Design and Implementation of an English Mobile Learning System Based on Weighted Naive Bayes

Ying Ye

School of Foreign Languages, Wuhan Business University, Wuhan, Hubei, 430000, China

Email: 20150670@wbu.edu.cn

Abstract. This study employs the Weighted Naive Bayes algorithm in order to enhance the efficiency of personalized English learning in a mobile learning environment. By introducing a weighting factor, the traditional Naive Bayes classifier is optimized for an English mobile learning system design. Meanwhile, the application of the Naive Bayes algorithm and weighting techniques in mobile learning is analyzed in details, including algorithm selection, optimization, weight distribution, and model training. The results indicate that the English mobile learning system, optimized with the Weighted Naive Bayes algorithm, significantly improves learning outcomes, accuracy of personalized recommendations, and security of the learning process. Thus, it can effectively support English teaching and learning in a mobile learning environment.

Keywords: Weighted Naive Bayes; English Mobile Learning System; Weight Distribution; Model Training.

1 Introduction

Mobile learning, as an important branch of modern educational technology, has broken the traditional time and space constraints of English education with the widespread use of mobile devices and the development of network technology, providing users with the possibility to learn anytime and anywhere. With the application of mobile learning, the challenges of insufficient personalized services and system security are becoming increasingly prominent [1]. Especially in the context of diverse learning needs and complex network environments, enhancing the adaptability and security of the learning system has become an urgent issue to address. The Naive Bayes algorithm, due to its simplicity and effectiveness, is widely used in classification problems across many fields. However, the standard Naive Bayes algorithm has limitations in handling personalized learning data, especially in situations of data imbalance and strong feature dependency. The openness of the mobile learning environment also makes the security of learning systems particularly important [2]. Therefore, this study proposes the use of the Weighted Naive Bayes algorithm to address these issues. By introducing a weighting mechanism, the model's adaptability to personalized learning behavior is enhanced, and the accuracy of the recommendation system is improved. The design of the English mobile learning system, fro m both the algorithmic and

M. Yu et al. (eds.), *Proceedings of the 2024 5th International Conference on Big Data and Informatization Education (ICBDIE 2024)*, Advances in Intelligent Systems Research 182,
https://doi.org/10.2991/978-94-6463-417-4_17

system design perspectives, creates an efficient and secure English mobile learning environment.

2 Research Methodology

2.1 Naive Bayes Algorithm and Weighting Technique

The Naive Bayes algorithm is a simple probability classifier based on Bayes' theorem, which assumes that each feature is independent of the others. This assumption simplifies the model's computational process. However, in practical applications, the interdependence of features can limit the algorithm's performance [3]. To address this issue, this study introduces a weighting technique. By assigning different weights to different features or samples, it reflects their importance in classification decisions, enhancing the model's adaptability to data characteristics and classification accuracy. Specifically, the Weighted Naive Bayes algorithm dynamically adjusts weights during the model training process, driven by data, to optimize model parameters. This enables it to better capture and utilize information in the training data [4]. This approach not only improves the model's ability to recognize specific user learning behaviors but also enhances the adaptability of the English mobile learning system to different learning environments. By precisely adjusting weights, the Weighted Naive Bayes algorithm can significantly improve the level of personalized service and user satisfaction in English mobile learning systems, while maintaining the advantages of the original algorithm.

2.2 Requirements for English Mobile Learning System

The requirements for an English mobile learning system focus on creating a flexible, efficient, and user-friendly learning platform that meets the diverse needs of learners and provides them with a wealth of learning resources and pathways. Firstly, the system should offer a personalized learning experience, including adaptive learning paths, personalized content recommendations, and customized learning plans to cater to the abilities and preferences of different learners [5]. Secondly, efficient interactivity is a key requirement for the English mobile learning system. The system should support real-time feedback, peer interaction, and communication with teachers to enhance the appeal of English learning. Thirdly, considering the characteristics of English learning, the system should include a variety of learning materials, such as texts, audio, and videos, and be equipped with voice recognition and natural language processing technologies to aid in pronunciation practice and language skill enhancement. To support flexible learning, the system should also be designed to be easily accessible and user-friendly, with good mobile adaptability and stable performance [6]. Data analysis and learning assessment functionalities are important components of the English mobile learning system, aimed at collecting and analyzing learning data to evaluate learning outcomes and provide valuable insights to learners. In summary, the requirements for the English mobile learning system are focused on providing a comprehensive, personalized, and highly interactive learning environment to foster continuous progress in English learners.

