Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 18, No. 2, 2025

A Sequential Recommendation Model Based on Convolutional Neural Networks and State Space Models

Chenfei Feng, Haitao Wang and Yuguo Wang

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454150, China

Abstract: Sequential recommendation aims to predict users' future preferences by analyzing their historical behavior sequences. In recent years, deep learning techniques have been widely applied to sequential recommendation tasks, achieving remarkable improvements in recommendation accuracy. However, existing methods often suffer from structural imbalance in modeling the rapidly changing short-term interests and the relatively stable long-term preferences of users. Moreover, these methods face limitations such as high computational costs and low inference efficiency when dealing with long behavior sequences. To address these challenges, this paper proposes a novel sequential recommendation model named CNN-Mamba, which integrates Convolutional Neural Networks (CNN) with State Space Models (SSM). Specifically, the CNN component leverages local receptive fields to efficiently extract short-term interest features from recent user interactions, thereby enhancing the model's ability to capture local behavior patterns. Meanwhile, the SSM component is introduced as a long-term interest modeling module to capture global dependencies within long sequences. Furthermore, an adaptive fusion layer is designed to dynamically integrate the short- and long-term modeling outputs, thereby improving the model's generalization ability. In addition, the implicit recurrence mechanism in the state space model effectively reduces computational complexity and enhances the efficiency of long-sequence modeling. Experimental results on three real-world datasets demonstrate that the proposed CNN-Mamba model outperforms state-of-the-art baselines in both recommendation accuracy and inference efficiency, validating its effectiveness and practicality.

Keywords: State Space Model; Convolutional Neural Network; Sequential Recommendation; Long- and Short-Term Interest Modeling.

1. Introduction

In today's era of information overload, recommendation systems have become a key technology for delivering personalized content [1]. As an important branch of recommendation systems, sequential recommendation predicts items that users may be interested in by analyzing their historical behavior sequences, thereby enhancing user experience and recommendation performance. Unlike traditional static recommendation methods, sequential recommendation focuses on the dynamic evolution of user interests over time and captures temporal dependency patterns within user behaviors to improve the accuracy of personalized recommendations [2]. However, due to the complexity of user interests and the highly dynamic nature of behavioral sequences, efficiently modeling short-term interest fluctuations and long-term interest stability remains a significant challenge in sequential recommendation. Section Headings

At present, sequential recommendation has evolved from traditional statistical methods to deep learning-based approaches. Early methods primarily relied on Markov chains and matrix factorization. These methods were able to mine latent features from user–item interaction matrices but did not explicitly account for the order of interactions, thus failing to capture the temporal changes in user preferences. Another class of traditional approaches employed Markov chains to model sequential dependencies. Rendle et al. [3] proposed the FPMC model, which combines matrix factorization with a first-order Markov chain to jointly model users' long-term and short-term preferences. Compared with traditional collaborative filtering methods, FPMC achieved superior performance in sequential prediction tasks. However, it still has limitations: Markov chain-based approaches rely heavily on the most recent interactions and struggle to capture global information. In addition, static models such as matrix factorization ignore the temporal evolution of user interests; when user preferences change rapidly, these models cannot update accordingly, leading to recommendations that lack timeliness and fail to adapt to dynamic user interests.

The rise of deep learning has injected new vitality into sequential recommendation. In particular, recurrent neural networks (RNNs) and their variants have been widely adopted for handling long sequences. Hidasi et al. [4] introduced the classical GRU4Rec model, which was the first to apply gated recurrent units (GRUs) to session-based recommendation tasks, demonstrating the powerful capability of RNNs in modeling user click sequences. RNN-based approaches can encode entire historical sequences into hidden states, theoretically capturing dependencies of arbitrary length and extracting richer long-term semantic information compared to Markov chains. In data-rich scenarios, RNN-based sequence models often significantly outperform simple Markov chain models. However, RNN-based models also have inherent drawbacks. First, the recursive computation pattern of RNNs makes parallelization difficult, as the computation of subsequent states depends on the results of previous steps. Second, RNN models contain a large number of parameters and complex training processes; under data-sparse conditions, they are prone to overfitting and gradient vanishing problems, requiring large amounts of training data to fully learn sequential patterns. Consequently, although recurrent networks enhance sequence modeling capabilities, their performance and efficiency remain limited in tasks involving long sequences, high sparsity, or strict real-time requirements.

Convolutional neural networks (CNNs), with their parallel

computing capabilities and local receptive fields, have emerged as a promising direction for sequential recommendation. Unlike RNNs that process sequences step by step, CNNs can apply convolution operations to fixed-length recent interaction segments to capture local patterns, thereby effectively modeling users' short-term dynamic interests. Tang et al. [5] proposed the Caser model, a representative of this approach. Caser stacks embeddings of the user's most recent L interactions into a two-dimensional "time–latent space" matrix, treating it as an image, and applies convolutional filters over this image to extract local sequential patterns. Through horizontal and vertical convolutions, Caser can simultaneously characterize users' general preferences and short-term sequential preferences within a unified network. Experiments have shown that Caser consistently outperformed other sequential recommendation algorithms on multiple public datasets. However, models purely based on convolutional neural networks also face certain limitations. Since the length L of the convolutional window is usually fixed, the range of historical interactions directly utilized by the model is limited; when user interest evolution involves long-term dependencies far beyond the window length, the model may fail to adequately capture earlier preference changes. Therefore, how to leverage the efficiency of CNNs in capturing short-term patterns while enhancing the modeling of long-term interests remains a key problem to be addressed in CNN-based sequential models.

