Hindawi
Complexity
Volume 2022, Article ID 4275868, 15 pages
https://doi.org/10.1155/2022/4275868

Research Article

An Improved Sequential Recommendation Algorithm based on Short-Sequence Enhancement and Temporal Self-Attention Mechanism

Jianjun Ni,[1,2] **Guangyi Tang**,[1] **Tong Shen**,[1] **Yu Cai**,[1] **and Weidong Cao**[1,2]

[1]*College of Internet of Things Engineering, Hohai University, Changzhou 213022, China*
[2]*Jiangsu Key Laboratory of Power Transmission & Distribution Equipment Technology, Hohai University, Changzhou 213022, China*

Correspondence should be addressed to Jianjun Ni; njjhhuc@gmail.com

Received 11 February 2022; Accepted 19 April 2022; Published 29 September 2022

Academic Editor: Jesus Vega

Sequential recommendation algorithm can predict the next action of a user by modeling the user's interaction sequence with an item. However, most sequential recommendation models only consider the absolute positions of items in the sequence, ignoring the time interval information between items, and cannot effectively mine user preference changes. In addition, existing models perform poorly on sparse data sets, which make a poor prediction effect for short sequences. To address the above problems, an improved sequential recommendation algorithm based on short-sequence enhancement and temporal self-attention mechanism is proposed in this paper. In the proposed algorithm, a backward prediction model is trained first, to predict the prior items in the user sequence. Then, the reverse prediction model is used to generate a batch of pseudo-historical items before the initial items of the short sequence, to achieve the goal of enhancing the short sequence. Finally, the absolute position information and time interval information of the user sequence are modeled, and a time-aware self-attention model is adopted to predict the user's next action and generate a recommendation list. Various experiments are conducted on two public data sets. The experimental results show that the method proposed in this paper has excellent performance on both dense and sparse data sets, and its effect is better than that of the state of the art.

1. Introduction

With the development of Internet technology, recommender systems have become one of the indispensable tools in people's daily life [1–4]. Compared with traditional methods, the sequential recommendation model performs well on the Top-N recommendation problem [5]. In recent years, with the development of deep learning technology, sequential recommendation models based on deep learning have been widely used, such as e-commerce shopping platforms, medical and health services [6, 7], and audiovisual platforms [8]. The user's interaction behavior with items in such application platforms can be regarded as a sequence of behaviors in chronological order. Based on this, researchers have proposed various sequential recommendation models to mine and analyze user-item interaction information. The purpose of these models is to provide users with a personalized recommendation list containing N items to help users filter out valuable information.

The recommendation model based on the Markov chain (MC) [9] method is one of the early methods of sequential recommendation, which assumes that the user's next action is determined by his historical behavior and transforms the recommendation problem into a sequence prediction problem. In recent years, with the continuous breakthroughs of the deep neural networks (DNN) in the field of artificial intelligence [10–12], researchers have tried to introduce a series of deep neural network models into the field of recommendation and have achieved a series of results [13–15]. For example, Huang et al. [16] combined the traditional MC

method and the recurrent neural network (RNN) to optimize the recommendation model and improve the recommendation accuracy. Based on long-short-term memory (LSTM) network, Xu et al. [17] combined self-attention network to capture users' complex and dynamic behavioral preferences. Inspired by the semantic understanding model, Sun et al. [18] applied the bidirectional attention model to sequential recommendation, combining user context information and making recommendations. The existing sequential recommendation models tend to perform poorly when there are a large number of short-sequence users in the data set [19]. In addition, most of the existing sequential recommendation models only consider the absolute position information of the user sequence and assume that each item has the same time interval, ignoring the impact of the time interval between items on the recommendation results, which cannot capture user preferences effectively [20].

To address these issues introduced above, some improved sequential recommendation models are proposed. For example, Zhao et al. [21] employed a deep bidirectional long-short-term memory network and attention mechanism to capture the changes in user preferences. Liu et al. [22] first used the method of reverse training short sequences to expand short sequences and fine-tuned the model through the enhanced short sequences, which can achieve certain results on sparse data sets. Ahmadian et al. [23] adopted a deep learning based trust- and tag-aware recommender system, to extract potential features through sparse automatic encoder, which can effectively solve the problem of data sparsity. Li et al. [24] adopted a time-aware self-attention mechanism to explore the effect of different time intervals on the prediction results. These methods lay a good foundation for the research of sequential recommendation models, but there are still some problems that are not well solved. For example, the pseudo-historical items generated by direct reverse training are not accurate enough, and the time interval information is not sufficiently mined to capture user preferences well.

Based on previous research, we propose a sequential recommendation model based on short-sequence enhancement and improved time-aware self-attentive mechanism to address the above-mentioned problems. In the proposed model, the data set is first preprocessed to divide users into long-sequence sets and short-sequence sets. Then, by reverse training the long-sequence set, a reverse prediction model is generated. Finally, the model is transferred for the short sequence, and a batch of pseudo-historical items is generated before the initial item of the short sequence, to enhance the short sequence and solve the problem of data sparsity. At the same time, the model adopts an improved time interval self-attention mechanism, which not only considers the influence of absolute location information on the recommendation effect but also considers the influence of the time interval information between any two items on the recommendation result.

The proposed model in this paper can fully reflect the changes of user preferences over time and improve the accuracy of the recommender system. In summary, the main contributions of this paper are as follows: (1) pretrain a reverse prediction model, use the transfer learning method to reverse predict short sequences, and generate a batch of pseudo-historical items before the initial items of the short sequence, so as to achieve the purpose of enhancing short sequences. (2) Combined with the absolute position information and time interval information of the item, an improved time-aware self-attention mechanism is used to give the absolute position weight and time interval weight of different items, fully exploit the change of user behavior preferences, predict the user's next action, and generate a list of recommendations. (3) Extensive experiments on two real data sets are conducted. The results demonstrate the effectiveness of the proposed model, which can outperform existing methods on two different metrics. In addition, the influence of each key component in the proposed model on the recommendation results is discussed through multiple experiments.

This paper is organized as follows. Section 2 introduces the related works. Section 3 gives out the details of the proposed model. Section 4 provides experiments and analysis of results. Section 5 discusses the parameters and important components of the proposed algorithm. Section 6 provides the conclusions.

2. Related Works

2.1. Sequential Recommendation Model. The earliest sequential recommendation models are mainly based on the MC method [25]. These MC-based models have a significant improvement over other types of recommendation algorithms in terms of short-term prediction. However, this type of model cannot capture the long-term behavioral features in the user sequence and has low accuracy and high computational complexity in long-term prediction.

As deep learning technology shines in the fields of machine vision and natural language processing [26, 27], the introduction of deep learning technology into recommender systems has also become the focus of researchers. For example, Zhang et al. [28] designed a new session-based recommendation method based on recurrent neural network, which fuses user's general preference information and dynamic preference information. Sun et al. [29] proposed a method based on temporal context awareness and RNN, which can effectively capture the correlation between items. In addition, long-short-term memory (LSTM) and gated recurrent unit (GRU) (two popular variants of RNNs) have also achieved results in the field of recommendation. For example, Yuan et al. [30] computed the global state transitions of user sequences to model user interest preference changes, based on an improved GRU model. Zhao et al. [31] proposed a content-aware movie recommendation model based on LSTM, which effectively utilizes the long-term and short-term information of the sequence for content perception and movie recommendation. However, most models assume that user behavior sequences are simple time-sequential sequences, without considering the time interval information between items. At the same time, existing models perform poorly on sparse data sets and short-sequence users.

2.2. Transformer-Based Model. Attention mechanism has achieved great results in a large number of works, such as image processing [32, 33] and natural language processing [34]. The essence of the attention mechanism can be understood as selecting some important information from a large amount of information and giving them weights, where the size of the weights represents the importance of the information. In recent years, transformer, a neural network architecture based on pure attention mechanism, has achieved excellent performance and effects in the field of machine translation [35]. Inspired by this, researchers introduced the transformer model into the recommender system [36] and achieved good results. The transformer-based model uses scaled dot-product attention, which is presented as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \tag{1}$$

where Q, K, V are three matrices representing queries, keys, and values, respectively; $\sqrt{d}$ is the scaling factor, which is used to avoid the inner product value being too large; and Softmax is the normalized function [33].

The attention function can be described as mapping a query and a set of key-value pairs to an output, where the queries, keys, values, and output are all vectors. The output is computed as a weighted sum of the values. The transformer model adopts the multihead attention mechanism, which executes the attention function in parallel, and connects each output value with item linearly again to obtain the final result. The multihead attention mechanism enables the model to focus on the information of different subspaces from different locations at the same time. The architecture of the transformer-based model is shown in Figure 1.

