

UNDERGRADUATE PROJECT REPORT

|  |  |
| --- | --- |
| **Project Title:** | **Sequential Recommendations with Graph Neural Networks** |
| **Surname:** | **Chen** |
| **First Name:** | **An Ju** |
| **Student Number:** | **202118010227** |
| **Supervisor Name:** | **Joojo Walker** |
| **Module Code:** | **CHC 6096** |
| **Module Name:** | **Project** |
| **Date Submitted:** | **May 6,2025** |

**Chengdu University of Technology Oxford Brookes College**

**Chengdu University of Technology**

**BSc (Single Honours) Degree Project**

Programme Name: **Computer Science**

Module No.: **CHC 6096**

Surname: Chen

First Name: An Ju

Project Title: Sequential Recommendations with Graph Neural Networks

Student No.: 202118010227

Supervisor: Joojo Walker

2ND Supervisor (if applicable): **Not Applicable**

Date submitted: **May 6, 2024**

*A report submitted as part of the requirements for the degree of BSc (Hons) in Computer Science*

*At*

**Chengdu University of Technology Oxford Brookes College**

# **Declaration**

**Student Conduct Regulations**:

Please ensure you are familiar with the regulations in relation to Academic Integrity. The University takes this issue very seriously and students have been expelled or had their degrees withheld for cheating in assessment. It is important that students having difficulties with their work should seek help from their tutors rather than be tempted to use unfair means to gain marks. Students should not risk losing their degree and undermining all the work they have done towards it. You are expected to have familiarised yourself with these regulations.

<https://www.brookes.ac.uk/regulations/current/appeals-complaints-and-conduct/c1-1/>

Guidance on the correct use of references can be found on www.brookes.ac.uk/services/library, and also in a handout in the Library.

The full regulations may be accessed online at <https://www.brookes.ac.uk/students/sirt/student-conduct/>

If you do not understand what any of these terms mean, you should ask your Project Supervisor to clarify them for you.

**I declare that I have read and understood Regulations C1.1.4 of the Regulations governing Academic Misconduct, and that the work I submit is fully in accordance with them**.

2025-5-1_22-24-472025-5-1_22-23-40

Signature …………………………………………. Date …………………………………………

REGULATIONS GOVERNING THE DEPOSIT AND USE OF OXFORD BROOKES UNIVERSITY MODULAR PROGRAMME PROJECTS AND DISSERTATIONS

Copies of projects/dissertations, submitted in fulfilment of Modular Programme requirements and achieving marks of 60% or above, shall normally be kept by the Oxford Brookes University Library.

2025-5-1_22-24-472025-5-1_22-23-40**I agree that this dissertation may be available for reading and photocopying in accordance with the Regulations governing the use of the Oxford Brookes University Library.**

Signature …………………………………………. Date ………………………………………….

# **Acknowledgment**

First, I would like to express my deepest gratitude and respect to my supervisor, Dr. Joojo Walker, for his selfless academic assistance, guidance, and spiritual encouragement. Next, I would like to thank my parents for their boundless love and support. Finally, I would like to thank my wonderful friends, who have inspired me to keep moving forward.

# **Table of Contents**

[Declaration 3](#_Toc25946)

[Acknowledgment 4](#_Toc31240)

[Table of Contents 5](#_Toc7983)

[Abstract 7](#_Toc15960)

[Abbreviations 8](#_Toc19569)

[Glossary 9](#_Toc22814)

[Chapter 1 Introduction 12](#_Toc24863)

[1.1 Background 12](#_Toc1587)

[1.2 Aim 12](#_Toc18975)

[1.3 Objectives 13](#_Toc26001)

[1.4 Project Overview 13](#_Toc7049)

[1.4.1 Scope 13](#_Toc26349)

[1.4.2 Audience 14](#_Toc27212)

[Chapter 2 Background Review 15](#_Toc26432)

[Chapter 3 Methodology 18](#_Toc29728)

[3.1 Approach 18](#_Toc14204)

[3.2 Technology 23](#_Toc4063)

[3.3 Project Version Management 24](#_Toc15872)

[Chapter 4 Implementation and Results 25](#_Toc29108)

[4.1 Implementation 25](#_Toc12418)

[4.1.1 Dataset statistics 25](#_Toc27080)

[4.1.2 Evaluation metrics 26](#_Toc4753)

[4.1.3 Parameter Settings 27](#_Toc14678)

[4.2 Results 28](#_Toc17626)

[Chapter 5 Professional Issues 40](#_Toc14660)

[5.1 Project Management 40](#_Toc5073)

[5.1.1 Activities 40](#_Toc9917)

[5.1.2 Schedule 41](#_Toc26737)

[5.1.3 Project Data Management 42](#_Toc19804)

[5.1.4 Project Deliverables 42](#_Toc5168)

[5.2 Risk Analysis 42](#_Toc17384)

[5.3 Professional Issues 44](#_Toc14680)

[5.3.1 Ethical Issues 44](#_Toc22447)

[5.3.2 Social issue 45](#_Toc7012)

[5.3.3 Legal impact 46](#_Toc25605)

[5.3.4 Environment Issues 47](#_Toc16151)

[Chapter 6 Conclusion 49](#_Toc32370)

[6.1 Findings & Reflections 49](#_Toc16478)

[6.2 Limitations 49](#_Toc26002)

[6.3 Future Work 50](#_Toc13379)

[References 52](#_Toc20405)

# **Abstract**

With the rise of the era of information, the number of pieces is exploding. The emergence of recommendation algorithms has become an important means for us to better select our favorite works in the field of navigation, which improves user satisfaction and platform competition. Accurate prediction of users' next interests is particularly important for sequences. Traditional sequence recommendation algorithms cannot effectively balance long- and short-term user needs, are not suitable for cases with less data samples, and are unable to capture slight changes in user behavior.

Aiming at the above problems, this project identified and studied three sequence recommendation models (HGN, SRGNN, and GCSAN) around graph neural networks. Firstly, HGN uses instance gating and feature gating to filter out noise, captures user's current preference and lasting interest at the same time, also models projects' relationship explicitly by using a project-project product module; Secondly, SRGNN makes full use of the graph neural network technology to transform the session sequences into graph-understandable forms so as to solve the problem of sparsity due to insufficient samples; Finally, GCSAN combines GNNs and self attention technique together, builds the graph adaptively with the latest self attention and all historical attentions to solve the traditional sequence problems and optimize the recommendation effect. This project will compare these three models and conduct performance analysis with three real datasets in reality.

***Keywords:Sequence recommendation, recommendation system, graph neural network, SRGNN, HGN, GCSAN***

# **Abbreviations**

|  |  |
| --- | --- |
| Abbreviations | Definitions |
| HGN | Hierarchical Gating Network |
| SRGNN | Session - based Recurrent Graph Neural Network |
| GCSAN | Graph - Convolutional Self - Attention Network |
| MRR | Mean Reciprocal Rank |
| Recall@k | Recall at rank k |
| Hit@k | Hit Rate at rank k |
| Precision@k | Precision at rank k |

# **Glossary**

|  |  |
| --- | --- |
| Term | Explanation |
| Sequential Recommendation | Predict users' next interactions based on historical interaction sequences, mine behavior patterns for accurate recommendations, applied in e-commerce, media, etc. |
| Recommendation System | Use algorithms to filter and recommend items according to users' behaviors and preferences, alleviating information overload and enhancing user experience and business value. |
| Graph Neural Network | Process graph-structured data, learn features through information propagation on nodes and edges, used to explore complex dependencies between items in sequential recommendation. |
| SRGNN | For session-based recommendation, model session sequences as graphs, learn node vectors, and predict user clicks by combining global and current session interests. |
| HGN | Comprising feature gating, instance gating, and item-item product modules, it captures users' long-term and short-term interests and characterizes item relationships to improve recommendation accuracy. |
| GCSAN | Combines self-attention techniques with graph neural networks, builds graphs dynamically to record local dependencies, and learns global dependencies using self-attention to forecast user behaviour. |
| NDCG | Examine the suggestion rankings' accuracy while taking item relevance rankings and scores into account. Rankings that are closer to real user interactions are indicated by higher values. |
| Recall | Measure the coverage of relevant items in recommendation results. Recall@k represents the proportion of relevant items actually interacted with by users among the top k recommended items. |
| MRR | Calculate the average of the reciprocals of the ranks of the first relevant item in the recommendation list for each query, reflecting the recommendation system's ranking ability for relevant items. |
| Hit Rate | Hit@k represents the probability that at least one item among the top k recommended items matches the user's actual interaction item. |
| Precision | Precision@k refers to the proportion of truly relevant items among the top k recommended items, reflecting the precision of recommendation results. |
| BPR | An optimization method based on pairwise comparison, used for implicit feedback data, maximizing the probability that positive samples are recommended before negative samples to learn user preferences. |
| GRU | A variant of RNN, solving gradient problems in long sequence training through reset and update gates, used for learning sequence features. |
| LSTM | An enhanced RNN that uses input, forget, and output gates to handle long-term dependencies in lengthy sequences improves user behaviour comprehension and prediction. |
| Markov Chain | Assumes that the future state of a system depends only on its current state, previously used to analyze item transition probabilities but with limitations. |
| CNN | Processes grid data, automatically extracts features, and can capture local patterns in sequences to assist with recommendations. |
| Self-Attention Mechanism | Dynamically assigns weights to each position in the sequence, captures long-range dependencies between items, and enhances the modeling ability of user interests. |
| Data Sparsity | Scarce user-item interaction data with numerous missing values in the interaction matrix, affecting the learning and quality of recommendation algorithms. |
| Item-Item Collaborative Filtering | Recommendation based on users' common preferences for items, calculating item similarity, but limited in handling data sparsity. |
| User-Item Interaction Graph | Represents user-item interaction relationships as a graph, mining behavior patterns and item associations to provide a data basis for recommendations. |
| Model Overfitting | A model performs well on training data but poorly on new data, with decreased generalization ability, affecting recommendation accuracy. |
| Hyperparameter Tuning | Manually modify the model's hyperparameters to maximise its performance on test or validation sets using techniques like grid search. |
| Model Regularization | Adopt L1, L2 regularization, Dropout, etc., to prevent model overfitting and improve generalization ability. |
| Early Stopping | Monitor validation set performance during training. Stop training if performance does not improve for multiple consecutive rounds to prevent overfitting. |
| Cross-Validation | Divide the dataset for multiple rounds of training and testing, comprehensively evaluating model performance to avoid biases from dataset division. |
| A/B Testing | Compare the effects of two recommendation schemes, randomly divide users into groups, and optimize the recommendation system based on user feedback. |
| IPR | Legal rights protecting intellectual labor achievements, covering the protection of algorithms, codes, etc., in recommendation model development. |
| Data Protection Regulations | Control data processing practices to safeguard the security and privacy of personal information. Recommendation systems have to abide by applicable laws. |
| Computer Misuse | Unauthorized access, modification, etc. of computer systems. Measures are needed to ensure the secure operation of recommendation systems. |

# **Chapter 1 Introduction**

## **1.1 Background**

With the increase of digital media and e-commerce platforms, the amount of information that each person can obtain is increasingly explosive, so personal recommendation systems are needed. Personalized recommendation systems allow individuals to filter useful contents from a large amount of content by screening, while at the same time allowing content providers to make better offers to their customers. Session-based recommendation using GNNs belongs to effective sequential recommendation systems based on graph neural networks (GNN). Using SRGNN [2] has been proven to have good results when dealing with complicated relationship situations between data such as user behavior series. With it comes another model called HGN [1] and its associate GCSAN [3], which provide novel ideas and methods for the field of sequential recommendations.

In conclusion, this article will study these three new models: HGN, SRGNN, and GCSAN. Finally, we construct a system used for sequence recommendation based on these models. Then we combine our recommended sequence product into a huge platform full of products, so users can quickly find things they like from hundreds or thousands of choices on the website/app. Sections 1.2 - 1.4 will briefly describe the plan of the whole project and the introduced product. Chapter 2 introduces current state-of-the-art recommendation systems and details the idea, advantages, disadvantages, designing process, etc., of the mentioned three models: HGN, SRGNN, and GCSAN. Chapters 3 & 4 will talk more about how these systems were implemented and how well these applications worked. We will do some comparisons of these systems' performances and evaluations. In Chapter 5, I will talk about potential risks and problems regarding real-world applications concerning the practical use of different types of models. In the last part, I will summarize my entire work and reflect upon my own contributions and deficiencies in building up a proper recommendation system using any other kind of sequential recommendation technology.

## **1.2 Aim**

In this project, we hope to explore the use of graph neural networks in building fast and accurate sequence recommendation systems and recording methods for users' behavior modes for correct suggestions. Our work is based on three models (HGN, SRGNN, GCSAN) that aim to improve the efficiency of recommendation systems.

## **1.3 Objectives**

The objectives of this project include:

1. Fully study three sequence recommendation models based on graph neural networks, namely HGN (Hierarchical Gate Network), SRGNN (session-based recommendation graph neural network), and GCSAN (graph contextualised self-attention network), investigate the relevant literature, database, and technical solution to explore its internal laws, structure, and basic technical problems.

2. Construct and adopt sequence recommendation systems based on the three models mentioned above respectively, and excavate hidden patterns from user behavior sequence data using graph neural networks.