Recently, the success of Transformers and other self-attention mechanisms in sequence modeling has drawn significant attention in the recommendation field. Kang et al. [6] pioneered the introduction of the Transformer encoder into sequential recommendation with the SASRec model, which leverages self-attention mechanisms to model user behavior sequences. SASRec captures long-range dependencies similar to RNNs while simultaneously emphasizing recent important behaviors like Markov chains, making the model highly adaptable to datasets with varying levels of sparsity. Experimental results on multiple benchmark datasets demonstrate that SASRec significantly outperforms previous sequential recommendation algorithms.

Building on SASRec, Sun et al. [7] proposed BERT4Rec, which incorporates a bidirectional Transformer encoder to fully utilize bidirectional contextual information. Unlike traditional unidirectional models that rely solely on historical information to predict the next item, BERT4Rec randomly masks portions of sequences during training, allowing the model to infer masked items by jointly leveraging both left and right contexts of the target item. This pretraining paradigm avoids the issue of target information leakage in bidirectional modeling and substantially enhances the precision of sequence representations. Experimental results show that BERT4Rec achieved superior recommendation performance on multiple public datasets compared to previous unidirectional sequential models. However, these models typically have a large number of parameters and high computational costs, making them less suitable for scenarios that demand high real-time performance or are constrained by limited computational resources. Thus, improving the efficiency of these models and achieving a more robust integration of short-term and long-term interest modeling remain significant challenges.

Unlike Transformer-based architectures with attention mechanisms, state space models (SSMs) have recently emerged as a promising paradigm in sequence modeling tasks. Gu et al. introduced the Mamba model [8], which employs a selective state space mechanism to eliminate the sequential processing bottleneck inherent in traditional recurrent structures, achieving linear time complexity while effectively capturing long-range dependencies. Mamba has demonstrated performance comparable to or even surpassing that of Transformers across multiple long-sequence tasks, such as natural language processing, while delivering better inference efficiency and substantial resource savings. Subsequently, Dao et al. proposed Mamba2 [9], improving the state transition structure and feature representation, thereby enhancing the stability and generalization ability of the model in complex sequential tasks.

Given the strong memory retention and global modeling capabilities of state space models in long-term interest modeling, this paper draws inspiration from the structural design of the Mamba family of models, using it as the long-term interest modeling module. Combined with the local feature extraction strength of CNNs in capturing short-term interests, we design a hybrid personalized recommendation model, CNN-Mamba, to enhance the representational power and performance of sequential recommendation in dynamic environments. Despite the significant breakthroughs achieved by the Mamba family in long-sequence modeling, they still face limitations. Since SSMs mainly rely on linear dynamic systems to model sequential information, they exhibit limited capability in capturing complex local nonlinear interest variations. This issue becomes particularly pronounced in recommendation scenarios where users' short-term interests fluctuate rapidly, making it challenging to precisely capture fine-grained behavioral patterns.

To address these challenges, this paper proposes a novel sequential recommendation model named CNN-Mamba. This model exploits the strength of convolutional neural networks in extracting local patterns to model short-term interest dynamics from recent user behaviors. Meanwhile, it leverages the state space model's capability in global dependency modeling to capture the evolving trends of users' long-term preferences. By integrating the local feature extraction ability of CNNs with the global sequence modeling power of SSMs, CNN-Mamba simultaneously models both short-term and long-term user interests, thereby improving the accuracy and diversity of sequential recommendations.

The key contributions of this paper can be summarized as follows:

1). We propose a hybrid sequential recommendation model that integrates CNN and the Mamba state space model. The architecture includes an embedding layer, the CNN-Mamba module, an adaptive fusion layer, and a prediction layer. This design enables the dynamic fusion of long-term and short-term interests, fully leveraging CNN's efficiency in short-term interest modeling while capitalizing on Mamba's capacity for global dependency modeling in long-term interest modeling.

2). We design a CNN-Mamba fusion strategy that achieves collaborative optimization of short-term and long-term interests, overcoming the information segmentation issue present in traditional models. This allows short-term interests to effectively influence long-term modeling while enabling long-term interests to stabilize the representation of short-term interests, thereby enhancing recommendation quality.

3). The model optimizes computational efficiency and improves recommendation effectiveness. Compared with Transformer-based architectures, CNN-Mamba reduces

computational complexity, making the model more scalable for long-sequence modeling tasks.

4). Extensive experiments conducted on three real-world datasets demonstrate that CNN-Mamba outperforms baseline models across multiple evaluation metrics.

2. Related Work

2.1. Sequential Recommendation

Sequential recommendation is a method that predicts users' future preferences by leveraging their behavioral sequences. Unlike traditional collaborative filtering methods, which are based solely on static user–item interaction data, sequential recommendation emphasizes temporal patterns, dynamically modeling user behavior to identify the evolution of user interests and generate more personalized recommendations. Early sequential recommendation methods were primarily based on Markov chains and collaborative filtering. Markov chains [10] build transition matrices to capture users' short-term behaviors and are well suited for modeling transitions between adjacent behaviors. However, because they rely only on the most recent interactions, they struggle to capture long-range dependencies within sequences. Collaborative filtering algorithms [11], which recommend items by identifying similar users or items, also perform poorly when handling the temporal dependencies and dynamic variations inherent in sequential data. With the growth of data volume, sequential recommendation models have increasingly evolved toward deep learning methods. Deep learning models offer strong capabilities for capturing complex patterns, allowing for more flexible modeling of both short-term and long-term user preferences. Against this backdrop, convolutional neural networks (CNNs) have emerged as an important direction for sequential recommendation research due to their advantages in short-term interest modeling.