3. Proposed Sequential Recommendation Model

In this paper, a sequential recommendation model (SeTsRec) based on short-sequence enhancement and temporal self-attention mechanism is proposed, which is shown in Figure 2. First, the original data are preprocessed, where users are divided into long-sequence user sets and short-sequence user sets; and then the long-sequence sets are reversely input into the transformer network to train a reverse prediction model. Subsequently, inspired by the transfer learning method [37], this reverse prediction model is transferred to short-sequence users to generate a batch of pseudo-historical items before the initial item of the short-sequence user behavior list. By combining pseudo-historical items and short-sequence user behavior lists, an augmented sequence of short sequences is generated to enhance short sequences. Finally, the long sequences and enhanced short sequences are used as input to train a time-aware self-attention recommendation model and predict the user's next action. The model proposed in this paper will be described in detail as follows.

3.1. Problem Description. In the sequential recommendation problem, we assume that $U = \{u_1, u_2, \ldots, u_n\}$ is the user set of the system, where n is the number of the users in the data set, and $I = \{i_1, i_2, \ldots, i_m\}$ is the item set of the system, where m is the number of the items in the data set. For a certain user u, $S^u = \{i_1, i_2, \ldots, i_w\}$ and $T^u = \{t_{i_1}, t_{i_2}, \ldots, t_{i_w}\}$ are the user behavior sequence and time series, respectively, indicating that the length of the behavior sequence of the user u is w. Each item in the behavior sequence S^u is arranged in the chronological order of the user's interaction with it, and each element in the time sequence T^u represents the actual interaction time between the user u and the item i. At a certain moment, given the user's behavior sequence S^u and time series T^u, the goal of the model is to predict the next item that the user u is most likely to interact with, which is expressed as

$$p(i_{w+1}) = f(i_1, \ldots, i_w, t_{i_1}, \ldots, t_{i_w}), \tag{2}$$

where $p(\cdot)$ is the output probability of a certain item and $f(\cdot)$ is the nonlinear function that needs to be learned.

Recommendation systems usually provide users with multiple recommendation results and finally generate a recommendation list containing N items. Set $Y^u = \{y_1^u, y_2^u, \ldots, y_N^u\}$ as the output possibility of all the candidates, according to the output probability of the candidates, select the previous N items for recommendation, which is the famous Top-N recommendation problem in the recommendation system.

3.2. Short-Sequence Enhancement. The sequential recommendation algorithm is a recommendation method that predicts the user's next action by mining the information contained in the user's behavior sequence. Therefore, the validity of user behavior sequence information is crucial. Existing sequential recommendation methods have achieved good results. However, most of the existing methods do not solve the short-sequence prediction problem well and often perform poorly on sparse data sets. To deal with the limitation problem of the sequential recommendation model on sparse data sets, the proposed method in this paper utilizes the transfer learning to enhance short sequences on the basis of existing research, which will be introduced in detail as follows.

3.2.1. Reverse Prediction Model. Ideally, in the field of machine learning, it is always expected that the data sets used for model training are dense and efficient. However, in the actual research process, the data sets often have a large amount of data sparse phenomenon. In sparse data sets, there are often a large number of missing or zero data, which makes the data availability very poor, and brings many difficulties to establish the recommendation models.

In this paper, the user set U is first divided into a long-sequence user set U_L and a short-sequence user set U_S according to the length of the user sequence. The long-sequence user set U_L is a dense datas et, and the short-sequence user set U_S is a sparse data set. For the long-sequence

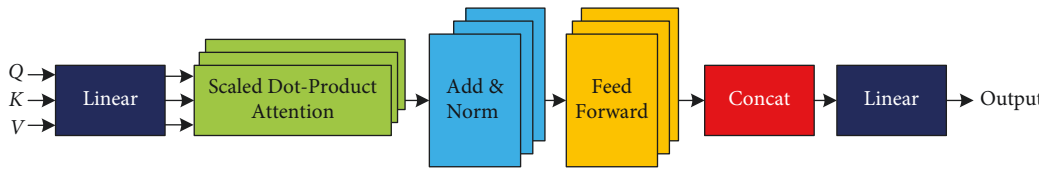

Figure 1: The architecture of the transformer-based model, where Q, K, V are three matrices representing queries, keys, and values, respectively.

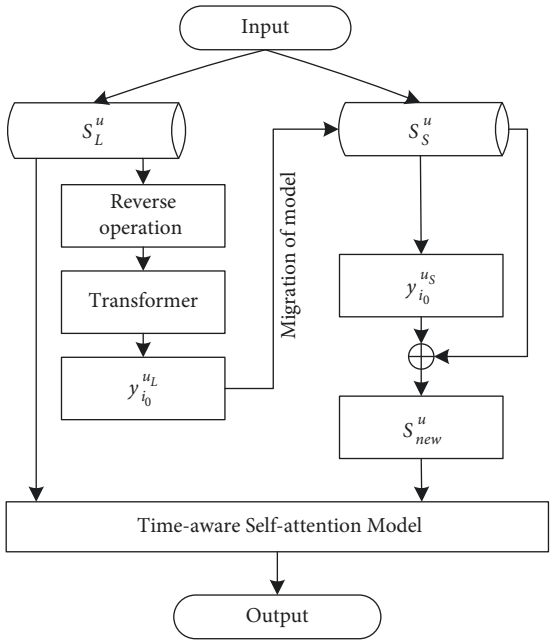

Figure 2: The framework of the proposed model, where S_L^u represents the behavior sequence of the long-sequence user set U_L; S_S^u represents the behavior sequence of the short-sequence user set U_S; $y_{i_0}^{u_L}$ represents the previous item of the item i_1 in the sequence S_L^u; $y_{i_0}^{u_S}$ represents the previous item of the item i_1 in the sequence S_S^u; and S_{new}^u is the generated enhanced short sequence.

user set U_L, the behavior sequence S_L^u can be obtained. In this paper, the long sequence is first reversed to obtain the reverse sequence S_r, and then the reverse sequence S_r is input into the transformer layer for training, to obtain a reverse prediction model. For the user u, the purpose of this reverse prediction model is to predict the previous item of the sequence $S_L^u = \{i_1, i_2, \ldots, i_n\}$:

$$y_{i_0}^{u_L} = f\left(S_L^u : i\right), \tag{3}$$

where $y_{i_0}^{u_L}$ represents the previous item of the item i_1 in the sequence S_L^u. Although the model is reverse-trained, the transformer network is also able to mine interitem correlations, which has been demonstrated in previous work [22].

3.2.2. Pseudo-Historical Item Generation. The existing methods often regard the data set as a whole for recommendation tasks, which ignore the different data quality of different users in the same data set. Specifically, some users have interacted more with the item, and the data of these users is relatively rich and reliable; while some users have little interaction data with the item, so the data of these users are sparse and can be poor usability. In order to solve this problem, this paper uses the transfer learning method to transfer the reverse prediction model of long-sequence users obtained above to short-sequence users [38]. The long-sequence reverse prediction task is taken as the source task, and the short-sequence reverse prediction task is taken as the target task. By fully mining the rich data information provided by the long-sequence users, the data sparsity problem of the short-sequence users is alleviated, which can improve the overall recommendation quality. Taking the short-sequence set as input, the reverse prediction model is used to generate pseudo-history items of short-sequence users, namely:

$$y_{i_0}^{u_S} = f\left(S_S^u : i\right), \tag{4}$$

where S_S^u represents the behavior sequence of the short-sequence user set U_S and $y_{i_0}^{u_S}$ represents the previous item of the item i_1 in the sequence S_S^u.

For a data set, we define the length L to represent the threshold for the short-sequence user set. Namely, if the length of a sequence (denoted by $|S^u|$) is less than L, this sequence is regarded as a short sequence; otherwise, the sequence is regarded as a long sequence.

