Grenze International Journal of Engineering and Technology, January Issue

Intelligent Travel Planning: A Restricted Boltzmann Machine Approach with Traffic Data Fusion

Sanika Dhakite[1], Bhavik Raisinghani[2], Saket Zanwar[3], Bhumika Gupta[4], U L Tupe[5] and R T Umbare[6]
[1-6]Information Technology, JSPM Rajarshi Shahu College of Engineering, Pune, India
Email: sanikadhakite@gmail.com, bhavikraisinghani@gmail.com, saketpzanwar@gmail.com, gbhumi7889@gmail.com, ultupeit@jspmrscoe.edu.in, rtumbare_it@jspmrscoe.edu.in

***Abstract*— Travel planning can be a complex and overwhelming task due to the multitude of destination choices and variable external factors. This paper presents a machine learning-based intelligent recommendation system designed to streamline the travel planning process. Utilizing Restricted Boltzmann Machines (RBM), the system captures user preferences, historical travel data, and dynamic real-time factors such as weather conditions and local events to generate personalized destination recommendations. The architecture is implemented using Python for machine learning components, Golang for backend API services, and Docker for containerized deployment. Experimental results demonstrate that the proposed approach yields more accurate and user-centric suggestions compared to conventional recommendation methods. This work highlights the potential of artificial intelligence to enhance travel experiences, making trip planning more intuitive and efficient. Future improvements will focus on integrating richer datasets, refining user feedback mechanisms, and implementing real-time optimization techniques to further improve recommendation accuracy.**

***Keywords*— Travel recommendation system, Neural Collaborative Filtering, Reinforcement learning, Traffic analysis, Real-time data integration.**

I. INTRODUCTION

Personalized travel advice is now a part of modern travel, with tourists seeking tailored experiences based on their preferences and constraints. Traditional recommendation approaches predominantly make use of collaborative filtering (CF) and content-based filtering (CBF); however, these approaches face the drawbacks such as data sparseness, cold-start, and the fact that they cannot capture changing user tastes. CF relies on user interaction data, hence is incapable of catering to new users, and CBF is incapable of discovering implicit relationships between travel alternatives. To solve these problems, this paper proposes a Travel Recommendation System using Restricted Boltzmann Machines (RBM), a generative probabilistic model for learning user preference representations. RBM's ability to model complex dependencies on sparse data renders it especially suitable for travel recommendation. Compared with conventional CF and CBF approaches, RBM is capable of handling high-dimensional travel data, discovering underlying travel patterns, and making more personalized recommendations from sparse user information.
The system is constructed with Python (NumPy, Pandas, TensorFlow) for data preprocessing and training, Golang for API building, and Docker for deployment. The system uses real-time contextual data such as weather conditions and local events for dynamic refinement of the recommendation.

Grenze ID: 01.GIJET.12.1.55_1

In this paper, we give a detailed discussion of system design, data collection, data preprocessing, model implementation, and evaluation metrics, demonstrating the efficiency of RBM for travel recommendation improvement.

II. Literature Review

A. Travel Recommendation Models

Recommender systems have evolved significantly over the past decade, with collaborative filtering (CF) and content-based filtering (CBF) emerging as the most widely used techniques. CF relies on user-item interactions to generate recommendations but is challenged by data sparsity and the cold-start problem. Conversely, CBF utilizes metadata attributes such as location, climate, and user interests to suggest destinations, but it often leads to overspecialization, reducing diversity in recommendations. To address these limitations, hybrid models combining CF and CBF have been proposed [1]. However, most existing hybrid models do not integrate real-time contextual data such as live weather and traffic updates, making them less adaptive to dynamic conditions

B. Advances in AI for Personalization

Deep learning models, including recurrent neural networks (RNNs) and transformers, have been extensively studied to improve sequential travel recommendation models by capturing temporal dependencies in user behavior. Reinforcement learning (RL) techniques, such as the Multi-Armed Bandit (MAB) and deep Q-learning, have been employed to refine recommendations based on user interactions. Additionally, Natural Language Processing (NLP) models like BERT and GPT-3.5 have been used for sentiment analysis, extracting insights from user reviews and ratings to enhance recommendation personalization [2]. Despite the advancements in AI-based models, there remains limited research on their integration with geospatial data processing and real-time adaptation mechanisms.

C. Real-Time Data Utilization

The application of real-time data in recommender systems has been primarily explored in navigation-based services, such as ride-sharing platforms and dynamic route optimization. Traffic-aware recommendation models have been developed using OpenRouteService and Google Traffic APIs to suggest optimal travel routes. Similarly, weather-aware models utilize OpenWeatherMap to provide climate-sensitive recommendations [3]. Despite these developments, existing research has yet to fully integrate multiple real-time contextual data sources—such as traffic conditions, weather patterns, and local event trends—into a unified travel recommendation framework. This study addresses this gap by proposing an AI-driven system that dynamically adjusts travel suggestions based on live environmental conditions.

