WaveRec: Frequency Decomposition and Resolution Augmentation with Wavelet Transform & Upsampler for Sequential Recommendation

Submitted as part of
CE901 MSc Project and Dissertation

Submitted by:
Kun Xie
ID 2310011

Supervisor:
Dr. Richard Sutcliffe

Assessor:
Dr. Hassan Malik

Abstract

In recent years, the field of sequential recommendation system has received increasing attention due to its capability of capturing temporal dependencies within the sequence of interactions between users and items. Sequential recommendation systems are considered particularly suitable for applications involving dynamic changes of user interests and time-sensitive scenarios. However, various challenges are identified to construct effective sequential recommendation systems, including appropriate balance of short-term and long-term preferences and severe sparsity issues in large-scale data environments.

This study focuses on the next-item prediction task, framing it as a time-series forecasting problem. A series of experiments explore the effectiveness of sequential recommendation methods in capturing user behavior patterns across three datasets: MovieLens-1M, Amazon Sports & Outdoors, and Amazon Movies & TV. A key innovation in this research is the incorporation of wavelet transforms to perform multi-resolution analysis of interaction sequences, allowing the model to differentiate between long-term trends and short-term fluctuations. Additionally, an Upsampler module is implemented to enhance the resolution of the wavelet transform outputs, improving the feature representation of different frequency components.

This research is built upon the RecBole library to design and implement innovative model architectures for sequential recommendation tasks. Leveraging RecBole's modular framework, various sequence representation methods and feature fusion strategies are explored. By conducting extensive experiments across multiple datasets, the study identifies optimal models that enhance recommendation accuracy and generalization, particularly addressing data sparsity and dynamic user behavior.

Contents

1 Introduction

In recent years, sequential recommendation systems have become an important topic in the field of personalized recommendation system. While traditional recommendation models typically rely on static relationships between users and items, sequential recommendation models focus on temporal dependencies within the interaction sequence between users and items. By leveraging the historical interaction sequence, the sequential recommendation system capture the dynamic changes of user interests, thereby to make prediction of the next potential item [1, 2]. Such characteristics make sequential recommendation systems suitable for time-sensitive and constantly changing application scenarios.

Various challenges have been identified to effectively perform sequence modelling for user behaviors. The most critical one among which is to accurately capture short-term and long-term dependencies within the sequence, user preferences usually exhibit immediate changes over short periods, but can also maintain stable and constant interests. Moreover, datasets in this domain are often sparse, with limited interaction records for each user and item, particularly in large-scale environments such as e-commerce or entertainment platforms [1, 12]. Therefore, sequential recommendation models not only need to demonstrate the competent generalization capability but also make balance between considering immediate and persistent interests in order to deliver accurate and timely recommendations.

This study focuses on the next-item prediction task, which is considered a time-series forecasting problem. Three datasets, including MovieLens - 1M, Amazon - Sports & Outdoors and Amazon - Movies & TV, are involved to explore the effectiveness of deep learning models to capture user behavior patterns. In this study, a critical aspect of the proposed approach is the integration of the wavelet transform operation to perform multi-resolution analysis for interaction sequences, allowing the model to distinguish the long-term trends and short-term fluctuations. Additionally, this study implements the UpSampler module to enhance the resolution of outputs from wavelet transform, thereby to facilitate feature representations of different frequency components.

This research builds upon the in-depth study of the open-source RecBole framework and the CFIT4SRec model, implementing the current model architecture based on their existing code. Specifically, data preprocessing, model training, evaluation, and testing are carried out using the unified infrastructure provided by the RecBole open-source framework. On the other hand, the definition and implementation of the loss functions are based on the methodology employed in the CFIT4SRec model, with appropriate optimizations and improvements. By extending and innovating on these existing frameworks, this study effectively explores and enhances the performance of sequential recommendation systems, particularly in the areas of feature fusion and sequence representation methods. For detailed description of contributions of this study, please see the section "Contribution Specification".

2 Task Definition

Sequential recommendation systems aim at predicting interested items based on the historical interaction sequence of the user. Compared to traditional recommendation tasks and systems, sequential recommendation specially focus on the temporal dependencies and sequential ordering of interaction sequence in order to capture the dynamics of user interests [1].

The sequential recommendation task of the current study aims to predict the next item that the user is likely to interact with, which is usually considered as a time-series forecasting task. Given the user's historical interaction sequence, the recommendation system needs to comprehensively capture the relationship and dependency among chronological interactions in both short-term and long-term scales [1, 2].

Additionally, a sequential recommendation system is usually required to cope with various

Table 1: Datasets Statistics

Dataset	users	avg$_u$	items	avg$_i$	interactions	sparsity
MovieLens - 1M	6041	165.50	3417	292.63	999611	95.157%
Amazon - Sports & Outdoors	331845	8.54	103912	27.28	2835125	99.992%
Amazon - Movies & TV	297378	11.46	59926	56.88	3408912	99.981%

avg$_u$: average interactions of per user
avg$_i$: average interactions of per item

challenges including dataset sparsity, user behavior diversity and dynamic preference changes. To this end, it is significant to design and implement a robust recommendation method with capabilities of effective learning and capturing the sequential dependency, thereby to improve the recommendation accuracy.

2.1 Datasets

In the experimental stages, this study focuses on the dataset MoveLens - 1M, but final tests will be conducted on three datasets: MovieLens - 1M [5], Amazon - Sports & Outdoors and Amazon - Movies & TV [12]. Each sample in the datasets represents a record of interaction between a user and an item, of which three attribute columns are concerned, including user id, item id and timestamp. However, the original data are not chronologically sorted, and not sorted by ids either. Therefore, data need to be transformed into the user-oriented format during the data preprocessing section, in which, each sample represents the historical behavior sequence of a user, chronologically sorted. The maximum of sequence length has been set to 50, and the sequence shorter than this number were padded with zeros. Additionally, a new attribute seq_len was created to track the sequence length of each sample, which as well indicates the position of the last item with which the user interacted with.

Dataset Statistics

According to dataset statistics, as shown in the **Table 1**, a significant challenge of sparsity is identified in the current sequential recommendation task. Especially for two Amazon datasets, there are high sparsity and very low average interaction counts of users and items. Given that this study sets the constraint of the length of user interaction sequences to 50, the average number of user interactions in the datasets appears to be quite low, it becomes more difficult for recommendation models to capture the user preference and behavior pattern, especially for users who have only interacted with a limited number of items. On the other hand, the recommendation model may struggle to adequately assess the relevance or popularity of items with fewer interactions, which could bring bias that items with more interactions are favored, while items with less interactions, which may still be highly relevant, are overlooked. The three datasets exhibit varying levels of sparsity, which could lead to differing performance outcomes for the recommendation system.

2.2 Task Modeling

This study uses the following definitions to describe the elements of the current sequential recommendation task.

User set $U = \{u_1, u_2, \ldots, u_{|U|}\}$ denotes the set of all users with no less than 5 interactions within the datasets. This study sets this constraint to prevent the model from adverse noises.

Item set $I = \{i_1, i_2, \ldots, i_{|I|}\}$ denotes the pool of all items with no less than 5 interactions within the datasets. This study sets this constraint as a minimum popularity for items.

By data preprocessing, this study constructs the collection of sequences of user historical interactions, for each user u, this study defines its interaction sequence as $S^u = [i_1^u, i_2^u, \ldots, i_{|I|}^u]$, in which S^u is a chronological interacted items sequence, the length of the sequence is restricted as no more than 50 items.

The current sequential recommendation task is a supervised learning task aimed at predicting the next item $i_{|S^u|+1}^u$ for a given user u based on the preceding sequence S^u. This study defines this process using the following mathematical formula:

$$\hat{i}_{|S^u|+1}^u = \arg\max_{i \in I} P(i \mid S^u)$$

This formula determines the item by maximizing the probability of a user interacting with it, based on the preceding sequence of interactions.

2.3 Evaluation Metrics

This study adopts three popular evaluation metrics to assess the performance of recommendation methods: Hit Rate (HR), Normalized Discounted Cumulative Gain (NDCG) and Mean Reciprocal Rank (MRR). These metrics provide different perspectives to comprehensively evaluation the experimental performance of models.

