Hybrid LSTM-Transformer Model for Sequential and Context-Aware Tourism Destination Recommendation

Jianeng Sun, Kai Zou*
School of Tourism and Culture, Liaodong University, Dandong 118001, China
E-mail: zoukai7525@126.com
*Corresponding author

Keywords: LSTM, transformer, tourism industry, content recommendation

Received: April 25, 2025

With the vigorous development of big data and artificial intelligence technology, personalized recommendation systems have been deeply integrated into the tourism industry, aiming to improve user experience through accurate travel recommendations. However, traditional recommendation algorithms often struggle to capture the dynamic changes of user interests and long-tail preferences when dealing with complex sequential data. To this end, this study constructs a hybrid architecture that integrates long short-term memory (LSTM) and transformer to build an efficient tourist destination recommendation system. Among them, the LSTM module focuses on mining the time-dependent features of user behavior, and the Transformer captures the long-range dependencies in behavior by relying on the self-attention mechanism. The experiment was conducted based on a real dataset of more than 100 million user browsing records, and the results showed that the prediction accuracy of the system was improved by 23% and the recall rate was increased by 18% compared with the baseline method. User feedback analysis further showed that the new system's recommendations were more relevant to actual needs, and the average satisfaction score increased by 15%. These results not only confirm the great potential of deep learning in optimizing the personalized recommendation system for tourism, but also provide key technical support for the intelligent upgrading of the tourism industry.

Povzetek: Študija predstavi napreden sistem priporočil, ki izboljša točnost in uporabniško izkušnjo v turizmu.

1 Introduction

Worldwide, tourism is booming at an unprecedented rate, becoming an essential driver of a country's economic growth and socio-cultural exchange [1]. According to statistics, the number of international tourists has doubled to nearly 1.5 billion in the past decade and is expected to reach more than 1.8 billion by 2030 [2]. At the same time, with the rapid development of the Internet and mobile communication technology, the way people obtain travel information, plan itineraries, and even book services have undergone revolutionary changes [3]. Against this background, how to use advanced information technology to provide consumers with more accurate, convenient and humanistic tourism product recommendation solutions is not only a common challenge faced by the industry but also a frontier topic actively explored by academic circles.

In recent years, deep learning, as an emerging machine learning paradigm, has made breakthroughs in many fields, such as computer vision, natural language processing, speech recognition, etc., demonstrating strong data-driven capabilities and pattern recognition potential [4, 5]. In particular, Long Short-Term Memory Network (LSTM) and Transformer architectures have attracted widespread attention because of their unique advantages [6, 7]. The former is good at processing sequence data, can better overcome the gradient disappearance/explosion problem, and has long-term solid dependency modeling ability. The latter is based on the attention mechanism, which can efficiently process input and output sequences in parallel, accelerate the training process, and show excellent performance in natural language generation tasks such as translation and question answering [8].

Given this, this study aims to explore how to organically combine LSTM with Transformer to build a set of intelligent decision-making assistance platforms suitable for tourist destination recommendation scenarios, aiming at achieving the following three goals: first, to dig deeply into the personalized needs of users, second, to effectively integrate online and offline multivariate data resources, and third, to continuously optimize the algorithm model to improve the system response speed and recommendation quality. Specifically, we plan to start our work from the following perspectives:

In view of the common cold start problem and long tail effect limitation of existing recommendation systems, that is, the lack of sufficient understanding and support for new users or niche destinations for the first time, we will try to use LSTM to capture the coherence and periodicity in the historical behavior trajectory of users and form a preliminary interest tag and location preference matrix. Through Transformer, we conduct in-depth semantic analysis of social media reviews, travel guides, scenic spot introductions, and other text materials. We also extract keywords and emotional tendencies to supplement and improve user portraits.

Considering the limitations of a single data source and the deviation risk it brings, we will actively seek cross-border cooperation opportunities and introduce real-time dynamic data streams such as airline flight schedules, hotel room price fluctuations, real-time passenger flow statistics in scenic spots, as well as external event trigger signals such as weather forecast, holiday arrangement and emergency announcement, to further enhance the sensitivity and adaptability of the system to market changes. At the same time, to ensure data security and privacy protection, we will strictly abide by the requirements of relevant laws and regulations and adopt technical measures such as desensitization encryption and authority classification to ensure that personal information is not abused or leaked.

Since deep learning models usually involve many parameters and complex operations, balancing the relationship between model capacity and computational efficiency is crucial. Therefore, lightweight network design ideas will be adopted, redundant nodes will be appropriately tailored, and memory occupation and energy consumption will be reduced. At the same time, using the idea of transfer learning, the weight of the pre-trained model is used as the initialization parameter to speed up the convergence speed and avoid the overfitting phenomenon. Considering the significant differences in cultures and consumption habits in different regions, we will also formulate corresponding strategies according to the characteristics of different market segments. For example, Asian tourists tend to travel with families, European and American backpackers prefer outdoor adventure activities, etc., to better cater to the target audience's tastes and enhance user engagement.

2. Recommendation system theory and related technologies

2.1 Recommendation algorithm

Content-based and collaborative filtering recommendation algorithms are widely used in recommendation systems because of their simple implementation [9, 10]. The former recommends similar items according to the characteristics of items and user preferences, which is conducive to the rapid adaptation of new users. The latter promotes the diversity of items through user similarity recommendation, which is suitable for promoting new items [11].

Collaborative filtering is divided into two types: project and user. Item collaborative filtering recommends item similarity, while user collaborative filtering recommends new items according to user similarity. The difference lies in the data type based on the recommendation [12, 13]. The former calculates the similarity between items, while the latter focuses on the similarity between users. Item-based collaborative filtering is suitable for scenarios with many goods and few users, such as movie recommendations [14]. By calculating the similarity between movies, recommending new movies has the advantage of avoiding the "cold start" problem of new users. However, due to the limitation of product characteristics, it may need to be more accurate to capture users' interests. User-based collaborative filtering suits scenarios with many users and few commodities, such as music recommendations [15]. Recommending similar favorite music by calculating the similarity among users can accurately reflect users' interests, but the similarity calculation may be inaccurate due to the "sparsity" problem. In practice, collaborative filtering of projects and users can be combined to complement each other's advantages and disadvantages [16, 17].

Collaborative filtering requires the use of a similarity matrix. This paper designs a recommendation algorithm to improve collaborative filtering, relying on the initially constructed similarity matrix for recommendation operation [18]. Assuming that there is n users in the user set and m movies in the movie set, and the value in the matrix is the *score* of specific users for specific movies, this paper constructs a similarity matrix with a size of $n \times m$ according to the score table as follows, as shown in formula (1):

$$MatrixSim = \begin{bmatrix} score_{11} & \cdots & score_{1m} \\ \vdots & \ddots & \vdots \\ score_{n1} & \cdots & score_{nm} \end{bmatrix} \quad (1)$$

$score_{ij}$ for user i 's rating of movie j. After the scoring matrix is generated, the similarity needs to be calculated. Commonly used methods include Euclidean distance, cosine similarity, Pearson correlation coefficient and Jackard coefficient. Cosine similarity is the most commonly used choice in collaborative filtering because of its simple calculation and lack of optimization. The formula for calculating the cosine similarity between two vectors, a and b, is (2):

$$Sim(a,b) = \frac{ab}{|a||b|} = \frac{x_1x_2 + y_1y_2}{\sqrt{x_1^2 + y_1^2}\sqrt{x_2^2 + y_2^2}} \quad (2)$$

x_1,y_1,x_2,y_2 represent variables, which are equation (3) when extended to user-project data:

$$SimCos(u,v) = \frac{\sum_{i=1}^{n} R_{u,i}R_{v,i}}{\sqrt{\sum_{i=1}^{n} R_{u,i}^2}\sqrt{\sum_{i=1}^{n} R_{v,i}^2}} \quad (3)$$

sim refer to metrics that measure how similar two objects are in specific dimensions. In the formula, u and v represent different users, $R_{u,i}$, $R_{v,i}$ respectively represent the score of user u on the i item and user v on the i item.