3. Use different datasets to deeply evaluate the three models, compare and analyze their performance in terms of suggestion accuracy, efficiency, and stability, and summarize their pros and cons.

4. Explore the evaluation results on many datasets to discover the available application occasions of each model and form a good premise for choosing the optimal model for practice, while collecting information from real-world applications to enhance model performance.

## **1.4 Project Overview**

This project investigates the utilization of graph neural network technology to enhance the effectiveness of sequence recommendation systems via a thorough analysis of three models: hierarchical gated networks (HGN), sequential recommendation graph neural network models (SRGNN), and graph contextualized self-attention networks (GCSAN). Recommendation systems that accurately discern customer preferences can provide more tailored services, which holds great significance for service-oriented enterprises reliant on user engagement and retention. This proposal will discuss the research foundation, methodologies for the three models, project management strategy, and specific implementation techniques of the project.

### **1.4.1 Scope**

This project aims to employ graph neural networks to examine three models: Hierarchical Gate Network (HGN), Session-based Recommendation Graph Neural Network (SRGNN), and Graph Contextualised Self-Attention Network (GCSAN), with the objective of developing an efficient sequence recommendation system that enhances recommendation precision and optimizes user experience. This research holds substantial significance, as it has the potential to substantially reduce user search time for relevant content, attract a larger user base to recommendation systems, and enable broader application of these systems across diverse domains, including e-commerce, media, and social networking. The project evaluates the performance of the three models in various scenarios, pinpointing their strengths and weaknesses, thereby establishing a solid foundation for optimizing recommendation systems to effectively address user needs and maximize their potential.

### **1.4.2 Audience**

This project aims to develop an efficient sequence recommendation system via thorough analysis of three graph neural network (GNN) models: hierarchical gateways network (HGN), session-based recommendation graph neural network (SRGNN), and graph contextualized self-attention network (GCSAN). The goal is to enhance recommendation accuracy and elevate user satisfaction. The research outcomes will directly impact content producers and various service platforms, such as social media networks, e-commerce sites, and video-on-demand services. These systems can attract and retain consumers, improve user engagement, and ultimately increase revenue through the use of more precise recommendation algorithms. Additionally, marketers and advertisers can utilize this information to target customers more accurately and execute targeted marketing campaigns.

The primary audience for this initiative consists of enterprise decision-makers, researchers, and technology developers. The technologies and resources of this project will facilitate technical developers in integrating advanced recommendation features into existing systems. The academic community will augment its understanding and expand the application of graph neural networks in recommendation systems. Additionally, business decision-makers may leverage the project outcomes to formulate more competitive strategies and augment market share. The project's findings possess substantial value for any enterprise or individual reliant on user engagement and the enhancement of personalized experiences.

# **Chapter 2 Background Review**

Sequence recommendation is an essential task aimed at forecasting the material users are probable to engage with next, based on their prior interaction sequences [4,5,6]. Various strategies have emerged in the advancement of sequence recommendation. One strategy is to incorporate group behaviour and utilise collective intelligence to improve the modelling of user preferences [2,7]. Concurrently, sequence recommendation models have progressed to emphasise multi-dimensional transformation links among items, aiming to alleviate the adverse effects of data sparsity [8,9]. Furthermore, by incorporating diverse information such as product categories and timestamps, models can reveal users' profound interests, thereby offering substantial support for accurate suggestions [10,11].

Initially, Markov chain-based models garnered considerable interest in the domain of sequence recommendation [13,14]. These models, characterised by their straightforward and intuitive state transition processes, first simulated user behaviour by examining the transition probabilities between contiguous items and forecasted the items with which users may next engage. The memoryless assumption of Markov chains confines them to utilising only first-order or finite-order historical data. Complex and dynamic user behaviour patterns provide challenges for such models, hindering their ability to capture long-term and intricate connections, which results in considerable restrictions in the accuracy and efficacy of recommendations [13,14].

To address this constraint, recurrent neural networks (RNNs) and their derivatives, including long short-term memory (LSTM) and gated recurrent units (GRUs), have been integrated into the sequence recommendation field [6,7,25]. RNNs retain information regarding users' historical behaviour sequences via hidden states, thus preserving long-term dependencies within user behaviour sequences [6,20]. In contrast to Markov chains, RNNs more effectively encapsulate the temporal progression of user interests, hence enhancing suggestion efficacy to a certain degree. LSTMs have the capability to capture intricate long-term connections, enhancing the precision and efficacy of suggestions in the context of complicated and changeable user behaviour patterns. LSTMs and GRUs enhance RNN architecture via gating mechanisms, mitigating the prevalent problems of gradient vanishing and explosion in long sequences, thereby improving the model's capacity to manage long-range dependencies and execute long-sequence recommendation tasks more efficiently [20,25].

Graph neural networks (GNNs) utilised in session recommendation (SR-GNN) models are crucial in the optimisation of recommendation systems. This approach converts users' historical interaction sequences into a graph structure, with each item depicted as a node and the sequences and interaction links between items illustrated as edges [2,15]. Utilising the robust representation capabilities of graph neural networks, SR-GNN may thoroughly explore the intricate relationships among user behaviours, facilitating a more nuanced examination of behavioural logic than conventional sequence models. In session-based recommendation contexts, SR-GNN adeptly models the transition patterns of items within a session, efficiently identifying user interest focal points while also accommodating abrupt shifts and variations in user preferences.

To mitigate data sparsity, SR-GNN employs a local subgraph information aggregation approach to deduce users' latent interests based on the local structure of user activity, even in contexts with limited samples [2]. By consolidating and refreshing node neighbourhood data, it perpetually augments node feature representations, hence improving the model's resilience to sparse data. Furthermore, SR-GNN facilitates the smooth incorporation of multi-dimensional heterogeneous data, including product categories and timestamps, into the graph structure learning process [2]. By integrating product category data as node attributes and modifying edge weights according to timestamp information, the recommendation outcomes can better represent users' enduring preferences and immediate requirements, thereby enhancing the precision and promptness of recommendations.

In practical applications, SR-GNN has attained notable success in product recommendations on e-commerce platforms and content recommendations on short video platforms [19,23]. In e-commerce contexts, it may precisely forecast products that users could find appealing next, based on their activity during shopping sessions, including clicks, cart additions, and purchases, hence enhancing user experience and platform conversion rates [19,23]. SR-GNN may assess sequences of consecutively seen videos and integrate this data with video tags and posting times to recommend material that corresponds more closely with users' interests, hence enhancing user engagement and activity. This technology offers a novel solution to address technical obstacles in intricate recommendation contexts and is anticipated to yield substantial performance enhancements in recommendation systems within e-commerce, social media, and content distribution sectors, establishing it as a pivotal focus for the advancement of future recommendation technologies. Moreover, the Hierarchical Gate Network (HGN) and the Graph Context Self-Attention Network (GC-SAN) have exhibited exceptional efficacy in sequence suggestion [1,3]. The HGN utilises feature gating and instance gating modules to selectively filter pertinent information at both the feature and instance levels, while the item-item product module elucidates item-item links, thereby enhancing the understanding of users' long-term and short-term interests [1,3]. GC-SAN integrates graph neural networks with self-attention techniques to dynamically create graph structures that capture local dependencies while learning global connections through self-attention, showcasing distinct improvements in recommendation efficacy [1,3]. Thorough investigation of these three models facilitates a thorough understanding of sequence recommendation methodologies utilising graph neural networks, establishing a robust foundation for the subsequent optimisation of recommendation systems [26].

# **Chapter 3 Methodology**

## **3.1 Approach**

The three models, SRGNN, HGN and GCSAN, are represented by their respective unique implementation processes when building a sequential recommendation system based on graph neural networks:

Constructing a session recommendation system based on SRGNN Constructing session graph structure data:

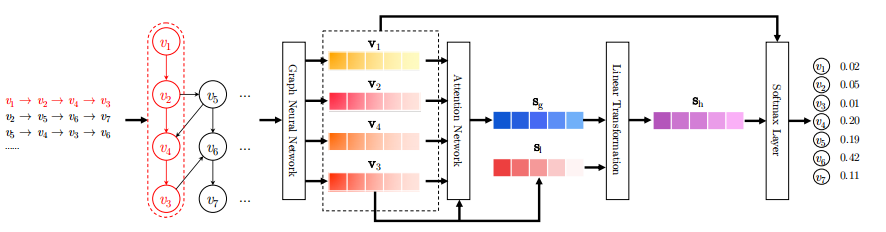


Figure 1: The workflow of the proposed SRGNN method [2].

We model all session sequences as session graphs. Then, each session graph is proceeded one by one and the resulting node vectors can be obtained through a gated graph neural network. After that, each session is represented as the combination of the global preference and current interests of this session using an attention net. Finally, we predict the probability of each item that will appear to be the next-click one for each session.

In a session-based recommendation scenario, each anonymized session sequence is significant. Using to represent the entire set of unique items involved in all sessions, the anonymous session sequence can be represented as a timestamped list , where represents the items clicked by the user in the session . When constructing the session graph, each session sequence is modeled as a directed graph . Each node in the graph represents an item , and each edge indicates that the user clicks before clicking in session . Since some items may appear repeatedly, each edge is given a normalized weight calculated as the number of occurrences of the edge divided by the starting node's out-degree. Subsequently, each item is embedded into a uniform embedding space, and the node vector represents the potential vector of the item learned by the graph neural network, and is the vector dimension. Based on these node vectors, each session can be represented by an embedding vector composed of the node vectors in the graph. Learning item embeddings using graph neural networks: graph neural networks are suitable for session recommendation, which can automatically extract session graph features and consider node connectivity information. For the node in the graph , the update function is as follows:

(1)

Among them, controls the weights, and are the reset and update gates, respectively, is the list of node vectors in the session , is the sigmoid function, is the element-by-element multiplication operator, and denotes the potential vectors of the node . The connection matrix determines how the nodes in the graph communicate, and is the two-column block corresponding to node in . It consists of two adjacency matrices and , which represent the weighted connections of outgoing and incoming edges in the session graph, respectively. For example, for the session , the corresponding graph and matrices are shown in Figure (corresponding to the original graph numbers). The final node vector is obtained by updating all nodes in the session graph until convergence.

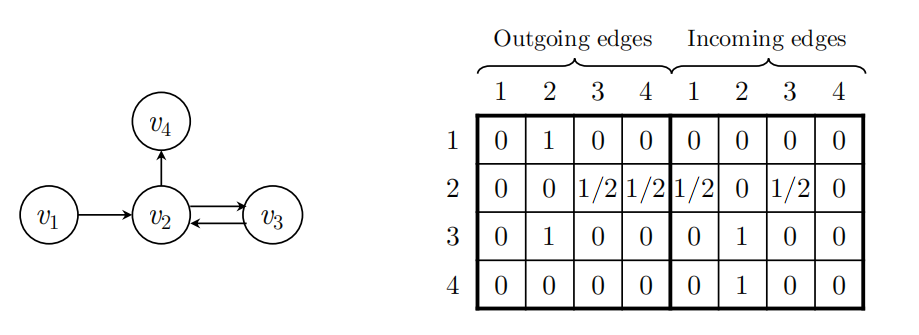


Figure 2: An example of a session graph and the connection matrix As [2].

Generating session embeddings and performing recommendation and model training: instead of assuming a unique user latent representation for each session, SR-GNN directly indicates the session live based on the nodes in the session. To predict the user's next click, SR-GNN combines the long-term preferences and current interests of the session and uses the combined embedding as the session embedding. The implementation of graph neural networks (GNNs) is located in the GNN class, where the GNNCell method implements node history and the forward method refreshes nodes several times until convergence. Input all the session graphs to the gated graph neural network to get all the node vectors. In order to get the session embedding vector ,first consider the local embedding of session ,for session ,the local embedding can be simply defined as the vector of the last clicked item , i.e. . Next, the global embedding of the session graph is considered by aggregating all node vectors ,using a soft-attention mechanism to better represent the global session preferences:

(2)

The parameters and , control the weights of the item embedding vectors. Finally, the hybrid embedding is computed by linearly transforming the concatenation of local and global embedding vectors

(3)

where the matrix compresses the two combined embedding vectors into the potential space . After obtaining the embeddings for each session, the score for each candidate item is computed by , and then the function is applied to obtain the model output vector . For each session graph, the loss function is defined as the cross-entropy between the predicted results and the true results, and the model is trained using the Back Propagation Through Time (BPTT) algorithm. In the compute\_scores method of the SessionGraph class, the logic for generating hybrid embeddings is implemented, and the train\_test function implements the training and testing process of the model.

Constructing a Sequential Recommender System Based on HGN:

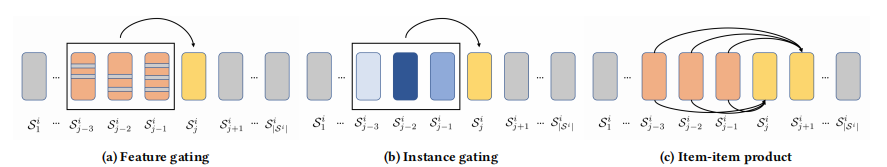


Figure 3:An illustrative example of the feature gating, instance gating, and item-item product modules [1].