2.3.1 Hit Rate (HR)

HR is a fundamental metrics widely used in recommendation task, which measures whether the target item of user's interest is included in the recommendation list. For each user, if the target item is present in the top k positions within the recommendation list, it is considered a 'hit'.

$$\text{HR@k} = \frac{1}{|U|} \sum_{u \in U} \mathbb{I}(\text{target}_u \in \text{Top-k}_u)$$

Where $\mathbb{I}$ is an indicator function, which takes 1 if the target item is present in the top k items within the recommendation list, and 0 otherwise. $|U|$ denotes the number of users.

2.3.2 Normalized Discounted Cumulative Gain (NDCG)

NDCG measures the ranking capability of the recommendation system, which takes into account not only the presence of the target item but also its ranking position within the list, which achieves this by assigning higher weights to the items ranking front. NDCG is defined based on Discounted Cumulative Gain (DCG), NDCG introduces normalization to ensure comparability across different users.

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{\text{rel}_i}{\log_i(i+1)}$$

$$\text{NDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}}$$

Where rel_i represent the relevance score of the item at position i, and IDCG@k denotes the ideal DCG value in which the most relevant items ranked first.

2.3.3 Mean Reciprocal Rank (MRR)

MRR focuses on the ranking position of the first relevant item within the recommendation list. For each user, it finds the ranking of the first relevant item, then calculates its reciprocal and take the average of the reciprocal ranks across all users.

$$\text{MRR@k} = \frac{1}{|U|} \sum_{u \in U} \frac{1}{\text{rank}_u}$$

Where rank_u denotes the position of the first relevant item for user u in the recommendation list.

3 Literature Review

3.1 Evolution

In recent years, prominent advancements have been made in the field of recommendation system. Traditional recommendation methods, such as collaborative filtering and content-based filtering, can effectively establish matching relationships between users and items [24]. However, as the complexity of interactions between users and items increases, static recommendation methods usually struggle to capture dynamic shifts in user interests. Therefore, the field of sequential recommendation system has been receiving increasing attention from researchers. Through analysis and process on chronological sequence of user-item interactions data, sequential recommendation systems achieve higher accuracy in capturing user behavior patterns and offering personalized recommendation [14].

The main target of sequential recommendation is to predict the next item that the user likely to be interested in. Early methods primarily rely on the Markov chain model, which captures short-term dependencies to make predictions [15]. Such methods perform well in modelling simple sequences, but have limitations in capturing long-distance dependencies and complicated user behavior patterns. To address these issues, the matrix factorization and implicit feedback-based methods have been introduced to facilitate sequence modelling. Through decomposing interaction matrix into latent factors, these methods exhibit the competent capability of long-distance modelling of user interests.

The advancements of deep learning techniques have been improving sequential recommendation systems in recent years, and become the mainstream solutions for complicated user behavior modelling. Recurrent neural networks, due to the inherent capability of sequence modelling, became one of the initially applied deep learning models in the field of sequential recommendation. Through proceeding and preserving the hidden states, RNNs can effectively capture long-term behavior patterns in the temporal sequence. However, limitations were also identified, including inadequate computational efficiency when handling long sequence data and problems of gradient vanishing. To address them, variants of the RNN structure were proposed, such as long short-term memory (LSTM) [30] and gated recurrent unit (GRU) [6].

In addition, Convolutional Neural Networks (CNNs) and Graph Neural Networks (GNNs) have also been gradually introduced into the field of sequential recommendation. By leveraging local receptive fields, CNNs effectively capture short-term patterns within sequences, and also exhibit prominent computational efficiency, benefiting from parallel computation [20]. On the other hand, GNNs have been employed to model complex relationships between users and items by utilizing graph structures, which are particularly effective in modelling higher-order connections and dependencies [23].

Furthermore, the emerging techniques based on the self-attention mechanism bring novel evolution to the field of sequential recommendation. The model based on the self-attention mechanism, such as SASRec and BERT4Rec, can effectively and flexibly capture dependencies across the user behavior sequence through assigning separate weights to each positions

[8, 18]. The successes of these models indicate that the self-attention mechanism carries significant potential in terms of capturing long-distance dependencies and improving recommendation accuracy.

3.2 Caser

Sequential recommendation systems have evolved from traditional approaches like Markov Chains and Factorized Personalized Markov Chains (FPMC) to advanced deep learning models. Based on the use of Convolutional Neural Networks (CNNs), Caser is proposed to model sequential patterns by treating user interaction sequences as two-dimensional images [20]. This allows the model to capture both point-level and union-level patterns, outperforming earlier methods such as FPMC and GRU4Rec, particularly in datasets with rich sequential behavior.

3.3 GRU4Rec

It is suggested that recommendation methods based on traditional matrix factorization are hard to exert in session-based recommendation scenarios due to absence metadata of users. The study of Hidasi applies the GRU-based RNN model for session-based recommendations [6]. Specifically, GRU4Rec applies 1-of-N encoding to represent the item according to the state input of the current session, and generates the next recommended item through processing sequential data.Additionally, GRU4Rec adopts the session-parallel mini-batch strategy to handle data with different session length, ensuring to reset the hidden state at the end of the session, thereby to efficiently understand the user behavior sequence.

The experiments demonstrated that GRU4Rec delivers excellent results in session-based recommendation tasks, outperforming baseline and latest models at that time. However, despite success achieved by GRU4Rec, there are differences between the session-based recommendation and the sequential recommendation that the current study focuses on [22].

Firstly, session-based recommendation tasks primarily focus on modelling user's behavior within a single session. Of which the principal objective is to generate immediate recommendation corresponding to the series of interactions within the session. This process typically do not rely on user's long-term historical data, but focus on the context of the current session. In contrast, sequential recommendation tasks emphasize modelling temporal sequences of user behaviors, typically taking into account the short-term and long-term interactions to generate recommendations. Sequential recommendation system not only consider interactions within a single session but only track that across multiple sessions over time, capturing shifts in user interests.

3.4 SASRec

In recent years, it has become an important topic for researchers to develop sequence modelling of user behaviors, traditional methods, including Markov Chains (MCs) and recurrent neural networks (RNNs), are commonly adopted techniques [1]. MCs assume that the next action of the user only depends on the previous one or several actions, which are suitable for processing short-term dependencies in sparse datasets [15]. In contrast, RNNs can capture dependencies over longer time spans through recursive proceeding of hidden states. However, their performances rely on extensive data, particularly demonstrating competent performance on dense datasets [6]. In addition, traditional methods have limitations when facing complex sequence patterns. MCs struggle to handle complicated dependencies of long-distance due to the simplified assumption it based on, while RNNs, although capable of handling long-distance dependencies, have lower computational efficiency, especially when dealing with long sequence data. Therefore, researchers attempt to introduce the attention-based mechanisms to improve model's performance and efficiency.

In this context, the SASRec model is proposed based on the Transformer module for sequential recommendation, which excels at capturing long-term semantic dependencies of user behavior sequences [8]. For detailed discussion about applying Transformer in the sequential recommendation model, please see the Methodology section. Compared to traditional methods, such as the MC and RNN, SASRec performs more flexibly when handling datasets of various density, moreover, exhibits higher computational efficiency thanks to its paralleled sequence processing.

In recent years, extensive emerging sequential recommendation models adopt the SASRec model as the foundational structure, and develop further improvements and optimizations [1, 2, 13, 29]. Such Transformer-based models demonstrate not only excellent accuracy for recommendation but also computational efficiency when dealing with large-scale data. Therefore, SASRec has become the mainstream framework to construct cutting-edge methods, which yield excellent performance on multiple standard datasets for sequential recommendation.

3.5 CFIT4SRec

Despite the success achieved by traditional sequential recommendation methods, such as models based on convolutional neural network (Caser), recurrent neural network (GRU4Rec) and transformer (SASRec), they exhibit inadequate performance in handling rapidly changing interest trends and noisy data. According to the theory of Frequency Principle, deep learning models typically tend to optimize learning low-frequency features, which probably lead to limited performance on learning high-frequency feature, such as rapidly changing interests [26]. Accordingly, CFIT4SRec is an innovative model proposed, which not only relies on feature extraction in the temporal domain but also introduces contrastive learning practice in the frequency domain to further improve the recommendation capability [29].