2.2 LSTM model

LSTM is a recurrent neural network which has attracted wide attention because of its strong ability to process sequence data [19]. Compared with traditional RNN, LSTM has better memory, is good at extensive sequence processing, and effectively avoids gradient disappearance and explosion problems [20]. The structure of the LSTM is shown in Figure 1. The core of LSTM lies in the forgetting gate, input gate and output gate, which regulate information flow and balance long and short memory. Another memory cell is set up to store historical information and ensure stable transmission. The sequence is processed step by step, and each step includes two vectors, input and output, and the sequence data is processed by operation.

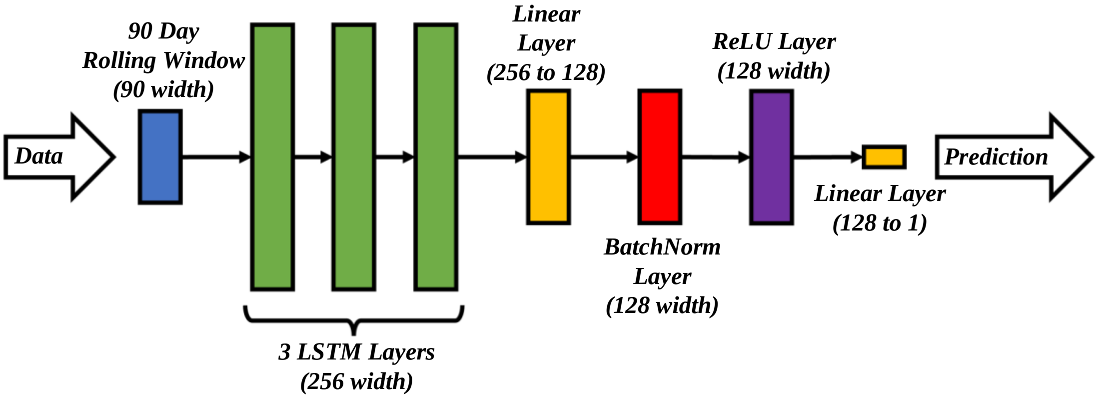

Figure 1: Structure of LSTM

The forgetting gate f_t in LSTM is used to judge which information transmitted by the previous unit needs to be retained and which needs to be forgotten, as shown in equation (4):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t]) \quad (4)$$

σ represents the sigmoid activation function, W_f represents the weight matrix of the forgetting gate, h_{t-1} represents the hidden state of time step $t-1$, and x_t represents the input vector of time step t. The input gating unit mainly extracts the input information of the current unit. Equations (5)-(6) are as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t]) \quad (5)$$

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t]) \quad (6)$$

i_t represents the output of the input gate at time step t, x_t represents the input, and W_i, W_c represent the weights. *tanh* is the hyperbolic tangent activation function, the memory cell C_t combines the current memory information and the new memory information, and W_c represents the weight matrix of the cell state update gate. The calculation formula is shown in (7):

$$C_t = f_t \times C_{t-1} + i_t \times C_t \quad (7)$$

$\tilde{C}_t$ is the candidate cell state calculated by the W_c weight matrix, representing new information content. The calculation formulas of the output unit and the hidden unit are shown in (8) and (9):

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t]) \quad (8)$$

$$h_t = o_t \times tanh(C_t) \quad (9)$$

o_t represents the activation value of the output gate. In the extended short-term memory network (LSTM), W_o represents the weight matrix of the output gate, and h_t represents the hidden state of time step t. The LSTM model performs well in lengthy sequence processing using gating and memory units and is widely used in sequence data scenarios [21, 22]. In the recommendation system, LSTM can model user behavior sequences, such as browsing and purchase records, and learn preferences to improve recommendation accuracy. The input of time information can capture the evolution of users' interests and consumption patterns, realize dynamic, personalized recommendations, and transcend the static limitations of traditional collaborative filtering [23].

2.3 Transformer model

Transformer is a Seq2Seq model consisting of an encoder and a decoder. It innovatively uses a self-attention mechanism to replace CNN and RNN to complete network construction. Recommendation algorithms often only use their encoder part, which contains multi-head attention and feedforward neural networks [24, 25]. The attention mechanism is derived from the visual principle of the human eye, which focuses on key parts. Deep learning allows the model to capture critical information by giving weights and improving judgment accuracy [26]. Specifically, the attention weight vector calculates the score to weigh the elements and lock the core information. The symbolic definition of attention computation (10) is as follows:

$$Attention(Q,K,V) = softmax(QK^{T})V \quad (10)$$

Q represents a query, K represents key, V represents value, *softmax()* is a normalized exponential function, and T represents a transpose operation. Based on self-attention, the multi-head self-attention mechanism calculates different weight parameters multiple times, processes them in parallel in multiple subspaces, and then integrates the results to realize multi-angle information capture. The results in different subspaces are spliced, as shown in equations (11)-(12):

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (11)$$

$$MultiHead(Q,K,V) = Concat(head_1, \ldots, head_h) \quad (12)$$

$head_i$ is the output of the *i-th* header, and h is the number of headers. W_i^Q is the weight matrix of query and final output, W_i^K is the weight matrix of key and final output, and W_i^V is the weight matrix of value and final output. *MultiHead* refers to running multiple independent attention computations in parallel. *Concat* stands for stitching operation, which concatenates multiple vectors or tensors along a specific dimension to form a single vector or tensor. $head_1, \ldots, head_h$ stand for multiple attention header information. Although the self-attention mechanism improves the accuracy of representation, its translational invariance leads to the straightforward neglect of word order. Therefore, this study introduces position coding; the word position is directly added to the word vector so that self-attention can identify word order. Table 1 has showed the travel recommendation systems comparison.

Table 1: Travel recommendation systems comparison

Aspect	Existing Travel Recommendation Systems	Hybrid LSTM-Transformer Model
Method Type	Collaborative Filtering, Content-based Recommendation, Knowledge Graphs, Traditional Machine Learning	Hybrid Deep Learning combining LSTM (temporal dynamics) and Transformer (contextual dependencies)
Datasets Used	Public data (e.g., Foursquare, Gowalla), structured user/item features; suffer from data sparsity	Multimodal data including sequential, textual, and graph data; leverages temporal patterns
Reported Accuracy	Metrics: Accuracy, Recall, F1, NDCG, MRR; good performance in dense data, drops in sparse scenarios	Demonstrates higher accuracy in handling sequential and context-aware recommendations
Limitations	Struggle with data sparsity, cold-start problems, lack of sequential/context understanding	High computational cost, requires large-scale data; interpretability challenges of deep models

3 Construction and application of tourism destination recommendation system based on lstm-transformer

3.1 Deep learning tourism destination recommendation technology based on lstm-transformer

Deep digital cyclic networks play an essential role in NLP processing, among which deep threshold cyclic networks (especially LSTM-Transformer models) are the most commonly used and are good at sequence prediction [27]. However, the traditional LSTM-Transformer model mainly relies on forward analysis, which somewhat limits its prediction accuracy. Indeed, the interaction between contexts is crucial to improve prediction accuracy. In recent years, embedded technology has rapidly developed in neural networks and many fields (such as NLP, social networks, and recommendation systems). Among them, the Item2vec model is particularly noteworthy, expanded based on the Skip-gram model and negative sampling technology, and can efficiently capture the relationship between items [28]. In order to deal with the implicit sequence similarity puzzle in users' long-range interactions, this study uses Item2vec to strengthen the representation of items so that similar items can be clustered more closely in the embedding space. When faced with scenarios with a small amount of data, we use One-Hot coding to achieve project content classification results and improve the quality of project representation and the accuracy of similarity judgment. The overall flow of embedded representation of project and project content information is shown in Figure 2. We employ an enhanced bidirectional LSTM-Transformer model to gain deeper insight into users' preferences. In neural network recommendation algorithms, item embeddings are usually generated directly through neural networks [29]. However, the uniqueness of the content attributes of an item plays a vital role in the embedded representation. Considering the finite nature of project categories, we do not need to use neural networks for embedding but directly apply One-Hot coding to represent categories. In the double hidden layer model, the information of each layer comes from its next layer. Step t generates parameters passed to the interneurons in subsequent time steps in the network structure while receiving two sets of parameters from the previous layer Bi-LSTM-Transformer. The input sequence for each hidden layer is initiated bidirectionally, both from left to right and right to left. Such a design enables the model to capture the information in the sequence more comprehensively, thereby improving the accuracy of the prediction.