2.2. CNN-based Sequential Recommendation Algorithms

Convolutional neural networks, known for their efficient local feature extraction capabilities [12], are widely used in sequential recommendation, particularly for short-term interest modeling. By leveraging local receptive fields, CNNs can efficiently extract localized patterns from user behavior sequences and progressively build more complex feature representations through stacked convolutional layers. Compared with traditional recurrent neural networks (RNNs), CNNs can process sequence data in parallel, avoiding the computational bottlenecks that occur when handling long sequences in traditional networks.

The Caser model is a classic CNN-based sequential recommendation model. Caser uses horizontal and vertical convolutions to capture user behavioral patterns along the temporal axis and interaction features between users and items, thereby enabling precise modeling of short-term interests [13]. Additionally, the S3Rec model [14] builds upon Caser by introducing multi-scale convolutional kernels, which further enhance the ability to extract multi-level features. This enables CNNs to simultaneously capture short-term fluctuations and long-term trends within user sequences. The multi-scale design allows S3Rec to excel in modeling complex user behaviors, as it considers changes in user interests across multiple scales, improving both accuracy and robustness.

However, despite their advantages in short-term interest modeling, CNN-based methods also have significant limitations. First, the local receptive fields of CNNs restrict their ability to model long sequences and capture global dependencies effectively. In addition, CNNs model temporal dependencies in an indirect manner; while stacking more convolutional layers can expand the receptive field, such methods are still limited in capturing complex temporal dependencies and cannot fully reflect the evolving nature of user interests.

2.3. Attention-based Sequential Recommendation

Attention mechanisms dynamically assign weights to each time step of user behaviors, enabling more flexible modeling of evolving user interests. By adaptively distributing attention weights, attention mechanisms capture global dependencies across time steps, providing stronger capabilities for modeling long sequences [15].

Kang et al. [6] introduced the SASRec model, which adopts a self-attention-based architecture. Unlike traditional recurrent neural networks, SASRec calculates relationships between each position in the sequence and all others, effectively capturing dynamic changes in short-term dependencies. Through stacked self-attention layers, SASRec incrementally extracts user interest features and produces accurate recommendations. This model also effectively avoids gradient vanishing and information loss issues that are common in RNNs. Sun et al. [7] further proposed the BERT4Rec model, which employs a bidirectional self-attention architecture. This structure enables the model to consider both forward and backward dependencies in user behavior sequences, thereby capturing complex dependencies comprehensively. BERT4Rec not only improves long-sequence modeling capabilities but also adopts a pretraining-and-finetuning framework. This allows the model to be pretrained on large-scale unlabeled datasets and then fine-tuned on specific recommendation tasks, significantly enhancing generalization performance. However, despite their strong ability in long-sequence modeling, attention-based models still face limitations in local feature extraction and computational complexity [16]. First, the computational complexity of attention mechanisms is high, especially when processing very long sequences, where the complexity of self-attention reaches $O(n^2)$. This can create computational bottlenecks and hinder performance on large-scale datasets.

To further improve efficiency and representational capacity in long-sequence modeling, Gu et al. [8] proposed a state space modeling framework. Among these, the Mamba model has gained significant attention for its linear-time complexity and global modeling capability. By constructing selective state space representations, the Mamba model maintains the advantages of parallel computation while effectively capturing long-range dependencies, thus overcoming the high complexity and low efficiency issues of attention mechanisms in long-sequence modeling. However, the Mamba model still shows limitations in capturing users' recent preferences.

To address these existing challenges, this study proposes a hybrid model that integrates convolutional neural networks and self-attention mechanisms, while further improving the architecture by introducing state space models. In this design, CNNs are employed to model users' recent preferences, while the state space model is leveraged to capture long-term preferences, enabling more comprehensive sequential recommendation.

3. CNN-Mamba

The framework of the proposed model is illustrated in Figure 1, where the input sequence represents the user's historical interaction records, denoted as $x_1, x_2, \ldots, x_i$, with each element x_i indicating the user's interaction behavior at time step i. The model mainly consists of the following components: an embedding layer, a short-term interest modeling module, a long-term interest modeling module, and an adaptive fusion layer. In the embedding layer, user behavior sequences are mapped into dense vector .

In this study, U represents the set of all users, where |U| denotes the number of users, and |V| denotes the number of items. The interaction sequence of a user u is denoted as $S_u = \{v_1, v_2, \ldots, v_n\}$, arranged in chronological order, where each $v_i \in V$ indicates an item the user u interacted with at time step i. The objective of sequential recommendation is to predict the next item with which user u will interact at time step n+1. Sequential recommendation achieves this by learning the conditional probability.$P(v \mid S_u^{1:n}; \theta)$,where θ represents the model parameters, and $v \in V$ is any candidate item in the item set. This conditional probability distribution characterizes the likelihood that the user will interact with each candidate item given their historical interaction behaviors. By maximizing this conditional probability distribution, the model produces optimal recommendation results, generating the most probable next-item recommendation for the user.