CFIT4SRec regards the user behavior sequence as images-like data, and draws on signal processing techniques from the computer vision field, introducing two-dimensional Fourier transform to construct self-supervised samples for contrastive learning. CFIT4SRec leverages three methods of data augmentation, low-pass, high-pass and band-stop, to remain or filter frequency-specific signal components, by which to construct self-supervised signals. This strategy allows the model to focus and adapt dynamics and changes of different frequency scales in the sequence.

This strategy employs the multi-task learning framework, by simultaneously optimize recommendation and contrastive learning tasks to improve model capabilities of generalization and user representation. Specifically, CFIT4SRec generate self-supervised sample pairs of different frequency components through a series of data augmentation operations, including low-pass, high-pass and band stop.

In order to implement effective modelling of user interest as the query vector, CFIT4SRec defines a comprehensive loss function, which put together the recommendation and contrastive learning loss. During the training process, model parameters not only update depending on recommendation loss, but also guided by contrastive learning loss. The loss definition of contrastive learning maximizes the similarity of the samples augmented from the same sequence (positive sample pair), and minimizes the similarity of the samples augmented from different sequence (negative sample pair), by which to facilitate the model to identify trends of user interests of different frequencies.

3.6 Frequency Principle

It has been considered an important direction for researchers to understand the training behavior and generalization capability of deep neural networks in recent years. In this regard, the frequency principle (F-Principle) proposed by Xu et al. [26] provides a new theoretical perspective for revealing how neural networks gradually fit the objective function from low to high frequencies. Based on the Fourier analysis, this theory suggests that neural networks tend to prioritize fitting the low-frequency component, and subsequently learn the high-frequency component. F-Principle reveals the inherent deviation of neural networks, providing the theoretical support to explain the model's generalization capability.

3.7 Emerging Themes

In the research context of sequential recommendation, several emerging themes have gained board and significant attention due to their potentials in addressing key issues, including long-term dependencies modelling, dataset sparsity and user behavior complexity. The following covers representative topics of them, including contrastive learning, graph neural network and cross-domain recommendation.

- **Contrastive Learning**

  Contrastive Learning has become increasingly popular for sequential recommendation due to it can learn robust representations through comparing positive and negative samples [13, 29]. By maximizing and minimizing the similarity between positive and negative sample pairs respectively, the contrastive learning technique facilitates the model to optimize feature space representations. This approach exhibit effective benefits in regards of improving the generalization capability of the model.

- **Graph Neural Network**

  The graph neural network technique has been considered a powerful approach for the field of sequential recommendation, which models the interaction between users and items as the graph structure, by which to represent the high-order links and dependent relationships between users and items [23, 28, 29]. This capability is particularly significant for sequential recommendation, as user behaviors are typically influenced by complicated and non-linear relationships, which is beyond the existence of the simple temporal sequence. By integrating the information between neighborhood nodes within the graph, GNNs bring enhanced capabilities to generate personalized and context-relevant recommendations.

- **Cross-Domain Recommendation**

  Cross-domain recommendation is an emerging field aiming at improve the recommendation capability through exploiting information from various domains or contexts [10, 32]. Traditional single-domain recommendation systems usually struggle to deal with the data sparsity issue, while cross-domain recommendation systems addresses it by transferring knowledge from the original domain to the target domain. This knowledge transfer approach exploits the auxiliary data from relevant domains to improve the performance of recommendation systems when dealing with the data sparsity issue. In addition, the cross-domain pattern brings more comprehensive and global modelling for user profiles, by which to generate accurate and diverse recommendations.

4 Methodology

Given the historical interaction sequence S_u of the user, the recommendation model needs to predict the appearance probability of each item across the item set I. To achieve this, the recommendation model focuses on effectively performing feature extraction. The items chronologically arranged through the user's behavior sequence are first transformed into embedding features, by which the embedding sequence is obtained. Subsequently, the model takes the embedding sequence as the input and perform feature extraction through multiple layers of neural networks, by which the query vector is generated as the feature representation of the target position. The query vector encompasses context information of the position and potential interests of the user, which is used to calculate the similarity score of each item from the pool.

By calculating the similarity between the query vector and item embeddings, the score is considered the probability of the item appearing on the target posion. The recommendation model will be trained and optimized over iteritations to improve the accuracy of such probability predictions.

4.1 Wavelet Transform

Effective feature extraction is the critical focus in the current task, as which significantly determines the capability of the recommendation model to precisely understand and predict user interests. This study employs the wavelet transform mathematical techniques to process sequential data in order to extract both short-term and long-term interests from the user's behavior sequence.

The wavelet transform is a mathematical transform technique for signal analysis and processing, particularly suitable for multi-resolution analysis for non-stationary signals [27, 3]. Through which, the signal can be divided into components of different frequencies, thereby to separately perform processing and analysis of low-frequency and high-frequency features [16]. The low-frequency component captures overall trends and consistent pattern, which reflects relatively stable interests and preferences in a long term scale. The low-frequency component is crucial for understanding persistent preferences of the user, which facilitates generating customized recommendations that align with constant interests. On the other hand, the high-frequency component plays an important role in identifying short-term changes and fluctuations in the user behaviors sequence, which usually indicates user's immediate demands or temporary interests. By focusing on these high-frequency elements, the model can respond more effectively to user's latest behaviors changes, thereby enhancing timeliness and relevance of recommendation generated.

This study leverages wavelet transform to isolate and analyze frequency-specific components, allowing the model to capture nuance and delicate dynamic in the sequence. This comprehensive feature extraction method ensures that persistent and immediate dynamics of user preference are both taken into account, thereby improving model's capability to deliver personalized and contextually relevant recommendations.

4.1.1 Discrete Wavelet Transform (DWT)

Discrete Wavelet Transform is employed to process the input embedding vectors. DWT decomposes the sequence signal into low-frequency and high-frequency components. Low-frequency components represent the overall trends and long-term preferences of user behavior, which are crucial for understanding sustained user interests. High-frequency components capture short-term fluctuations and rapid changes in user behavior, which are evidence for identifying immediate user interests and needs.

Discrete wavelet transform is a powerful instrument for signal analysis and feature extraction, especially suitable for multi-resolution analysis for non-stationary signals [27]. Mathematically, DWT decomposes a signal $x(t)$ into a series of wavelet coefficients by applying a pair of filters: a low-pass filter and a high-pass filter, which are derived from a mother wavelet function $\psi(t)$ and a scaling function $\phi(t)$. The decomposition process can be expressed as:

$$c_j[n] = \sum_k x[k] \cdot \phi_{j,n}[k]$$

$$d_j[n] = \sum_k x[k] \cdot \psi_{j,n}[k]$$

Where $c_j[n]$ represents the approximation coefficients (low-frequency components), and $d_j[n]$ represents the detailed coefficients (high-frequency components). The scaling function $\phi_{j,n}(t)$