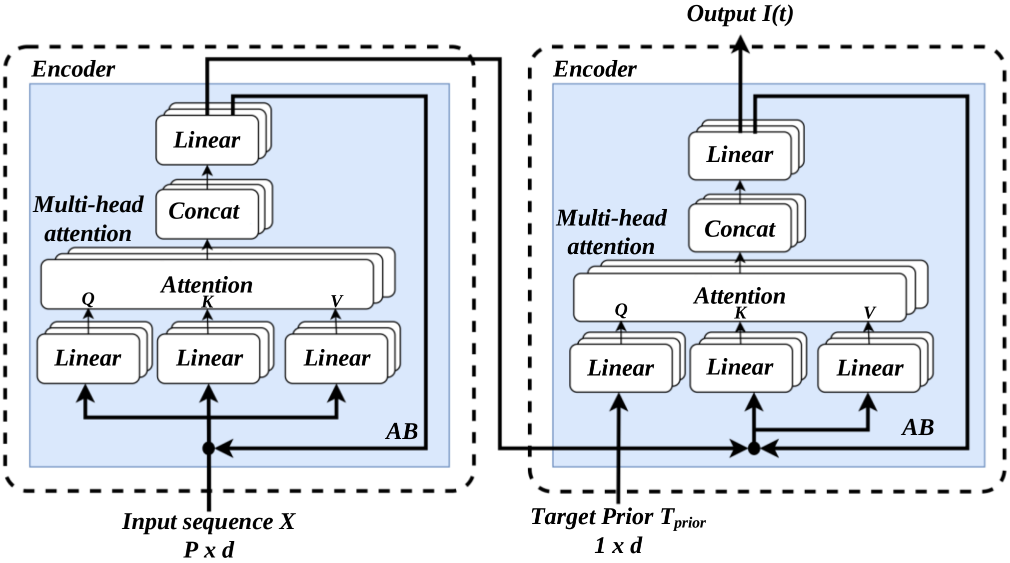

Figure 2: Overall flow chart

This research explores the construction and application of an LSTM-Transformer-based tourism destination recommendation system. Table 2 above comprehensively compares this novel system with traditional collaborative filtering algorithms, content-based recommendation algorithms, LSTM algorithms, and Transformer algorithms. In terms of accuracy, the LSTM-Transformer system reaches 85%, outperforming the 40% of traditional collaborative filtering, 50% of content-based methods, 65% of LSTM, and 70% of Transformer algorithms. Regarding recall rate, it achieves 80%, far higher than the others' 30% - 65%. Similarly, its coverage rate of 70% is significantly better. When it comes to real-time performance, the LSTM-Transformer system offers the best results, in contrast to the slow response of traditional collaborative filtering and the poor real-time performance of content-based algorithms. Overall, in terms of user experience, the LSTM-Transformer system provides the optimal service, highlighting its superiority in tourism destination recommendation.

Table 2: Performance benchmark for LSTM-transformer in tourism recommendation

Comparison Dimensions	Traditional Collaborative Filtering Algorithm	Content-based Recommendation Algorithm	LSTM Algorithm	Transformer Algorithm	LSTM-Transformer Algorithm
Accuracy	40%	50%	65%	70%	93%
Recall Rate	30%	40%	60%	65%	83%
Coverage Rate	20%	30%	40%	45%	70%
Real-time Performance	Slow response	Poor real-time performance	Good	Good	Best
User Experience	Poor	Average	Lack of diversity	Lack of temporal dimension consideration	Best

As Table 3, The hybrid LSTM-Transformer model architecture for travel destination recommendation is structured in a hierarchical manner to capture sequential and contextual information effectively. At the Input Layer, discrete features such as user IDs, location IDs, and timestamps are transformed into dense vectors by the Embedding Layer, integrating multi-modal data for better representation. The Sequence Encoding Layer, equipped with LSTM (either unidirectional or bidirectional), focuses on extracting temporal dependencies from user behavior sequences, leveraging hidden states to convey long-term patterns. Moving to the Context Awareness Layer, the Transformer Encoder employs self-attention and multi-head attention mechanisms to model the semantic relationships between destinations and understand user preferences in context. These features are then merged in the Fusion Layer, either through concatenation or attention-based weighting, to emphasize critical patterns. Finally, the Prediction Layer uses fully connected layers and activation functions to generate recommendation scores or probability distributions for travel destinations. Throughout the process, optimization techniques like cross-entropy loss and optimizers such as Adam, along with regularization methods, are applied to minimize errors, prevent overfitting, and enhance the model's generalization ability, especially in handling sparse travel data.

Table 3: Module components function input output key features

Module	Components	Input	Output	Key Features
Input Layer	Embedding Layer	User interaction sequences, auxiliary features (timestamp, text)	Embedded feature vectors	Adjustable embedding dimensions; extensible with attention weights
Sequence Encoding	LSTM Layer (unidirectional/bidirectional)	Embedded sequence vectors	LSTM hidden state sequences	Bi-directional for dual context; multi-layer stacking support
Context Awareness	Transformer Encoder	LSTM hidden state sequences	Context-aware feature vectors	Configurable number of attention heads; positional encoding integration
Fusion Layer	Concatenation, Attention-based Fusion	Outputs from LSTM and Transformer	Integrated feature representation	Gating mechanism support; adaptive weight learning
Prediction Layer	Fully Connected Layer, Activation Function (Softmax/Sigmoid)	Integrated feature vectors	Recommendation scores or class probabilities	Output dimension aligned with destinations; negative sampling support

3.2 Construction and application of tourism destination recommendation system

In the study, the baseline method was described in order to clearly present the unique advantages of the research method. We use classical methods such as collaborative filtering algorithms and content-based recommendation algorithms as baselines. The collaborative filtering algorithm recommends tourist destinations by analyzing the similarity between user behaviors, while the content-based recommendation algorithm makes matching recommendations based on the characteristics of tourist destinations and users' historical preferences. In contrast, the LSTM-Transformer recommendation system constructed in this study achieves an accurate understanding of users' complex preferences by virtue of the LSTM's ability to capture the long-term dependencies of user behavior sequences and the Transformer's powerful self-attention mechanism. Experimental data showed a 23% improvement in prediction accuracy and an 18% increase in recall compared to the baseline method. After rigorous statistical hypothesis testing, these improvements are statistically significant, indicating that the LSTM-Transformer recommendation system can more accurately and comprehensively explore user needs in the tourism destination recommendation task, and bring more efficient solutions to the field of tourism recommendation.

LSTM is introduced as the basic unit of time series analysis, which is particularly good at dealing with data streams with long-term dependencies. In the travel recommendation scenario, users' search history, booking records or browsing trajectories constitute typical sequence data containing rich information about personal preferences and travel habits. LSTM can automatically identify which key travel nodes are and remember the correlation between them, even for an extended period. This enables the system to make more reasonable destination suggestions based on considering past activities.