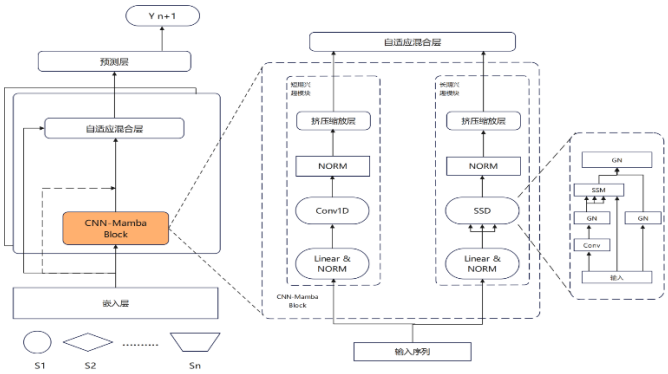

Figure 1. CNN-Mamba

3.1. Embedding Layer

The item embedding laye $E_{item} \in \mathbb{R}^{\mathcal{V}}$ r maps each interacted item $v_i^{(u)} \in \mathcal{V}$ into a ddd-dimensional dense vecto $e_i^{(u)} \in \mathbb{R}$ r. In this way, the item embedding layer effectively reduces the sparsity of the input data and enhances the expressiveness of the features. The positional $P_{pos} \in \mathbb{R}$ embedding layer provides temporal position information for the user's behavior sequence. By generating learnable embeddings with the same dimensionality as the item embeddings, the positional embedding layer enables the model to capture temporal patterns within the sequence. Finally, by summing the item embeddings and positional embeddings, the embedding layer transforms the user's historical behavior sequence $H_{(u)}^{l-1} = \{h_1, h_2, \ldots, h_T\}$ into a dense feature matrix $\mathcal{E}$ ^{1}).

3.2. CNN-Mamba

3.2.1. Short-Term Interest Module

The short-term interest module aims to capture user behavior sequences through convolution operations and dynamically adjust the importance of features by incorporating the attention mechanism. The input features of the module are as follows: $H_{(u)}^{l-1}$, The input features of the first layer $H_{(u)}^{0}$ are the feature representations generated by the embedding layer: $H_{(u)}^{0} = \mathcal{E}$, where $S_{(u)}^{1:n}$ denotes the historical behavior sequence of user u, and $\mathcal{E}$ represents the embedding operation, which includes the combination of item embedding and positional embedding. The features generated by the embedding layer serve as the input of the first layer, while the input features of the subsequent layers are provided by the output of the previous layer."In the short-term interest module, is first $H_{(u)}^{l-1}$ subjected to a linear transformation and the feature distribution is stabilized through layer normalization, with the formula as follows:

$$\mathcal{E} \quad) = \mathrm{LN}(\mathrm{Dpt}(\mathrm{Linear}_{en}(H_{(u)}^{l-1}))) \tag{1}$$

Linear denotes a fully connected linear transformation, Dpt denotes the Dropout operation, and LN refers to the layer normalization operation, which is used to improve the training stability of the model.

The normalized features $\mathcal{E}$) are extracted through multiple one-dimensional convolution operations (Conv1D) with kernel size k = 3 to obtain local features. For time step j, the output of the m-th convolution kernel is given by the following formula:

$$C_j^m = \Phi\left(\mathcal{E} \quad)_{[j:j+k-1]} \cdot F^m\right) \tag{2}$$

$\mathcal{E}$ $)_{[j:j+k-1]}$ denotes the features within the sliding window from position j to j+k-1; F^m denotes the m-th convolution kernel; Φ is the activation function.

At each position j, the outputs of the convolution kernels are integrated through concatenation to generate the local feature representation, as shown in the following formula:

$$\mathrm{Conv}_{mid}^{k=3}(\mathcal{E} \quad))_j = \mathrm{Concat}(C_j^1, \ldots, C_j^m) \in \mathbb{R} \tag{3}$$

Here, Concat indicates concatenating the results of multiple convolution kernels along the feature dimension. Finally, the convolution results are concatenated across the entire sequence into a holistic representation and further optimized through dropout and layer normalization, as shown in the following formula:

To further emphasize key features, the convolution results are dynamically weighted through a squeeze-attention mechanism, and the final short-term interest feature representation is obtained as follows:

$$\mathrm{Conv}_{mid}^{k=3}(\mathcal{E} \quad \mathcal{E} \tag{4}$$

$$L_{\mathrm{Conv}} = \mathrm{SEAtt}(\mathrm{Conv}_{mid}^{k=3}(\mathcal{E} \quad \mathcal{E} \tag{5}$$

$\otimes$ denotes the element-wise multiplication operation, and represents the attention mechanism that generates dynamic weights using the global context.

3.2.2. Long-Term Interest Module

Since long-term interest modeling in recommendation systems requires handling the global dependencies of users' historical behavior sequences, the traditional self-attention mechanism captures the correlations among features through three matrices: Query (Q), Key (K), and Value (V). However, the Softmax operation in self-attention has high computational complexity, making it difficult to meet the modeling requirements for long sequence data. In contrast, the structured state space model achieves efficient modeling through linearized computation.