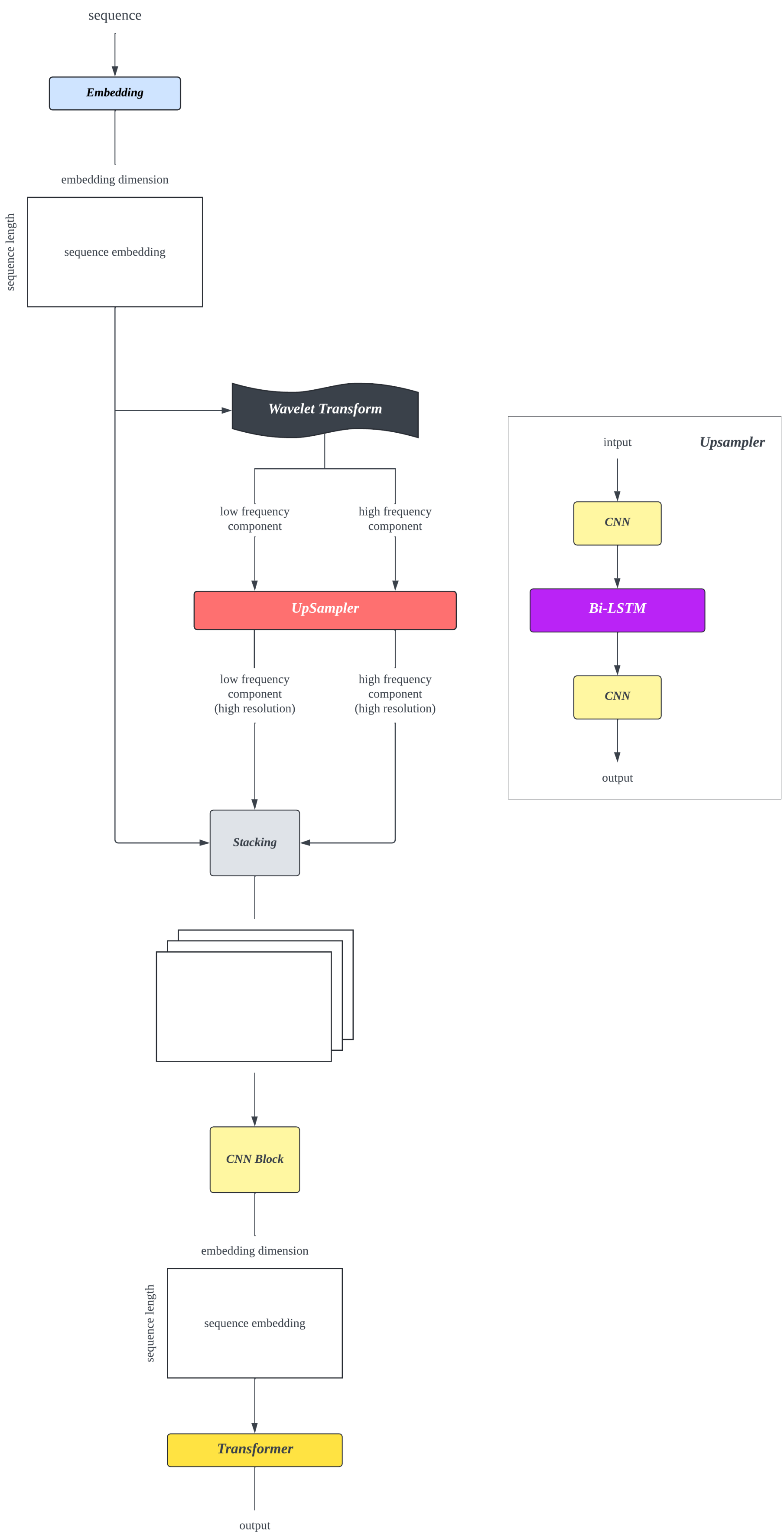

Figure 1: Architecture of WaveRec

and $\psi_{j,n}(t)$ capture the low-frequency component and high-frequency component, respectively, of the signal at scale j. These coefficients are calculated recursively by applying the filters to progressively downsampled versions of the original signal, decreasing the resolution over each level. The decomposition can be illustrated as a series of convolutions:

$$c_j[n] = \sum_k h[k-2n] \cdot x[k]$$

$$d_j[n] = \sum_k g[k-2n] \cdot x[k]$$

Where h and g denote the impulse responses of the low-pass and high-pass filters, respectively.

4.1.2 Inverse Wavelet Transform (IDWT)

Inverse wavelet transform is employed to reconstruct the signal from the wavelet coefficients by reversing the decomposition process. IDWT combines the approximate coefficient (low-frequency component) and detailed coefficient (high-frequency component) at each scale to progressively reconstruct the signal with the same size of original one [27]. Mathematically, this process can be expressed as:

$$x[k] = \sum_n (c_j[n] \cdot \phi_{j,n}[k] + d_j[n] \cdot \psi_{j,n}[k])$$

4.1.3 Hyperparameters

When incorporating wavelet transform into the neural network, the selection of hyperparameters is critical for the performance and effectiveness. The following is an introduction of commonly focused hyperparameters of wavelet transform.

Wavelet Type

The wavelet type determines the basis function adopted in the wavelet transform process, namely the mother wavelet. Various wavelet types have different characteristics and application scenarios, the following are wavelet types commonly adopted.

- **Haar:** The simplest type of wavelet, suitable for processing signals with transient changes.
- **Daubechies:** The type with better smoothness and flexibility, usually adopted for processing complicated signals.
- **Symlets:** A series of symmetric wavlets, similar to Daubechies wavelets but exhibit higher symmetry in the calculation.
- **Coiflets:** The wavelets suitable for signal reconstruction, as they show better accuracy in processing high-frequency details.

Order

The order of the wavelet determines the complexity and smoothness of the mother wavelet. Higher order comes the longer length that the wavelet supports, meanwhile, allows to capture more signal details, but brings increased computational complexity. For example, 'db3' denotes the Daubechies wavelet in order 3.

Level

The level of decomposition determines the depth that wavelet transform perform decomposition to the signal. The higher level, the more scales that the signal is decomposed into, namely more components of different frequencies.

4.1.4 The PyWavelets Library

In order to efficiently implement wavelet transform method in the current model, this study employs the third library called PyWavelets. PyWavelts is a python package built upon the PyTorch framework, allowing highly flexible and efficient implementation to be integrated in the deep learning models, and as well compatible with GPU acceleration for trainign and prediction [9]. A wide variety of predefined wavelets are supported and it is available for users to specify and custom wavelets filter options. Thanks to the PyWavelets package, users can simply customize different hyperparameters configurations and conduct experimental evaluations for wavelet transform implementation.

4.2 Upsampler

the upsampler module is implemented in this study in order to enhance the feature representation of separate frequency components, it enhances the resolution of outputs of wavelet transform re-composition by combination of convolution and bidirectional LSTM layers. The upsampler improves the capabilities of the model in terms of detail representation and feature capturing of temporal sequences [11].

4.2.1 Architecture

Upsampling and resolution augmentation are considered important steps in many types of tasks, particularly those involving temporal sequence and image processing topics. This study employs the encoder-decoder architecture for the upsampler, which adopts two symmetric convolutional blocks as the encoder and decoder, enclosing the bidirectional LSTM layer in the middle.

Traditional methods of upsampling usually rely on interpolation techniques, but in recent years, deep learning methods, based on sequence to sequence (Seq2Seq) models, exhibit distinctive advantages in such fields. The core advantage of Seq2Seq models lie in their ability to flexibly handle length-variable sequences, and capture complicated relationships between inputs and outputs [19], which makes them suitable for the current sequential recommendation task. In the upsampling module, the Seq2Seq model encodes low-frequency signals to high-dimensional feature representations, and generate output in the higher frequency. This approach not only preserves the key features of input signals but also enhance the resolution and quality of the output through multi-step predictions in the decoding stage. Specifically, the encoder first map the low-resolution input into a high-dimensional representation of hidden-state, which captures the global information of the signal. Subsequently, from which the decoder generates the high-resolution output at a finer spatiotemporal scale. Compared to traditional interpolation methods, this learning-based upsampling techniques handle the complicated context relationships more effectively. Seq2Seq models offer a flexible and powerful solution for upsampling and resolution augmentation, particularly in tasks involving complex context dependencies.

Convolutional Blocks

Convolutional Blocks are particularly effective at reducing spatial redundancy and identifying key features through weight sharing and local connectivity, making them effective for tasks involving structured data, such as images or sequential data with spatial dependencies [4]. By employing symmetric convolutional blocks, the encoder can extract relevant features and compress the input into a meaningful representation, while the decoder can use these extracted features to reconstruct the target output with high fidelity, ensuring both efficiency and accuracy in the overall model.

Bi-LSTM Layers

The bidirectional long short-term memory network plays a crucial role in the upsampler module. While traditional LSTM networks effectively capture long-term and shot-term dependencies in sequential data through the gating mechanisms, they learn from the sequence in a unidirectional pattern [7]. In many tasks, particularly those involving temporal data, utilizing both forward and backward information can provide a more comprehensive understanding of feature, to this end, the Bi-LSTM is introduced.

The LSTM structure, as a variant of recurrent neural network (RNN), exhibits the excellent advantage in handling the issue of long short-term dependencies in temporal sequences [17]. LSTM effectively alleviates the gradient vanishing problem encountered by standard RNNs in long sequences through its unique memory gate mechanism. By simultaneously considering the forward and backward information in the sequence, the bidirectional approach helps to reconstruct information at a more detailed level while preserving global information accurately. Moreover, the gating mechanism of LSTM can effectively process long short-term dependencies and reduce information loss, which plays a critical role to improve the accuracy of the upsampling operation. Specifically, in the upsampler module, Bi-LSTM handles the low-frequency and high-frequency components, enhancing their representational capacity. By extracting information from both the beginning and the end of the sequence, Bi-LSTM enables the model to reconstruct sequence features with greater precision.