In Figure 1a, the gray lines on items denote those latent features are masked off. In Figure 1b, the darker blue means the item is more important. In Figure 1c, the line linked between two items denotes the inner product, which captures the relations between the items users have accessed and the items users will access in the future.

Feature-Instance Gating and Item Relationship Capture: HGN consists of a feature gating module, an instance gating module and an item-item product module. In the feature gating module, the input item sequences are transformed into low-dimensional vector representations by the embedding layer. Inspired by the gated linear unit (GLU), the HGN realizes the selection of salient potential features of items based on user preferences by modifying the GLU and introducing the user embedding, which is given by :

(4)

The instance gating module then selects significant items that are helpful in predicting future items from the sequence processed by feature gating based on user preferences, with the formula

(5)

The item-item product module captures inter-item relationships by computing the inner product of the input and output item embeddings, with the equation:

(6)

The prediction and training: at the prediction level, combined with the classical matrix decomposition term, HGN is analyzed by the formula:

(7)

The user's prediction score for items is computed, where the different terms capture the relationship between the user's long-term interest, short-term interest, and item pairs, respectively.The HGN is optimized using a Bayesian Personalized Ranking (BPR) objective by minimizing the objective function, calculating the partial derivatives of the parameters using gradient descent and back-propagation, and adapting the learning rate adaptively using an Adam optimizer.

Building a Sequential Recommender System Based on GCSAN:

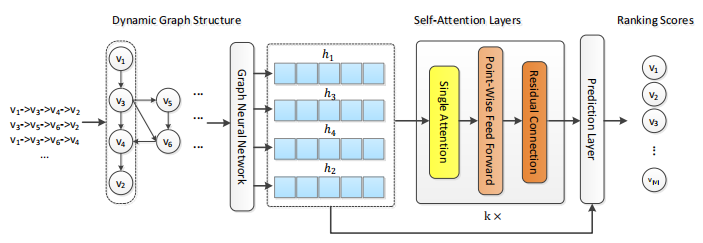


Figure 4: The general architecture of the proposed model [3].

We first construct a directed graph of all session sequences. Based on the graph, we apply graph neural network to obtain all node vectors involved in the session graph. After that, we use a multi-layer self-attention network to capture long-range dependencies between items in the session. In prediction layer, we represent each session as a linear of the global preference and the current interest of that session. Finally, we compute the ranking scores of each candidate item for recommendation.

Dynamic graph structure construction and node vector update: GCSAN first models each session sequence as a directed graph. For the session ,each item is used as a node, is used as an edge, and the weight of the edge is normalized according to the number of occurrences and the starting node out degree. With the constructed session graph, the node vectors are updated using graph neural network. For the node in the graph ,it is first passed through the equation :

(8)

The neighborhood context information is extracted and a series of computations are performed to obtain the final node vector .

Self-attention layer and prediction: After obtaining the node vectors from the graph neural network, the GCSAN captures the global dependencies through the self-attention layer. The self-attention layer formula is:

(9)

Afterwards, the modeling capability is enhanced by pointwise feed-forward networks and residual connections. After multi-layer self-attention mechanism, the long-range self-attention representation is obtained. At the prediction layer, the session representation is obtained by combining the long-term preferences and current interests of the session

(10)

And accordingly predict the probability of the next click on the item: the

(11)

The models are trained by minimizing an objective function containing cross-entropy and regularization terms. Each of these three models has its own characteristics in graph neural network-based sequence recommendation, SRGNN focuses on capturing intra-session item hopping patterns and dealing with data sparsity; HGN portrays user interests through the gating mechanism and item relationship capture; and GCSAN combines the advantages of graph neural network and self-attention mechanism to comprehensively improve the recommendation performance, which provide diversified They provide diverse solutions for the development of sequence recommendation technology.

## **3.2 Technology**

The technologies used to implement the projects are in table 1:

Table 1: Technologies used for product

|  |  |  |
| --- | --- | --- |
| **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 |

## **3.3 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

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

The saved files include:

1. All the previous documents
2. Project report,
3. Original datasets,

2. processing datasets codes,

3. processed datasets,

4. Training and testing configuration codes,

5. Training and testing codes,

6. Training and testing log.

# **Chapter 4 Implementation and Results**

## **4.1 Implementation**

### **4.1.1 Dataset statistics**

We evaluated three graph neural network-based sequence recommendation models—SRGNN, HGN, and GCSAN—using three representative real-world datasets: MovieLens-1M, DIGINETICA, and Retailrocket. MovieLens-1M is a classic movie rating dataset containing rich user-movie interaction information; Diginetica originated from the 2016 CIKM Cup, whose transaction data effectively reflects user behaviour in an e-commerce context; the Retailrocket dataset focuses on clickstream data from e-commerce platform users.

To ensure fairness in comparison, we performed uniform data preprocessing. In all three datasets, we removed all sessions of length 1 and items appearing fewer than 5 times. After preprocessing, the MovieLens-1M dataset comprises 573,726 user-movie rating records, including 5,951 users and 3,126 movies; the DIGINETICA dataset contains 204,690 sessions and 42,255 items; and the Retailrocket dataset has 381,092 sessions and 86,975 items. At the same time, we split the input sequences to generate the sequences and corresponding labels required for model training. Specifically, for the MovieLens-1M dataset, we use user rating data from a portion of the time period as the test set; for the Diginetica dataset, we use sessions from the following weeks as the test set; and for the Retailrocket dataset, we select sessions from subsequent dates as the test set. For example, for an input session sequence, we generate a series of sequences and labels, where the sequences are processed user behaviour sequences and the labels are the items the user clicks on next, providing an accurate reference standard for model evaluation. The statistics of the datasets are summarized in Table 2:

Table 2:Datasets Introduction

|  |  |  |  |
| --- | --- | --- | --- |
| **Datasets Name** | **Sessions or Users** | **items** | **inters** |
| **Diginetica** | 204,690 | 42,255 | 990,151 |
| **Retailrocket** | 381,092 | 86,975 | 1,531,562 |
| **MovieLens-1M** | 5951 | 3126 | 573,726 |

### **4.1.2 Evaluation metrics**

In session-based recommendation systems, this project uses normalised loss cumulative gain (NDCG) and recall rate to evaluate the performance of three models: session-based recommendation graph neural network (SRGNN), hierarchical gated network (HGN), and graph contextualized self-attention network (GCSAN).

NDCG Calculation Formula:

NDCG is called Normalized Discounted Cumulative Gain, and is often used to evaluate sorting results as a measure of the accuracy of the sort. In recommender systems, a list of items is usually returned for a particular session, and assuming that the length of the list is , then is used to evaluate the gap between this sorted list and the actual list of user interactions in the session.

Gain represents the relevance score of each item in the list. Assuming that the relevance score of the th item in the recommendation list is , usuall y =1 for relevant items and =0 for irrelevant items. Discounted Cumulative Gain() takes into account the sorting order factor, which increases the gain of the items at the front of the list, and decreases the loss of the items at the back of the list, and is computed by the following formula:

(12)

Due to the differences in the length of the actual user interaction lists in different sessions, it is not meaningful to directly compare the of different sessions. Therefore, in order to normalize the metrics across different sessions, it is necessary to compute the score of the real list for each session, denoted by (Ideal Discounted Cumulative Gain), and is the value of when the recommendation list is sorted in perfect order (i.e., exactly the same order as the list of the user's actual interactions). The Normalized Discounted Cumulative Gain ( ) is calculated as

(13)

Finally, the scores for all sessions are averaged to obtain the final value used to evaluate the performance of the recommendation algorithm.

Recall (Recall) Calculation Formula:

The calculation of recall is simpler compared to NDCG. The in NDCG represents the number of the first recommended results returned. Among the returned results, the correct results are called positive samples and the incorrect results are called negative samples. The formula is:

(14)

### **4.1.3 Parameter Settings**

The parameter settings for SRGNN are represented in Table 3.

Table 3: Parameter Settings for SRGNN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datasets Name** | **epochs** | **train\_batch\_size** | **eval\_batch\_size** | **inter** |
| **Diginetica** | 500 | 4096 | 40960 | [session\_id, item\_id, timestamp] |
| **Retailrocket** | 500 | 4096 | 10240 | [timestamp, visitor\_id, item\_id] |
| **MovieLens-1M** | 500 | 4096 | 40960 | [user\_id, item\_id, rating, timestamp] |

The parameter settings for HGN are represented in Table 4:

Table 4: Parameter Settings for HGN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datasets Name** | **epochs** | **train\_batch\_size** | **eval\_batch\_size** | **inter** |
| **Diginetica** | 500 | 4096 | 20480 | [session\_id, item\_id, timestamp] |
| **Retailrocket** | 500 | 4096 | 10240 | [timestamp, visitor\_id, item\_id] |
| **MovieLens-1M** | 500 | 4096 | 40960 | [user\_id, item\_id, rating, timestamp] |

The parameter settings for GCSAN are represented in Table 5:

Table 5: Parameter Settings for GCSAN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datasets Name** | **epochs** | **train\_batch\_size** | **eval\_batch\_size** | **inter** |
| **Diginetica** | 500 | 4096 | 20480 | [session\_id, item\_id, timestamp] |
| **Retailrocket** | 500 | 4096 | 10240 | [timestamp, visitor\_id, item\_id] |
| **MovieLens-1M** | 500 | 4096 | 40960 | [user\_id, item\_id, rating, timestamp] |

In addition, NDCG@10 was set as the evaluation metric for early termination during model training to prevent overfitting and improve the model's generalisation ability.

## **4.2 Results**

The following shows the performance of the SRGNN model on Diginetica:

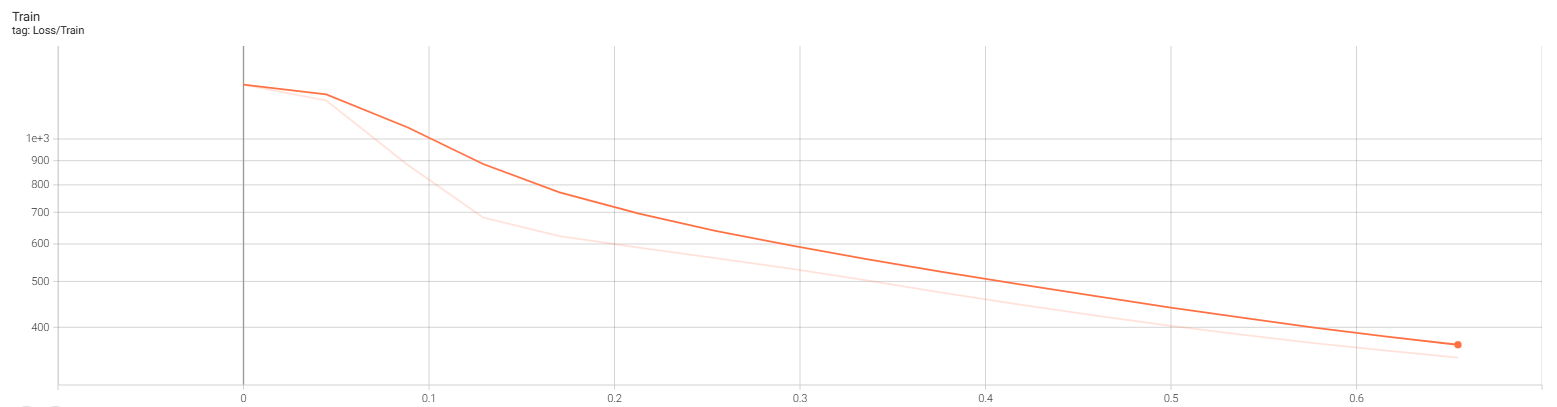


Figure 5:Lose of SRGNN model on Diginetica

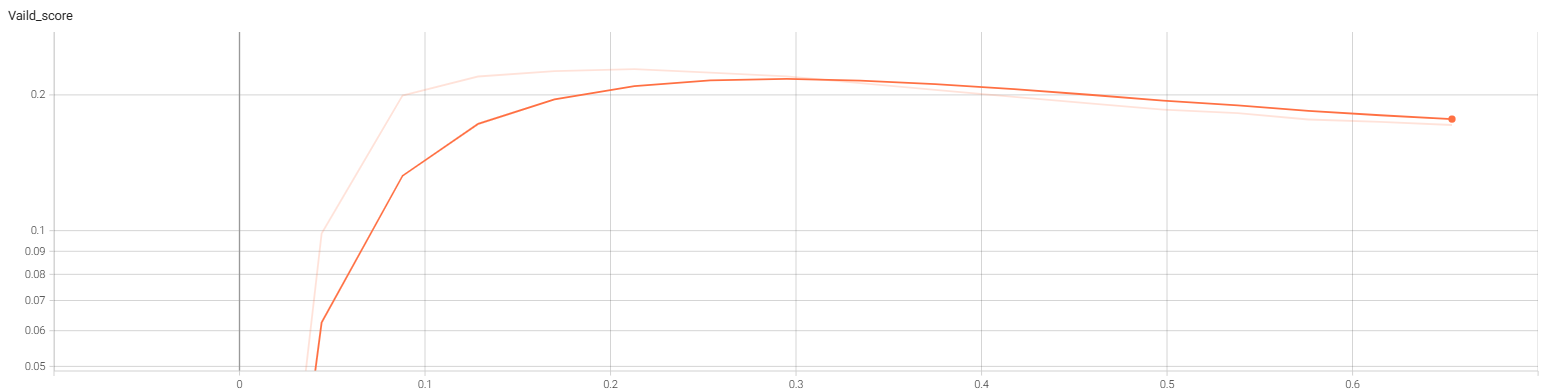


Figure 6:Vaild\_score of SRGNN model on Diginetica

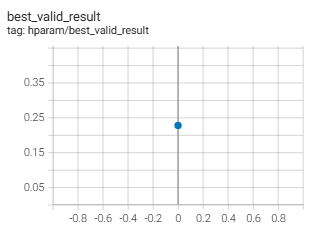


Figure 7:Hparam of SRGNN model on Diginetica

The following shows the performance of the SRGNN model on Retailrocket:

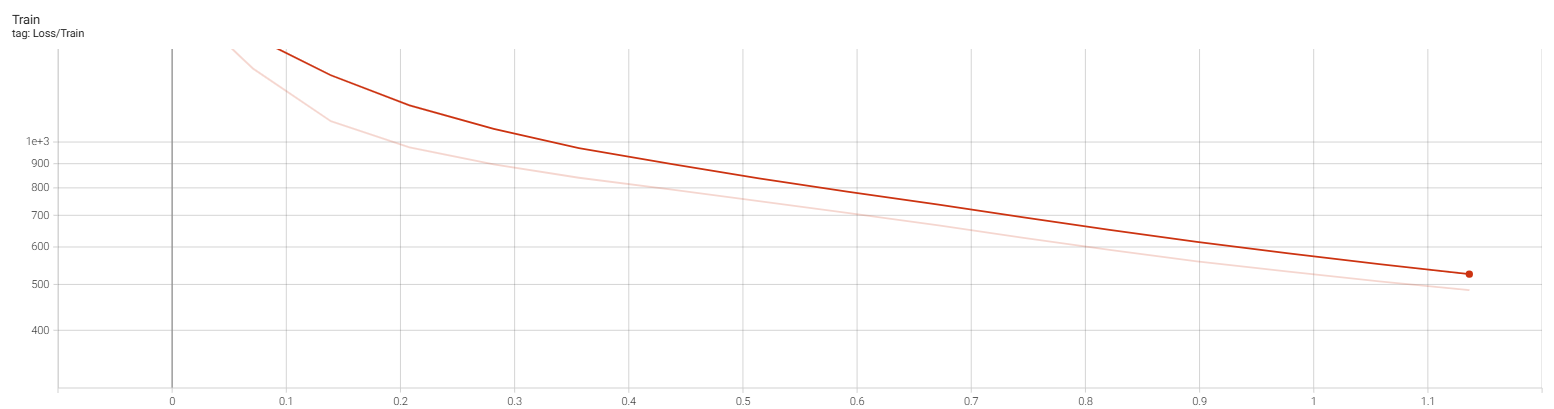


Figure 8:Lose of SRGNN model on Retailrocket

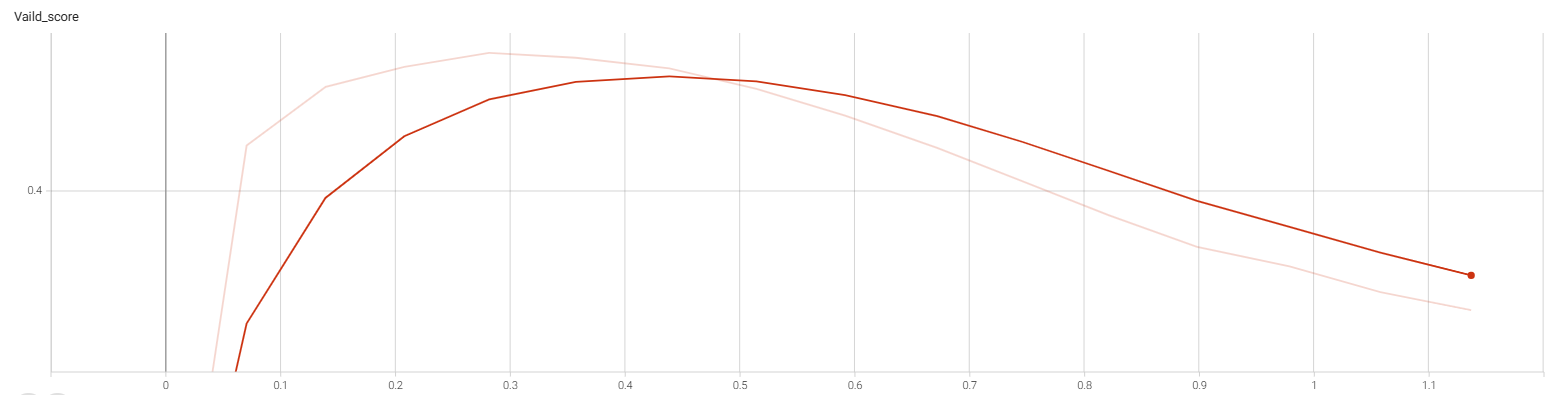


Figure 9:Vaild\_score of SRGNN model on Retailrocket

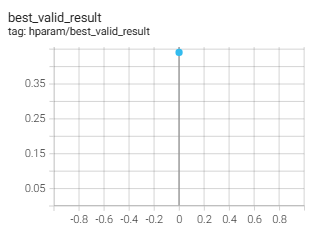


Figure 10:Hparam of SRGNN model on Retailrocket

The following shows the performance of the SRGNN model on MovieLens-1M:

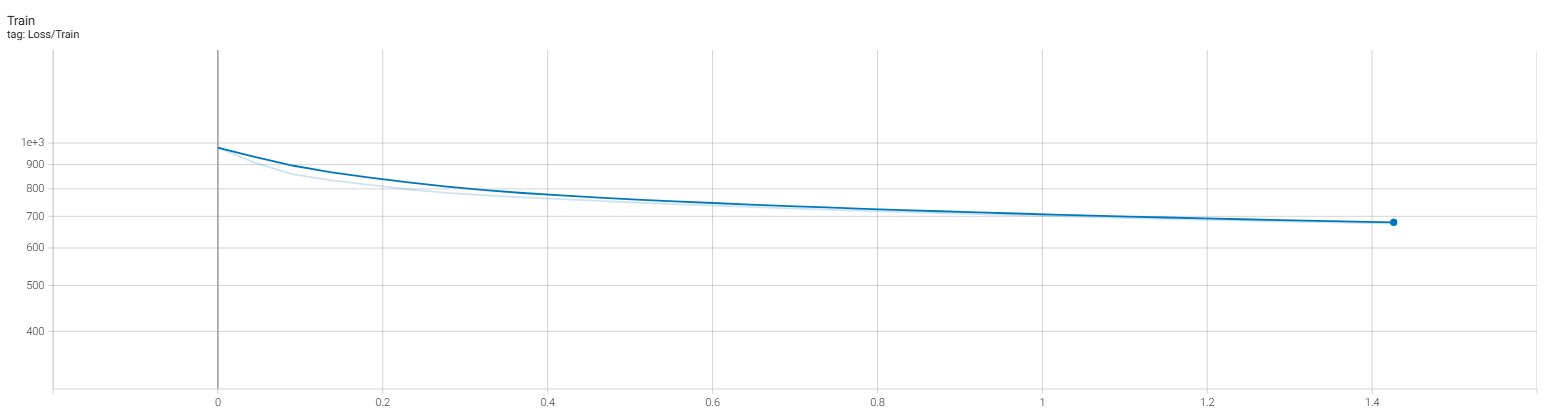


Figure 11:Lose of SRGNN model on MovieLens-1M

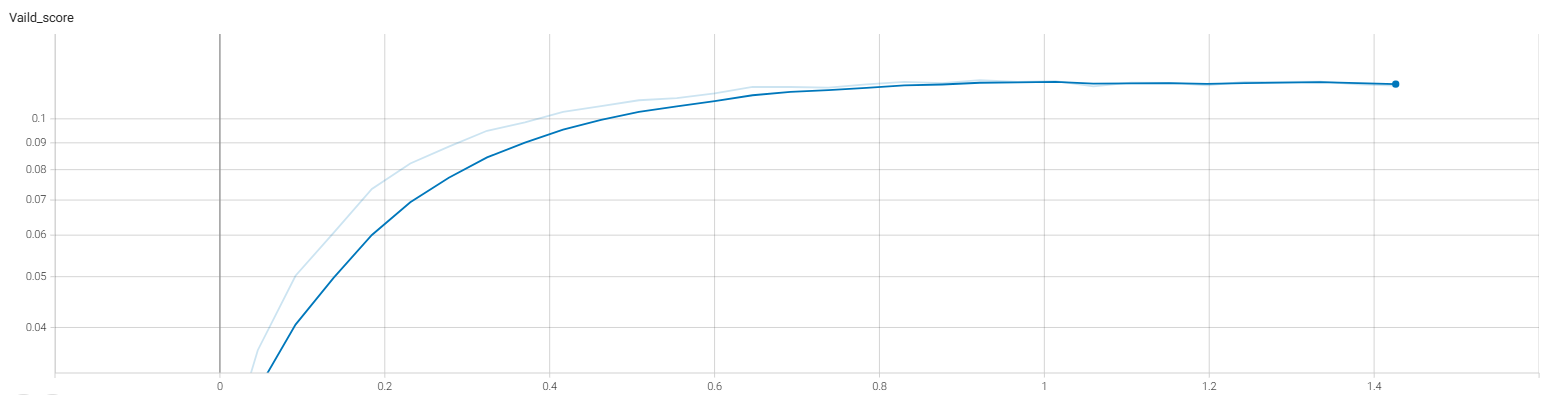


Figure 12:Vaild\_score of SRGNN model on MovieLens-1M

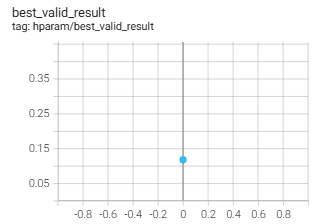


Figure 13:Hparam of SRGNN model on MovieLens-1M

The following shows the performance of the HGN model on Diginetica:

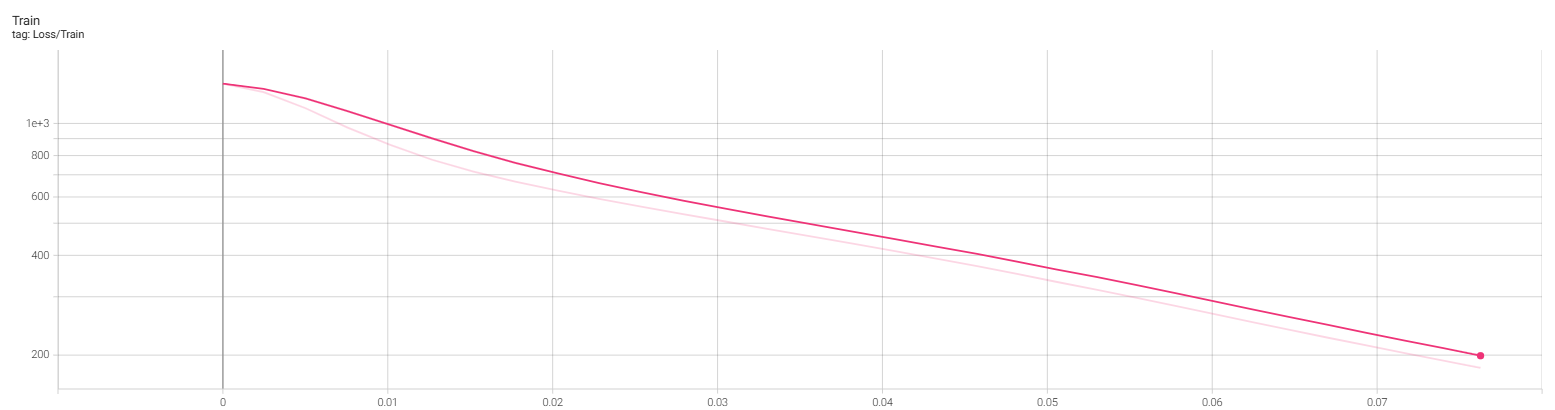


Figure 14:Lose of HGN model on Diginetica

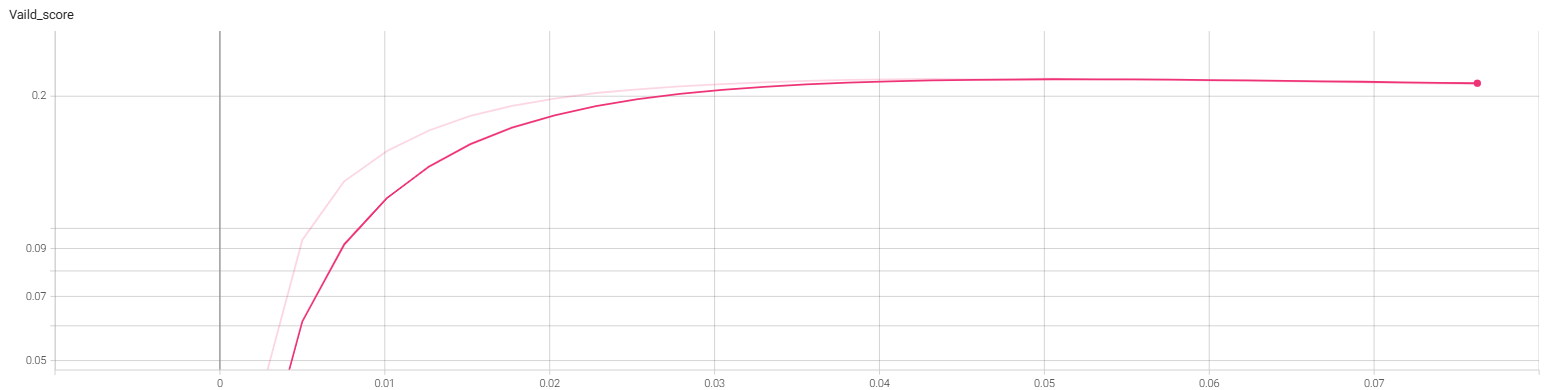


Figure 15:Vaild\_score of HGN model on Diginetica

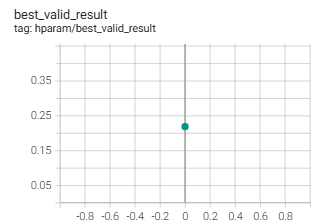


Figure 16:Hparam of HGN model on Diginetica

The following shows the performance of the HGN model on Retailrocket:

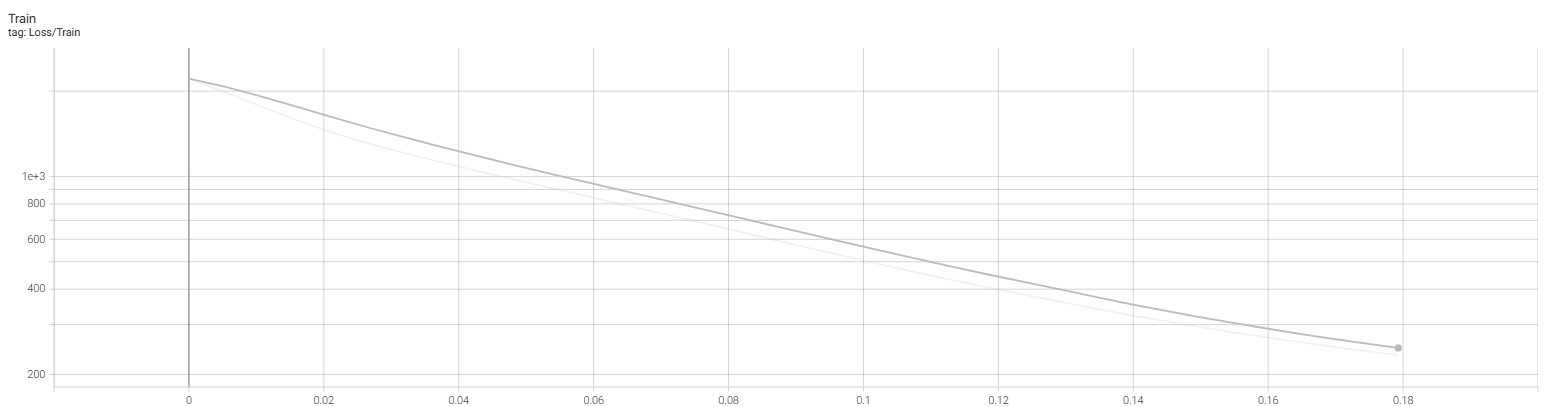


Figure 17:Lose of HGN model on Retailrocket

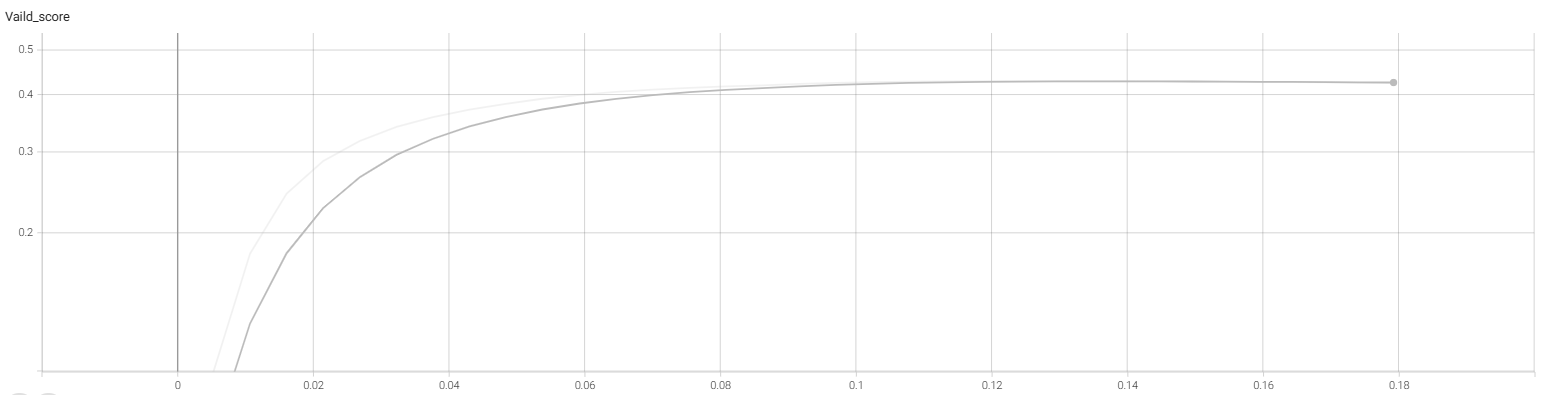


Figure 18:Vaild\_score of HGN model on Retailrocket

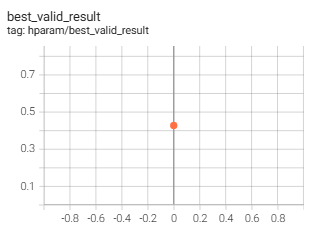


Figure 19:Hparam of HGN model on Retailrocket

The following shows the performance of the HGN model on MovieLens-1M:

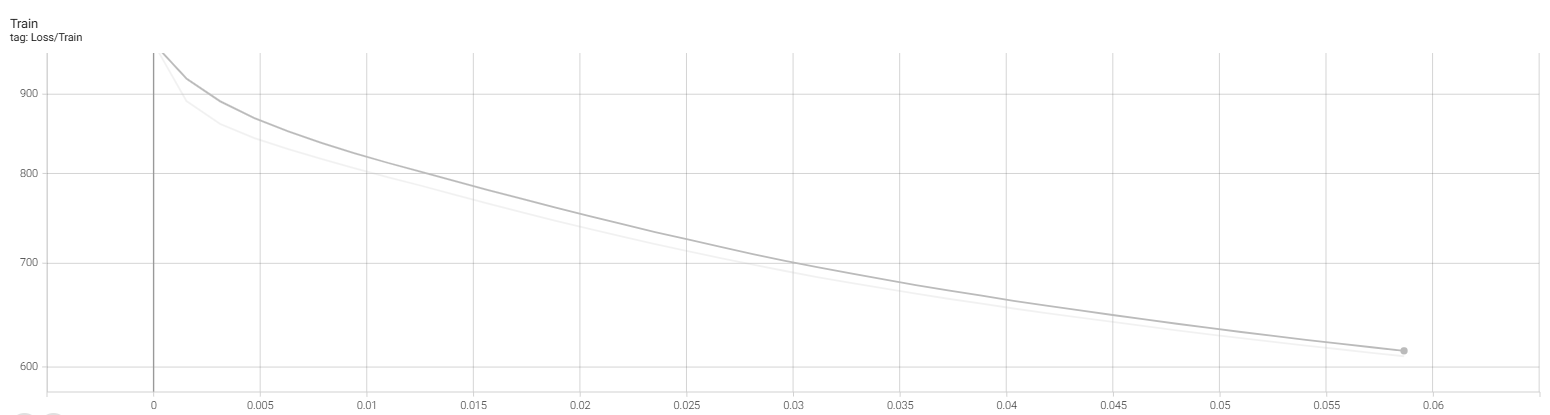


Figure 20:Lose of HGN model on MovieLens-1M

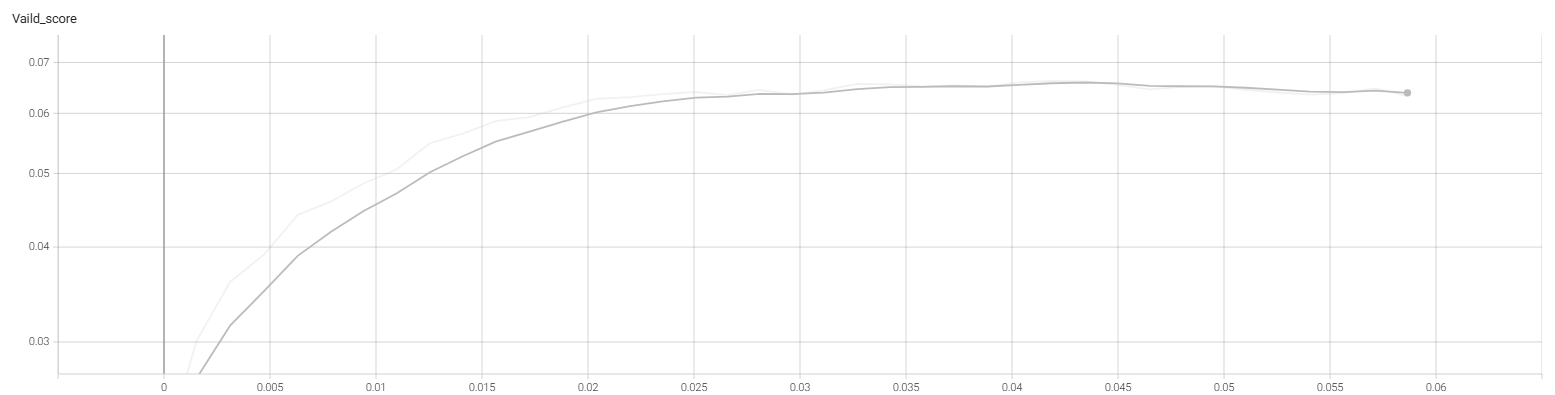


Figure 21:Vaild\_score of HGN model on MovieLens-1M

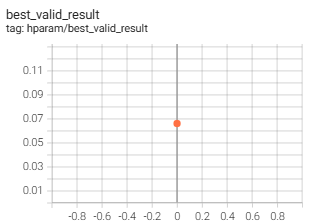


Figure 22:Hparam of HGN model on MovieLens-1M

The following shows the performance of the GCSAN model on Diginetica:

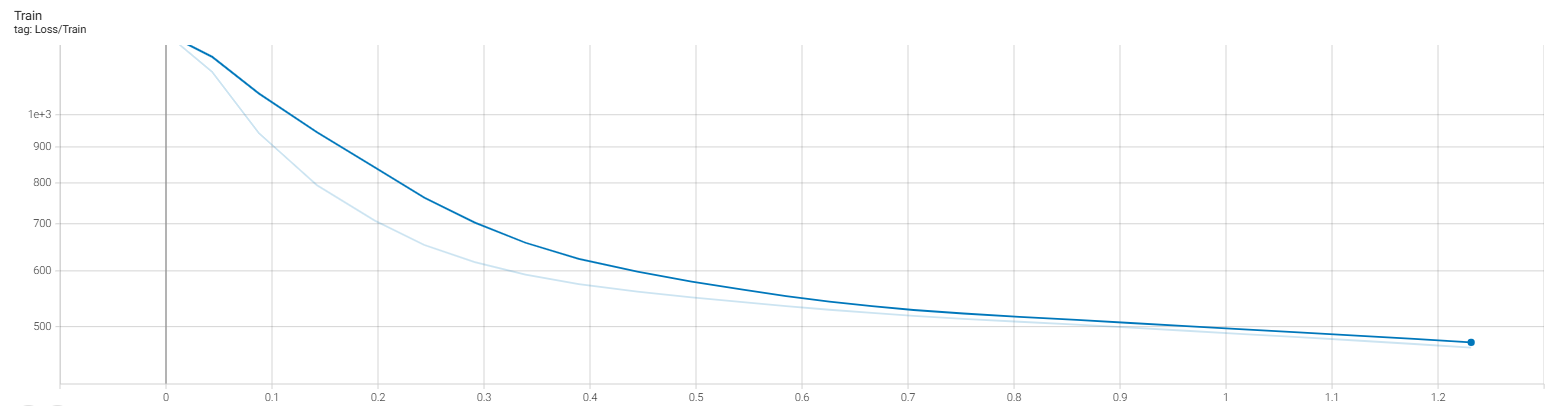


Figure 23:Lose of GCSAN model on Diginetica

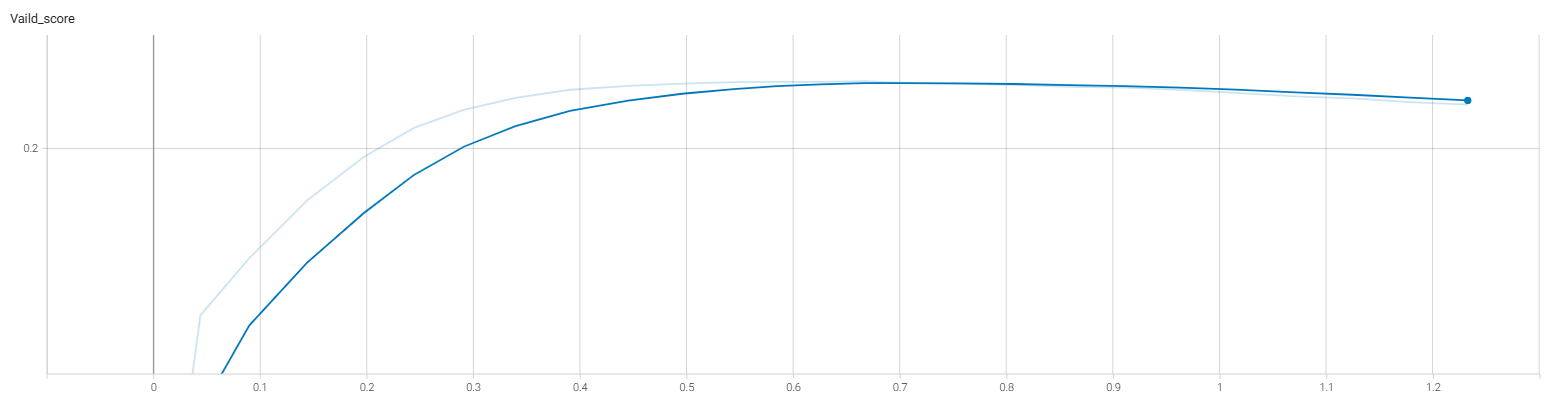


Figure 24:Vaild\_score of GCSAN model on Diginetica

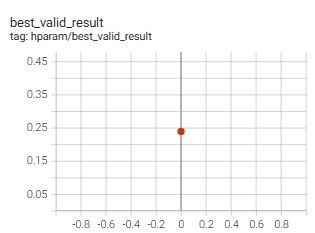


Figure 25:Hparam of GCSAN model on Diginetica

The following shows the performance of the GCSAN model on Retailrocket:

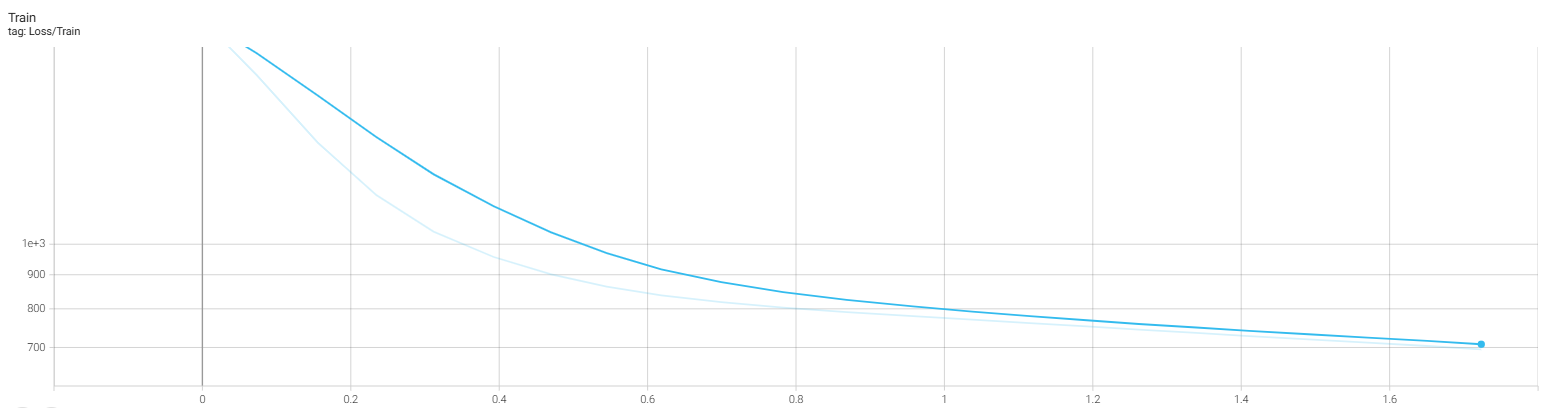


Figure 26:Lose of GCSAN model on Retailrocket

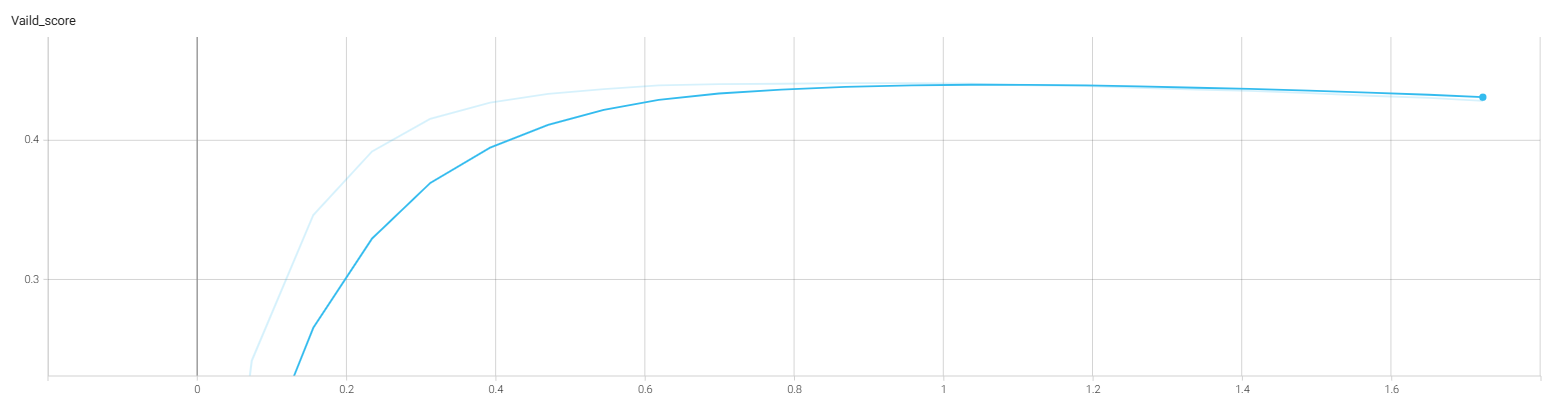


Figure 27:Vaild\_score of GCSAN model on Retailrocket

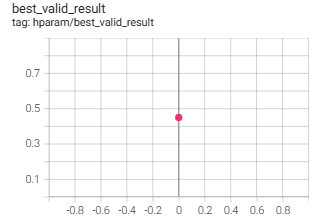


Figure 28:Hparam of GCSAN model on Retailrocket

The following shows the performance of the GCSAN model on MovieLens-1M:

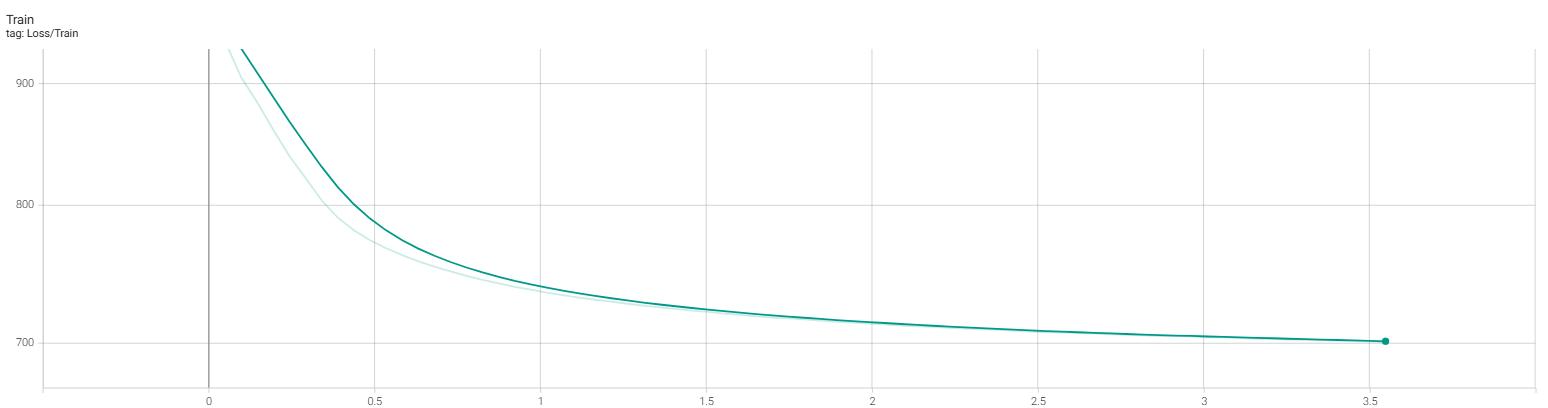


Figure 29:Lose of GCSAN model on MovieLens-1M

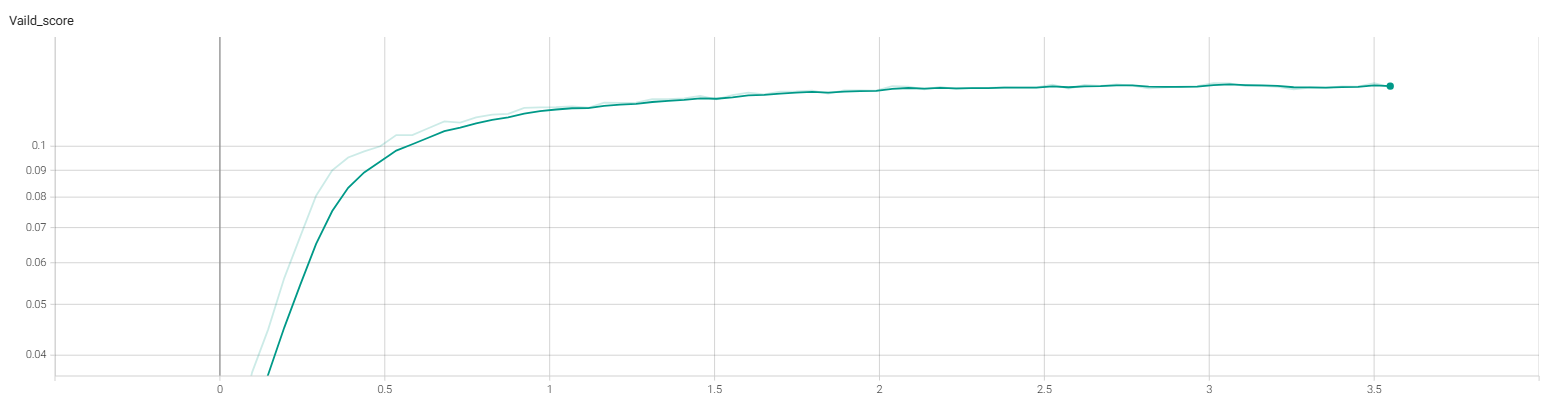


Figure 30:Vaild\_score of GCSAN model on MovieLens-1M

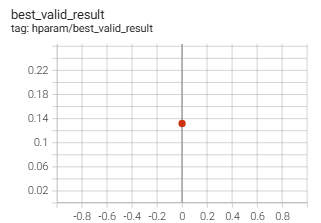


Figure 31:Hparam of GCSAN model on MovieLens-1M

Here is the training performance table 6 of SRGNN on those prepared datasets:

Table 6: Training performance of SRGNN on datasets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datasets Name** | **Recall @10** | **MRR@10** | **NDCG@10** | **Hit@10** |
| Diginetica | 0.4038 | 0.1886 | 0.2392 | 0.4038 |
| Retailrocket | 0.5955 | 0.4074 | 0.453 | 0.5955 |
| MovieLens-1M | 0.2072 | 0.0888 | 0.1163 | 0.2072 |

Here is the training performance table 7 of HGN on those prepared datasets:

Table 7: Training performance of HGN on datasets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datasets Name** | **Recall @10** | **MRR@10** | **NDCG@10** | **Hit@10** |
| Diginetica | 0.3668 | 0.1707 | 0.2168 | 0.3668 |
| Retailrocket | 0.5404 | 0.378 | 0.417 | 0.5404 |
| MovieLens-1M | 0.1071 | 0.0427 | 0.0576 | 0.1071 |

Here is the training performance table 8 of GCSAN on those prepared datasets:

Table 8: Training performance of GCSAN on datasets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datasets Name** | **Recall @10** | **MRR@10** | **NDCG@10** | **Hit@10** |
| Diginetica | 0.4217 | 0.1994 | 0.2518 | 0.4217 |
| Retailrocket | 0.6062 | 0.4228 | 0.4672 | 0.6062 |
| MovieLens-1M | 0.2306 | 0.0989 | 0.1297 | 0.2306 |

Performance exhibits considerable variability across distinct datasets (Diginetica, Retailrocket, MovieLens - 1M). For instance, the performance metrics of all models on the Retailrocket dataset are generally superior to those on the MovieLens - 1M dataset. The user-item interaction patterns in the Retailrocket dataset may be more consistent or varied, facilitating model learning; conversely, the MovieLens - 1M dataset likely exhibits significant data sparsity and intricate user behaviour patterns, resulting in constraints on model performance.