3 System Design and Implementation

3.1 Overall System Architecture

The English mobile learning system adopts a layered architectural design to ensure maximization of modularity and scalability. This architecture is divided from top to bottom into several layers: User Interface Layer, Application Logic Layer, Content Management System Layer, Security and Privacy Protection Layer, and Server-Side Technology Layer, （As shown in Figure 1.） Specifically: The User Interface Layer, at the top, is responsible for providing a diverse and responsive interactive interface. It optimizes user interaction experience and adapts to various devices. The Application Logic Layer undertakes the personalized recommendation engine and interactive feedback mechanism. It utilizes the Weighted Naive Bayes algorithm for in-depth learning and analysis of user behavior, achieving personalized adaptation of teaching content. The Content Management System Layer is dedicated to processing and distributing rich educational resources, ensuring real-time updates and efficient management of learning content [7].The Data Analysis and Evaluation Layer provides quantitative assessments of learning outcomes and data support for the iterative optimization of teaching strategies by synthesizing learning data. The Security and Privacy Protection Layer focuses on data encryption, access control, and security auditing to ensure the integrity and privacy of user data[8]. The Server-Side Technology Layer, at the bottom, includes database management, server programming, and network security, providing stable and reliable technical support for the entire system. The overall architecture tightly integrates each layer through precisely defined interfaces and protocols, ensuring overall system performance, reliability, and maintainability, thereby constructing a professional, efficient, and secure English mobile learning system.

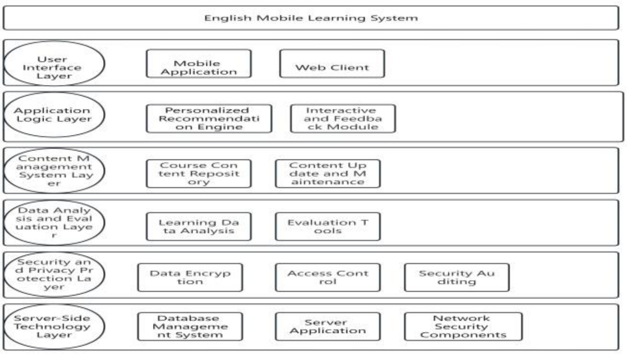

Fig. 1. Overall Architecture Diagram of the English Mobile Learning System

3.2 Construction of the Weighted Naive Bayes Model

1) Algorithm Selection and Optimization.

The standard Naive Bayes algorithm is chosen as the base due to its conditional independence assumption. This assumption considers the contribution of each feature independently, thereby simplifying the model's computational complexity. The basic form of the Naive Bayes algorithm can be expressed as shown in (1):

$$P(C_k|x) = \frac{P(C_k)\prod_{i=1}^{n} P(x_i|x)}{P(x)} \tag{1}$$

Where: C_k Target category" or "desired class label, x Represents a feature vector, x_i Is a specific feature, $P(C_k|x)$ Is given the feature vector belongs to the category C_k The probability of.

To adapt to the English learning environment, the study introduces a weighting mechanism, modifying the algorithm to consider the varying importance of different features. The Weighted Naive Bayes model can be expressed as shown in (2):

$$P(C_k|x) = \frac{P(C_k)\prod_{i=1}^{n} P(x_i|C_k)^{w_i}}{P(x)} \tag{2}$$

In the model, w_i and x_i represent the weights associated with the respective features. These weights reflect the importance of each feature in the classification decision, which can be determined through the performance of features in the training data or through expert knowledge.

During the optimization process, it is necessary to determine the optimal value for weight w_i. Cross-validation can be used to ensure that the selected weight maximizes overall classification performance. Particularly in the context of English learning, different areas of language learning (such as vocabulary, grammar, listening, etc.) require different weight settings. Considering the user's personal learning history and preferences, dynamic weight adjustment is required to further enhance the system's personalization and accuracy. The Weighted Naive Bayes model not only retains the simplicity and efficiency of the original Naive Bayes algorithm but also enhances its capability to handle complex English learning data by introducing weights. This provides robust algorithmic support for constructing highly personalized mobile English learning systems.

2) Weight Distribution.

The key to weight distribution is to determine the importance of each feature in the classification decision and adjust its influence accordingly. The system's weight distribution is based on statistical information of features, and the formula for calculating feature weights is shown in (3):

$$w_i = \log \frac{P(x_i|C_1)}{P(x_i|C_2)} \tag{3}$$

Where: w_i is the weight of feature x_i, $P(x_i|C_1)$ and $P(x_i|C_2)$ are the conditional probabilities of under two different categories of C_1, C_2.