This bidirectional processing mechanism is particularly beneficial for modelling the complex patterns in user behavior sequences, where short-term actions (high-frequency components) and long-term preferences (low-frequency components) often exhibit interdependencies. By incorporating both forward and backward information, the Bi-LSTM accurately models these dynamic features, effectively improving the resolution of frequency components during the upsampling phase. This provides richer and more precise feature representations for subsequent recommendation tasks.

4.2.2 Resolution Augmentation

The upsampler exerts remarkable resolution augmentation in multiple aspects , it not only refines and enhances representations for components of different frequencies but also improves the overall performance of the model though multi-level mechanisms for information processing and feature fusion. Firstly, in terms of low-frequency component refinement, the upsampler finely reconstructs low-frequency information, improving its representation precision. Which enables the mode to capture and reflect the long-term behavior trends more accurately. This refinement processing helps the model maintain constant and stable prediction performance across long temporal spans, ensuring accurate identifications and long-term preference predictions as processing complicated historical data. Simultaneously, in terms of high-frequency augmentation, the upsampler combines advantages of convolutional and recurrent layers, further improving resolutions of high-frequency components of spatial and temporal aspects. The convolutional layers help the model capture localized feature in user behavior, while recurrent layers excel at processing temporal dependencies of sequential data. Such combination allows the model sensitively detect subtle fluctuation and dynamics in short-term user behaviors, by which to realize more flexible and dynamic recommendations. In addition to refinement and augmentation to frequency-specific components, the upsampler exerts critical functions for feature fusion. Through high-resolution sampling, the upsampler ensures consistent fusion of low-frequency and high-frequency components embedded in the same feature dimension as the original input. Such dimensional consistency offer the fundamental condition for detailed information integration, allowing multiple features can be mapped and processed in the common feature space, thereby improving the accuracy and stability of recommendations.

4.3 Transformer

The Transformer architecture has become one of the most influential advancements in deep learning, particularly in sequential data modelling. The core idea of Transformer is based on the self-attention mechanism, combined with the multi-head attention mechanism and position embedding techniques. This pattern of designing allows the Transformer model to outperform recurrent neural networks (RNNs) and convolutional neural networks (CNNs) in aspects of processing long-distance dependencies parallel computation [21]. Unlike traditional recurrent neural networks (RNNs) that rely on sequential processing, the transformer structure employs a self-attention mechanism that enables it to process entire sequences in parallel. This parallelization not only improves efficiency but also allows the model to better capture long-range dependencies within the data.

The self-attention mechanism is the most core idea of Transformer, considered one of the most critical characteristics, distinguished from traditional models for sequence processing. This mechanism fundamentally aims to grant separate weights for each elements of the sequence, allowing them to perform focusing towards other elements in the global scope. Through this approach, the self-attention mechanism can effectively capture the interdependence between different elements in the sequence, without being limited in a localized neighborhood or their positions. The self-attention mechanism aggregates information across the entire sequence, enabling the model to dynamically attend to different positions, regardless of their distance in the sequence.

Traditional recurrent neural networks adopt the step-wise approach as sequential data processing, in which the calculation of the current step relies on the output of previous time step. Although such sequential-process approach enables the model to capture temporal dynamics in the sequence, it brings inevitable limitation, particularly for extended sequences. As the sequence length increases, the gradual-processing characteristic of RNNs make it difficult to capture long distance dependencies, resulting in problems of vanishing or exploding gradients, which then affect the training efficiency and performance of the model.

In contrast, the self-attention mechanism does not rely on sequential processing procedures, which allows the model simultaneously perform global calculation for all elements at a single time step. The self-attention mechanism processes the entire sequence by assigning a group of specific vectors of query, key and value for each elements, and subsequently get the relevance weights between each elements through dot production calculations. This practice not only excels at capturing long-distance dependencies between elements, but also perform overall modelling for the sequence with excellent computation efficiency.

For recommendation systems, the attention mechanism offers significant gains. By leveraging self-attention, the model can effectively capture both short-term and long-term user preferences without being constrained by the fixed order of interactions. This flexibility is particularly advantageous for capturing complex patterns in user behavior, where the importance of specific interactions may vary based on the overall context of the sequence.

Additionally, the attention mechanism's ability to weigh and integrate information from multiple points in the sequence enhances the model's capacity to generate more accurate and personalized recommendations. This dynamic feature extraction helps the Transformer excel in scenarios where user behavior is multifaceted and influenced by a range of factors across different timescales. Ultimately, the attention mechanism amplifies the model's ability to focus on the most pertinent details, thereby boosting overall performance.

5 RecBole Framework

This study is conducted based on the RecBole framework, which is an open-source library specially designed for recommendation system, aiming to provide a uniformed and unified experimental platform in light of the great concern of the reproducibility of recommendation strategies [31] [25]. Recbole supports an extensive number of recommendation methods from traditional collaborative filtering algorithms to cutting-edge deep learning models. It integrates a great number of datasets and benchmark in the recommendation system field. RecBole targets to improve the efficiency of procedures of recommendation system including implementation, training and evaluation, which offers a highly modular and extensible environment for developers and academics.

- **Unified Framework:**

  The RecBole library is a comprehensive framework built upon the PyTorch deep learning framework and is designed to support the development of principal modules, including data processing, model implementation and evaluation metrics. It offers a user-friendly and flexible environment to facilitate procedures of development and experimentation across an extensive range of recommendation system tasks and topics.

- **Datasets and Models:**

  RecBole integrates comprehensive options of benchmark datasets and models, encompassing principal methodologies ranging from traditional collaborative filtering algorithms to emerging deep learning techniques. By using RecBole, users can efficiently conduct experiments on popular datasets with existing models through

- **Evaluation protocols:**

  RecBole supports a series of standard evaluation protocols for experimental comparison between different methodologies. Users can customize evaluation settings from comprehensive options of different metrics, which supports measurements of different capabilities of recommendation systems, including retrieval, ranking and sorting.

6 Experiments

The experimental process in this study can be divided into three phases. In the first phase, a series of state-of-the-art models for sequential recommendation have been tested, several widely experimented and applied models were selected as benchmarks for comparison and reference. Meanwhile, the foundation and fundamental framework of the proposed model is determined in this phase. In the second phase, preliminary design and innovation has been incorporated. In the third phase, an extensive number of experiments have been conducted, during this process, the final structure of the proposed model is determined. Moreover, the appropriate configuration and combination of hyper-parameters has been obtained in terms of the current datasets.

The research and experiments was guided by several key questions aimed at enhancing the performance of sequential recommendation systems:

- **Q1:** How can wavelet transforms improve the performance of sequential recommendation models?
- **Q2:** What is the impact of using the UpSampler module on feature representation and recommendation accuracy?
- **Q3:** How do different feature fusion strategies affect the model's ability to capture both short-term and long-term user behavior?

By answering these research questions, the study provides new insights into optimizing sequential recommendation systems through wavelet transform integration, feature up-sampling, and advanced fusion techniques.

6.1 Experiment I: Evaluation across Existing Methods

Aim

Conducting experimental evaluations across existing methods for sequential recommendation, and determine the foundation module for the proposed model.