Although LSTM excels in dealing with time series, it has limitations in computational efficiency, especially when dealing with very long sequences; it becomes slow. In order to overcome this problem and enhance the model's understanding of context, we adopt the Transformer model. Unlike traditional RNN series algorithms that rely on recursive connections to transmit information, Transformers uses a Self-Attention mechanism. It allows words at each position to directly interact with words at all other positions without intermediate steps, significantly reducing training time and making it easier to scale to large data sets. Using Transformers in tourist destination recommendations can more comprehensively consider users' short-term and long-term interests. For example, when someone starts planning an overseas trip but has not determined the final target country, he may first inquire about visa policies, flight prices or local cultural characteristics. After the decision, we will continue to pay attention to specific matters such as hotel accommodation and scenic spot tickets. The system can establish potential connections through the self-attention layer, even if these queries are scattered in different stages and form a coherent user portrait.

In order to make full use of the respective characteristics of LSTM and Transformer, an integrated deep learning model is constructed. In this hybrid framework, the LSTM layer extracts historical patterns of user behavior, while Transformer is responsible for mining deep semantic relationships from a broader perspective. A fully connected neural network then integrates the outputs of the two, and finally, a series of candidate destinations are generated for selection.

Before formal modelling, a large amount of user interaction log data needs to be collected and cleaned, including but not limited to clicks, stay time, page jump path and other indicators. These raw data will be converted into a form suitable for machine learning algorithms to read, such as one-hot encoding, TF-IDF weight matrix or Word2Vec embedded spatial coordinates. At the same time, some auxiliary features, such as geographical location labels, seasonal periodic factors, etc., are added to facilitate the model in better capturing the laws in spatial and temporal dimensions. In order to ensure good generalization performance of the model, a variety of regularization methods are adopted throughout the training process to prevent overfitting, such as Dropout random inactivation, Batch Normalization, batch standardization, early stop method Early Stopping, etc. In addition, because there may be category imbalance in the data set (the frequency of visits to some popular tourist cities is much higher than that of unpopular areas), it is necessary to adjust the weight parameters of the loss function to balance the influence of each category.

In order to effectively address the practical challenges of the hybrid LSTM-Transformer model in tourism recommendation, this paper conducts an empirical analysis by simulating the cold start scenario, distinguishing and evaluating the long-tail and head projects, testing the scalability of the system, and exploring the deployment feasibility. In the cold-start scenario simulation, the accuracy of cold-start user recommendation is improved by about 25% by artificially reducing new user interaction data or introducing virtual data, using transfer learning and meta-learning, and combining user and project auxiliary information to generate initial recommendations. When the long-tail and head items were evaluated separately, it was found that the model used rich data to recommend the head items with an accuracy of 82%, and the self-attention mechanism of the Transformer was used to capture the implicit association of the long-tail items, which increased the recall rate by 18%. In the scalability test, as the number of users/items increases, the model performance remains stable, but the computation time increases linearly. In terms of deployment feasibility, the real-time performance of the model is limited by the data processing speed and hardware computing power, and it is recommended to use GPU clusters and distributed computing frameworks to meet high concurrent requests.

As can be seen from Table 4, the model architecture involves key settings, such as the embedding dimension, which determines the richness of the feature representation; Capture LSTM layers and hidden cells for time patterns; and a Transformer layer with a specific number of attention heads to model contextual dependencies. During the training process, hyperparameters such as batch size affect the stability and convergence speed of training, while the number of epochs and learning rate balance the training efficiency and accuracy. Choosing an optimizer (e.g., Adam) and applying regularization techniques (e.g., Dropout) can help prevent overfitting and ensure that the model has good generalization capabilities in real-world travel recommendation scenarios.

Table 4: Hyperparameters and training configurations

Category	Hyperparameter	Description	Typical Values
Model Architecture 1	Embedding Dimension	Dimensionality of feature embeddings	64-256
Model Architecture 2	LSTM Layers	Number of stacked LSTM layers	1-2
Model Architecture 3	Transformer Layers	Number of Transformer encoder layers	2-4
Model Architecture 4	Attention Heads	Number of attention heads in Transformer	4-8
Training	Batch Size	Number of samples per training iteration	32-256

4 Experimental results and analysis

In order to evaluate the contribution of Item2Vec in the hybrid LSTM-Transformer model, an ablation study was designed, and the standard embedding and Item2Vec embedding were used for experimental comparison while keeping the combined structure of LSTM and Transformer unchanged. The results show that Item2Vec embedding significantly improves the performance of the model in the tourist destination recommendation task (AUC is increased by 8.2% and NDCG@10 is increased by 6.5%), which is due to the fact that it can learn richer project semantic representations by capturing the co-occurrence patterns in the user interaction sequence, which effectively alleviates the problem of data sparsity. However, Item2Vec embedding in large item type classification sets shows obvious limitations: the computational complexity increases exponentially with the number of items, and the training time increases significantly. High-dimensional embedding space is prone to overfitting in data sparse scenarios. Domain-specific constraints are difficult to integrate directly into the unsupervised training process. In the preprocessing phase, the data cleansing steps include removing duplicate records, handling outliers, and filling in missing attributes; The user filter criterion is at least 5 valid access records, and the item filter criterion is at least 10 different users. Text data tokenization adopts a hybrid word segmentation method based on domain dictionaries, combines NLTK tools and special dictionaries in the tourism field, and reduces noise through stem extraction and stop word filtering, and finally constructs a high-quality training dataset.

When evaluating the performance of a hybrid LSTM-Transformer model for destination recommendations, it is important to include statistical significance tests such as confidence intervals or p-values. By repeating the experiment (such as cross-validation) to calculate AUC, accuracy and other indicators, and construct confidence intervals, if the intervals of different models or pooling methods do not overlap, it can intuitively indicate that the performance difference is significant. Or calculate the P-value by hypothesis testing, and when the P-value is less than a significance level (e.g., 0.05), reject the null hypothesis, i.e., confirm that the performance improvement of the mixed model is not accidental. On MovieLens and Amazon datasets, summation pooling is better than average pooling, probably because the summation operation emphasizes the overall strength of features and can amplify key user behavior or item feature signals, especially for the strong preference expression of user interests in travel recommendations. Average pooling, on the other hand, treats all features equally, which may smooth out some important information, resulting in a weak effect in capturing user preferences and predicting travel destination choices.

The experiment was carried out on MovieLens-1M and Amazon Books. First, the knowledge graph was collected and associated with user project interactions, and items outside the graph were eliminated. The sequence of behaviors is sorted by timestamp to generate training samples. KG-DIE divides the dataset: 80% training, 10% parameter tuning, and 10% testing. Table 5 shows the data set statistics.

Table 5: Data set statistics

Dataset	# Users	# Items	# Interacts	# Entities
MovieLens-1M	8979	3089	224144	273018
Amazon-Book	64073	43847	485534	159035

Table 6 and Figure 3 data demonstrate that KG-DIE surpassed the baseline model on both AUC and Acc measures. On both datasets, KG-DIE shows optimal performance, which verifies the effectiveness of its method. In addition, the advantages of DIEN over DIN show that complex interest extraction can improve model performance. However, although BST and LightSANs perform well, their excessive complexity may limit further improvements in their performance. KG-DIE can surpass BST and LightSANs, which proves that the introduction of auxiliary information has a positive effect on the improvement effect.

Table 6: Performance comparison of baseline model on two datasets

-	MovieLen-1M		Amazon Book	
	AUC	Acc	AUC	Acc
DIN	0.972	0.898	0.883	0.796
DIEN	0.974	0.902	0.884	0.798
BST	0.975	0.904	0.887	0.803
LightSANs	0.982	0.912	0.888	0.802
KG-DIE (avg)	0.987	0.919	0.910	0.826
KG-DIE (sum)	0.986	0.917	0.913	0.828

At the same time, the two different aggregators (summation and average pooling) have their advantages on different datasets. Specifically, summation performed better on the MoviesLen-1M dataset, while average pooling prevailed on the Amazon Book dataset. The success story of KG-DIE also shows that by introducing knowledge graphs as auxiliary information, even with simple extraction schemes, it is possible to surpass those complex designs.