In this paper, we denote the generated set of pseudo-historical items as $\{i_{-q+1}^u, \ldots, i_{-1}^u, i_0^u\}$ and place this set before the initial items i_1^u of the original short sequence to form an augmented short sequence, where q is the total number of pseudo-historical items generated by short sequences. Figure 3 shows the enhanced short-sequence set, in which the yellow part represents the generated pseudo-historical item, and the green part represents the original short sequence. In Figure 3, it is assumed that $q = 3, L = 4$. The generated enhanced short sequence is denoted by

$$S_{\text{new}}^u = \left\{y_{-q+1}, \ldots, y_{-1}, y_0, i_1, i_2, \ldots, i_n\right\}. \tag{5}$$

3.3. Time-Aware Self-Attention Model. The existing sequential recommendation model simply regards the user's behavior list as a sequential sequence according to the interaction time between the user and the item. In addition, the items are regarded as having the same time interval. Specifically, as shown in Figure 4(a), if the user A and the user B have been exposed to the same item, the traditional method will regard the time interval between the items in the two sequences as a fixed value of N days, which will lead to the same result for the two different users. However, such a result is unreasonable because the actual time that the user A and the user B have access to these items is different.

In the actual application scenarios, the time interval between items will be different even if the user's behavior list

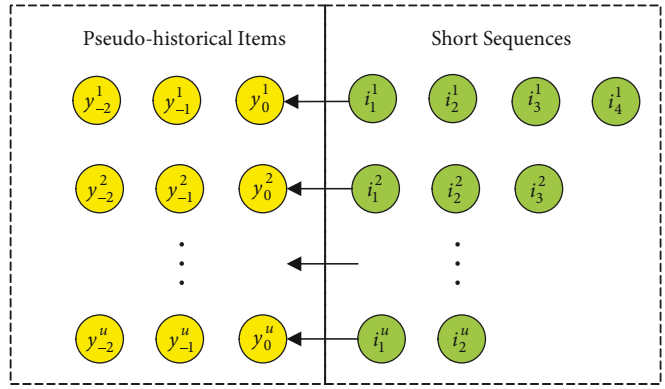

Figure 3: The example of the short-sequence enhancement process.

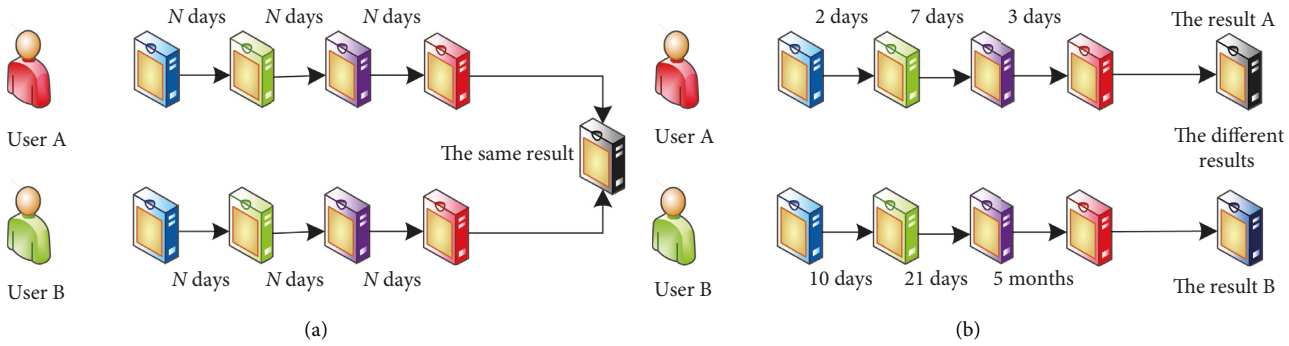

Figure 4: The effect of different time intervals on recommendation results. (a) Traditional sequential recommendation. (b) Our proposed method.

is exactly the same due to the different actual interaction time between users and items. As shown in Figure 4(b), although the user A and the user B have been exposed to the same items, the time interval between items is different. In this case, if the model can combine the different time interval information between the items, it is possible to make more accurate recommendation results. To solve the above problems, this paper adopts an improved time-aware self-attention model. The overall framework of the proposed model is shown in Figure 5, which will be introduced in detail as follows.

3.3.1. Time Interval Matrix. After getting the augmented sequence of user behavior S^u_{new}, it is used as model input together with the long-sequence S^u_L. First, the two different types of behavior sequences are converted into a fixed-length sequence S:

$$S^u_{\text{new}} \cup S^u_L \longrightarrow S = (s_1, s_2, \ldots, s_m), \tag{6}$$

where m represents the maximum sequence length of the input model. If the length of the sequence S^u_{new} or S^u_L is greater than m, only the latest m items are considered; otherwise, padding items are added to the left of the sequence S until its length reaches m.

Similarly, for the time series T^u_{new} and T^u_L, they can be converted to a fixed sequence t:

$$T^u_{\text{new}} \cup T^u_L \longrightarrow t = (t_1, t_2, \ldots, t_m). \tag{7}$$

If the length of the sequence T^u_{new} or T^u_L is greater than m, only the latest m items are considered; otherwise, the time corresponding to the first item t_1 is used on the left side of the sequence t, and padding it until its length reach m. In this study, for the time of the pseudo-historical items generated in Section 3.2.2, the average time interval $t_{\text{avg}} = \sum_m^{i=1} \sum_m^{j=1} r^u_{ij}/2$ is used to define them in turn, which are calculated as follows:

$$\begin{cases} t_{y_0} = t_1 - t_{\text{avg}}, \\ t_{y_{-1}} = t_{y_0} - t_{\text{avg}}, \\ \vdots \\ t_{y_{-q+1}} = t_{y_{-q+2}} - t_{\text{avg}}. \end{cases} \tag{8}$$

After obtaining the user's fixed time series $t = (t_1, t_2, \ldots, t_m)$, define the time interval between any items as $\triangle t = |t_i - t_j|$. Due to the different frequency of interaction between different users and items, this paper adopts the relative length of the time interval between items, which is defined as follows [24]:

$$r^u_{ij} = \left| \frac{\Delta t}{r_{\min^u}} \right|, \qquad r_{\min^u} = \min(\Delta t). \tag{9}$$

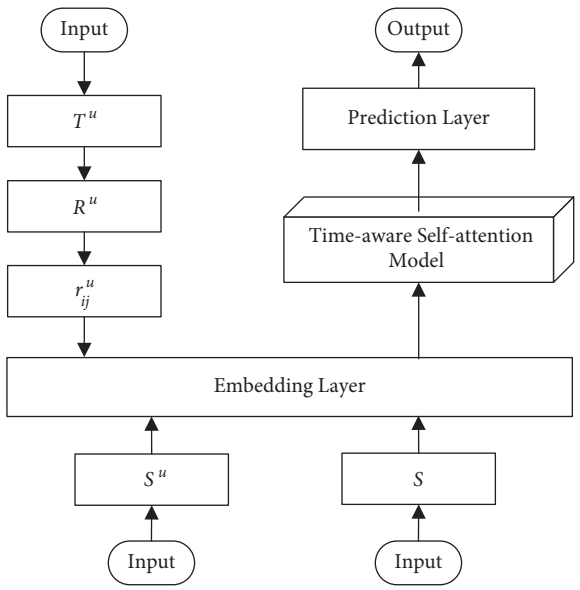

Figure 5: The framework of the proposed model, where T^u represents the time series of user u; R^u is the time interval matrix of user u; r_{ij}^u is the element in the time interval matrix R^u; S^u represents the behavior sequence of user u; and S is the absolute position information sequence of the item.

Finally, the time interval matrix R^u of the user u can be obtained:

$$R^u = \begin{bmatrix} r_{11}^u & r_{12}^u & \cdots & r_{1m}^u \\ r_{21}^u & r_{22}^u & \cdots & r_{2m}^u \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1}^u & r_{m2}^u & \cdots & r_{mm}^u \end{bmatrix}. \tag{10}$$

3.3.2. Time-Aware Self-Attention Module. (1) Time-Aware Self-Attention Layer. For each input sequence, an embedding layer is applied to convert the user behavior sequence into item embedding matrix $E^I \in \mathfrak{R}^{m\times d}$, absolute position information into position embedding matrix $E_K^P, E_V^P \in \mathfrak{R}^{m\times d}$, and time interval information into time interval embedding matrix $E_K^{\mathfrak{R}}, E_V^{\mathfrak{R}} \in \mathfrak{R}^{m\times m\times d}$ (d is the latent dimension):

$$\begin{aligned} E^I &= \left[m_{s1}, m_{s2}, \ldots, m_{sm}\right]^T, \\ E_K^P &= \left[p_1^k, p_2^k, \ldots, p_m^k\right]^T, \\ E_V^P &= \left[p_1^v, p_2^v, \ldots, p_m^v\right]^T, \\ E_K^{\mathfrak{R}} &= \begin{pmatrix} r_{11}^k & r_{12}^k & \cdots & r_{1m}^k \\ r_{21}^k & r_{22}^k & \cdots & r_{2m}^k \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1}^k & r_{m2}^k & \cdots & r_{mm}^k \end{pmatrix}, \\ E_V^{\mathfrak{R}} &= \begin{pmatrix} r_{11}^v & r_{12}^v & \cdots & r_{1m}^v \\ r_{21}^v & r_{22}^v & \cdots & r_{2m}^v \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1}^v & r_{m2}^v & \cdots & r_{mm}^v \end{pmatrix}. \end{aligned} \tag{11}$$