By combining the linear state space modeling of SSD, the input features establish causal dependencies between time steps via a mask matrix, generating global feature representations, as shown in the following formula:

$$\mathrm{SSD}(\mathcal{E} \qquad \mathcal{L} \ \mathcal{E} \tag{6}$$

Here, is the mask matrix, which is used to simulate the state transition relationships, ensuring that the current time step depends only on the previous time steps and preventing the leakage of future information. The matrix $\mathcal{L}$ is defined as follows:

$$\mathcal{L} \quad \begin{cases} \prod_{k}^{i} a_k, & \text{if } i \geq j, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Next, the input features $\mathcal{E} \qquad ^{1})$ are further mapped into the $Q=\mathcal{E} \qquad ^{1})W^{Q}$, $K=\mathcal{E} \qquad ^{1})W^{K}$, and $V=\mathcal{E} \qquad ^{1})W^{V}$. Based on the SSD attention mechanism, linearized computation is used to replace the traditional Softmax operation, as expressed in the following formula:

$$\mathrm{Att}(Q,K,V)=\mathcal{L} \qquad , \ , \tag{8}$$

$Q\cdot K^{\top}$ is used for the dot-product operation between the Query and Key to measure the similarity between time steps; V is the Value matrix, through which the final global feature representation is obtained by weighted computation. Finally, by incorporating the dynamic weighting mechanism SEAtt, the generated global feature representation is further dynamically adjusted, as shown in the following formula:

$$G_{long}=\mathrm{SEAtt}(\mathrm{SSD}(\mathcal{E} \qquad \mathcal{E} \tag{9}$$

Here, SEAtt refers to the squeeze-and-excitation attention mechanism, which enhances the adaptability of feature representations through channel-wise weighted adjustment.

3.3. Adaptive Fusion Layer

The adaptive fusion layer achieves personalized modeling of user behavior sequences by dynamically integrating short-term and long-term interest features. First, the output features of the historical behavior $H_{(u)}^{l-1}$ sequence are subjected to global average pooling to obtain the global representation, as shown in the following formula $\mathcal{P} \qquad)$:

$$\mathcal{P} \qquad)=\frac{1}{n}\sum_{t=1}^{n}\mathrm{Out}(H_{(u),t}^{l-1}) \tag{10}$$

Subsequently, the global representation is passed through a linear transformation and a sigmoid activation function to compute the adaptive fusion weight $\mathrm{Ada}(u)$, as shown in the following formula:

$$\mathrm{Ada}(u)(H_{(u)}^{l-1})=\Omega\left(\mathrm{Linear}_{mid}\left(\mathcal{P} \qquad)\right)\right) \tag{11}$$

Ω is the activation function, which is used to map the weights into the range [0, 1]. Finally, the adaptive weights $\mathrm{Ada}(u)$ are applied to the short-term interest features L_{Conv} and long-term interest features G_{long} for weighted fusion, producing the final feature representation, as shown in the following formula:

$$M_{long}^{\mathrm{final}}=\mathrm{Ada}(u)(H_{(u)}^{l-1})\cdot M_{\mathrm{Conv}}^{long}+\left(1-\mathrm{Ada}(u)(H_{(u)}^{l-1})\right)\cdot G_{long} \tag{12}$$

$1-\mathrm{Ada}(u)(H_{(u)}^{l-1})$ is the complementary value, representing the weight of the long-term interest features. When $\mathrm{Ada}(u)$ approaches 0, the weight of long-term interest features is higher; when $\mathrm{Ada}(u)$ approaches 1, the weight of short-term interest features is higher.

3.4. Prediction Layer

After aggregating the information from all positions across the L layers, we obtain the final output of all items in the input sequence $H_{(u)}^{L}$. Then, the feature at the n-th time step is taken $H_{(u)}^{L}[n]$ and mapped into a high-dimensional space through a linear transformation and an activation function. Subsequently, the mapped feature is dot-multiplied E_{table} with the shared item embedding table b^{O} and added with a bias term to obtain the scores of all candidate items, as shown in the following formula:

$$\mathcal{P} \qquad)=\sigma\left(\delta\left(H_{(u)}^{L}[n]W^{\mathrm{Pred}}+b^{\mathrm{Pred}}\right)E_{\mathrm{table}}^{T}+b^{O}\right) \tag{13}$$

σ denotes the softmax function, which normalizes the item scores into a probability distribution, ultimately outputting the probability of the item that the user is most likely to choose in the next step.

4. Experiments and Results Analysis

To validate the effectiveness of the proposed model, relevant experiments were conducted. In this section, we first introduce the datasets used in the experiments and the data preprocessing methods; then, we describe the baseline models for comparison, the evaluation metrics, and the related experimental parameters.

4.1. Datasets

In this paper, the performance of the CNN-Mamba model is evaluated on three publicly available real-world recommendation system datasets: Amazon Toys, Sports, and Beauty. These datasets are derived from large-scale product review data on the Amazon platform and are widely used in recommendation system research. The datasets are divided according to the top-level categories of products, specifically 'Toys and Games,' 'Sports and Outdoors,' and 'Beauty,' covering user purchase behavior records in different domains. The data preprocessing method in this paper is consistent with previous studies. Specifically, each rating or review is regarded as evidence of a user–item interaction, and each

dataset is converted into an implicit dataset. Then, interactions are grouped by user ID and sorted according to timestamps to form a sequence for each user.

Table 1. Dataset Statistics

Dataset	Inter	Users	Items	Len_U	Len_I	Sparsity
Toys	167597	19411	11926	8.5	14.2	99.94%
Sports	296338	35599	18456	8.2	16.2	99.91%
Beauty	198500	22360	12002	8.8	16.5	99.93%

4.2. Baseline

To demonstrate the effectiveness of the proposed model, the CNN-Mamba model was compared with the following baseline models:

GRU4Rec: A recommendation system based on recurrent neural networks that captures the dynamics of user session behaviors through gated recurrent units (GRU). It is a session-based recommendation model capable of handling the temporal dependencies in user behavior sequences.

Caser: A model that employs horizontal and vertical convolutional layers of convolutional neural networks (CNN) to capture dynamic features in user behavior sequences and extends user interests into multi-layer perceptrons to generate the final recommendation results.