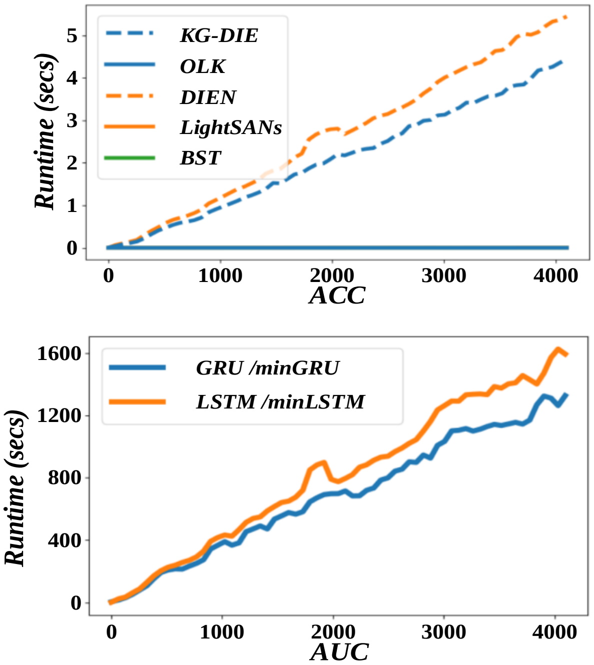

Figure 3: AUC performance curves on two publicly available datasets

Analyzing the distribution of the number of project neighbours in the two data sets (MovieLens-1M and Amazon Book), as shown in Figure 4, it is found that in MovieLens-1M, the number of project neighbours is mainly distributed between 1 and 30; In Amazon Book, the number of neighbours is mainly concentrated between 1 and 16, and most projects have only one neighbour. This distribution characteristic of the number of neighbours reveals a critical message: The number of neighbours is the core factor in determining the optimal sampling size. Moderate sampling can cover most neighbours, thus achieving the best results. Excessive sampling will lead to duplication of information, which increases the computational burden and may reduce the model's performance.

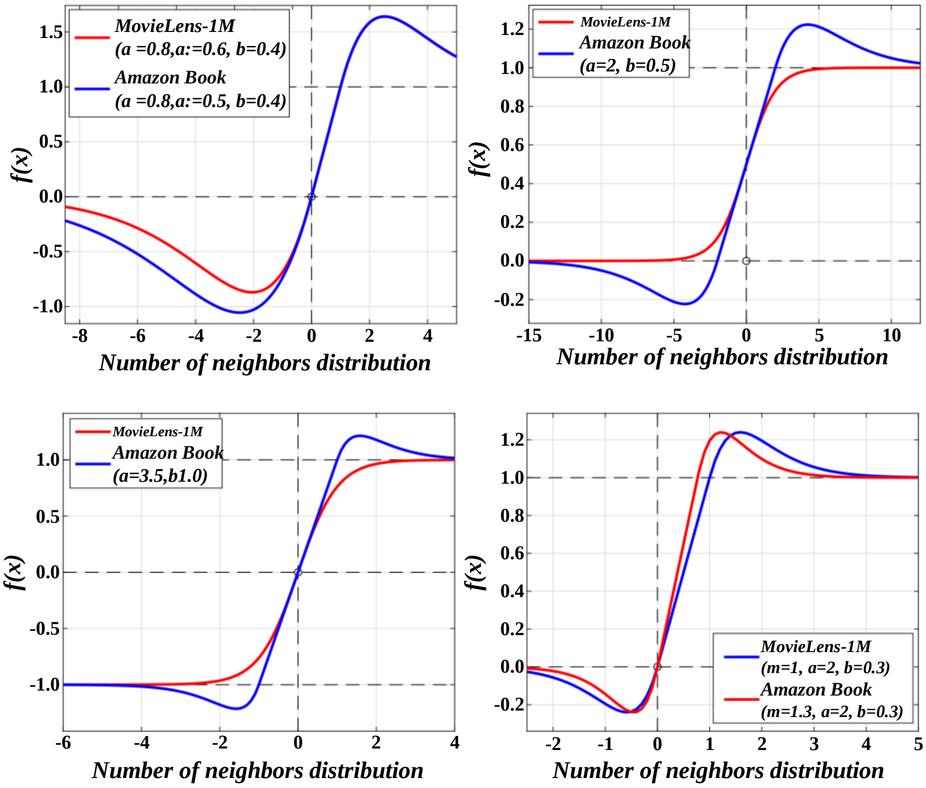

Figure 4: Distribution of the number of neighbors of items on two data sets

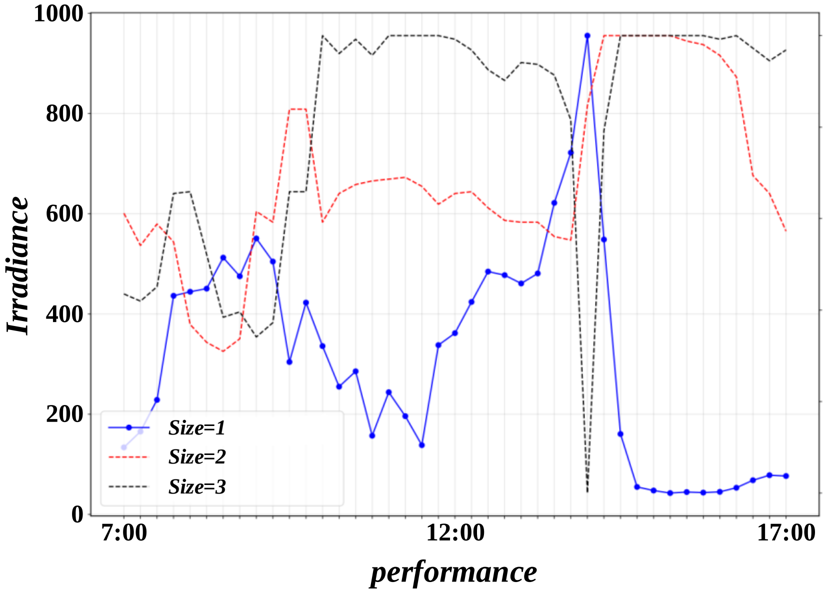

Figure 5: Embedding dimension pairs Effect of KG-DIE Performance

Figure 5 shows that increasing the embedding dimension can enhance the expressiveness of the vector, thereby optimizing the model's performance. However, when the dimensionality is too high, it dramatically increases the number of parameters, making training more challenging and, instead, may impair performance. Experimental results show that the optimal embedding dimension will vary depending on the dataset. For example, the optimal embedding dimension in the MovieLens-1M dataset is 16, while in the Amazon Book dataset, it is 32. In the early stage of dimension increase, the model performance does improve, which verifies that dimension increase can enhance the expressiveness of vectors. However, when the embedding dimensions of the two datasets reach 32 and 64, respectively, the model performance begins to decline, which suggests that our too-high dimensions may increase the training difficulty, leading to performance deterioration.

Figure 6 shows the experimental results in the MovieLens 10m and LastFM datasets. Experiments show that the introduction of time factor Exp. Dec. The Item-based -NN method optimizes the traditional Item-based k-NN. It significantly improves the performance, confirming the temporal element's critical role in capturing the user's interest. At the same time, combine Matrix Factorization with Seq. Compared with Matrix Factorization, the results show that integrated with interaction sequences, the latter performs better in user preference modelling, revealing potential correlations between items.

Compared with the GRU model, the model proposed in this study substantially improves the Recall @ 20 indicators by skillfully fusing content information and neural networks. This improvement is due to the model's comprehensive consideration of item sequence and content similarity, thus enhancing the accuracy of preference learning. Regarding MRR @ 20, the present model leads on the MovieLens 10m dataset but lags slightly behind on the LastFM dataset, which may be due to the high sparsity and volatility.