Then a new sequence $Z \in \mathfrak{R}^{m\times d}$ is calculated:

$$Z = \left[z_1, z_2, \ldots, z_m\right]^T, \tag{12}$$

where $z_i \in \mathfrak{R}^d$ is obtained by the input item embedding, absolute position embedding, and time interval embedding, namely

$$\begin{aligned} Z_i &= \sum_m^j \alpha_{ij}\left(m_{sj}W^V + r_{ij}^V + p_j^V + b_i\right), \\ \alpha_{ij} &= \frac{\exp\left(e_{ij}\right)}{\sum_m^{k=1}\exp\left(e_{ik}\right)}, \\ e_{ij} &= \frac{m_{sj}W^Q\left(m_{sj}W^K + r_{ij}^k + p_j^k\right)^T}{\sqrt{d}}, \end{aligned} \tag{13}$$

where W^V, W^Q, $W^K \in \mathfrak{R}^{d\times d}$, respectively, represent the input item matrix of the value, query and key; $\sqrt{d}$ is the scale factor, which is used to prevent the inner product from being too large; and b_i is the bias term.

(2) Point-Wise Feed-Forward Network. The self-attention layer of the model is mainly based on linear combination to realize the combination of absolute position information and relative time interval information of items. In order to make the model have nonlinear characteristics and consider the interaction between different latent dimensions, we apply a point-wise feed-forward network to each output of the self-attention layer:

$$\mathrm{FFN}\left(z_i\right) = G\left(\left(z_iW_1 + b_1\right)W_2\right) + b_2, \tag{14}$$

where $G(\cdot)$ is an activation function, which is ELU in this paper. The main reason of using ELU function is that it can solve the Dead Relu problem, while reducing the influence of the bias term offset, and the learning rate is faster. $W_1, W_2 \in \mathfrak{R}^{d\times d}$ represents the weight matrix and $b_1, b_2 \in \mathfrak{R}^d$ represents the bias term.

(3) Stacking Self-Attention Blocks. With the continuous stacking of self-attention layers and point-wise feed-forward networks, problems such as overfitting and long training time will occur. In order to solve these problems, this paper adopts the residual connection, dropout, and layer normalization processing methods [36]:

$$\begin{aligned} Z_i &= z_i + \mathrm{Dropout}\left(\mathrm{FFN}\left(LN\left(z_i\right)\right)\right), \\ \mathrm{LN}\left(x\right) &= \frac{x-\mu}{\sqrt{\sigma^2+\varepsilon}}\odot\gamma + \beta, \end{aligned} \tag{15}$$

where $\odot$ is the element-level product; μ, σ represents the mean and variance of x; and γ, β represents the learned scale factors and bias terms.

The specific workflow is as follows: for each self-attention block, layer normalization is first applied to each input z_i, which is beneficial to stabilize and speed up the training process of the neural network. Then, the output of the self-attention layer is applied to the point-wise feed-forward network, to give the model nonlinearization features. Finally, a dropout regularization technique is applied to the

output of the position feed-forward network, to alleviate the overfitting problem that occurs in deep neural networks. The main reason for using the dropout regularization technology is that it can control overfitting by artificially destroying data, which has been proven to be effective in various neural network architectures [39, 40].

3.3.3. Prediction Layer. After the prediction layer obtaining the final representation of the absolute position of the item and the time interval, in order to predict the possible next action of the users, we use a Softmax function to calculate the user's interaction probability with the candidate item $y_{i,t}$, namely:

$$p\left(y_{i,t}\right) = \text{Softmax}\left(Z_t M_i^I\right), \tag{16}$$

where M_i^I represents the embedding vector of items i and Z_t represents the first given sequence $(i_1, i_2, \ldots, i_t)$ containing t items and their time interval $(r_{1(t+1)}, r_{2(t+1)}, \ldots, r_{t(t+1)})$ between the $t+1$th item.

3.3.4. Model Inference. This paper uses user implicit interaction data. In implicit feedback, the interaction between the user and the item can be regarded as a binary classification problem, where 1 means that the user likes an item and 0 means that the user does not like or has not touched the item. Therefore, the items in the user behavior sequence can be regarded as positive samples. At the same time, all the items that the user has not touched are regarded as negative feedback, which is sampled as negative samples. In this paper, the sampling is carried out according to the ratio of 1: 1. When negative sampling is performed for each user, the principle is to select those items with higher popularity, which are more representative. The loss function is as follows [30]:

$$\text{Loss} = \sum_{(u,i,j)\in S} \log\left(1 + e^{-\left(p\left(y_i^u\right) - p\left(y_j^u\right)\right)}\right) + \lambda\|\ominus\|_F^2, \tag{17}$$

where i represents the predicted candidate items; j represents the negative sample; $\ominus = \{E^I, E_K^P, E_V^P, E_K^{\mathfrak{R}}, E_V^{\mathfrak{R}}\}$ is the set, which represents the embedding matrix; and λ is the regularization parameter, which is used to prevent the model from overfitting. In the training process, the Adam optimizer is used to optimize the model, which is a variant of the stochastic gradient descent (SGD) algorithm [41]. As an adaptive learning rate optimization algorithm, the Adam optimizer is usually used for tasks in sparse data scenarios, and its convergence speed is fast [42].

In summary, the process of the proposed algorithm is shown in Algorithm 1.

4. Experiments

4.1. Setting of Experiments

4.1.1. Data set. In order to verify the effectiveness of the proposed algorithm in this paper, experiments were carried out on two public data sets, namely Movielens-1M data set (denoted by ML-1M, see https://files.grouplens.org/datasets/movielens/ml-1m.zip) and Amazon Beauty data set (denoted by AM-BE, see https://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Beauty_5.json.gz). Among them, the ML-1M data set is a dense data set, and the AM-BE data set is a sparse data set. The dense data set ML-1M in this paper is used to evaluate the effectiveness of the time self-attention improvement in the proposed model, while the sparse data set AM-BE is used to evaluate the effectiveness of the improved short-sequence enhancement method of the proposed model. The statistics of the two data sets are listed in Table 1, which contains the information such as users, items, and timestamps.

Before the experiments, the two data sets are preprocessed [17]. For all data sets, we treat the rating behaviors as implicit feedback, where "1" means that there is an interaction between the user and the item, on the contrary, "0" means that there is no interaction between the user and the item. Then, the behaviors are sorted according to the chronological order of the actual interaction between users and items to generate the historical behavior sequence of users.

In this paper, the leave-one-out method is used to train and test the model [43]. Namely, the user's last behavior-producing item is taken as the true value, which is used as the test set. The last second-behavior-producing item is taken as the validation set, and all other remaining items are used as the training set. The advantage of the leave-one-out method is that it is not affected by the random sample division method and can use as large a sample as possible for training. It is suitable for sparse data sets.

4.1.2. Evaluation Metrics. This paper adopts two commonly used metrics in Top-N recommendation problem, namely hit rate (HR) and normalized discounted cumulative gain (NDCG) [29] to evaluate the recommendation performance of the model.

Hit rate (HR) is a common indicator for measuring recall rate, which can intuitively measure whether the predicted item exists in the first k items of the real list. The larger the hit rate (HR), the more accurate the recommendation. The calculation of HR is as follows:

$$\text{HR@}k = \frac{\text{Number of Hits @}k}{|GT|}, \tag{18}$$

where $|GT|$ represents all items in the test set, Number of Hits @k represents in the user's recommendation list, and the number of the top k items belonging to the test set.