NextItNet: A generative network constructed by stacking convolutional layers to capture deep patterns in user sequential behaviors. It can efficiently perform next-item recommendation and exhibits good scalability.

SASRec: A model that adopts the Transformer architecture, leveraging the self-attention mechanism to model long-range dependencies in user behavior sequences, making it suitable for long-term interest modeling.

BERT4Rec: A model based on the Transformer that introduces a bidirectional modeling mechanism to simulate the contextual information of user behaviors, thereby enhancing the global modeling capability of user behavior sequences.

GCSAN [17]: A model that combines graph contextual information with the self-attention mechanism, enhancing the modeling of graph-structured data by capturing the relationships between users and items in session-based recommendation.

SINE [18]: A sparse interest network designed to address the sparsity of user interest distributions, which optimizes sequential recommendation performance through an adaptive mechanism.

4.3. Evaluation Metrics

This paper adopts two popular performance evaluation metrics in recommendation systems to measure the effectiveness of recommendation algorithms: Recall@K and NDCG@K (Normalized Discounted Cumulative Gain).

Recall@K: This metric is used to evaluate the accuracy of recommendation algorithms. It is defined as the ratio of the number of positive samples T(u) in the recommendation list to the total number of samples (N).

$$\mathrm{Recall@K} = \frac{T(u)}{N}$$

NDCG@K (Normalized Discounted Cumulative Gain): This metric not only measures the accuracy of the recommendation system but also evaluates the rationality of the ranking in the recommendation results, emphasizing the contribution of correctly recommended items at higher ranks.

$$\mathrm{NDCG@K} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\log_2(\mathrm{rank}+1)}$$

In this paper, the values of K are set to 1, 5, and 10 for comparison with the baseline methods. The higher the values of these evaluation metrics, the better the prediction results.

4.4. Experimental Details

The open-source framework RecBole was used to evaluate the baseline models, and grid search was adopted to optimize the hyperparameters of each model. For each baseline algorithm, the following parameter settings were applied: hidden layer dimension h = {16, 32, 64, 128}, convolution kernel size k = {3, 5, 7, 9, 11}, activation functions selected from $\Phi = \{\mathrm{Re}LU, GELU, \mathrm{Sigmoid}\}$, dropout rate for the attention mechanism set to $d_2 \in [1e^{-1}, 1e^{-1}]$, and learning rate range set to $\eta \in [1e^{-8}, 1e^{-1}]$. All models used the Adam optimizer with hyperparameters $\beta_2 = 0.999$, and L2 regularization ($\lambda = 0.001$). The learning rate was initialized at $\eta = 0.001$ and decayed during training. The training epochs were set to 20. In terms of data processing, all datasets were uniformly set with a maximum sequence length of N = 50, and padding was used for sequence completion. To ensure consistency of the experiments, all models adopted the same training settings.

4.5. Overall Performance Comparison

Table 2. Performance Comparison between CNN-Mamba and Baseline Models

	Mrtric	(a)	(b)	(c)	(d)	(e)	(g)	(h)	(j)	(k)
		GRU4Rec	Caser	NextItNet	SASRec	BERT4Rec	GCSAN	SINE	MAmba	improve
Dataset	Recall@5	0.3536	0.3145	0.2894	0.3960	0.3042	0.3531	0.3679	**0.4133**	+4.40%
	Recall@10	0.4593	0.4232	0.3955	0.4880	0.4097	0.4468	0.4663	**0.5080**	+4.11%
	NDCG@5	0.2631	0.2237	0.2020	0.3081	0.2196	0.2719	0.2750	**0.3188**	+3.94%
	NDCG@10	0.2971	0.2589	0.2361	0.3382	0.2536	0.3020	0.3068	**0.3510**	+3.97%
	Recall@5	0.3689	0.3365	0.3365	0.4015	0.3178	0.3629	0.3867	**0.4201**	+4.78%
	Recall@10	0.4687	0.4372	0.4371	0.4940	0.4165	0.4555	0.4807	**0.5170**	+4.52%
	NDCG@5	0.2808	0.2469	0.2469	0.3127	0.2350	0.2770	0.2934	**0.3239**	+3.62%
	NDCG@10	0.3130	0.2793	0.2788	0.3426	0.2666	0.3066	0.3238	**0.3551**	+3.76%
	Recall@5	0.3388	0.3174	0.3275	0.3768	0.3038	0.3304	0.3752	**0.4090**	+8.82%
	Recall@10	0.4610	0.4418	0.4580	0.4992	0.4270	0.4459	0.4931	**0.5409**	+8.68%
	NDCG@5	0.2438	0.2226	0.2291	0.2777	0.2128	0.2368	0.2612	**0.3010**	+8.68%
	NDCG@10	0.2829	0.2628	0.2708	0.3172	0.2525	0.2737	0.3008	**0.3440**	+8.56%

The experimental results are shown in Table 2, where the bold numbers indicate the best results for each metric, and the underlined numbers indicate the second-best results. The results demonstrate that the proposed model achieves

improvements in the Recall@5, Recall@10, NDCG@5, and NDCG@10 metrics across the three datasets (Toys, Beauty, and Sports).