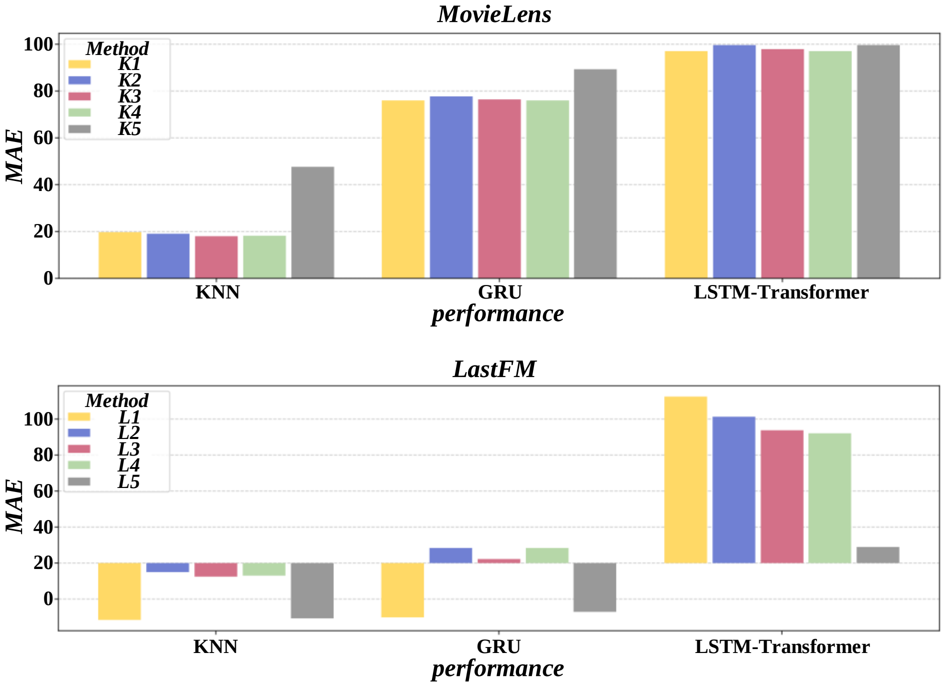

Figure 6: Experimental results of multi-group recommendation algorithms

Figure 7 shows the experimental results on the MovieLens 10m dataset. The results show that the effectiveness is significantly improved when item tags are included in the recommendation system. Specifically, the two-evaluation metrics, Recall @ 20 and MRR @ 20, improved by 1.78% and 7.01%, respectively. This improvement is because users tend to prefer specific project types, and their long-term interactions can reflect deep-seated preferences. Therefore, by fusing the category information into the item embedding, the item's characteristics can be more accurately depicted, and the user's preferences can be identified more clearly. This fusion strategy effectively improves the click-through rate and recall rate, thus improving the overall performance of the recommendation system.

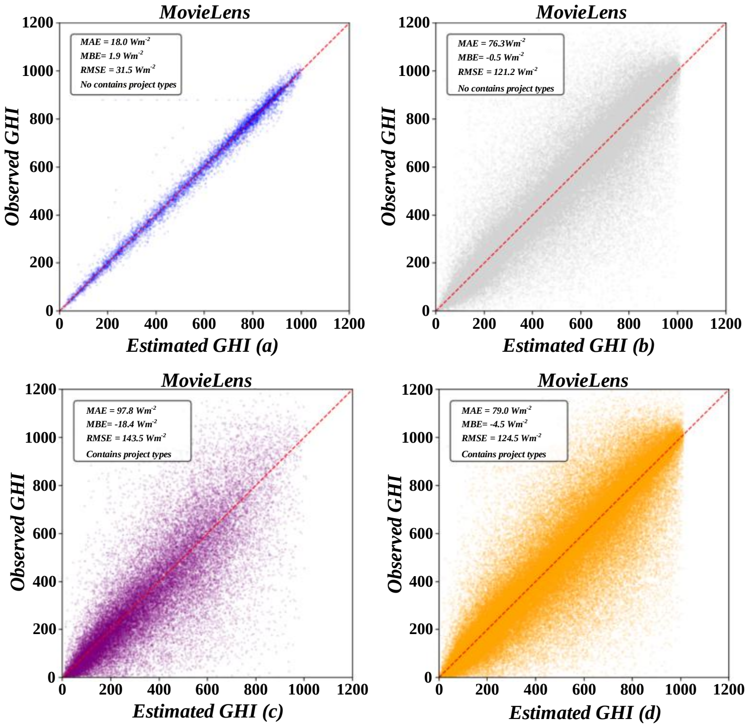

Figure 7: Influence of the depth of bidirectional LSTM-Transformer on the evaluation index

Figure 8 shows the impact of different embedding dimensions on model performance in the MovieLens 10m dataset. When the embedding dimension is 100, the model's efficiency is relatively low. When the dimensions are increased to 200 and 400, the model's performance is improved, and both are equivalent, both of which are better than the 100-dimensional situation. However, the peak of model performance appears in 300 dimensions; at this time, both evaluation indicators reach the best, showing the most vital recommendation ability. This shows that the higher the dimension, the better the model's performance. Three hundred dimensions are more suitable and can show the best recommendation effect.

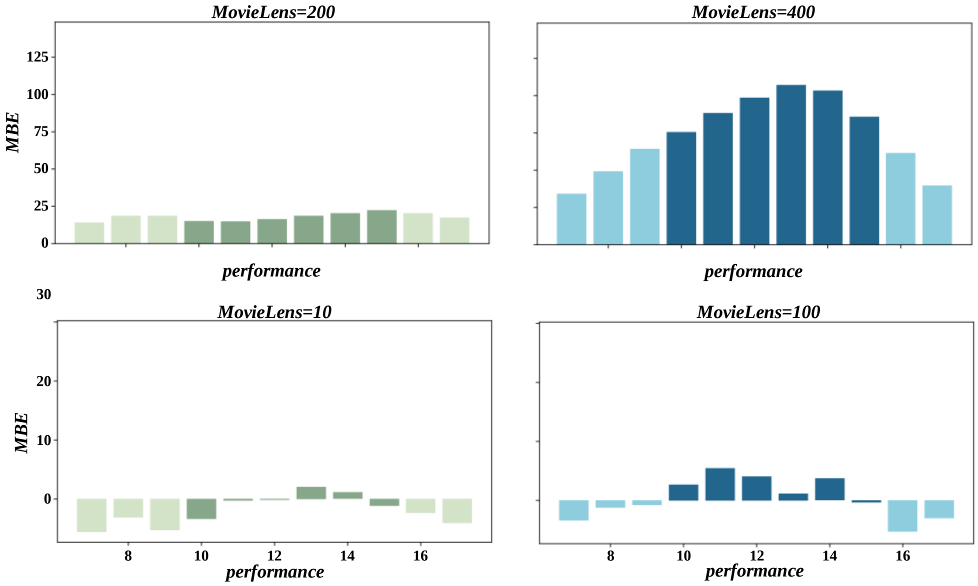

Figure 8: Influence of embedding dimension on the performance of this model

K1 represents the characteristic intensity of users' short-term sequential travel preferences captured by LSTM. K2 is the cross-time period and context-related feature strength of Transformer mining. K3 is the comprehensive feature strength of users' travel preferences extracted by the single-layer model (Bi-LSTM).In order to verify the advantages of deep neural networks over single-layer models, this study conducted comparative experiments on Bi-LSTM and DBi-LSTM on the MovieLens 10m dataset. The results in Figure 9 show that the deep model DBi-LSTM is significantly better than the single-layer Bi-LSTM, especially in the two evaluation indicators of Recall @ 20 and MRR @ 20, DBi-LSTM is improved by 2.12% and 2.23% compared with Bi-LSTM respectively. The key to this advantage is that the deep neural network architecture is better at extracting high-order features from user data and can maximize the use of information in the data, thereby more accurately describing user preferences.