NDCG is often used to evaluate the accuracy of ranking of recommendation results [44]. NDCG introduces a location influence factor to discount lower ranked recommendations. The calculation of NDCG is as follows:

$$\text{NDCG@}k = z_k \sum_{i=1}^{k} \frac{2^{r_i} - 1}{\log_2(i+1)}, \tag{19}$$

where z_k is the normalization factor, which is used to make the value of NDCG between 0 and 1 and r_i

Input: The behavior sequence S^u of user u
Output: The recommendation list result of user u, denoted as Y^u
(1) **for** u in length $(|U|)$**do**
(2) **if**length $(|S^u|) > L$**then**
(3) $S^u = S_L^u$
(4) **else**
(5) $S^u = S_S^u$ %Date preprocessing
(6) **end if**
(7) **end for**
(8) **for** u in S_L^u**do**
(9) $y_{i_0}^{u_L} = f(S_L^u: i)$ %Reverse prediction model training
(10) **end for**
(11) **for** u in S_S^u**do**
(12) $y_{i_0}^{u_S} = f(S_S^u: i)$ %Short sequence enhancement
(13) **end for**
(14) **for** u in $S_{\text{new}}^u \cup S_L^u$**do**
(15) Generate time interval matrix;
(16) Calculate time-aware self-attention model;
(17) Apply the point-wise feed-forward network and further processed;
(18) Calculate prediction and loss;
(19) **end for**
(20) **return** Y;

Algorithm 1: The sequential recommendation algorithm proposed in this paper.

Table 1: The information of the two data sets.

Data set	Number of			
	Users	Items	Actions	Average actions
ML-1M	6,040	3,900	1,000,209	165.6
AM-BE	51,369	19,369	225,509	4.4

represents the predicted correlation of the item at position i in the sequence. If the item is in the test set, $r_i = 1$ otherwise, $r_i = 0$.

The above two indicators can well reflect the performance of the recommendation list. This paper intercepts the top 10 of the recommendation list, namely $k = 10$, and uses HR@10 and NDCG@10 to evaluate the performance of the recommendation model.

Remark 1. The proposed method in this paper is based on the deep neural network and transfer learning technology, which needs more time in the process of the model training. However, the proposed model is trained offline and the computational time of the prediction is very fast. Thus, the computational complexity is not used as the evaluation metric in this study, which is a common way in the literature about the recommendation problem [19, 44].

4.1.3. Comparison Methods. To show the efficiency of the proposed methods (denoted by SeTsRec), various methods are used for comparison in this paper, including the recommendation method without considering the order, the classic order recommendation method, and the latest order recommendation method. In the experiments, the settings of these comparison methods are made by their optimal parameters according to the respective paper declarations. The comparison methods are listed as follows:

(a) POP [28]: POP is a simple baseline method that generates recommendation lists based on item popularity rankings, namely more popular items rank higher.

(b) BPR [45]: Bayesian personalized ranking method, which is a classic nonsequential recommendation method using matrix factorization.

(c) FPMC [46]: A sequential recommendation method that combines matrix factorization and Markov chains method.

(d) GRU4Rec+ [47]: An RNN-based deep sequential recommendation model for user sessions.

(e) Caser [48]: A CNN-based sequential recommendation method that captures higher order Markov chains by applying a convolution operation to the embedding matrix of the nearest term.

(f) SASRec [36]: One of the state-of-the-art sequential recommendation methods, which is the first method using a self-attention-based sequential recommendation model.

(g) TiSASRec [24]: A state-of-the-art sequential recommendation model that applies a multiorder

Table 2: The parameters of the proposed model and the experimental environment.

Model	Learning rate	0.001
	Momentum	0.9
	Dropout rate	0.2
	Batch size	128
	Maximum iterations	200
	Validation interval	20
	Regularization	0.00005
	Short-sequence threshold	20
	Maximum sequence length for ML-1M	70
	Maximum sequence length for AM-BE	30
	Latent dimension for ML-1M	50
	Latent dimension for AM-BE	20
	Pseudo-historical item for ML-1M	5
	Pseudo-historical item for AM-BE	15
Environment	Programming software	Python3.6
	Deep learning framework	Pytorch
	Computer system	Windows 10
	Cpu	E5-2620 v4
	RAM	32.0 GB
	Gpu	GeForce RTX 2080

attention mechanism to capture personal and item relatedness.

4.1.4. Other Settings. The experiments are conducted on a computer with Windows 10 system and the programming language used in the experiment is Python3.6. In this study, the model uses two self-attention blocks. Because different data sets have different sparsity, some parameters are different, such as the maximum sequence length (m) and the latent dimension (d). The setting of these parameters will be discussed in Section 5 for details. The parameters of the proposed deep network and the experimental environment are listed in Table 2.

4.2. Experimental Results and Analysis. Table 3 shows the experimental results of the proposed algorithm and all baseline methods on two different data sets with different indicators. The results in Table 3 show that the best recommendation effect is achieved by the proposed method in this paper, which prove the superiority of the model in this paper. The results are analyzed in details as follows:

(1) In most cases, the sequential recommendation methods FPMC, GRU4Rec+, and Caser outperform the nonsequential recommendation methods POP and BPR. This indicates the necessity of considering the order of user behavior lists in recommender systems. The user's behavior sequence order information can effectively characterize the user's preference change to a certain extent and can effectively improve the performance of the recommendation system.

(2) Compared with the three classical sequential recommendation methods, the latest attention-based SASRec and TiSASRec methods outperform all other baseline methods on the two different types of data sets, which indicates that the attention mechanism can effectively improve the performance of recommender systems.

Table 3: The experimental results.

Models	ML-1M		AM-BE	
	HR@10	NDCG@10	HR@10	NDCG@10
POP	0.4386	0.2389	0.3215	0.1758
BPR	0.5952	0.3421	0.2554	0.1523
FPMC	0.6182	0.3917	0.3771	0.2477
GRU4Rec+	0.6522	0.4334	0.3949	0.2556
Caser	0.7517	0.5011	0.4064	0.2547
SASRec	0.7929	0.5524	0.4185	0.2722
TiSASRec	0.8038	0.5706	0.4345	0.2818
Ours (SeTsRec)	**0.8127**	**0.5805**	**0.4754**	**0.3036**

(3) The algorithm SeTsRec proposed in this paper is improved on the basis of the existing algorithm. Through short-sequence enhancement and the use of an improved time-aware self-attention mechanism, it not only works well on dense data sets but also has the best results on sparse data sets. On the dense data set ML-1M, the HR@10 and NDCG@10 of the proposed method are improved by 1.1 % and 1.7 %, respectively, compared with the second best method TiSASRec. On the sparse data set AM-BE, the performance of the proposed method is improved by 9.4 % and 7.7 %, respectively, compared with TiSASRec. Compared with the baseline method, our model adopts an improved time-aware self-attention mechanism, which can adaptively adjust the item absolute position information and time-interval information to assign different weights in two different types of data sets.

5. Discussions

The results of the experiments in Section 4 show that the proposed model has better performance than that of the state of the art. The influence of the key parameters is discussed in

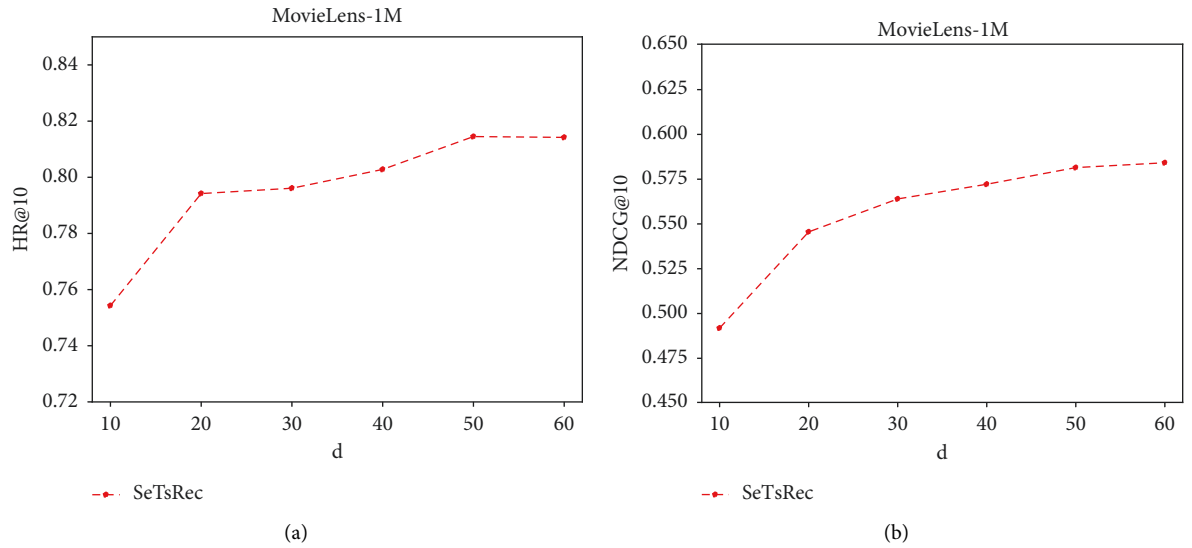

Figure 6: Experiment results of different latent dimension d on ML-1M. (a) Results on HR@10. (b) Results on NDCG@10.