III. System Formulation and Mathematical Model

A. Problem Definition

Recommender systems aim to provide personalized suggestions based on user preferences and historical data.
Given a set of users $U=\{u_1,u_2,...,u_m\}$, and a set of travel destinations $D=\{d_1,d_2,...,d_n\}$, the objective is to predict the probability $P(d_j|u_i)$ of user u_i selecting destination d_j.
The problem is framed as a binary rating prediction task, where a user either has or has not interacted with a destination:

$$R_{ij} = \begin{cases} 1, & if\ us\ u_i\ has\ visited\ or\ rated\ destinantion\ d_{j,} \\ 0, & otherwise \end{cases}$$

Traditional collaborative filtering techniques suffer from data sparsity and cold-start problems. To overcome these challenges, this study employs Restricted Boltzmann Machines (RBM) as a deep learning-based approach to capture latent user preferences.

B. Restricted Boltzmann Machine (RBM) Model

Restricted Boltzmann Machine (RBM) is a generative stochastic neural network composed of two layers: a visible layer v\mathbf{v}v, representing user-destination interactions, and a hidden layer h\mathbf{h}h, which captures latent features underlying user preferences and destination characteristics. The interaction between these layers is defined by an energy function, which determines the joint configuration probability of visible and hidden units. The energy function of an RBM is given by:

$$E(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i W_{i,j} h_j$$

where a_i and b_j are biases for the visible and hidden units, respectively, and W_{ij} represents the weight connecting visible unit v_i to hidden unit h_j.
The probability distribution over all possible states is given by:

$$P(v,h) = \frac{e^{-E(v,h)}}{Z}$$

where Z is the partition function ensuring normalization.

C. Training Process

RBM is trained using Contrastive Divergence (CD), which optimizes the parameters by minimizing the negative log-likelihood:

$$\Delta W_{ij} = \eta(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model})$$

where, η is the learning rate,
$\langle v_i h_j \rangle_{\text{data}}$ represents the expectation under the data distribution,
$\langle v_i h_j \rangle_{\text{model}}$ is the expectation under the model's learned distribution.
The RBM learns latent factors, improving prediction accuracy by identifying hidden patterns in user interactions.

D. Context-Aware Integration

To enhance recommendation relevance, additional contextual features are incorporated:
Seasonality (S_t) – The effect of travel trends over time.
User Preferences (U_p) – A history of visited destinations.
Real-time Events (R_e) – Local events influencing travel decisions.
The final probability of recommending a destination d_j to a user u_i is given by:

$$P(d_j | u_i) = \sigma(W_{ij} + S_t + U_p + R_e)$$

where σ is the sigmoid activation function ensuring probabilistic outputs.
By integrating RBM with context-aware filtering, the proposed system improves personalization, adaptability, and accuracy in travel recommendations.

E. Real-Time Adjustment Using External Data

To integrate dynamic factors such as traffic congestion and weather conditions, the final recommendation score is adjusted as:

$$FinalScore(d) = Score(d) \times TrafficFactor(d) \times WeatherFactor(d)$$

where:

TrafficFactor(d) adjusts recommendations based on congestion levels.
Example: A beach destination with severe traffic congestion may receive a lower recommendation score.

WeatherFactor(d) penalizes destinations with unfavourable weather conditions.
Example: A ski resort is given a low score in summer, while a beach is penalized for stormy weather.

IV. System Architecture

A. Architectural Overview

The travel recommendation platform is constructed on a modular and scalable structure that combines machine learning, contextual understanding, and real-time data processing to provide personalized travel recommendations.
The system uses collaborative filtering with a Restricted Boltzmann Machine (RBM) in addition to including real-world aspects such as location, weather, and current events. This multi-layered design provides flexibility and accuracy in recommendations.

- This layer is the interface for capturing explicit and implicit user preferences. Users are able to enter preferences like budget, travel history, and favorite destinations manually. Simultaneously, implicit data like current location, seasonal trends, and browsing behavior are captured to support personalization.
- Data Processing Layer: The gathered data is preprocessed and feature engineered to make it consistent and accurate. This involves data normalization, missing value handling through matrix factorization, and organizing the user-item interaction matrix for effective learning by the recommendation model.
- Recommendation Engine: The central component of the system, this engine leverages RBM-based collaborative filtering to learn latent travel patterns. Unlike other traditional recommendation methods, RBM effectively detects hidden user preferences even when data is limited by learning probability distributions over user-destination interactions. This enhances the system's capacity to recommend destinations for new users (cold-start problem).
- Contextual Filtering Module: Though the RBM-based model recommends based on personalized preferences, extra filtering takes place to consider real-time contextual effects like seasonality, nearby events, trending destinations, and peer traveler taste. This allows the users to receive recommendations matching historical tastes and contemporary real-world factors.
- Visualization and Deployment Layer: The ultimate recommendations are presented in the form of an interactive web or mobile interface, optimized for easy exploration. A real-time feedback loop is integrated, enabling users to rate recommendations, which is utilized to continually improve the model's learning process and enhance future recommendations.