In the English mobile learning system, weight distribution must consider specific needs for English learning. For instance, for beginners, a higher weight would be assigned to vocabulary and basic grammar, as these are key parts of foundational learning. For advanced learners, more emphasis is needed on understanding and applying complex sentence structures, thus increasing the weight for these features. Weight distribution is not limited to static settings but can also be dynamically adjusted according to the user's learning progress. By tracking user learning activities and outcomes, the system can adjust weights to reflect the user's current learning needs and preferences. For example, if a user shows rapid progress in a certain area, the weight for related features can be appropriately reduced, prompting the system to recommend new learning content [9]. Through the weight distribution mechanism, the Weighted Naive Bayes model can more precisely adapt to the personalized learning needs of users, enhancing the instructional effectiveness and user experience of the English mobile learning system.

3) Model Training.

During the training process of the Weighted Naive Bayes model, it is necessary to collect a large amount of English learning-related data from real-world application scenarios. This data includes user learning behavior records, quiz results, interaction logs, etc. Using this data, initial parameters are set, calculating the prior probability of categories $P(C_k)$ and the conditional probability of each feature under each category $P(x_i|C_k)$, and determining the initial feature weights w_i. The model is trained using the gradient descent algorithm, iteratively updating the parameter values, especially the feature weights, to maximize the model's predictive performance.

Throughout the model training process, cross-validation is used for model evaluation to avoid overfitting and ensure the model has good generalization capabilities. The computational efficiency of the model is also taken into account during training to ensure that the model can operate efficiently in a real mobile learning environment. As new data accumulates and user needs change, the model is regularly updated to adapt to the dynamic changes in the English learning environment, ensuring long-term efficiency and accuracy in personalized recommendations.

Through this comprehensive training approach, the Weighted Naive Bayes model can be effectively applied in the English mobile learning system, providing users with a high-quality personalized learning experience.

3.3 System Implementation

The first step in system implementation involves building the framework and choosing an appropriate technology stack. Modern frameworks like React and Flutter are used to develop responsive front-end interfaces, while back-end services utilize technologies such as Node.js, Python Flask, or Django to ensure interactive and efficient data processing. For database management, solutions like MySQL are chosen to store user data, learning content, and interaction records. In terms of model integration, the Weighted Naive Bayes model is embedded into the back-end services, processing user data to implement personalized learning path recommendations. System implementation also involves user interface design, ensuring it is intuitive and user-friendly, and adaptable to various screen sizes and operating systems. Core functionalities provided include course browsing, learning activities, and interactive testing. Security implementation includes data encryption, secure authentication, and access control to ensure user data security [10].For system deployment, suitable cloud service platforms are selected to leverage their elastic computing and storage resources, ensuring system stability and scalability. Throughout the implementation process, continuous testing and optimization are conducted, including functional testing, performance testing, and user experience testing, to ensure stable system operation and a positive user experience. Through this comprehensive design approach, the English mobile learning system effectively supports user learning activities, offering a personalized, interactive, and secure learning environment.

4 System Testing

4.1 Experimental Environment and Tools

The experimental environment encompasses multiple versions of Android and iOS devices, covering a wide range of smartphones and tablets, and simulating Wi-Fi, 4G, and 5G network conditions. This multi-device and multi-network setup ensures comprehensive system testing under various conditions. For testing tools, Selenium is used for interface and functionality testing, JMeter for performance testing, and OWASP ZAP for security testing. These automated testing tools are capable of simulating user operations, assessing system performance, and identifying potential security vulnerabilities. Additionally, data analysis tools like Google Analytics are deployed to monitor system operation and user behavior in real time, collecting feedback for subsequent optimization. Through these professional tools and meticulous environment configurations, the thoroughness of the system testing is ensured, providing a solid foundation for the stable operation and optimization of the English mobile learning system.

4.2 Experimental Procedure Design

In the testing of the English mobile learning system, for functional testing, the Selenium automation testing framework is used to simulate user operations. Specific testing metrics include the correctness of functionalities such as course access, interactive test submissions, and user settings adjustments. In performance testing, JMeter is employed to simulate 1,000 concurrent users accessing the system, assessing the system's response time and resource consumption under high load to ensure stable operation in high-concurrency scenarios. For security testing, OWASP ZAP is used to conduct penetration tests on the system to detect common security vulnerabilities such as SQL injection and cross-site scripting attacks, ensuring the security of system data.

4.3 Experimental Results Analysis

System Functional Testing Results

As shown in Table 1, the results indicate that the English mobile learning system performs excellently across multiple core functionalities. All tests for course access functionality passed, demonstrating that the system is stable and reliable in providing learning content, allowing users to smoothly access and browse courses. The success rate for interactive test submission functionality is 98%, with one test failing, indicating compatibility or response issues under certain specific conditions or devices, necessitating further investigation and optimization. The success rate for user settings adjustment functionality is 99%, showing that the system effectively responds to user personalization settings, providing a good user experience. Overall, the system's total success rate reached 99%, indicating strong functional stability.

