin Zhang, Kaishun Wu, Liangjie Zhang, Victor C.M. Leung, Huan Wang, and Zhiguo Gong. ‘Unveiling User Interests: A Deep User Interest Exploration Network for Sequential Location Recommendation’. *Information Sciences* 689 (January 2025): 121416. <https://doi.org/10.1016/j.ins.2024.121416>.

[2] Chen, Ruixin, Jianping Fan, and Meiqin Wu. ‘MC-RGN: Residual Graph Neural Networks Based on Markov Chain for Sequential Recommendation’. *Information Processing & Management* 60, no. 6 (November 2023): 103519. <https://doi.org/10.1016/j.ipm.2023.103519>.

[3] Hao, Yongjing, Jun Ma, Pengpeng Zhao, Guanfeng Liu, Xuefeng Xian, Lei Zhao, and Victor S. Sheng. ‘Multi-Dimensional Graph Neural Network for Sequential Recommendation’. *Pattern Recognition* 139 (July 2023): 109504. <https://doi.org/10.1016/j.patcog.2023.109504>.

[4] He, Xiangnan, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. ‘LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation’. arXiv, 7 July 2020. <http://arxiv.org/abs/2002.02126>.

[5] Huang, Zhen, Zhongchuan Sun, Jiaming Liu, and Yangdong Ye. ‘Group-Aware Graph Neural Networks for Sequential Recommendation’. *Information Sciences* 670 (June 2024): 120623. <https://doi.org/10.1016/j.ins.2024.120623>.

[6] Li, Xiao, Li Sun, Mengjie Ling, and Yan Peng. ‘A Survey of Graph Neural Network Based Recommendation in Social Networks’. *Neurocomputing* 549 (September 2023): 126441. <https://doi.org/10.1016/j.neucom.2023.126441>.

[7] Wang, Xiang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. ‘Neural Graph Collaborative Filtering’. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 165–74, 2019. <https://doi.org/10.1145/3331184.3331267>.

[8] Wu, Shu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. ‘Session-Based Recommendation with Graph Neural Networks’. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, no. 01 (17 July 2019): 346–53. <https://doi.org/10.1609/aaai.v33i01.3301346>.

[9] Xu, Chengfeng, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. ‘Graph Contextualized Self-Attention Network for Session-Based Recommendation’. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 3940–46. Macao, China: International Joint Conferences on Artificial Intelligence Organization, 2019. <https://doi.org/10.24963/ijcai.2019/547>.

[10] Yin, Fulian, Tongtong Xing, Meiqi Ji, Zebin Yao, Ruiling Fu, and Yuewei Wu. ‘Multipath-Guided Heterogeneous Graph Neural Networks for Sequential Recommendation’. *Computer Speech & Language* 87 (August 2024): 101642. <https://doi.org/10.1016/j.csl.2024.101642>.

[11] Zhang, Yihu, Bo Yang, Haodong Liu, and Dongsheng Li. ‘A Time-Aware Self-Attention Based Neural Network Model for Sequential Recommendation’. *Applied Soft Computing* 133 (January 2023): 109894. <https://doi.org/10.1016/j.asoc.2022.109894>.