Among the baseline models, Caser and SASRec represent methods based on convolutional neural networks and self-attention mechanisms, respectively. Caser introduces a convolutional structure on top of GRU4Rec, effectively capturing local interests and performing particularly well in short-term interest modeling. SASRec, on the other hand, leverages the self-attention mechanism to effectively capture global dependencies in user behaviors, showing significant advantages especially with long sequence data. Compared with the RNN-based GRU4Rec and CNN-based Caser, SASRec and BERT4Rec, which are based on self-attention, achieve superior performance in capturing global interests. Methods based on graph neural networks (GCSAN and HGNN) also perform well in modeling complex relationships between users and items, as they can capture interdependencies among items; however, they are relatively weaker in modeling local interests.

The CNN-Mamba model proposed in this paper effectively models both short-term and long-term user interests by combining convolutional neural networks with state space models. Thanks to the convolutional layers for capturing local interests and the state space model for global dependency modeling, CNN-Mamba provides more accurate and diverse recommendation results.

4.6. Ablation Study Analysis

To investigate the impact of the key components of CNN-Mamba on recommendation performance, two different model variants were designed: CNN-only and SSM-only, with NDCG@10 chosen as the primary evaluation metric. CNN-only refers to removing the state space model (SSM) from the complete model while retaining only the CNN structure, whereas SSM-only refers to removing the CNN structure while retaining only the SSM.

The performance of these different variants on the Toys, Beauty, and Sports datasets is shown in Table 3.

Table 3. Impact of Model Components on Recommendation Performance

Model	NDCG@10		
	Toys	Beauty	Sports
CNN-only	0.3430	0,3465	0.3340
SSM-only	0.3440	0.3492	0.3360
CNN-Mamba	0.3510	0.3551	0.3440

Overall, the proposed CNN-Mamba structure outperforms other variants across all datasets, indicating that the combination of CNN and SSM is the key to the model's success. Removing either component significantly degrades recommendation performance, demonstrating the complementarity of short-term and long-term interest modeling. By integrating the local feature extraction capability of CNN with the global dependency modeling capability of SSM, CNN-Mamba is able to handle both short-term and long-term interests simultaneously, thereby achieving superior performance across multiple datasets.

4.7. Imapct of Key Hyperparameters

In this section, experiments are conducted on the Beauty dataset, using NDCG@10 and Recall@10 as evaluation metrics to analyze the impact of convolution kernel size K, activation functions, and hidden layer dimensions d_{model} on the experimental results.

4.7.1. Impact of Convolution Kernel Size k

Keeping other parameters unchanged, the convolution kernel sizes were set to [3, 5, 7, 9, 11], and experiments were conducted under different kernel sizes using the Beauty dataset to obtain the NDCG@10 values. The experimental results are shown in Figure 2. As can be seen from Figure 2, as the convolution kernel size increases, the model performance exhibits a fluctuating trend, with the best performance achieved at k = 7. Increasing the kernel size allows the model to capture more local features, but when the kernel size continues to grow, the performance improvement becomes marginal, and at k = 11 a slight decline is observed. This indicates that, on the Beauty dataset, an excessively large convolution kernel leads to overfitting, and thus a moderate kernel size can achieve better performance.

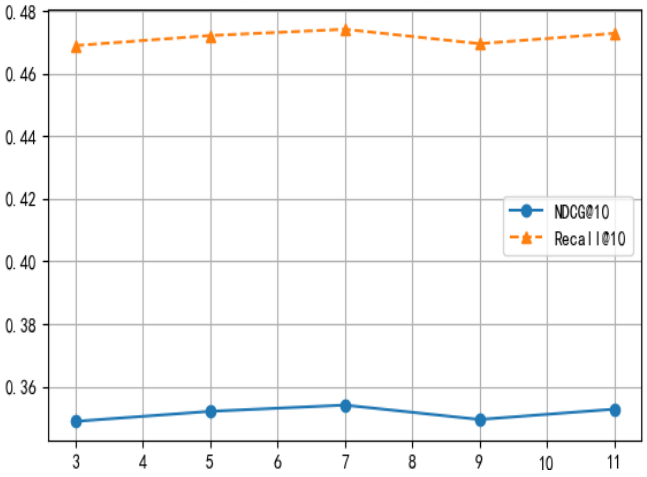

Figure 2. Analysis of Kernel Size k

4.7.2. Activation Function

The activation functions were set to [GELU, ReLU, Swish, TanH, Sigmoid], and the results are shown in Figure 3. As the activation functions varied, the recommendation performance of the model exhibited different trends. When using the GELU activation function, the model achieved the best performance, with the NDCG@10 metric reaching the highest value. As the activation function switched from ReLU to other forms, the evaluation metrics of the model declined, indicating that the choice of an appropriate activation function is also crucial to the model's final performance.

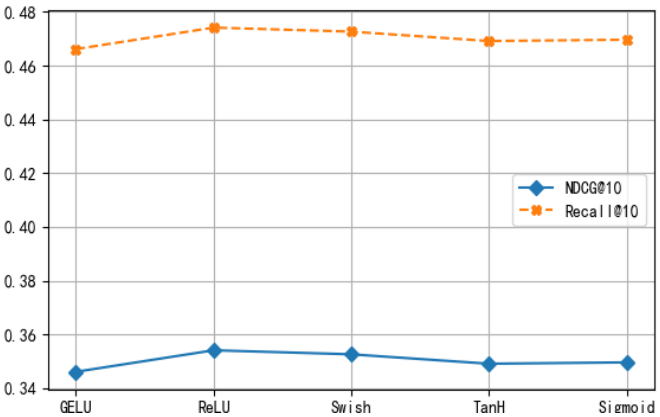

Figure 3. Analysis of Different Activation Functions

4.7.3. Hidden Layer Dimensions

The hidden layer dimensions were set to [16, 32, 64, 96,

128], and the results are shown in Figure 4. As the hidden layer dimension h increases, the model performance gradually declines, with a particularly sharp drop observed when the hidden layer dimension exceeds 32. This indicates that, in this experiment, larger hidden layer dimensions lead to a decrease in performance, showing a negative effect. This trend suggests that excessively large hidden layer dimensions make the model more complex and prone to overfitting. Therefore, selecting a moderate dimension can effectively improve the model's representation capability.