Description

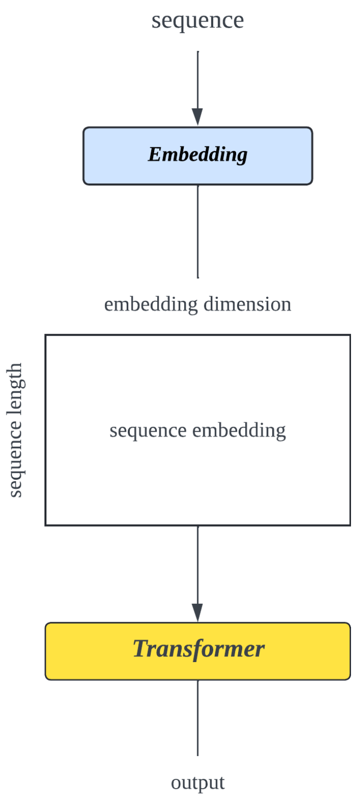

Figure 2: SASRec Architecture

In the first experiment stage, a comprehensive evaluation and test of several state-of-the-art sequential recommendation models was conducted, including Caser, GRU4Rec, SASRec and CFIT4SREC. For experimental results, see table 4. Thanks to the powerful reproducibility feature that the RecBole framework brings, most of these models could be directly obtained and executed on standard datasets for experimental test without manual implementation, which significantly improves the efficiency of experiment procedures. Moreover, the source code of of CFIT4SRec is publicly released by its authors, allowing for downloading for experimental evaluation. Throughout the preliminary experiments, these sequential recommendation models have shown promising performance. Particularly GRU4Rec and SASRec, despite their simple and easily accessible architectures, they exhibited strong and robust recommendation capabilities. After extensive experiments and consideration, SASRec was selected as the base model for the current study, which delivers a robust and reliable foundation thanks to the transformer module based on the multi-head self-attention mechanism.

6.2 Experiment II: Incorporation of Wavelet Transform

Aim

The goal of the second experimental stage was to incorporate wavelet transform to enable the model to learn from different frequencies in sequential data, and to explore effective feature fusion methods for combining the original, low-frequency, and high-frequency components.

Description

In the second experiment stage, the wavelet transform technique is initially incorporated to grant the model with capability to learn from different frequencies of the sequential data. Two different options of feature fusion was formulated to combine the original feature, the low frequency component and the high frequency component. The initial approach is to concatenate the three signals along the embedding dimension, resulting in a tensor with triple the original

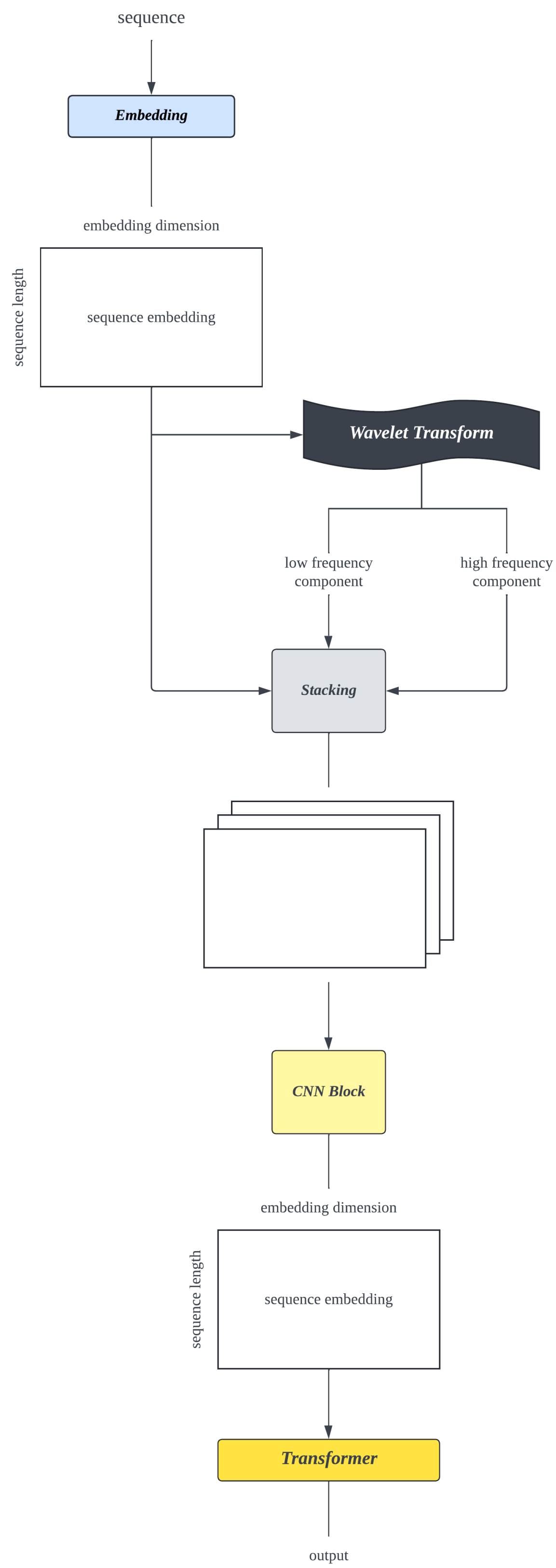

Figure 3: Model architecture in the Experiment II

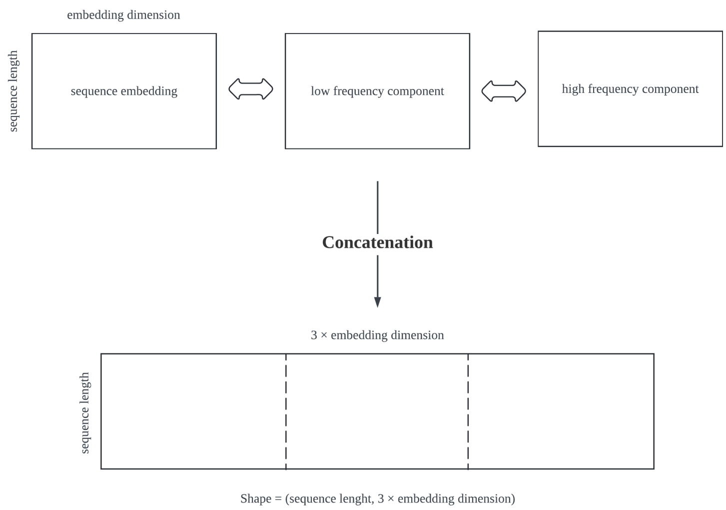

Figure 4: Experiment II - Concatenation

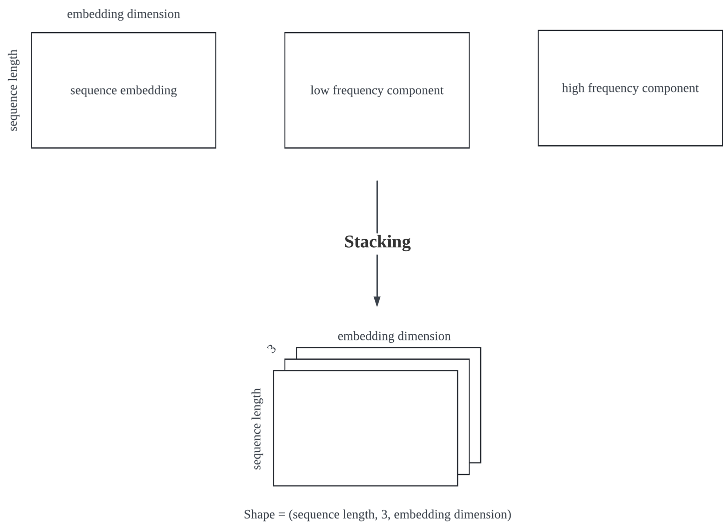

Figure 5: Experiment II - Stacking

embedding size. This tensor was then transformed through a linear layer back into a tensor with the original embedding size. However, this design did not perform well in the experiments, and instead introduced noise, which degraded the model's overall performance.

Upon further analysis, it was hypothesized that the lateral concatenation operation was neglecting the parallel relationship among the three signals and introducing an unreasonable spatial structure, then brought interference to the effective fusion of features. Considering that three signals are inherently parallel in nature, a new strategy was proposed to address this problem: stacking the three signals along the channel dimension, then performing feature fusion by employing the convolutional layer.This approach treats the original embedding, the low frequency component and the high frequency component as paralleled existence by combining them using stacking method. The experimental results showed that the performance of the model has been improved by such feature fusion method, the combination of stacking and the convolutional layer outperforms that of concatenation and the linear layer.

Justification of Feature Fusion

This study adopts the combination of stacking operation and convolutional layers to perform feature fusion. This design aims at sufficiently preserving and exploiting the distinctive

Method	HR@5	HR@10	NDCG@5	NDCG@10	MRR@5	MRR@10
CL	0.203	0.291	0.138	0.166	0.117	0.128
SC	0.212	0.305	0.143	0.173	0.119	0.131

Table 2: Experiment II: comparison between CL and SC

characteristic of each signal, and performing effective feature extraction and fusion through convolutional layers.

