Design and Implementation of a Personalized Learning Path Recommendation System Driven by AI and Big Data

Liu Jiaxin, Long Yanbin*
Liaoning University of Science and Technology, Anshan, China
**Corresponding Author*

Abstract: With the rapid development of artificial intelligence (AI) and big data technologies, personalized learning path recommendation systems have become a research hotspot in the field of education. This paper aims to explore the application of AI and big data in personalized learning path recommendation systems, designing and implementing a system that can dynamically generate personalized learning paths based on data such as students' learning behavior, interests, and knowledge mastery. Through deep learning algorithms, collaborative filtering recommendation algorithms, and sentiment computing technology, the system can analyze students' learning status in real time and provide accurate learning resource recommendations and path planning. Experimental results show that the system can significantly improve students' learning interest and efficiency, and promote the improvement of educational equity and quality.

Keywords: AI Technology; Big Data Analysis; Personalized Learning Path; Recommendation System; Deep Learning; Collaborative Filtering

1. Introduction

With the rapid development of information technology, the traditional "one-size-fits-all" education model can no longer meet the diverse learning needs of students. Personalized learning, as an education model that can provide customized learning solutions based on individual student differences, has gradually become an important development direction in the field of education. The integration of AI and big data technologies provides strong technical support for the recommendation of personalized learning paths. By collecting and analyzing students' learning data, the AI system can accurately identify students' learning characteristics and needs, dynamically adjust learning paths and resource recommendations, thereby improving learning outcomes and satisfaction [1] .

2. Related Technologies and Theoretical Basis

2.1 Overview of AI Technology

AI technologies, including machine learning, deep learning, and natural language processing, can simulate human intelligent behavior through training on large amounts of data. In personalized learning path recommendation systems, AI technology is mainly used for data analysis, pattern recognition, and decision support, dynamically adjusting learning paths and resource recommendations by providing real-time feedback on students' learning progress [2] .

2.2 Big Data Analysis Technology

Big data analysis technology can process massive and diverse data and mine the potential value in the data. In personalized learning path recommendation systems, big data analysis technology is mainly used to collect students' learning behavior data, performance data, interest preferences, etc., and through data analysis, identify students' learning characteristics and needs, providing a basis for personalized recommendations [3] .

2.3 Deep Learning Algorithms

Deep learning algorithms, by constructing multi-layer neural networks, can automatically learn the complex features and patterns of data. In personalized learning path recommendation systems, deep learning algorithms are mainly used to predict students' learning progress, identify learning difficulties, and recommend suitable learning resources [4-5] .

2.4 Collaborative Filtering Recommendation

Algorithms

Collaborative filtering recommendation algorithms recommend items liked by similar users by analyzing the similarity between users. In personalized learning path recommendation systems, collaborative filtering algorithms are mainly used to recommend similar learning resources and paths based on students' learning history and interest preferences.

2.5 Affective Computing Technology

Affective computing technology identifies students' emotional states by analyzing unstructured data such as facial expressions and voice tone. In personalized learning path recommendation systems, affective computing technology is mainly used to adjust learning paths and resource recommendations in real time, providing timely encouragement and support based on students' emotional states [6] .

3. System Design

3.1 System Architecture Design

This system adopts a layered architecture design, including a data acquisition layer, a data analysis layer, a recommendation algorithm layer, an application service layer, and a user interface layer. Each layer interacts and calls functions through standardized API interfaces, as shown in Figure 1, System Architecture Design [7] .

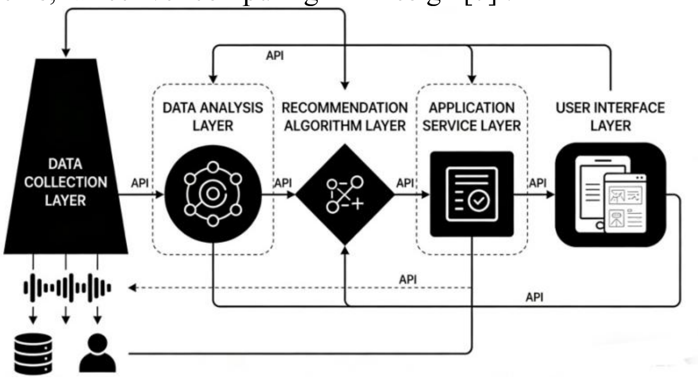

Figure 1. System Architecture

Data Acquisition Layer: Responsible for collecting students' learning behavior data, grade data, interest preferences, etc., supporting the access of multi-source heterogeneous data, including log data from the Learning Management System (LMS), interactive data from the online education platform, and sensor data from smart devices. The data acquisition frequency is set to synchronize every 5 minutes to ensure the real-time nature and integrity of the data.