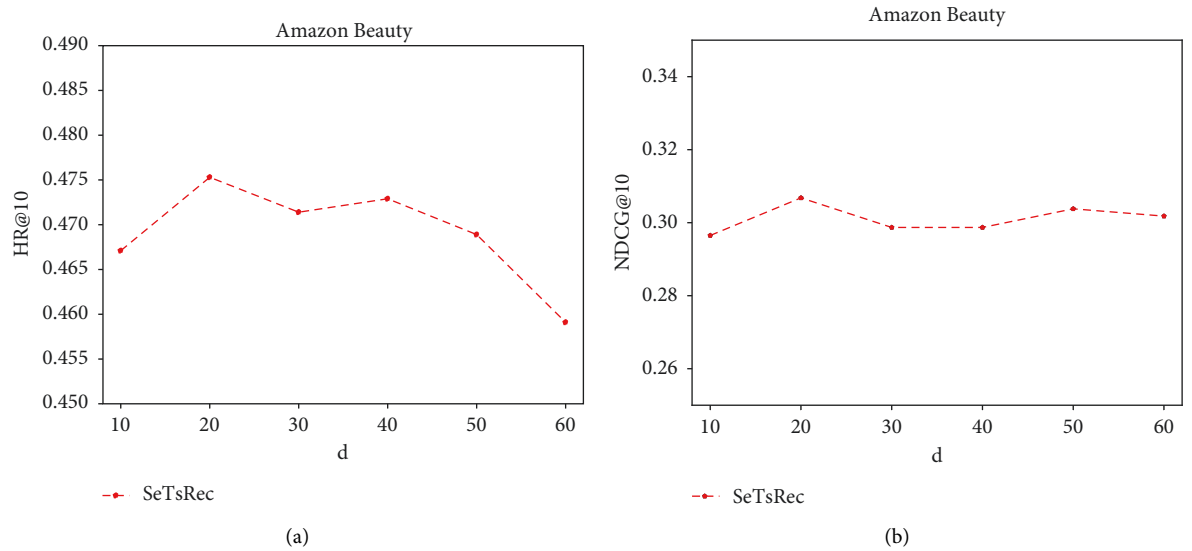

Figure 7: Experiment results of different latent dimension d on AM-BE. (a) Results on HR@10. (b) Results on NDCG@10.

this section. In addition, the ablation experiments are conducted in this section to further discuss the effectiveness of the improvement in the proposed model.

5.1. About the Latent Dimension. First, the influence of latent dimension d on the performance of the recommendation results of our model is discussed, and some experiments are conducted, where other hyperparameters are kept unchanged while the latent dimension d is changed within the range $[10, 60]$. The experimental results are shown in Figures 6 and 7.

It can be observed from Figure 6 (on the dense data set ML-1M) that the overall performance of the model improves with increasing potential dimensionality and tends to converge gradually, as the latent dimension increases. However, on the sparse data set AM-BE, the larger latent dimensions do not lead to better performance. The reason is that too many latent dimensions will lead to overfitting and thus degrade the model performance in a sparse data set. On the ML-1M data set, the algorithm in this paper tends to converge when $d \geq 50$. Considering the performance and time cost of the model, this paper sets the potential dimension $d = 50$ on the ML-1M data set and sets $d = 20$ on the AM-BE data set.

5.2. About the Maximum Sequence Length. Another important parameter of the proposed model is the maximum sequence length m. To discuss the influence of the maximum sequence length m of the input model on the performance of recommendation results, some experiments are conducted,

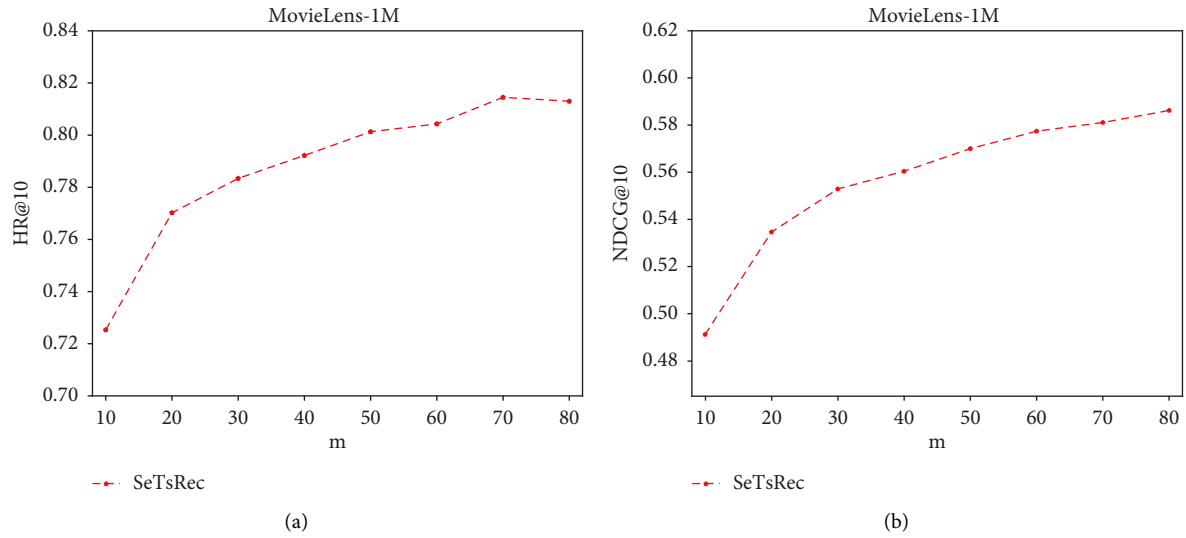

Figure 8: Experiment results of different maximum sequence length m on ML-1M. (a) Results on HR@10. (b) Results on NDCG@10.

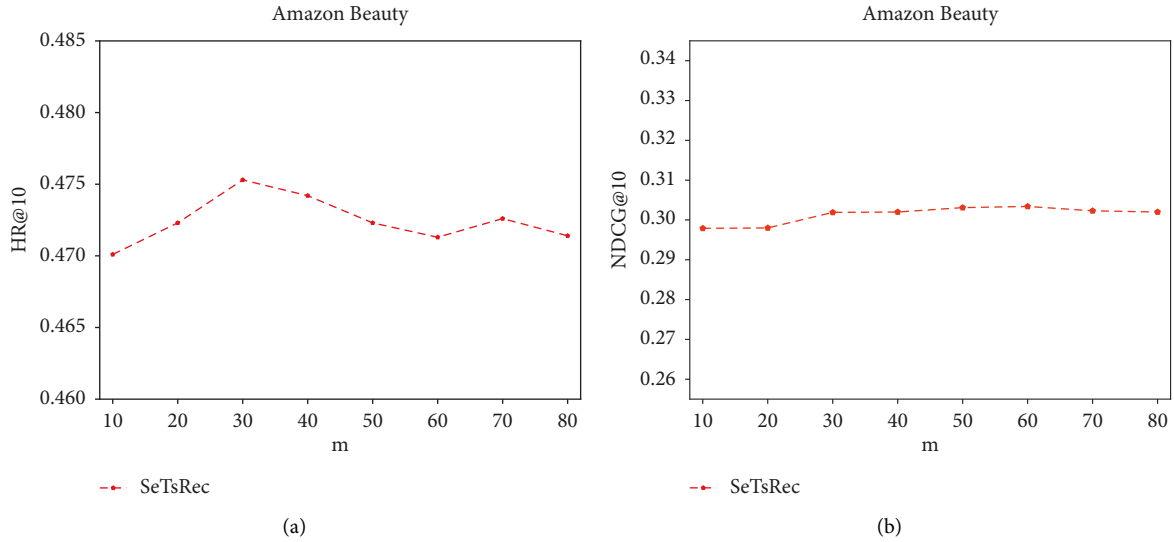

Figure 9: Experiment results of different maximum sequence length m on AM-BE. (a) Results on HR@10. (b) Results on NDCG@10.

where the maximum sequence length m is changed in the range [10,80], while keeping other hyperparameters unchanged. The experimental results are shown in Figures 8 and 9.