Diverse models demonstrate disparate performance on identical datasets. Taking the Diginetica dataset as an example, the Recall@10, MRR@10, NDCG@10, and Hit@10 metrics of GCSAN are 0.4217, 0.1994, 0.2518, and 0.4217, respectively, which outperforms HGN (corresponding metrics: 0.3668, 0.1707, 0.2168, and 0.3668), indicating that the GCSAN model structure has advantages in capturing the data features of this dataset and performing recommendations. Nonetheless, in specific datasets, SRGNN surpasses GCSAN on certain measures, indicating that various models are appropriate for distinct application contexts and data attributes.

Recall@10 and Hit@10 produce similar values across all models and datasets, as both metrics fundamentally assess the ratio of relevant items recovered. MRR@10 and NDCG@10 are not perfectly positively associated with Recall@10 and Hit@10. For instance, in the MovieLens - 1M dataset, SRGNN's Recall@10 (0.2072) surpasses HGN's (0.1071); however, the enhancement in MRR@10 (0.0888 versus 0.0427) is comparatively modest, indicating that while augmenting the quantity of retrieved items, the models may exert disparate influences on the precision of item ranking (assessed by MRR@10 and NDCG@10).

GCSAN exhibits outstanding performance across various datasets, leading in several key metrics on the Diginetica and Retailrocket datasets; SRGNN also shows commendable performance on the Retailrocket dataset; HGN performs marginally weaker than the other two models across all datasets but possesses distinct advantages in specific metrics on different datasets.

In practical applications, datasets resembling Retailrocket should prefer the GCSAN and SRGNN models. For more complex datasets such as MovieLens - 1M, additional model optimisation or investigation of other appropriate model architectures may be required to enhance recommendation efficacy.

# **Chapter 5 Professional Issues**

## **5.1 Project Management**

### **5.1.1 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 9 : Activities table

### **5.1.2 Schedule**

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

March 28th 2025.

The Gantt graph was shown below:

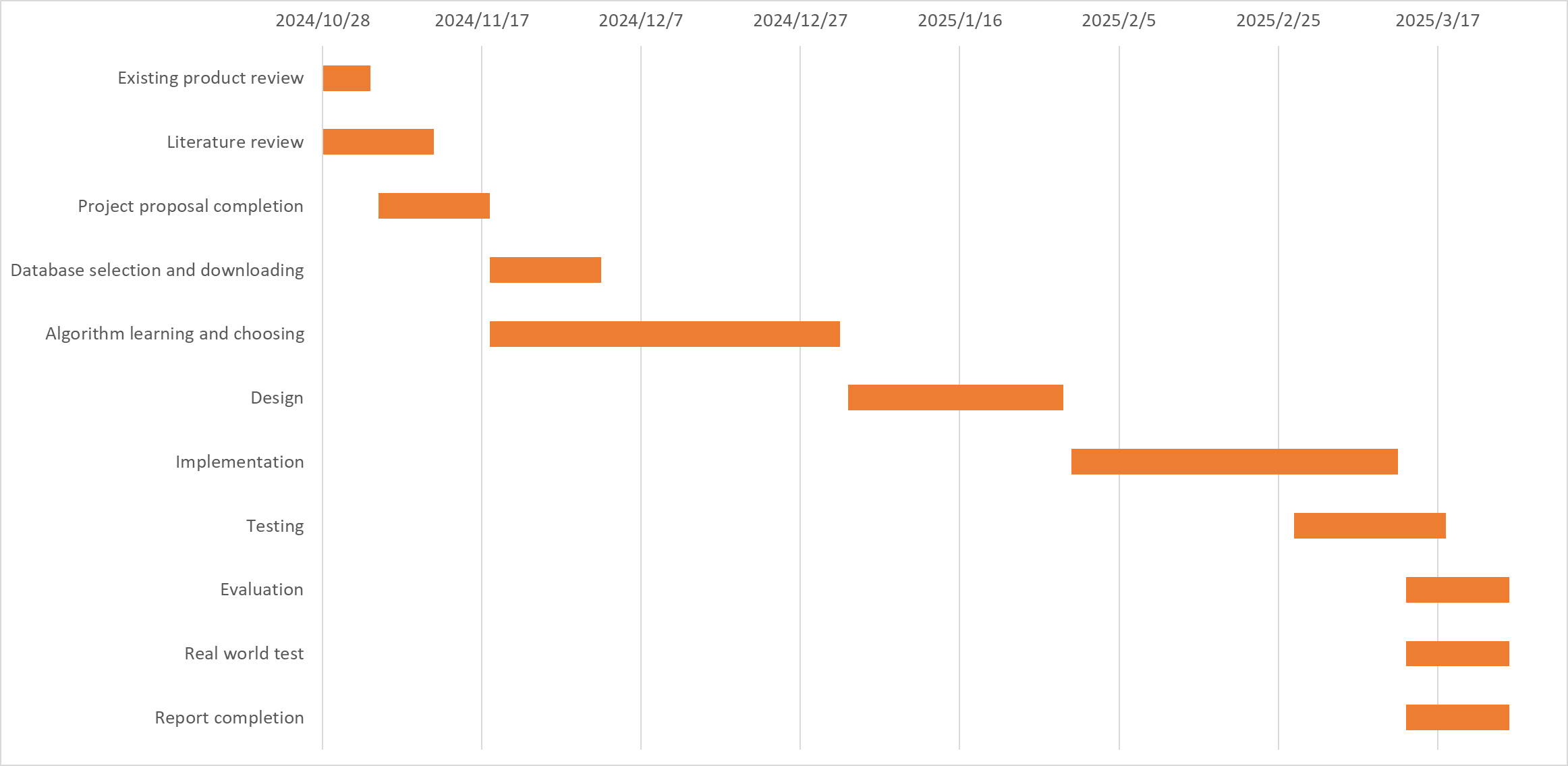


Figure 32: Gantt chart of the schedule

### **5.1.3 Project Data Management**

Project materials are classified and stored by establishing hierarchical folders based on project names. Functions such as batch upload, sharing links, and multi-person shared folders are utilized to achieve convenient storage and team collaboration. When using GitHub, project repositories are created, and code version control and team development are carried out through branch management and Pull Request. The search and Fork functions are used to participate in open source projects for learning and communication. The two complement each other and can improve project management efficiency.

### **5.1.4 Project Deliverables**

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

## **5.2 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 10: Risk Analysis

A detailed risk analysis table has been created in the project to detect, assess, and manage various risks during execution. The framework includes columns such as Risk ID, Potential Risk Causes, Severity, Likelihood, Risk Priority, and Risk Mitigation, allowing the project team to thoroughly analyse and address risks.

Within the "Missed deadline (R.1.1)" category, three potential causes exist. For "Illness (C1.1.1)", with a severity rating of 1, likelihood of 3, and priority of 3, documenting exceptional circumstances serves as the mitigation strategy. For "Poor time management (C1.1.2)" with a severity of 4, likelihood of 3, and a priority of 12, adherence to the Gantt chart is underscored. The issue of "Insufficient training data (C1.1.3)" has a severity of 3, a likelihood of 3, and a priority of 9. Although no particular mitigation measures are yet identified, it requires attention since it may impede 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.

The risk analysis table provides an effective and pragmatic risk management framework for the project team by outlining potential causes, precisely evaluating severities and probabilities, and suggesting specific mitigations, thereby allowing the project to anticipate risks and implement proactive measures for seamless advancement.

## **5.3 Professional Issues**

### **5.3.1 Ethical Issues**

The process of sequential recommendation, which is to say the method by which recommendations are made in a sequence, must be presented to users in a manner that is clear, or at least somewhat understandable, by utilizing certain approaches that are deemed appropriate. Furthermore, it is of utmost importance, and indeed necessary, to adhere to the codes of conduct established by organizations such as the ACM and BCS. These codes instruct us to steer clear of any unethical practices throughout the entirety of the project. This is to ensure that all activities related to research and development are conducted with a sense of integrity, a quality that is, one might say, essential. Additionally, any claims that are made regarding the performance or capabilities of the sequential recommendation system should be based upon data that is accurate, or at least verifiable, in some sense. Now, one must consider the implications of using graph neural networks, especially when these networks are trained on historical data that may be, how should one put it, biased. Such a scenario could very well lead to outcomes that are not equitable, and hence, it becomes crucial, or at least very important, to implement mechanisms that are aimed at detecting and reducing bias. This is not merely a suggestion but a necessity, one might argue, to ensure fairness and equity in the outcomes produced by these systems. Thus, the need for careful consideration in this regard cannot be overstated, as it is vital for the integrity of the entire sequential recommendation process.

Intellectual Property Rights, which are often referred to as IPR, hold a significant importance when we talk about the development and progress of graph-based neural network models that are used for making recommendations in a sequential manner. This importance is largely due to the fact that these models may involve the creation of new and original components, which makes it very necessary, or perhaps imperative, to think about protecting intellectual property. Now, when it comes to adhering to copyright regulations, it is essential to obtain the appropriate licenses that are suitable for the use of existing algorithms, as well as code snippets and also open-source libraries. It is also important to give proper attribution when using these things, as failing to do so could lead to issues down the road. At the same time, one cannot forget the necessity of ensuring compliance with data protection regulations, which is equally crucial, if not more so. Thus, we see that there are many layers to consider, and one must navigate through these complexities with care and diligence to ensure that all aspects are adequately addressed.This project, you see, is going to use some user data for the purposes of modeling, which is quite important. Therefore, it is necessary to acquire, or you might say obtain, consent from the users for the collection of their data, which is, after all, a very crucial step. Furthermore, one must ensure that the storage of this data is secure, so that it is kept safe and sound, away from prying eyes and potential mishaps. Moreover, there is the matter of anonymizing or perhaps pseudonymizing the data, which is another layer of protection, I suppose. Consent, you see, must be gathered before one can even think about collecting any interaction data from the users, which is a fundamental requirement. It is just the way things should be done in a proper manner, ensuring that all ethical considerations are taken into account, as they should be, because we must not overlook such important aspects. In conclusion, the process is quite involved, requiring multiple steps and considerations, but it is all in the name of using data responsibly, which is something we should all strive for? In order to ensure that the data is safe and not accessible to those who should not be able to access it, it is essential, very much so, that encryption is employed. This is important, particularly because, you see, there exists a risk, a potential risk, of computers being misused in various ways. Therefore, the recommendation system that the project has must have protection measures that are quite strict, you know, to prevent unauthorized individuals from accessing it. This is in line with the Computer Misuse Act that was established back in 1990, which is a law that, in essence, aims to protect computer systems from misuse. So, the measures that are to be put in place should be strong and effective, in order to stop any hackers, who might want to interfere with the system, from doing so. These hackers, if they get in, could undermine how the system works, or they could change the data that is there, which would be quite problematic, or even use it for purposes that are not ethical at all. It is crucial, you see, to have these security measures in place, and they must be stringent and robust, so that any unauthorized access is thwarted effectively, and thus, the integrity of the system is maintained.

### **5.3.2 Social issue**

Social responsibility, it must be said, is a requirement, a necessary demand, that the sequential recommendation system, which is to be utilized in this particular project, be designed in such a way that it promotes, or at least aims to promote, positive outcomes for society. This includes, but is not limited to, influencing user behavior, particularly in areas like e-commerce, where such influences could lead to impacts on what products are purchased and, in turn, how businesses succeed or fail. Furthermore, we should also consider that the effects on communities can be significant, since there exists the risk of developing what some might call echo chambers, which is a situation where people only hear what they already believe, and this could lead to various issues. Therefore, it becomes quite essential, if not crucial, to employ certain strategies that would promote, in a way, serendipity and diversity in the recommendations made, thus leading to improvements in user experiences overall and, in a broader sense, benefiting society as a whole.

### **5.3.3 Legal impact**

Intellectual Property Rights, often referred to as IPR, play an important role in the development and progress of models that are based on graphs, particularly when it comes to making recommendations in a sequential manner. This is because such models may involve the creation of new and original components, which makes it very important to think about how to protect intellectual property. It is, therefore, necessary to ensure compliance with copyright laws, which means that one must acquire the appropriate licenses and also give proper attribution whenever existing algorithms, code snippets, or libraries that are open-source are used. At the same time, it is crucial to keep in mind that one must also comply with the requirements related to data protection. This dual compliance is essential, as neglecting either aspect could lead to problems, which is something that should be avoided. The importance of adhering to these legal frameworks cannot be overstated, as they serve to uphold the integrity of the innovations and contributions made in this field.This project, which is going to use data from users, for the purposes of modeling, is something that requires careful consideration and attention to various aspects. It is necessary, absolutely necessary, to obtain consent from users before we even think about collecting any interaction data, which is, as you might understand, a crucial step in this whole process. Consent must be acquired beforehand, before any data is collected, because without it, well, we really should not proceed. Then, there is the matter of ensuring that the data we collect is stored in a secure manner. This means we have to think about security—yes, security is key here, and we must ensure that the data is safe, kept away from prying eyes or any unauthorized access. We need to be careful with this, very careful. Furthermore, it is also important, and I cannot stress this enough, to anonymize or pseudonymize the data that we gather. This means that the data should be altered in such a way that it cannot easily be traced back to any specific individual. We want to protect the identity of those individuals who are providing their data, which is essential for maintaining trust and confidentiality. This step is not just a formality; it is a fundamental aspect of handling user data responsibly. So, in summary, the collection of interaction data must be preceded by obtaining consent, and we must ensure that this data is stored securely and anonymized or pseudonymized appropriately. This is the process we must follow, and it is of utmost importance that we adhere to these guidelines closely, as they form the backbone of our approach to handling user data in this project.To ensure that the data, which is quite important, remains safe from those who should not have access to it, it is necessary to implement some form of encryption. This is crucial, you see, because there is always the potential for computer misuse, which is a serious concern. Therefore, the recommendation system of the project, which is meant to help users, must be safeguarded against unauthorized access. This safeguarding should follow the guidelines set forth by the Computer Misuse Act of 1990. In order to achieve this, it is essential to adopt strict security measures that are very effective in preventing hackers, who may wish to disrupt its operation, from compromising its functionality. These measures are not just for show; they are meant to stop anyone from altering the data that the system uses. It is equally important to prevent any exploitation of the system for unethical purposes, which could lead to misuse that is not acceptable. Thus, the project must take these precautions seriously, because in this age of technology, the risks are many and varied, and it is better to be safe than sorry. So, all in all, the security of the recommendation system is of utmost importance, and we must be vigilant in our efforts to protect it from potential threats, which are, unfortunately, quite common in today's world.