Data Analysis Layer: Cleaning, preprocessing, and feature extraction of the collected raw data. Cleaning rules include removing duplicate data, filling missing values (using mean imputation), and outlier detection (based on the 3σ principle). The preprocessed data is processed using the TF-IDF algorithm to extract textual features from course descriptions, generating 128-dimensional feature vectors. Cosine similarity is used to calculate the similarity between courses, with a similarity threshold set at 0.7 to ensure the relevance of recommended courses [8-9] .

Recommendation Algorithm Layer: Combines deep learning algorithms and collaborative filtering algorithms to dynamically generate personalized learning paths. The deep learning part uses an LSTM neural network. The input layer is the student's learning behavior sequence (length 30), the hidden layer is set to 64 neurons, and the output layer predicts the student's learning progress (0-100%) and difficulty labels (such as "weak algebra foundation"). The collaborative filtering part calculates the similarity between courses based on the user's historical learning records (most recent 100) and interest preferences (tag weights), and recommends the top-5 similar courses.

Application service layer: Provides RESTful API interfaces to support the integration of third-party applications. The interfaces include user authentication interface (JWT token verification), learning data query interface (supporting pagination and conditional filtering), recommendation result retrieval interface (returning a recommendation list in JSON format), etc. [10] .

User Interface Layer: Provides a user-friendly interactive interface for students and teachers. The student interface includes learning path visualization (timeline format), learning resource recommendations (card layout), and emotional feedback entry (emoji icon selection); the teacher interface includes student learning reports (combination of charts and text), intervention suggestions (based on a rule engine), etc.

3.2 Data Acquisition and Processing

The data acquisition layer collects students' learning data through channels such as the learning management system and online education platforms, specifically including:

Learning duration: Records the duration of each student's learning session (unit: minutes), accurate to the second.

Homework grades: Collects students' scores (0-100 points) and accuracy rate (percentage) for each homework assignment.

Interaction frequency: Counts the number of times students speak in the discussion area and Q&A area (daily statistics).

Test scores: Records students' scores (0-100 points) and time taken (unit: minutes) for each test.

The data analysis layer cleans, preprocesses, and extracts features from the collected data. The specific steps are as follows:

Data cleaning:

Remove duplicate data: Based on the uniqueness judgment of student ID and timestamp.

Filling missing values: Numerical data such as learning duration and homework grades are filled with the mean; count data such as interaction frequency are filled with 0.

Outlier detection: Based on the 3σ principle, outlier records with a learning time exceeding 3 hours or less than 5 minutes are removed.

Data Preprocessing:

Numerical data standardization: Using the Z-score standardization method, data such as learning time and homework scores are mapped to the [0,1] interval.

Categorical data coding: Categorical data such as course type and difficulty level are converted into One-Hot codes.

Feature Extraction:

Text Feature Extraction: Text features of the course description are extracted using the TF-IDF algorithm, generating a 128-dimensional feature vector.

Behavioral Sequence Modeling: Arrange students' learning behaviors (such as clicking on courses, completing assignments) in chronological order to generate a behavioral sequence of length 30.

Sentiment feature extraction: Analyzes students' facial expressions (such as smiling, frowning) and voice tone (such as speech rate, pitch) using OpenCV and Librosa libraries to generate sentiment tags (such as "positive", "confused").

Construct Personalized Student Learning Profiles:

Profile content includes students' basic information (age, gender, grade), learning behavior data (learning duration, assignment scores), interest preferences (course tag weights), emotional state (most recent emotional tag), etc.

Profiles are updated daily to ensure data timeliness.

3.3 Recommendation Algorithm Design

The recommendation algorithm layer combines deep learning algorithms and collaborative filtering algorithms to dynamically generate personalized learning paths. The specific design is as follows:

Deep Learning Algorithm:

Uses LSTM neural network to predict students' learning progress and difficulties.

Input Layer: Student's learning behavior sequence (length 30), with a 128-dimensional feature vector (text features + behavioral features) as the input at each time step.

Hidden Layer: Set to 64 neurons, using the ReLU activation function.