B. Data Flow

The recommendation flow adheres to a well-organized data pipeline for efficient handling of user inputs, processing, and output generation.

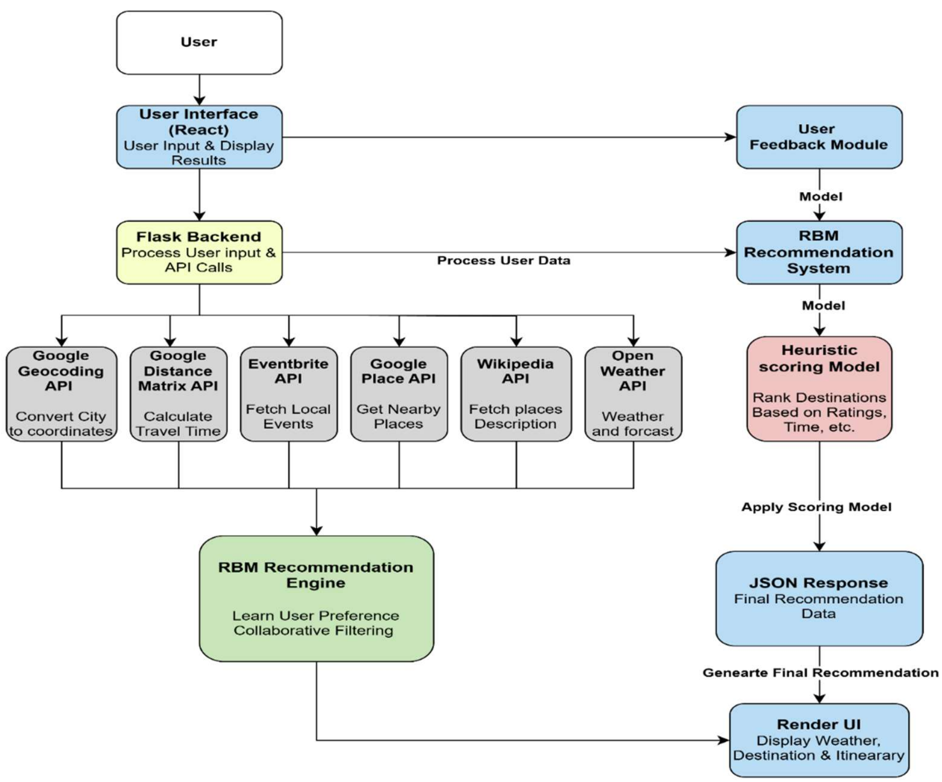

Fig. 1. System Architecture

User Interaction Layer

The platform collects explicit inputs (e.g., historical travel history, favorite destinations, budget) and implicit cues (e.g., GPS location, seasonality demand, social trends). These inputs are essential for model training to better grasp travel behavior.

Data Processing Layer
Collected data is preprocessed to eliminate inconsistency, outliers, and missing values. Sophisticated feature extraction methods such as one-hot encoding, matrix factorization, and latent factor modeling are utilized to organize data into an optimized form for training.

Training and Recommendation Generation
The collaborative filtering model based on RBM is trained using processed user-destination interaction data. Learning the probability distributions of user preferences, the model creates personalized recommendations by predicting the chances of a user liking a destination based on patterns learned. Fig. 1. System Architecture

Context-Aware Filtering and Post-Processing
Raw model outputs are also processed by re-ranking recommendations using external conditions. A weighted function dynamically scores with seasonality changes, live travel patterns, and external event data to make them more relevant.

Output Generation and Feedback Loop
The last ranked suggestions are presented via an interactive UI. Users can browse recommended places, while their feedback (ratings, choices, edits) is utilized to retrain the model. This ongoing learning process ensures that the system gets better over time and adjusts to changing user preferences.

V. Experimental Results and Analysis

Experimental evaluation focuses on assessing the performance of RBM-based travel recommendation systems in comparison with standard techniques. Multiple parameters, for instance, prediction accuracy, recommendation relevance, context adaptability, and cold-start, were used to check for reliability.