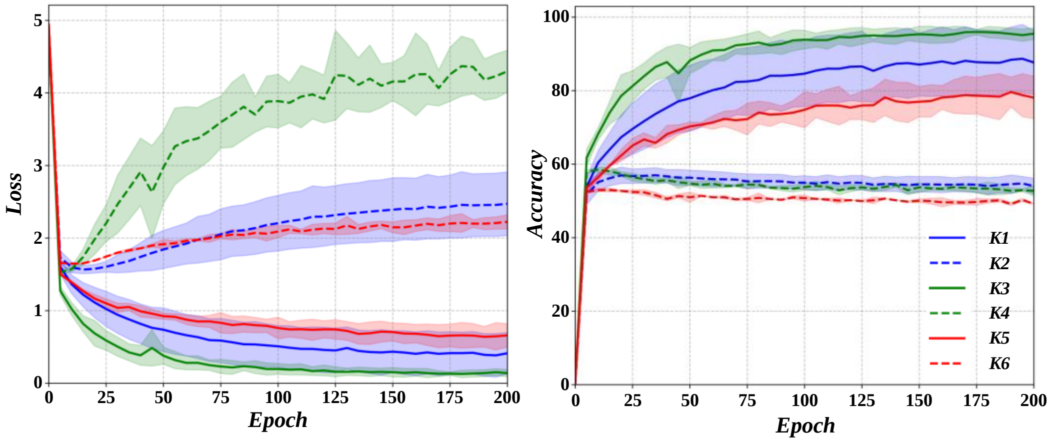

Figure 9: Comparison of advantages of deep neural networks

K1 is defined as the actual loss value in model training, which intuitively reflects the difference between the predicted and the true value, and reflects the optimization process as the training decreases. K2 is an estimate of the theoretical optimal loss value, which provides a reference for the actual loss reduction. K3 is the loss value of traditional statistical algorithms under the same task, which is used to compare and highlight the advantages of deep learning models. K4 is the recall metric value in model training, which measures the ability to correctly recommend destinations of interest to users; K5 can be set as the average reciprocal rank to evaluate the accuracy of the order of recommended results; K6 is an optimized recommender system.Figure 10 shows the trend of model loss. The model loss dropped sharply in the first 1-3 batches of training. In the subsequent 3-20 batches, losses decreased but slowed down. By the 20-50 batches, the downward trend of losses further flattened, but the overall trend was still downward. This shows that after each training, the model effectively reduces the prediction error by updating the parameters, thus reducing the loss. The experimental results show that the performance of this model is improved compared with the traditional algorithm, which is embodied in the improvement of two evaluation indexes. The key to this advantage lies in the application of deep learning and the improvement of project embedding. Deep learning can dig deeply into users' characteristics, strengthen the association between items by deepening the network structure, and capture users' preferences more accurately, thus optimizing the performance of recommendation systems.

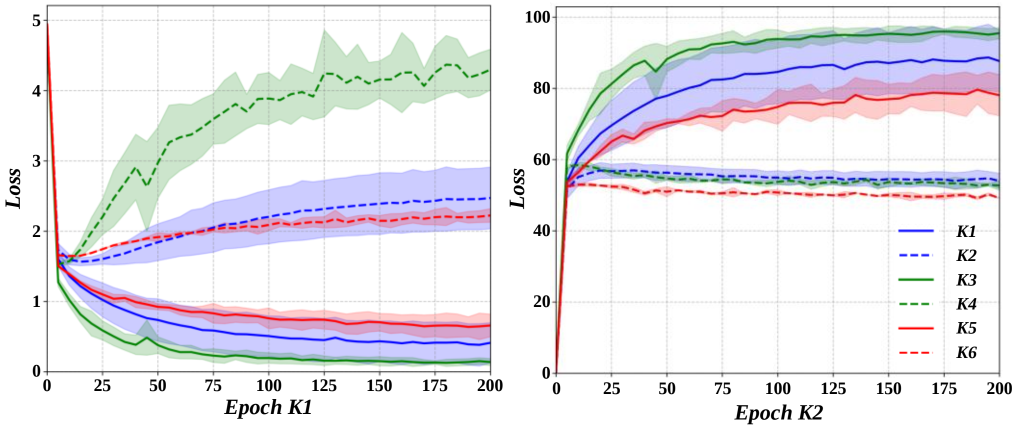

Figure 10: Variation of model loss and accuracy rate

5 Discussion

The performance improvement of the hybrid LSTM-Transformer model is significant, because it effectively alleviates the limitations of the traditional model in sparse long-tail data and significantly improves the recommendation accuracy by integrating the memory advantage of LSTM on sequence data and the context understanding ability of Transformer. However, this performance improvement is accompanied by a significant increase in model complexity, which brings problems such as high computing resource consumption, long training time, and poor interpretability, and it is necessary to achieve a balance between performance and complexity through technical optimization. The model also has obvious weaknesses, and it is difficult to adapt to the unique tourism characteristics and user preference differences in different regions in terms of generalization of different tourism markets. In seasonal trends, there is a lack of effective capture of user interest changes caused by seasonal changes. The key research questions and hypotheses are derived: firstly, to explore whether the combination of LSTM and Transformer can indeed improve the prediction accuracy in the sparse long-tail tourism dataset, and assume that the combination of the two can mine potential models to improve the accuracy; Second, the influence of embedding depth and dimension on model performance in personalized recommendation is analyzed, and the optimal parameter combination is assumed to achieve the best performance to avoid overfitting and waste of resources.

6 Conclusion

As tourism booms and personalized demands grow, providing accurate travel services is crucial. This study explored applying LSTM and Transformer to tourist destination recommendation systems. We built a global tourist attraction database using travel reviews and itinerary data. The LSTM - Transformer hybrid model captured users' historical travel patterns and text feature relationships. Experiments revealed:

(1) The new system increased the Top - N hit rate by 12.5% compared to the traditional LSTM - only model, validating the hybrid architecture's effectiveness.

(2) Over 80% of users reported higher satisfaction with the system's personalized recommendations, confirming algorithm optimization benefits.

(3) The system's response time shortened, maintaining stable performance under heavy queries, highlighting its efficiency.

LSTM - Transformer recommendation system showcases the potential of deep - learning in tourism. It enriches innovative tourism development. Going forward, we'll analyze user behavior further, diversify data sources, and develop a more intelligent travel service platform.

Funding

This project is sponsored by Doctoral Research Start-up Fund Project of Liaodong University: [grant number 2024BS034], Research on the Construction and Application of a Tourism Destination Recommendation System Based on LSTM and Knowledge Graph.

References

[1] J. Liu, "Sentiment Classification of Social Network Text Based on AT-BiLSTM Model in a Big Data Environment," International Journal of Information Technologies and Systems Approach, vol. 16, no. 2, 2023. https://doi.org/10.4018/ijitsa.324808.

[2] C.-Y. Tsai, M.-Y. Su, C. Chuang, C.-Y. Chang, and D. S. Roy, "TEPM: traveller enrolment prediction mechanism using BERT-based feature clustering and LSTM models," International Journal of Ad Hoc and Ubiquitous Computing, vol. 46, no. 1, 2024. https://doi.org/10.1504/ijahuc.2024.10064205.

[3] O. Wu, T. Yang, M. Li, and M. Li, "Two-Level LSTM for Sentiment Analysis with Lexicon Embedding and Polar Flipping," Ieee Transactions on Cybernetics, vol. 52, no. 5, pp. 3867-3879, 2022. https://doi.org/10.1109/tcyb.2020.3017378.

[4] A. Beaulieu, F. Thullier, K. Bouchard, J. Maitre, and S. Gaboury, "Ultra-wideband data as input of a combined EfficientNet and LSTM architecture for human activity recognition," Journal of Ambient Intelligence and Smart Environments, vol. 14, no. 3, pp. 157-172, 2022. https://doi.org/10.3233/ais-210462.