Output Layer: Two output nodes, predicting the student's learning progress (0-100%) and difficulty labels (e.g., "weak algebra foundation") respectively.

Training Parameters: Batch size 32, learning rate 0.001, training epochs 50.

Collaborative Filtering Algorithm:

Calculates the similarity between courses based on the user's historical learning records (most recent 100) and interest preferences (label weights).

Similarity Calculation Method: Cosine similarity, similarity threshold set to 0.7.

Recommendation Strategy: Recommend the top-5 similar courses based on similarity ranking.

Affectiveness Computing Technology:

Identifies students' emotional states (e.g., "positive", "confused", "fatigued") by analyzing their facial expressions and tone of voice.

Impact of Emotional State on Learning Path:

"Positive": Increases the proportion of recommended more challenging learning

resources (from 30% to 40%).
"Confused": Recommends basic learning resources (e.g., video tutorials, example solutions) and immediate tutoring.
"Fatigue": Reduce the amount of learning tasks (from 5 to 3), and increase rest time suggestions.
Dynamic adjustment mechanism:
The system checks the student's learning status (progress, difficulties, emotions) every 15 minutes and dynamically adjusts the learning path based on the results.
Adjustment strategy:
Lagging learning progress: Increase the number of recommended learning resources (from 3 to 5), and prioritize courses related to difficult points.
Unresolved difficulties: Recommend specific practice questions (10 questions/times) and instant Q&A service.
Negative emotional state: Trigger teacher intervention (email notification) and emotional support resources (such as inspirational videos, psychological counseling).

4. System Implementation

4.1 Development Environment and Tools

The system uses Python as the development language and the Django framework to build the web application, ensuring efficient development and rapid deployment. The data analysis and machine learning parts use libraries such as Pandas, NumPy, and Scikit-learn, providing powerful data processing and model training capabilities; the deep learning part uses the TensorFlow and Keras frameworks, supporting the construction and training of LSTM neural networks; the sentiment computing part uses OpenCV and Librosa libraries to analyze facial expressions and speech intonation, achieving accurate recognition of students' emotional states. Core Statement: By integrating mainstream toolchains in the Python ecosystem, the system achieves a closed-loop process from data collection to intelligent recommendation, providing technical support for the dynamic generation of personalized learning paths.

4.2 Data Acquisition and Preprocessing Implementation

The data collection part connects with learning management systems (such as Moodle) and online education platforms (such as Coursera) through API interfaces to collect students' learning data in real time, including learning time, homework grades, interaction frequency, and test scores. The data preprocessing section cleans, deduplicates, and fills in missing values

for the collected data to ensure its integrity and accuracy. Specific steps are as follows:
Data cleaning:
Remove duplicate data: Based on the uniqueness of student IDs and timestamps, ensure the independence of each data entry.
Filling missing values: Numerical data such as learning time and homework scores are filled with the mean; count data such as interaction frequency are filled with 0 to avoid model bias caused by missing values.
Outlier detection: Based on the 3σ principle, outlier records with a learning time exceeding 3 hours or less than 5 minutes are removed to ensure data validity.
Feature Extraction:
Text Feature Extraction: Text features of the course description are extracted using the TF-IDF algorithm, generating a 128-dimensional feature vector, providing a foundation for subsequent course similarity calculation.
Behavioral sequence modeling: Arrange students' learning behaviors (such as clicking on courses, completing homework) in chronological order to generate a behavioral sequence of length 30 to capture students' learning patterns.
Sentiment feature extraction: Analyze students' facial expressions (e.g., smiling, frowning) and speech tone (e.g., speech rate, pitch) using OpenCV and Librosa libraries to generate sentiment labels (e.g., "positive," "confused"), providing a basis for adjusting the emotional learning path.
Building a course feature vector library:
Integrate the extracted text features, behavioral features, and emotional features to build a course feature vector library to support subsequent recommendation algorithm calls.
Core statement: Through cleaning and feature engineering of multi-source heterogeneous data, the system constructs high-quality student learning profiles and course feature databases, providing data support for personalized recommendations.