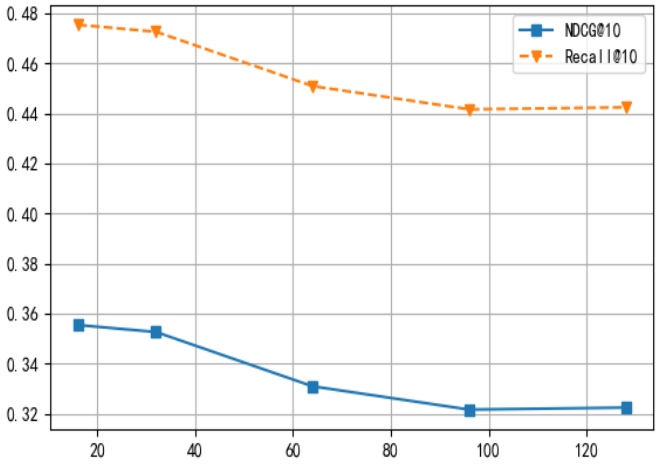

Figure 4. Analysis of Hidden Layer Dimensions

5. Conclusion

To address the insufficient capability of long-term interest modeling and the inadequate modeling of short-term interest variations in sequential recommendation, this paper proposes a recommendation framework that integrates convolutional neural networks with state space models. The CNN structure is employed to model users' recent interaction behaviors and extract local interest dynamics, while the Mamba model is introduced as the long-term interest modeling module to capture global features in user behavior sequences. On this basis, a collaborative modeling structure for long-term and short-term interests is designed to achieve multi-scale representation of user interests. Experimental results demonstrate that the proposed model outperforms existing methods on multiple public datasets, validating its effectiveness and practicality in modeling dynamic user interests.

References

[1] Chen K., Chen H. L., Peng C., et al. User profiling enhanced personalized recommendation system based on large model world knowledge [J]. Computer Applications, 2024, 44(8). doi: 10.11772/j.issn.1001-9081.2024060775

[2] Wang L., Zhu Z. Q., Zhou H. A survey on sequential recommendation based on deep learning [J]. Journal of Computer Research and Development, 2023, 46(1): 1-26. DOI:10.11897/SP.J.1016.2023.00001.

[3] Rendle S, Freudenthaler C, Schmidt-Thieme L. FPMC: Factorizing personalized Markov chains for next-basket recommendation[C]//Proceedings of the 19th International Conference on World Wide Web. ACM, 2010: 811–820.

[4] Hidasi B, Karatzoglou A, Baltrunas L, Tikk D. Session-based recommendations with recurrent neural networks [C] // Proceedings of the International Conference on Learning Representations (ICLR). 2016.

[5] Tang J, Wang X. Personalized top-n sequential recommendation via convolutional sequence embedding[C]//Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM). ACM, 2018: 565–573.

[6] Kang W C, McAuley J. Self-attentive sequential recommendation [C]//Proceedings of the IEEE International Conference on Data Mining (ICDM). IEEE, 2018: 197–206.

[7] Sun F, Liu J, Wu J, et al. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer[C]//Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM). 2019: 1441–1450.

[8] Gu A, Dao T, Ermon S, Ré C. Mamba: Linear-time sequence modeling with selective state spaces[C]//Advances in Neural Information Processing Systems. NeurIPS, 2023.

[9] Dao T, Liu T, Zhang T, et al. Mamba-2: A linear-time state space model for language modeling[J]. arXiv preprint arXiv:2404.03063, 2024.

[10] Feng L., Wang Y. F., Zhang X. Y. A survey on personalized sequential recommendation based on Transformer [J]. Journal of Frontiers of Computer Science and Technology, 2022, 16(6): 925-938.

[11] Quadrana M, Karatzoglou A, Hidasi B, Cremonesi P. Personalizing session-based recommendations with hierarchical recurrent neural networks[C]//Proceedings of the 11th ACM Conference on Recommender Systems (RecSys). 2017: 130–137.

[12] Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th International Conference on World Wide Web. 2001: 285–295.

[13] Wang Q, Zhang T, Yu Y, Liu Y, Zhang H. Learning disentangled user preferences for sequential recommendation[C]//Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM). ACM, 2022: 1839–1848.

[14] Wang Y, Li J, Zhao W X, et al. A survey on session-based recommender systems[J]. ACM Transactions on Information Systems, 2022, 40(3): 1–38. DOI:10.1145/3510428.

[15] Chen W, Zhang X, Zhang H, et al. S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization[C]//Proceedings of the 29th ACM International Conference on Information and Knowledge Management. ACM, 2020: 1893–1902.

[16] Zhou T, Xu Y, Zhang H, et al. Dynamic intent-aware recommendation with temporal attention networks[J]. Information Processing & Management, 2022, 59(3): 102949.

[17] Xu Y, Zhao Z, Liu B, et al. Graph contextualized self-attention network for session-based recommendation[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2019: 2298–2306.

[18] Fang H, Sun F, Zhang M, et al. SINE: Sequential recommendation with sparse user interest[C]//Proceedings of the 27th ACM SIGKD.