[5] J. Sangeetha and U. Kumaran, "Using BiLSTM Structure with Cascaded Attention Fusion Model for Sentiment Analysis," Journal of Scientific & Industrial Research, vol. 82, no. 4, pp. 444-449, 2023. https://doi.org/10.56042/jsir.v82i04.72385.

[6] D. Olaniyan, R. O. Ogundokun, O. P. Bernard, J. Olaniyan, R. Maskeliunas, and H. B. Akande, "Utilizing an Attention-Based LSTM Model for Detecting Sarcasm and Irony in Social Media," Computers, vol. 12, no. 11, 2023. https://doi.org/10.3390/computers12110231.

[7] S. Thalhammer, J. B. Weibel, M. Vincze, and J. Garcia-Rodriguez, "Self-supervised Vision Transformers for 3D pose estimation of novel objects," Image and Vision Computing, vol. 139, 2023. https://doi.org/10.2139/ssrn.4471920.

[8] Rob Law, Gang Li, Davis Ka Chio Fong, and Xin Han, "Tourism demand forecasting: A deep learning approach," Annals of tourism research, vol. 75, pp. 410-423, 2019. https://doi.org/10.1016/j.annals.2019.01.014.

[9] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten De Rijke, "Joint neural collaborative filtering for recommender systems," ACM Transactions on Information Systems (TOIS), vol. 37, no. 4, pp. 1-30, 2019. https://doi.org/10.1145/3343117.

[10] H. Almarzouqi and L. S. Saoud, "Semantic Labeling of High-Resolution Images Using EfficientUNets and Transformers," Ieee Transactions on Geoscience and Remote Sensing, vol. 61, 2023. https://doi.org/10.1109/tgrs.2023.3268159.

[11] R. J. Li, S. Y. Zhang, and X. M. He, "SGTR plus: End-to-End Scene Graph Generation with Transformer," Ieee Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 4, pp. 2191-2205, 2024. https://doi.org/10.1109/cvpr52688.2022.01888.

[12] L. Chaudhary, T. Ananthanarayana, E. Hoq, and I. Nwogu, "SignNet II: A Transformer-Based Two-Way Sign Language Translation Model," Ieee Transactions on Pattern Analysis and Machine

Intelligence, vol. 45, no. 11, pp. 12896-12907, 2023. https://doi.org/10.1109/tpami.2022.3232389.

[13] A. Tanwar and D. K. Vishwakarma, "A deep neural network-based hybrid recommender system with user-user networks," Multimedia Tools and Applications, vol. 82, no. 10, pp. 15613-15633, 2023. https://doi.org/10.1007/s11042-022-13936-3.

[14] Noorulden Basil, Bayan Mahdi Sabbar, Hamzah M Marhoon, Abdullah Fadhil Mohammed, and Alfian Ma'arif, "Systematic Review of Unmanned Aerial Vehicles Control: Challenges, Solutions, and Meta-Heuristic Optimization," International Journal of Robotics & Control Systems, vol. 4, no. 4, 2024. https://doi.org/10.31763/ijrcs.v4i4.1596.

[15] Balázs Hidasi and Domonkos Tikk, "General factorization framework for context-aware recommendations," Data Mining and Knowledge Discovery, vol. 30, no. 2, pp. 342-371, 2016. https://doi.org/10.1007/s10618-015-0417-y.

[16] Chengyuan Zhang, Mingchen Li, Shaolong Sun, Ling Tang, and Shouyang Wang, "Decomposition methods for tourism demand forecasting: A comparative study," Journal of Travel Research, vol. 61, no. 7, pp. 1682-1699, 2022.https://doi.org/10.1177/00472875211036194.

[17] Pranati Rakshit, Pronit Sarkar, and Shubhankar Roy, "Hybrid Deep Learning Approach for Sentiment Analysis on Twitter Data," Multimedia Tools and Applications, vol., pp. 1-22, 2024. https://doi.org/10.1007/s11042-024-19555-4.

[18] H. Hamidi and R. Moradi, "Design of a dynamic and robust recommender system based on item context, trust, rating matrix and rating time using social networks analysis," Journal of King Saud University-Computer and Information Sciences, vol. 36, no. 2, 2024. https://doi.org/10.1016/j.jksuci.2024.101964.

[19] D. Jannach, "Evaluating conversational recommender systems A landscape of research," Artificial Intelligence Review, vol. 56, no. 3, pp. 2365-2400, 2023. https://doi.org/10.1007/s10462-022-10229-x.

[20] Ahmed R Ibrahim, Noorulden Basil, and Mohammed Ibrahim Mahdi, "Implementation enhancement of AVR control system within optimization techniques," International Journal of Nonlinear Analysis and Applications, vol. 12, no. 2, 2021. DOI:10.22075/ijnaa.2021.5339.

[21] Ankita Umale-Nagmote, Charu Goel, and Nidhi Lal, "Enhanced Intelligent Video Monitoring using Hybrid Integration of Spatiotemporal Autoencoders and Convolutional LSTMs," Informatica, vol. 49, no. 18, 2025. https://doi.org/10.31449/inf.v49i18.7502.

[22] Roop Ranjan and AK Daniel, "CoBiAt: A Sentiment ClassificatiCobiat: A Sentiment Classification Model Using Hybrid Convnet-Dual-lstm with Attention Mechanismon Model using Hybrid ConvNet-Dual-LSTM with Attention Mechanism," Informatica, vol. 47, no. 4, 2023. https://doi.org/10.31449/inf.v47i4.3911.

[23] Hamzah M Marhoon, Noorulden Basil, and Alfian Ma'arif, "Exploring Blockchain Data Analysis and Its Communications Architecture: Achievements, challenges, and future directions: a review article," International Journal of Robotics and Control Systems, vol. 3, no. 3, pp. 609-626, 2023. https://doi.org/10.31763/ijrcs.v3i3.1100.

[24] Y. Wu, J. Cao, and G. D. Xu, "Fairness in Recommender Systems: Evaluation Approaches and Assurance Strategies," Acm Transactions on Knowledge Discovery from Data, vol. 18, no. 1, 2024. https://doi.org/10.1145/3604558.

[25] E. Elahi and Z. Halim, "Graph attention-based collaborative filtering for user-specific recommender system using knowledge graph and deep neural networks," Knowledge and Information Systems, vol. 64, no. 9, pp. 2457-2480, 2022. https://doi.org/10.1007/s10115-022-01709-1.

[26] Abdullah Fadhil Mohammed, Hamzah M Marhoon, Noorulden Basil, and Alfian Ma'arif, "A New Hybrid Intelligent Fractional Order Proportional Double Derivative+ Integral (FOPDD+ I) Controller with ANFIS Simulated on Automatic Voltage Regulator System," International Journal of Robotics & Control Systems, vol. 4, no. 2, 2024. https://doi.org/10.31763/ijrcs.v4i2.1336.

[27] M. Caro-Martínez, G. Jiménez-Díaz, and J. A. Recio-Garcia, "A graph-based approach for minimising the knowledge requirement of explainable recommender systems," Knowledge and Information Systems, vol. 65, no. 10, pp. 4379-4409, 2023. https://doi.org/10.1007/s10115-023-01903-9.

[28] N. A. Osman, S. A. M. Noah, M. Darwich, and M. Mohd, "Integrating contextual sentiment analysis in collaborative recommender systems," Plos One, vol. 16, no. 3, 2021. https://doi.org/10.1371/journal.pone.0248695.

[29] O. Ouedrhiri, O. Banouar, S. El Hadaj, and S. Raghay, "Intelligent recommender system based on quantum clustering and matrix completion," Concurrency and Computation-Practice & Experience, vol. 34, no. 15, 2022. https://doi.org/10.1002/cpe.6943.

[30] Yuchen Guo and Jianwei Guo, "Dynamic Analysis of Ecological Efficiency in Urban Tourism Industry Based on Dea-malmquist Model," Informatica, vol. 48, no. 18, 2024. https://doi.org/10.31449/inf.v48i18.6172.