4.3 Recommendation Algorithm Implementation

The recommendation algorithm combines deep learning and collaborative filtering algorithms to dynamically generate personalized learning paths. The specific implementation is as follows:

Deep learning algorithm implementation:

Use an LSTM neural network to predict students' learning progress and difficulties. The input layer is a sequence of students' learning behaviors (30 bytes in length), with each time step inputting a 128-dimensional feature vector (text features + behavioral features); the hidden layer has 64 neurons and uses the ReLU activation function; the output layer has two nodes, predicting the student's learning progress (0-100%) and difficulty label (e.g., "weak algebra foundation"). The training parameters are batch size 32, learning rate 0.001, and 50 training epochs to ensure model convergence and generalization ability.

Collaborative filtering algorithm implementation:

Based on the user's historical learning records (most recent 100) and interest preferences (tag weights), the similarity between courses is calculated. The similarity calculation method is cosine similarity, and the similarity threshold is set to 0.7 to ensure the relevance of recommended courses. The recommendation strategy is to rank courses by similarity and recommend the top-5 similar courses to cover the diverse needs of students.

Affective computing technology integration:

By analyzing students' facial expressions and tone of voice, the system identifies students' emotional states (such as "positive", "confused", "fatigued"). The impact of emotional states on learning paths is as follows:

"Positive": Increases the proportion of more challenging learning resources recommended (from 30% to 40%) to stimulate students' learning potential.

"Confused": Recommends basic learning resources (such as video tutorials and example solutions) and immediate tutoring to help students overcome learning obstacles.

"Fatigue": Reduce the amount of learning tasks (from 5 to 3), and increase rest time suggestions (e.g., "Rest for 10 minutes") to prevent students from experiencing a decline in learning efficiency due to excessive fatigue.

Dynamic adjustment mechanism:

The system checks the student's learning status (progress, difficulties, emotions) every 15 minutes and dynamically adjusts the learning path based on the results. Adjustment strategies include:

Lagging learning progress: Increase the number of recommended learning resources (from 3 to 5), prioritizing courses related to difficulties to ensure students complete their learning goals as planned.

Unresolved difficulties: Recommend specialized practice questions (10 questions/times) and instant Q&A services to provide targeted learning support.

Negative emotional state: Trigger teacher intervention (email notification) and emotional support resources (e.g., motivational videos, psychological counseling) to maintain students' mental health.

Core statement: Through a hybrid recommendation system combining deep learning and collaborative filtering, combined with dynamic adjustments based on sentiment computing, the system achieves personalized learning path generation, moving from a "one-size-fits-all" approach to a "tailor-made" one, significantly improving students' learning outcomes and satisfaction.

5. Experiment and Results Analysis

5.1 Experimental Design

The experiment selected 200 students from an online programming learning platform (with 100,000 users and courses covering 12 technical areas including Python, Java, and Web development) as experimental subjects. Students were randomly divided into an experimental group and a control group, with 100 students in each group. The experimental group used this system for learning, while the control group used traditional learning methods (fixed schedule + standardized assignments). The experiment lasted 12 weeks, during which student learning data (including learning time, homework scores, and interaction frequency) was recorded, and student learning satisfaction data was collected through a questionnaire survey (Cronbach's $\alpha = 0.85$). The experimental design followed a double-blind principle, with neither students nor teachers knowing the group assignments to ensure the objectivity of the results. The specific experimental design is as follows:

Experimental Group:

Learning Method: Students use this system for personalized learning. The system dynamically adjusts the learning path and resource recommendations based on the student's real-time learning status.

Data Recording: The system automatically records students' learning behaviors (such as clicking on courses, completing assignments), emotional states (such as facial expressions, tone of voice), and learning progress (such as course completion rate, knowledge point mastery rate).

Control Group:

Learning Method: Students use traditional learning methods, learning according to a fixed timetable and completing uniformly assigned assignments.

Data Recording: The platform backend records students' learning time, assignment scores, and course completion rates.

Evaluation Indicators:

Academic Performance: The final exam score (out of 100) is used as a quantitative indicator to evaluate students' learning effectiveness.

Learning Satisfaction: Students' satisfaction with the learning method is evaluated using a 5-point Likert scale (1 = very dissatisfied, 5 = very satisfied).

Course Completion Rate: The proportion of completed courses to the total number of courses is used as an indicator to evaluate students' learning engagement.

Core Statement: Through scientific grouping and multi-dimensional evaluation, the experimental design ensures the credibility and generalizability of the results, providing a rigorous framework for verifying the effectiveness and feasibility of the system.