### **5.3.4 Environment Issues**

The progress and improvement of various systems, you see, can lead to a situation where hardware becomes outdated, and this, in turn, might create a lot of electronic waste, which is, of course, a concern. However, it is very important to think about the environmental effects that come with the upgrading of hardware and the disposal of old components, which includes the recycling of those outdated parts or maybe the adoption of technologies that are more sustainable. This is something that we must keep in mind, especially when considering the guidelines laid out in the BCS Code of Conduct, which, as you may know, emphasizes the responsibility we have towards our environment. It is all interconnected, really. So, while we improve and advance, we must also pay attention to what happens to the old equipment and how we manage it, as this is crucial in today’s world, and we should not overlook these aspects that are intertwined with our technological progress.It is very, very important, indeed, to take a good look at the environmental factors that are relevant to this particular project we are discussing. We must, without a doubt, put into action some proactive measures that will help in reducing the environmental impact that might arise. It is also necessary to keep in mind, at all times, the best practices in the industry as well as the emerging standards that are being suggested by organizations like IEEE concerning what is known as sustainable computing. This is to ensure that we are indeed conforming to the environmental objectives that we have set for ourselves. Now, let us talk about energy consumption, which is quite a big deal. The training and deployment of those graph neural network models, particularly for the purpose of making sequential recommendations, tends to require a lot of resources and, you guessed it, energy. Therefore, it becomes absolutely essential to think about and explore various methods that could enhance the model architecture. Not to forget, we also need to look into the training algorithms that we are using, all in an effort to reduce this energy consumption issue that we are facing. It is a serious matter that cannot be overlooked.

# **Chapter 6 Conclusion**

## **6.1 Findings & Reflections**

This project performed a comprehensive investigation and experimental evaluation of three sequence recommendation models utilising graph neural networks: HGN, SRGNN, and GCSAN, yielding a range of significant outcomes. Experimental findings across several datasets indicate that each model has unique performance traits. GCSAN excels in various critical indicators, attaining premier positions on the Diginetica and Retailrocket datasets. It excels in measures including Recall@10, MRR@10, and NDCG@10. This results from its capacity to integrate the benefits of graph neural networks and self-attention mechanisms, adeptly capturing both local and global interdependence to significantly improve recommendation performance. SRGNN has strong performance on the Retailrocket dataset, showcasing distinct advantages in addressing data sparsity and identifying intra-session item transition patterns. It can deduce possible interests from the local patterns of user behaviour, offering robust support for recommendations. While HGN is marginally less effective than the previous two in overall performance, it possesses distinct benefits in particular measures and diverse datasets. It efficiently filters information utilising feature gating and instance gating modules, and integrates an item-item product module to delineate item relationships, thus enhancing the capture of users' long-term and short-term interests.

These results illustrate the varying responsiveness of distinct models to diverse data properties. The dataset's properties, including the consistency of user-item interaction patterns, data sparsity, and the intricacy of user behaviour, significantly influence model performance. The interaction patterns in the Retailrocket dataset may be more conducive to GCSAN and SRGNN, allowing them to capitalise on their strengths, whereas the MovieLens-1M dataset, characterised by significant data sparsity and intricate user behaviour patterns, presents constraints on model performance. This indicates that, in real applications, it is crucial to choose suitable models according to unique data attributes and business needs to get optimal recommendation efficacy.

## **6.2 Limitations**

This work has yielded specific results in the investigation of sequence recommendation models; nonetheless, shortcomings persist. Initially, while managing extensive datasets, the computational complexity and training duration of the model present significant challenges that must be resolved. As data volume escalates, the training and inference processes of the HGN, SRGNN, and GCSAN models may become exceedingly time-intensive, adversely impacting the real-time efficacy of the recommendation system and augmenting the consumption of computational resources. For instance, when handling datasets with a substantial quantity of users and objects, model training may require several hours or even days, complicating the fulfilment of real-time recommendation demands in actual applications.

Secondly, models have not yet comprehensively grasped intricate user behaviour. Although these models can partially identify patterns and connections in user behaviour, actual user conduct is frequently affected by several factors, including emotions and contextual changes. Existing models have not sufficiently addressed these intricate issues. This may lead to recommendation outcomes that inadequately align with users' genuine demands in specific circumstances, thereby diminishing suggestion precision and user contentment.

Moreover, there are deficiencies in the models' interpretability. While technologies like graph neural networks and self-attention processes have enhanced model performance, they have simultaneously increased structural complexity, complicating intuitive explanations of the model's recommendation conclusions. This may restrict the model's applicability in contexts demanding high interpretability of recommendation outcomes, such as medical and financial suggestions.

## **6.3 Future Work**

Considering the aforementioned restrictions, subsequent research may be pursued in the following avenues. Efforts must be undertaken to optimise the model architecture and methods to diminish computational complexity and enhance training efficiency and real-time performance. For instance, more efficient graph neural network topologies may be investigated, including lightweight graph convolution operations or enhanced self-attention processes, to minimise model parameters and computational complexity. Simultaneously, distributed computing technologies can be used to allocate model training duties over numerous computing nodes, expediting the training process and enhancing its capacity to address large-scale data issues.

Conversely, we can explore methods to enhance our comprehension and modelling of intricate user activity. We may include more contextual information, such users' geographical position, temporal data, and social ties, into the model to more thoroughly capture users' interests and intentions. Furthermore, we can employ technologies like reinforcement learning to allow the model to dynamically modify its suggestion approach based on real-time user feedback, thus enhancing the precision and adaptability of recommendations.

Examine the advancement of visualisation tools and techniques for elucidating the model's decision-making process in an intuitive and comprehensible manner about model interpretability. For instance, illustrate the significance of nodes and edges in graph neural networks or exhibit the weight distribution of self-attention mechanisms across various places to aid users and developers in comprehending the model's suggestion process. This enhances user trust in the recommendation system and aids developers in model debugging and optimisation.

In the end, it is important to consider that the application contexts of the approach can be broadened, which means that there are many possibilities. Nowadays, a lot of research, perhaps too much, seems to focus mainly on e-commerce and video recommendation systems, which is quite common. However, looking forward into the future, it might be worthwhile to think about examining these models in a variety of other fields. For instance, one could think about news recommendation, or perhaps music recommendation, or even intelligent education, which is a field that has been gaining attention. It would be interesting to see how effective these models could be in these different scenarios. Furthermore, based on the specific characteristics and unique attributes of each field, targeted optimizations and enhancements could be implemented. This way, there might be a chance to truly assess how well they function in these various contexts, thus allowing for a broader understanding of their efficacy, which is something that could be very beneficial.

# **References**

[1] C. Ma, P. Kang, and X. Liu, “Hierarchical Gating Networks for Sequential Recommendation,” Jun. 21, 2019, arXiv: 1906.09217. doi: 10.48550/arXiv.1906.09217.

[2] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, “Session-Based Recommendation with Graph Neural Networks,” AAAI, vol. 33, no. 01, pp. 346–353, Jul. 2019, doi: 10.1609/aaai.v33i01.3301346.

[3] C. Xu et al., “Graph Contextualized Self-Attention Network for Session-based Recommendation,” in Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China: International Joint Conferences on Artificial Intelligence Organization, Aug. 2019, pp. 3940–3946. doi: 10.24963/ijcai.2019/547.

[4] Rendle, S., Freudenthaler, C., & Schmidt - Thieme, L., “Factorizing personalized markov chains for next - basket recommendation,” in Proceedings of the 19th international conference on World wide web, Apr. 2010, pp. 811 - 820. ACM.

[5] Shani, G., Brafman, R. I., & Heckerman, D., “An MDP - based recommender system,” Journal of Machine Learning Research, vol. 3, pp. 1331 - 1366, 2002.

[6] Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D., “Session - based recommendations with recurrent neural networks,” in International Conference on Learning Representations, 2016.

[7] Tan, Y. K., Xu, X., & Liu, Y., “Improved recurrent neural networks for session - based recommendations,” in Proceedings of the 10th ACM conference on Recommender systems, Sep. 2016, pp. 17 - 22. ACM.

[8] He, X., Zhang, H., Kan, M. Y., & Chua, T. S., “Fast matrix factorization for online recommendation with implicit feedback,” in Proceedings of the 39th international ACM SIGIR conference on Research and development in information retrieval, Jul. 2016, pp. 549 - 558. ACM.

[9] He, X., He, Z., Song, J., Liu, Z., Jiang, Y., & Chua, T. S., “NAIS: Neural attentive item similarity model for recommendation,” ACM Transactions on Information Systems (TOIS), vol. 30, no. 12, pp. 1 - 23, 2018.

[10] Kang, W. C., & McAuley, J., “Self - attentive sequential recommendation,” in Proceedings of the 18th IEEE International Conference on Data Mining (ICDM), Nov. 2018, pp. 197 - 206. IEEE.

[11] Li, J., Ren, P., Chen, Z. M., Ren, Z. C., Lian, T., & Ma, J., “Neural attentive session - based recommendation,” in Proceedings of the 26th ACM international conference on Conference on information and knowledge management, Nov. 2017, pp. 1419 - 1428. ACM.

[12] Wang, Z., Zhang, Y., & Zhao, M., “Global context enhanced graph neural networks for session - based recommendation,” arXiv preprint arXiv:2006.03593, 2020.

[13] Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt - Thieme, L., “BPR: Bayesian personalized ranking from implicit feedback,” in Proceedings of the 25th conference on Uncertainty in artificial intelligence, Jun. 2009, pp. 452 - 461. AUAI Press.

[14] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J., “Item - based collaborative filtering recommendation algorithms,” in Proceedings of the 10th international conference on World Wide Web, May 2001, pp. 285 - 295. ACM.

[15] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G., “The graph neural network model,” IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61 - 80, 2009.

[16] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I., “Attention is all you need,” in Advances in neural information processing systems, 2017, pp. 5998 - 6008.

[17] Wang, X., He, X., Wang, M., Feng, F., & Chua, T. S., “Neural graph collaborative filtering,” in Proceedings of the 42nd International ACM SIGIR conference on Research and development in information retrieval, Jul. 2019, pp. 165 - 174. ACM.

[18] Lin, Z., Feng, M., Santos, C. N. D., Yu, M., Xiang, B., Zhou, B., & Bengio, Y., “A structured self - attentive sentence embedding,” in International Conference on Learning Representations, 2017.

[19] Liu, Q., Zeng, Y., Kusi, R. M., & Zhang, H., “STAMP: Short - term attention / memory priority model for session - based recommendation,” in Proceedings of the 24th ACM SIGKDD international conference on Knowledge discovery and data mining, Aug. 2018, pp. 1831 - 1839. ACM.

[20] Liu, P., Chang, S. C., Huang, X. J., Tang, J., & Cheung, J. C. K., “Contextualized non - local neural networks for sequence learning,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 6290 - 6297, Jul. 2019.

[21] Marino, K., Salakhutdinov, R., & Gupta, A., “The more you know: Using knowledge graphs for image classification,” in Proceedings of the IEEE conference on computer vision and pattern recognition, Jul. 2017, pp. 20 - 28.

[22] Bonnin, G., & Jannach, D., “Automated generation of music playlists: Survey and experiments,” ACM Computing Surveys (CSUR), vol. 47, no. 2, pp. 1 - 26, 2015.

[23] Huang, X., Qian, S., Fang, Q., Sang, J., & Xu, C., “CSAN: Contextual self - attention network for user sequential recommendation,” in Proceedings of the 26th ACM international conference on Multimedia, Oct. 2018, pp. 447 - 455. ACM.

[24] Li, Z., Ding, X., & Liu, T., “Constructing narrative event evolutionary graph for script event prediction,” arXiv preprint arXiv:1805.05081, 2018.

[25] Li, X. P., Song, J., Gao, L. L., Liu, X., Huang, W. B., He, X., & Gan, C., “Beyond RNNs: Positional self - attention with co - attention for video question answering,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 7406 - 7413, Jul. 2019.

[26] Zhao, W. X., Li, Y., Chen, Y., Wang, M., & Wang, F., “RecBole: Towards a unified, comprehensive and efficient framework for recommendation algorithms,” in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Aug. 2019, pp. 2902 - 2910. ACM.