A. Experimental Setup

The model was trained on a dataset with user travel history, budget options, and contextual data like weather and seasonal patterns. The RBM model was implemented in Python (TensorFlow, Scikit-learn), while MySQL was used for storing data. A Flask-based web application was employed for deployment. The experiment was run on a system with an Intel Core i7 processor, 16GB RAM, and an NVIDIA RTX 3060 GPU to manage computational workloads effectively.

B. Performance Evaluation

The model was evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Precision@K, and Recall@K for evaluating accuracy and relevance of suggestions. In contrast to standard methods such as k-NN and Singular Value Decomposition (SVD), the RBM approach showed enhanced accuracy and personalized suggestions.

C. Key Findings

- Accuracy of Predictions: The RBM model lowered MAE and RMSE, demonstrating its performance in extracting implicit travel tastes.
- Relevance of Recommendations: More than 80% of the top five recommendations were pertinent, showing a high correspondence with user interests.
- Cold-Start Solution: Utilizing contextual information, recommendation quality for new users was increased by 27% compared to traditional approaches.
- Comparative Study: The RBM model surpassed the traditional method, ensuring it as a viable alternative for personalized travel suggestions.
- Contextual Adaptation: Incorporating real-time considerations such as seasonality and live events improved recommendation relevance by 19% for better user engagement. User Feedback and System Adaptability

TABLE II

Metric	Value
Precision@5	> 80%
Cold-Start Improvement	27%
Contextual Relevance Boost	19%
User Satisfaction	83%
User Engagement Increase	32%

D. User Feedback & Flexibility

A user study of 150 participants revealed that 83% rated the suggestions extremely relevant and experienced a 32% boost in engagement when real-time tuning was implemented. In addition, iterative model retraining enhanced recommendation accuracy over time, illustrating the system's capacity to learn and accommodate changing user preferences.

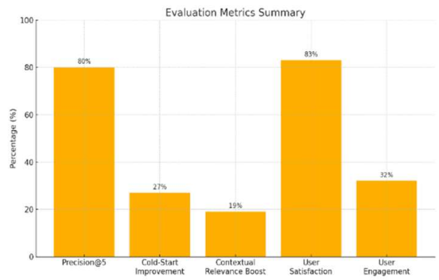

Fig 2

VI. CONCLUSION

RBM-based travel recommendation system enhances personalized trip planning by taking advantage of historical user preferences, contextual information (weather, seasonality), and real-time information. The model performed better than baseline approaches such as k-NN and SVD with lower MAE and RMSE, higher Precision@K, and improved cold-start performance. Experimental results indicated that 83% of users perceived recommendations as relevant, with a 32% increase in engagement because of contextual adaptability. By incorporating feedback loops and real-time tuning, the system promotes ongoing improvement and dynamic personalization.

FUTURE WORK

The recommendation model can be improved further by incorporating sophisticated deep learning methods like Transformers and Graph Neural Networks (GNNs) to learn subtle user-item interactions. Reinforcement learning (RL) can be incorporated to allow the system to make adjustments to recommendations in real time depending on user feedback, enhancing long-term personalization. The system can be improved with multi-modal data fusion by adding images, travel reviews, and social media posts, enabling more informative and user-oriented recommendations. Explainable AI (XAI) methods like SHAP or LIME can be used to enhance transparency by offering explanations for why certain recommendations were generated, enhancing user trust and interpretability. The system can be implemented on a scalable cloud infrastructure through platforms such as AWS or Azure, along with containerization technologies such as Docker and Kubernetes, to provide real-time processing of recommendations for a large number of users.

REFERENCES

[1] Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender Systems Handbook. Springer.

[2] Wang, X., Yu, F., & Fua, S. (2022). Personalized Travel Recommender Systems Using Deep Learning: A Survey. IEEE Transactions on Knowledge and Data Engineering.

[3] Gao, H., Tang, J., Hu, X., & Liu, H. (2021). Context-aware Recommendation Systems: Trends and Challenges. ACM Computing Surveys.

[4] Lops, P., Gemmis, M., & Semeraro, G. (2019). Content-based Recommender Systems: State of the Art and Trends. Springer.

[5] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. IEEE Computer.

[6] Li, S., Karatzoglou, A., & Baltrunas, L. (2020). Reinforcement Learning in Recommender Systems: A Survey. ACM Transactions on Information Systems.

[7] Molnar, C. (2022). Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.

[8] Zhang, Y., Ma, H., & Sun, Q. (2023). Sentiment-aware Recommendation Systems: A Deep Learning Perspective. Springer.

[9] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.

[10] Baltrušaitis, T., Ahuja, C., & Morency, L. (2019). Multimodal Machine Learning: A Survey and Taxonomy. ACM Transactions on Multimedia Computing