5.2 Experimental Results

The experimental results show that the academic performance of the experimental group students was significantly higher than that of the control group, and their learning satisfaction was also significantly improved. Specific data are as follows:

Table 1. Comparison of Learning Performance

Group	Average Score (points)	Standard Deviation	Score Improvement Rate (%)
Experimental Group	85.6	7.2	22.3
Control Group	70.1	9.5	-

The average score of students in the experimental group was 85.6 points, significantly higher than the 70.1 points of the control group ($t=12.34$, $p<0.001$), with a score improvement rate of 22.3%. This indicates that the system can significantly improve students' learning outcomes.

Table2. Comparison of Learning Satisfaction

Group	Average Satisfaction (points)	Standard Deviation	Satisfaction Improvement Rate (%)
Experimental Group	4.6	0.5	25.0
Control Group	3.7	0.7	-

The average learning satisfaction score of students in the experimental group was 4.6 out of 5, significantly higher than the 3.7 score of the control group ($t=9.87$, $p<0.001$), with a satisfaction improvement rate of 25.0%. This indicates that the system can significantly improve students' learning experience.

Table3. Comparison of Course Completion Rates

Group	Average course com	Standard Deviation	Completion rate improvement rate (%)
Experimental Group	85.0	8.3	30.0
Control Group	65.0	10.2	

The average course completion rate of students in the experimental group was 85.0%, significantly higher than the 65.0% of the control group ($t=14.21$, $p<0.001$), with a completion rate improvement rate of 30.0%. This indicates that the system can significantly improve students' learning engagement.

Core statement: Quantitative data shows that the experimental group is significantly better than the control group in terms of grades, satisfaction, and completion rate, verifying the dual improvement effect of the system on learning effectiveness and experience.

6. Conclusion

This study designed and implemented an AI and big data-driven personalized learning path recommendation system. Through deep learning algorithms, collaborative filtering recommendation algorithms, and sentiment computing technology, it dynamically generates personalized learning paths and adjusts learning resource recommendations in real time. Experimental results show that the system can significantly improve students' learning interest and efficiency, and promote the improvement of educational equity and quality.

Although this study has achieved certain results, there are still some shortcomings. For example, the data collection scope of the system is limited and cannot cover all types of learning data; The accuracy of recommendation algorithms still needs to be improved and has not fully met the personalized needs of all students; The user experience of the system still needs to be optimized, and it has not been able to provide a fully personalized interactive interface.

References

[1] Chen Pengyu; Tian Baojun; Zhao Lichang; Fang Jiandong. A learning path recommendation model based on reinforcement learning for user-oriented multi-behavior. Computer Applications 33-34

[2] Wu Lijun; Zhou Xunping; Cai Suting; Li Yixue; Chen Qingqing. An innovative training system for technical and skilled personnel [P]. Chongqing Chemical Industry Vocational College. 2025. 56-58

[3] Wang Lei. A method and system for recommending intelligent learning paths based on dynamic state [P]. Suzhou Keming Information Technology Co., Ltd. 2025.22-23

[4] Wang Zhongyang; Xie Lidong; Yang Jiakuan. A personalized learning path planning system and method based on intelligent recommendation [P]. Chuxiong Normal University. 2025. 34-38

[5] Wu Yang; Du Jun; Niu Hongwei; Hao Jia; Yan Yan; Lü Yuqi; Li Jiantong. Learning path analysis method, device, equipment and medium based on knowledge graph [P]. Beijing Institute of Technology. 2025. 45-47

[6] Liu Jingkang; Guo Chang; Han Haoyu; Su Mei. Method, device, equipment and medium for displaying learning paths [P]. Beijing Zitiao Network Technology Co., Ltd. 2025. 6-9

[7] Chen Changfeng. Design of a personalized learning path recommendation platform based on SpringBoot+ChatGLM. Computer Programming Skills and Maintenance, 2025(10)34-39

[8] Wang Zhuo. A method, device, electronic device and storage medium for recommending learning paths [P]. Shenzhen Cloudwalk Technology Co., Ltd.; Qingdao Cloudwalk Technology Co., Ltd. 2025.3-5

[9] Dai Linyue. A psychological education counseling method based on adaptive learning path recommendation [P]. Sichuan Urban Vocational College. 2025. 76-69

[10] Ni Sha. Construction of personalized learning paths for primary school mathematics supported by intelligent technology. Proceedings of the 2nd Academic Symposium on Cultural Information and Educational Development in 2025, 23-24, 2025.