The stacking operation allows the model to treat the original embedding, low-frequency and high-frequency signals as independent channels. This approach supports the model to process information from each channel separately without confusing signals from different sources like how the concatenation operation did in the previous experiment. Specifically, the original embedding carries the fundamental context information, the low-frequency signal generally capture the dynamic of user's long-term interests, while the high-frequency signal reflects the dynamic of user's short-term behaviors. By the stacking operation, the model ensure such different and distinctive signals are not roughly mashed together, but appear separately to facilitate feature extraction. On the other hand, convolutional layers remarkably exert the capability of feature fusion in such multiple-channels signal, through its design of sharing weights and local receptive fields, makes the feature fusion process efficient and hierarchical. Convolutional layers not only captures the local feature within each channel but also extracts mutual effects and patterns across different channels. Through the convolutional operation, the model can fuse information from different channels, thereby, to generate richer feature representations.

The combination of the stacking operation and convolutional layers not only remains characteristics of each channel but also effectively performs feature extraction and fusion, which improves the capability of the recommendation model. See table 2 for the experiment result of comparison between CL (Concatenation + Linear) and SC (Stacking + Convolution).

6.3 Experiment III: Adoption of Upsampler

Aim

Further refining the model structure by improving the handling of output signals from the wavelet transform, focusing on processing low-quality signals to reduce noise.

Description

In the third phase, this study dedicated to further refine the model structure upon previous working. After the strategy of the combination of stacking and convolution for feature fusion was established, this study shifted its focus on how to appropriately handle the output signals generated by the wavelet transform module. In the previous stage, the original embedding sequence is stacked with unprocessed low frequency and high frequency components, which neglected the quality of such output signals from the wavelet transform operation. As the result, the stacked tensor was passed through convolutional layers without appropriate handling on poor quality signals individually, which could bring noise into the model pipeline.

To address this issue, two strategies of sequential feature processing were formulated to process the output signals generated by the wavelet transform. The first approach employed a multi-head attention layer before the stacking operation, which processed the output signals to bring the impact of attention mechanism. The experimental result showed that the learning ability of the model is enhanced, which suggested that focusing on improve the quality of output signals is a promising direction and worth exploring further with additional sequential signal enhancement techniques.

Based on such insight, the second approach is proposed to further improve the low-quality signal processing section. This study implemented the upsampler focusing on sequential feature enhancement. For detailed discussion and illustration, please see literature review and method-

Upsampler	HR@5	HR@10	NDCG@5	NDCG@10	MRR@5	MRR@10
x	0.212	0.305	0.143	0.173	0.119	0.131
✓	0.219	0.309	0.150	0.179	0.127	0.139

Table 3: Experiment III: Upsampler testing

Dataset	Metric	Caser	GRU4Rec	SASRec	CFIT4SRec	WaveRec	Improve
MovieLens - 1M	HR@5	0.151	<u>0.195</u>	0.178	0.183	**0.215**	10.26%
	HR@10	0.230	<u>0.282</u>	0.263	0.275	**0.304**	7.80%
	NDCG@5	0.099	<u>0.131</u>	0.118	0.124	**0.147**	12.21%
	NDCG@10	0.125	<u>0.159</u>	0.145	0.154	**0.176**	10.69%
	MRR@5	0.082	<u>0.110</u>	0.098	0.105	**0.124**	12.73%
	MRR@10	0.093	<u>0.122</u>	0.109	0.117	**0.136**	11.48%
Amazon - S & O	HR@5	0.065	0.086	0.085	<u>0.089</u>	**0.092**	3.37%
	HR@10	0.071	0.094	0.095	<u>0.099</u>	**0.102**	3.03%
	NDCG@5	0.058	<u>0.079</u>	0.076	<u>0.079</u>	**0.084**	6.33%
	NDCG@10	0.060	<u>0.082</u>	0.079	<u>0.082</u>	**0.087**	6.10%
	MRR@5	0.056	<u>0.077</u>	0.073	0.076	**0.081**	5.19%
	MRR@10	0.057	<u>0.078</u>	0.074	0.077	**0.082**	5.13%
Amazon - M & T	HR@5	0.113	<u>0.154</u>	0.149	<u>0.154</u>	**0.161**	4.55%
	HR@10	0.135	0.180	0.175	<u>0.183</u>	**0.190**	3.83%
	NDCG@5	0.096	<u>0.134</u>	0.126	0.130	**0.138**	2.99%
	NDCG@10	0.103	<u>0.142</u>	0.134	0.139	**0.147**	3.52%
	MRR@5	0.091	<u>0.128</u>	0.118	0.122	**0.130**	1.56%
	MRR@10	0.094	<u>0.131</u>	0.122	0.126	**0.134**	2.29%

Table 4: Experiment IV

ology sections. The experimental result demonstrated that the encoder-decoder structure which combines convolutional block and bidirectional LSTM significantly improves the performance of the model, and grants better learning ability towards sequential signals and higher recommendation capability.

6.4 Experiment IV: Final Test

6.4.1 MovieLens - 1M Dataset

On the MovieLens - 1M dataset, WaveRec outperforms other baseline models across all evaluation metrics, particularly, WaveRec achieves HR@5 and NDCG@5 scores of 0.215 and 0.147, respectively, as the improvements of 10.26% and 12.21% upon the best baseline model GRU4Rec. Which indicates the effectiveness of WaveRec to deliver more accurate recommendations by capturing user interests in both short-term and long-term periods. Moreover, WaveRec achieves 12.73% and 11.48% improvements for MRR@5 and MRR@10, which implies that WaveRec demonstrates the remarkable capability of recommendation ranking, able to locate relevant items in the top positions within the recommendation list.

6.4.2 Amazon Datasets

WaveRec maintains the leading performance on two Amazon datasets as well, but there are decreased relative improvements compared to the MovieLens - 1M dataset. As for the HR metric, WaveRec brings approximate 3% improvements for two Amazon datasets, while as for the NDCG metric, it brings approximate 6% for the Sports & Outdoors and 3% for Movies & TV, which is considered a moderate improvement scale compared to that of the MovieLens - 1M. This phenomenon indicates that although WaveRec maintains the leading performance , it still suffers hindering from the severe sparsity of two Amazon datasets.

6.5 Discussion

the WaveRec model has achieved performance improvements on multiple datasets by introducing wavelet transform and feature fusion strategies. This indicates that the model can effectively extract features of different frequencies from user behavior sequences, thereby achieving better performance in complex recommendation tasks. However, it is also noticed that on the sparser Amazon dataset, although WaveRec still outperforms other models, its improvement is relatively limited. This may be due to the sparsity of the dataset, which brings tough challenges for sequential recommendation systems. Therefore, future research should consider further optimizing WaveRec to cope with significantly sparse data environments, or alleviating sparsity issues through methods such as data augmentation or transfer learning.

On the other hand, based on the performance improvements brought by the UpSampler module's enhancement of the wavelet transform outputs, this study proposes an important perspective: any output derived from non-deep learning methods, whether traditional feature extraction techniques or other forms of signal processing, should undergo sampling through neural network layers before feature fusion. By doing so, the neural network layers can effectively enhance the representation capabilities of the original features, leading to improved overall model performance.

The UpSampler module, through the combination of convolutional layers and bidirectional long short-term memory (Bi-LSTM) networks, significantly improved the representation of low-frequency and high-frequency signals produced by the wavelet transform. This result indicates that applying neural network layers to up-sample and refine outputs from non-deep learning methods before fusion not only captures both local and global patterns within the original data but also mitigates noise and deficiencies inherent to traditional signal processing techniques. As a result, this study proposes the insight that future models prioritize applying neural network layers to sample features prior to fusion, as a step of "DeepLearningization" in the pipeline, towards outputs coming out from traditional processing techniques.

6.6 Answering

- **Q1: How can wavelet transforms improve the performance of sequential recommendation models?**

  Through the introduction of wavelet transforms, the WaveRec model was able to conduct multi-resolution analysis, effectively capturing both short-term fluctuations and long-term trends in user behavior. The results demonstrated a significant improvement in model performance, particularly on complex datasets such as MovieLens-1M.

- **Q2: What is the impact of using the UpSampler module on feature representation and recommendation accuracy?**

  By refining low-frequency and high-frequency signals through neural network layers, the UpSampler module enhanced the quality of feature representations, leading to more accurate and reliable predictions. This finding suggests that incorporating neural networks to up-sample traditional signal outputs can further boost the model's capabilities.