It can be observed from Figure 8 (on the dense data set ML-1M) that the model achieves satisfactory performance when the sequence length is $m \geq 70$. Therefore, under the consideration of balancing model performance and time cost, we set the maximum sequence length $m = 70$ on the ML-1M data set. On the sparse data set AM-BE, it can be observed that the model performance does not change much when m changes, this is because the average sequence length of the AM-BE data set is 4.4 (see Table 1), even after a certain degree of short-sequence enhancement, the longer sequence input does not provide more useful information, but will increase the time cost of the model. Therefore, the maximum sequence length is set as $m = 30$ on the data set AM-BE.

5.3. Ablation Experiments. This section discusses the impact of two major improvements in the proposed model, namely the short-sequence enhancement and time-aware self-attention mechanism. In these ablation experiments, the method based only on short-sequence enhancement is defined as SeTsRec-Se, and the method based only on time-aware self-attention is defined as SeTsRec-Ts, and they are compared with the existing SASRec method and our proposed algorithm SeTsRec. The experimental results are shown in Table 4, Figures 10 and 11. The results are analyzed in details as follows.

(1) On the dense data set ML-1M, the SeTsRec-Ts method outperforms the SeTsRec-Se method and the SASRec method. The experimental results show that the improvement of the model by the short-sequence enhancement method is limited. In this case, the

Table 4: Results of ablation experiment.

Methods	ML-1M		AM-BE	
	HR@10	NDCG@10	HR@10	NDCG@10
SASRec	0.7929	0.5524	0.4185	0.2722
SeTsRec-Se	0.7648	0.5297	0.4503	0.2907
SeTsRec-Ts	0.8038	0.5706	0.4345	0.2818
Ours (SeTsRec)	**0.8127**	**0.5805**	**0.4754**	**0.3036**

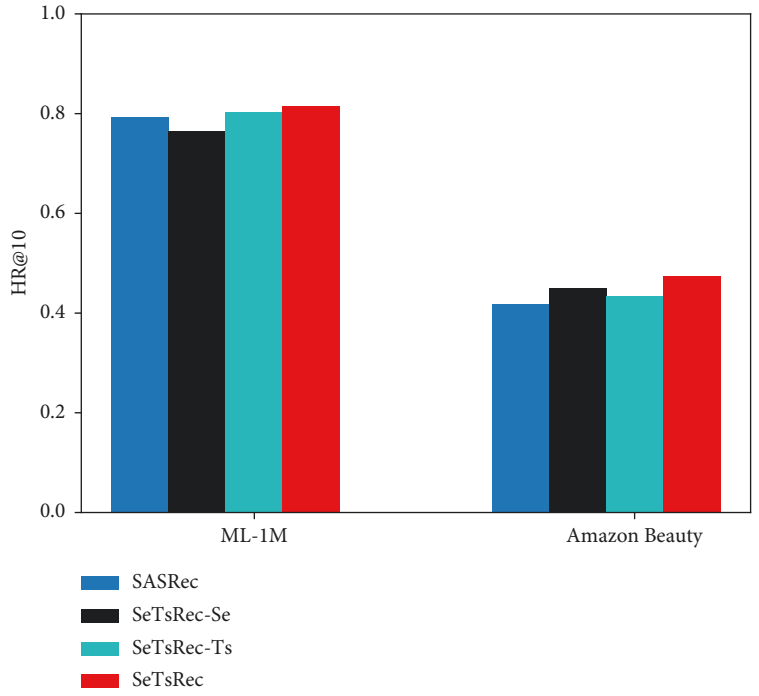

Figure 10: Results of ablation experiment on HR@10.

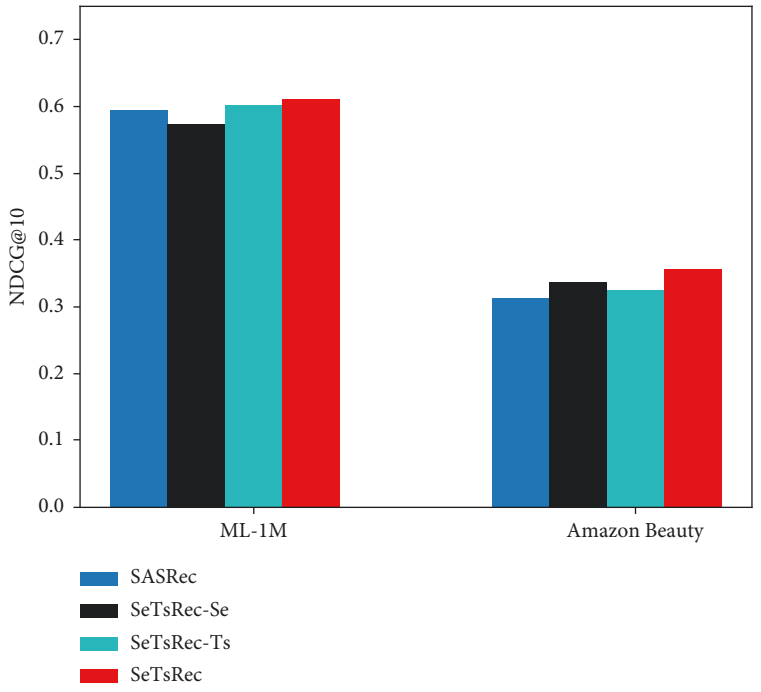

Figure 11: Results of ablation experiment on NDCG@10.

SeTsRec-Ts method can achieve good results, and its performance is improved by 1.4 %and 3.3 %, respectively, on the HR@10 and NDCG@10, compared with SASRec, which is close to the improved algorithm SeTsRec in this paper.

(2) On the sparse data set AM-BE, the SeTsRec-Se method is better than the SeTsRec-Ts method and the SASRec method, and its performance is improved by 3.6 % and 3.2 %, respectively, on the HR@10 and NDCG@10, compared with SeTsRec-Ts. The experimental results show that it is necessary to use the method of enhancing short sequences on sparse data sets. In addition, the model effect would not improve much if only the time-aware self-attentive mechanism approach is used. This is due to the large proportion of short-sequence users in the sparse data set, which limits the overall recommendation effect of the model.

(3) In summary, the proposed algorithm SeTsRec in this paper not only considers short-sequence enhancement to alleviate the problem of data sparsity but also combines the time-aware self-attention mechanism to fully consider the change of user preferences over time. Thus, it outperforms existing methods on both dense and sparse data sets.

6. Conclusions

This paper proposes a sequential recommendation algorithm based on improved short-sequence enhancement and temporal self-attention mechanism. The proposed algorithm first trains a reverse prediction model through the long-sequence users in the data set, to predict the reverse recommendation in the user sequence. Then, the model is transferred to short-sequence users, and pseudo-historical items of short-sequence users are generated to enhance short sequence. After enhancing short sequences, an improved time-aware self-attention model is adopted, which adaptively assigns different weights by combining the time interval information and absolute position information between items. It can deeply mine the changes of user preferences over time. Experimental results show that our method outperforms the existing sequential recommendation methods on different data sets. In the future, it can be considered to generate more accurate pseudo-historical items by improving the reverse prediction model to improve the recommendation effect further.

Data Availability

Publicly available data sets were analyzed in this study. These data can be found at https://files.grouplens.org/datasets/movielens/ml-1m.zip and https://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Beauty_5.json.gz.

Conflicts of Interest

The authors declared that they have no conflicts of interest to this work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61873086) and the Science and Technology Support Program of Changzhou (CE20215022).

References

[1] S. Ahmadian, N. Joorabloo, M. Jalili, and M. Ahmadian, "Alleviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach," *Expert Systems with Applications*, vol. 187, 2022.

[2] Z. Sun, Q. Guo, and J. Yang, "Research commentary on recommendations with side information: a survey and research directions," *Electronic Commerce Research and Applications*, vol. 37, 2019.

[3] A. Pujahari and D. Singh Sisodia, "Handling dynamic user preferences using integrated point and distribution estimations in collaborative filtering," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, In press, 2022.

[4] F. Tahmasebi, M. Meghdadi, S. Ahmadian, and K. Valiallahi, "A hybrid recommendation system based on profile expansion technique to alleviate cold start problem," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 2339–2354, 2021.

[5] Xu Chen, H. Xu, and Y. Zhang, "Sequential recommendation with user memory networks," in *Proceedings of the 11th ACM International Conference on Web Search and Data Mining, WSDM 2018*, pp. 108–116, Marina Del Rey, CA, USA, February 2018.

[6] E. Mutabazi, J. Ni, G. Tang, and W. Cao, "A review on medical textual question answering systems based on deep learning approaches," *Applied Sciences*, vol. 11, no. 12, p. 5456, 2021.