- **Q3: How do different feature fusion strategies affect the model's ability to capture both short-term and long-term user behavior?**

The combination of convolutional layers and Bi-LSTM networks for feature fusion successfully balanced local and global patterns in the data, demonstrating that carefully designed feature fusion mechanisms can significantly improve the model's ability to handle complex, dynamic recommendation tasks.

7 Contribution Specification

The overall framework of this research is built upon the RecBole library, which provides a unified and flexible infrastructure covering the standard pipeline of recommendation tasks, such as the implementations of data preprocessing, model training and evaluation metrics. The development process is simplified thanks to the modular design and implementation that the RecBole framework brings, upon which user are allowed to fucos on innovation and optimization on recommendation methods. In addition, the source code of CFIT4SRec was studied and learnt, and upon which the current study is developed. Building on these foundation, the contribution of this research primarily involves general exploration and experimentation with various sequence representation methods and feature fusion strategies for sequential recommendation tasks. This study designed and implemented a series of different model architectures, which incorporates cutting-edge approaches in the field of sequential recommendation, and conducted extensive experiments to evaluate the performances across three datasets. Through systematic analysis, the most effective model architectures is identified, which enhances recommendation accuracy and generalization capabilities, when coping with challenges such as data sparsity and dynamic shifts in user preferences.

8 Future studies

Due to time constraints, several promising approaches for this study have not been fully explored and implemented. Although the current contribution has achieved certain successes with the WaveRec model, there are still multiple anticipated works worth further investigation and development to fully harness the model's potential.

First, this study attempted to integrate Contrastive Learning (CL) methods into the WaveRec model. This study explored the use of WaveRec's wavelet transform module as a data augmentation method to generate different sample pairs for contrastive learning. However, despite a series of strategies were contemplated and implemented to construct sample pairs, there was not a satisfactory experimental results received. As the result, contents of CL are not included in the final version of the WaveRec model. Nevertheless, the relevant code for CL-integrated WaveRec is available in the wavereccl.py file within the git repository for future research.

Although the experiments did not yield satisfactory performance gains through contrastive learning, this study does not deny the potential of integrating contrastive learning into the WaveRec model. More time is required in the future research to delve deeper into the relevant literature and explore more optimized methods for constructing sample pairs and designing data augmentation strategies, which could eventually lead to meaningful improvements. Moreover, state space models (SSMs) is considered as another promising frontier in sequence modelling. In recent years, SSMs have made substantial progress and gained broad support for their effectiveness in handling sequential data. Compared Transformer, SSMs offer promising improvements in both computational efficiency and modelling capacity. However, due to the limited time, this research did not succeed to fully investigate and implement these emerging techniques. Nonetheless, it is believed that in future research, state space models could serve as strong alternatives to the Transformer model, offering new solutions for sequential recommendation tasks. Therefore, I will closely follow advancements in the field of SSMs.

In addition, Graph Neural Networks (GNNs) and the integration of metadata learning are two critical research area. GNNs have demonstrated the powerful capabilities in modelling complex graph-structured data, and are applied to model the relationships between users and items in sequential recommendation systems to receive significant performance improvements, leading to more personalized and accurate recommendations.

In conclusion, although some potential technological approaches were not fully implemented due to time and resource constraints, this research has delivered an effective solution for multi-frequency processing for sequential data. By exploring the integration of contrastive learning, state space models, GNNs, and metadata, future studies will continue to advance the development of the WaveRec model and related technologies, contributing new ideas and breakthroughs in the field of sequential recommendation systems.

9 Conclusion

In conclusion, this study proposes a promising solution for the use of multi-resolution analysis in sequential recommendation systems. The integration of wavelet transforms and neural network-based resolution enhancement techniques, has proven to be effective in improving recommendation accuracy. Future research will build on these contributions by further investigating methods to address data sparsity, exploring the potential of contrastive learning and state space models, and continuing to refine the WaveRec framework to better understand and predict user behavior in dynamic environments.

10 Appendices

10.1 Source Code

- **WaveRec (current study)**
  https://cseegit.essex.ac.uk/22-24-ce901-ce911-cf981-su/22-24_CE901-CE911-CF981-SU_xie_kun

- **RecBole framework**
  https://github.com/RUCAIBox/RecBole.git

- **CFIT4SRec**
  https://github.com/zhangyichi1Z/CFIT4SRec

10.2 Datasets

- **MovieLens - 1**
  https://grouplens.org/datasets/movielens/

- **Amazon 2018**
  https://nijianmo.github.io/amazon/index.html3YQ5ZraG7cNs/view?usp=drive_link

- **processed datasets**
  https://nijianmo.github.io/amazon/index.html
  Note: this is the collection of processed datasets provided by RecBole.

References

[1] T. F. Boka, Z. Niu, and R. B. Neupane. A survey of sequential recommendation systems: Techniques, evaluation, and future directions. *Information Systems*, page 102427, 2024.

[2] H. Fang, D. Zhang, Y. Shu, and G. Guo. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)*, 39(1):1–42, 2020.

[3] M. Farge et al. Wavelet transforms and their applications to turbulence. *Annual review of fluid mechanics*, 24(1):395–458, 1992.

[4] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017.

[5] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

[6] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[7] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[8] W.-C. Kang and J. McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.

[9] G. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, and A. O'Leary. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 2019.

[10] W. Liao, Q. Zhang, B. Yuan, G. Zhang, and J. Lu. Heterogeneous multidomain recommender system through adversarial learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8965–8977, 2022.

[11] J. Lyu, J. Yang, J. Kim, W. Lim, W. Ahn, D. Kang, M. Kim, and N. S. Kim. Multi-resolution sequence aggregation and model-agnostic framework for time-series forecasting. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[12] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52, 2015.

[13] R. Qiu, Z. Huang, H. Yin, and Z. Wang. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 813–823, 2022.

[14] M. Quadrana, P. Cremonesi, and D. Jannach. Sequence-aware recommender systems. *ACM computing surveys (CSUR)*, 51(4):1–36, 2018.

[15] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820, 2010.

[16] M. M. Richter, S. Paul, V. Këpuska, and M. Silaghi. *Signal processing and machine learning with applications.* Springer, 2022.

[17] A. Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[18] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.

[19] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[20] J. Tang and K. Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 565–573, 2018.

[21] A. Vaswani. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[22] S. Wang, Q. Zhang, L. Hu, X. Zhang, Y. Wang, and C. Aggarwal. Sequential/session-based recommendations: Challenges, approaches, applications and opportunities. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 3425–3428, 2022.

[23] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019.

[24] L. Wu, X. He, X. Wang, K. Zhang, and M. Wang. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4425–4445, 2022.

[25] L. Xu, Z. Tian, G. Zhang, J. Zhang, L. Wang, B. Zheng, Y. Li, J. Tang, Z. Zhang, Y. Hou, et al. Towards a more user-friendly and easy-to-use benchmark library for recommender systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2837–2847, 2023.

[26] Z.-Q. J. Xu, Y. Zhang, T. Luo, Y. Xiao, and Z. Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

[27] D. Zhang and D. Zhang. Wavelet transform. *Fundamentals of image data mining: Analysis, Features, Classification and Retrieval*, pages 35–44, 2019.

[28] M. Zhang, S. Wu, X. Yu, Q. Liu, and L. Wang. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4741–4753, 2022.

[29] Y. Zhang, G. Yin, and Y. Dong. Contrastive learning with frequency-domain interest trends for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 141–150, 2023.

[30] C. Zhao, J. You, X. Wen, and X. Li. Deep bi-lstm networks for sequential recommendation. *Entropy*, 22(8):870, 2020.

[31] W. X. Zhao, S. Mu, Y. Hou, Z. Lin, Y. Chen, X. Pan, K. Li, Y. Lu, H. Wang, C. Tian, et al. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *proceedings of the 30th acm international conference on information & knowledge management*, pages 4653–4664, 2021.

[32] F. Zhu, Y. Wang, C. Chen, J. Zhou, L. Li, and G. Liu. Cross-domain recommendation: challenges, progress, and prospects. *arXiv preprint arXiv:2103.01696*, 2021.