[7] L. Quijano-Sánchez, I. Cantador, M. E. Cortés-Cediel, and O. Gil, "Recommender systems for smart cities," *Information Systems*, vol. 92, Article ID 101545, 2020.

[8] J. Ni, Yu Cai, G. Tang, and Y. Xie, "Collaborative filtering recommendation algorithm based on TF-IDF and user characteristics," *Applied Sciences*, vol. 11, no. 20, p. 9554, 2021.

[9] H. Jia and J. Yang, "Research on joint ranking recommendation model based on Markov chain," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 6, Article ID e5191, 2020.

[10] H. Kang, S. Yang, J. Huang, and J. Oh, "Time series prediction of wastewater flow rate by bidirectional lstm deep learning," *International Journal of Control, Automation and Systems*, vol. 18, no. 12, pp. 3023–3030, 2020.

[11] J. Park and D.-J. Jung, "Deep convolutional neural network architectures for tonal frequency identification in a lofargram," *International Journal of Control, Automation and Systems*, vol. 19, no. 2, pp. 1103–1112, 2021.

[12] J. Ni, K. Shen, Y. Chen, W. Cao, and S. X. Yang, "An improved deep network-based scene classification method for self-driving cars," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–14, 2022.

[13] A. Yengikand, M. Meghdadi, and S. Ahmadian, "Deep representation learning using multilayer perceptron and stacked autoencoder for recommendation systems," in *Proceedings of the 2021 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2485–2491, Melbourne, Australia, October 2021.

[14] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: a survey and new perspectives," *ACM Computing Surveys*, vol. 52, no. 1, 38 pages, 2020.

[15] M. Ahmadian, M. Ahmadi, and S. Ahmadian, "Integration of deep sparse autoencoder and particle swarm optimization to develop a recommender system," in *Proceedings of the 2021 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2021*, pp. 2524–2530, Melbourne, Australia, October 2021.

[16] L. Huang, M. Fu, F. Li, H. Qu, Y. Liu, and W. Chen, "A deep reinforcement learning based long-term recommender system," *Knowledge-Based Systems*, vol. 213, Article ID 106706, 2021.

[17] C. Xu, J. Feng, P. Zhao, F. Zhuang, D. Wang, and Y.V. Liu, "Long-and short-term self-attention network for sequential recommendation," *Neurocomputing*, vol. 423, pp. 580–589, 2021.

[18] F. Sun, J. Liu, and J. Wu, "Bert4rec: sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1441–1450, Beijing, China, November 2019.

[19] X. Wang, Y. Sheng, H. Deng, and Z. Zhao, "Top-n-targets-balanced recommendation based on attentional sequence-to-sequence learning," *IEEE Access*, vol. 7, pp. 120262–120272, 2019.

[20] Q. Tan and F. Liu, "Recommendation based on users' long-term and short-term interests with attention," *Mathematical Problems in Engineering*, vol. 2019, pp. 1–13, 2019.

[21] C. Zhao, J. You, X. Wen, and X. Li, "Deep bi-lstm networks for sequential recommendation," *Entropy*, vol. 22, no. 8, p. 870, 2020.

[22] Z. Liu, Z. Fan, Yu Wang, and P. S. Yu, "Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1608–1612, Canada, July 2021.

[23] S. Ahmadian, M. Ahmadian, and M. Jalili, "A deep learning based trust- and tag-aware recommender system," *Neurocomputing*, vol. 488, 2021.

[24] J. Li, Y. Wang, and J. McAuley, "Time interval aware self-attention for sequential recommendation," in *Proceedings of the 13th International Conference On Web Search And Data Mining*, pp. 322–330, Houston, TX, USA, February 2020.

[25] R. He and J. McAuley, "Fusing similarity models with Markov chains for sparse sequential recommendation," in *Proceedings of the 2016 IEEE 16th International Conference On Data Mining (ICDM)*, pp. 191–200, IEEE, Barcelona, Spain, December 2016.

[26] J. Ni, Y. Chen, Y. Chen, J. Zhu, D. Ali, and W. Cao, "A survey on theories and applications for self-driving cars based on deep learning methods," *Applied Sciences*, vol. 10, no. 8, p. 2749, 2020.

[27] S. J. Lee, H. Choi, and S. S. Hwang, "Real-time depth estimation using recurrent cnn with sparse depth cues for slam system," *International Journal of Control, Automation and Systems*, vol. 18, no. 1, pp. 206–216, 2020.

[28] J. Zhang, C. Ma, X. Mu, P. Zhao, C. Zhong, and A. Ruhan, "Recurrent convolutional neural network for session-based recommendation," *Neurocomputing*, vol. 437, pp. 157–167, 2021.

[29] Ke Sun, T. Qian, Xu Chen, and M. Zhong, "Context-aware seq2seq translation model for sequential recommendation," *Information Sciences*, vol. 581, pp. 60–72, 2021.

[30] W. Yuan, H. Wang, X. Yu, N. Liu, and Z. Li, "Attention-based context-aware sequential recommendation model," *Information Sciences*, vol. 510, pp. 122–134, 2020.

[31] W. Zhao, B. Wang, M. Yang et al., "Leveraging long and short-term information in content-aware movie recommendation via adversarial training," *IEEE Transactions on Cybernetics*, vol. 50, no. 11, pp. 4680–4693, 2020.

[32] J. Li, H. Huo, C. Li, R. Wang, and Q. Feng, "Attentionfgan: infrared and visible image fusion using attention-based generative adversarial networks," *IEEE Transactions on Multimedia*, vol. 23, pp. 1383–1396, 2021.

[33] P. Shi, X. Fang, J. Ni, and J. Zhu, "An improved attention-based integrated deep neural network for pm2.5 concentration prediction," *Applied Sciences*, vol. 11, no. 9, p. 4001, 2021.

[34] I. Lopez-Gazpio, M. Maritxalar, M. Lapata, and E. Agirre, "Word n-gram attention models for sentence similarity and inference," *Expert Systems with Applications*, vol. 132, pp. 1–11, 2019.

[35] A. Vaswani, N. Shazeer, and N. Parmar, "Attention is all you need," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 5998–6008, Long Beach, CA, USA, December 2017.

[36] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *Proceedings of the 2018 IEEE International Conference On Data Mining (ICDM)*, pp. 197–206, IEEE, Singapore, November 2018.

[37] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[38] K.-H. Ahn and J.-B. Song, "Image preprocessing-based generalization and transfer of learning for grasping in cluttered environments," *International Journal of Control, Automation and Systems*, vol. 18, no. 9, pp. 2306–2314, 2020.

[39] R. Moradi, R. Berangi, and B. Minaei, "A survey of regularization strategies for deep models," *Artificial Intelligence Review*, vol. 53, no. 6, pp. 3947–3986, 2020.

[40] S. Wager, S. Wang, and P. Liang, "Dropout training as adaptive regularization," in *Proceedings of the 27th Annual Conference on Neural Information Processing Systems, NIPS 2013*, Lake Tahoe, NV, USA, December 2013.

[41] J. Mills, J. Hu, and G. Min, "Multi-task federated learning for personalised deep neural networks in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 630–641, 2022.

[42] J. David Camacho, C. Villaseñor, Y. Alma, C. Lopez-Franco, and N. Arana-Daniel, "Kadam: using the kalman filter to improve Adam algorithm," in *Proceedings of the 24th Iberoamerican Congress on Pattern Recognition, CIARP 2019*, pp. 429–438, Havana, Cuba, October 2019.

[43] C. Lonjarret, R. Auburtin, C. . Robardet, and M. Plantevit, "Sequential recommendation with metric models based on frequent sequences," *Data Mining and Knowledge Discovery*, vol. 35, no. 3, pp. 1087–1133, 2021.

[44] X. Wang, Q. Tan, and L. Zhang, "A deep neural network of multi-form alliances for personalized recommendations," *Information Sciences*, vol. 531, pp. 68–86, 2020.

[45] Y.-C. Lee, T. Kim, J. Choi, X. He, and S. W. Kim, "M-bpr: a novel approach to improving bpr for recommendation with multi-type pair-wise preferences," *Information Sciences*, vol. 547, pp. 255–270, 2021.

[46] 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, pp. 811–820, Raleigh, NC, USA, April 2010.

[47] B. Hidasi and A. Karatzoglou, "Recurrent neural networks with top-k gains for session-based recommendations," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 843–852, Torino, Italy, October 2018.

[48] D. Yang, J. Zhang, S. Wang, and X. D. Zhang, "A time-aware cnn-based personalized recommender system," *Complexity*, vol. 2019, Article ID 9476981, 11 pages, 2019.