Table 1. Functionality Test Results

Functionality	Total Tests	Passed	Failed	Success Rate
Course Access	50	50	0	100%
Interactive Test Submission	50	49	1	98%
User Setting Adjustment	100	99	1	99%
Total	200	198	2	99%

Performance Testing Results

As indicated in Table 2, the average response time of the system is 3 seconds, with a maximum of no more than 5 seconds. This demonstrates that the system can respond to user requests within a reasonable time frame, ensuring a smooth user experience. The system's CPU usage reached 70%, and memory usage was at 65%, indicating that under high concurrent stress, the system's resource utilization is high but does not exceed the safety threshold, maintaining a certain margin and stability in processing. In terms of concurrent handling capability, the system did not experience service denial or system crashes under the stress test of 1,000 concurrent users, showing good concurrent processing ability. The error rate was controlled within 1%, suggesting that the system can stably handle requests in most cases. Overall, the system demonstrates good performance under high concurrency conditions.

Table 2. Performance Test Results

Metric	Result
Average Response Time	3 seconds (max 5 seconds)
System CPU Usage	70%
System Memory Usage	65%
Concurrency Capability	No service denial or crash under 1000 concurrent users
Error Rate	Less than 1%

Security Testing

As shown in Table 3, during the security testing of the English mobile learning system, OWASP ZAP was used for penetration testing, covering 200 test points. The test results showed that the system performed excellently in defending against SQL injection attacks, with no vulnerabilities found in 20 potential risk points. In terms of Cross-Site Scripting (XSS) attack protection, none of the 30 test points detected script execution, effectively preventing XSS attacks. For Cross-Site Request Forgery (CSRF) protection, all operations implemented CSRF token validation, with no vulnerabilities found. In password cracking protection tests, the system enforced complex password policies and restricted consecutive login attempts, significantly enhancing security. Session management tests indicated that the system effectively manages sessions and auto-logouts, with all transmitted data encrypted using SSL/TLS, and sensitive information like passwords being stored using hashing, enhancing data security. In the overall vulnerability detection, 2 medium-risk vulnerabilities were identified and fixed. These results demonstrate that the English mobile learning system possesses robust security capabilities across multiple dimensions, effectively protecting user data and system security. However, they also emphasize the importance of continuous optimization and monitoring of security to adapt to the ever-changing cyber security environment.

Table 3. Security Test Results

Metric	Total Checked	Failures/Vulnerabilities
SQL Injection Test Points	20	0
XSS Test Points	30	0
CSRF Test Points	All Forms	0
Password Policy Requirements	Enforced	0
Session Timeout	30 min	0
Vulnerabilities Detected	200	2

5 Conclusions

This study explored the design and implementation of an English mobile learning system based on the Weighted Naive Bayes model. The system's layered architectural design ensured high modularity and scalability, and the application of the Weighted Naive Bayes method optimized the personalized learning experience. Comprehensive evaluations were conducted through a series of functional, performance, and security tests. The test results showed that the system met the expected goals in core functionalities, demonstrating good performance and high stability. The English mobile learning system has shown its potential as an effective learning tool, capable of meeting the personalized English learning needs of users and providing a richer and more convenient learning environment for a wide range of learners.

References

1. Rie K, and Yo I, Structural Equation Modeling of Vocabulary Size and Depth Using Conventional and Bayesian Methods. [J].Frontiers in psychology., 2020,11618.
2. Wang Y, and Zeng R, Study of Learning System based on Naive Bayes Classification Model[J]. Journal of Convergence Information Technology., 2012,7(22):746-753.
3. Xiaojing H, Design and Application of English Assisted Learning System Based on Mobile Learning Platform [J]. Procedia Computer Science., 2023, 228 231-240.
4. Jinxiang X, Design of language assisted learning model and online learning system under the background of artificial intelligence [J]. Applied Mathematics and Nonlinear Sciences., 2022, 8 (2): 711-720.
5. Zhang J, and Hao G, The Design and Function of Online Learning System Based on Artificial Intelligence Technology [J]. Automation and Machine Learning., 2022, 3 (1):
6. Yeram K, Hyeongmook L, and Wonjoon K, Design of an IoT-Based Remote Learning System for Medical Skill Training in the Age of COVID-19: Focusing on CPR Skill Training [J]. Applied Sciences., 2022, 12 (17): 8840-8840.
7. Yanqiu X, An Adaptive Learning System for English Vocabulary Using Machine Learning [J]. Mobile Information Systems., 2022, 2022.
8. Bin L, Shaofen W, Design of the Online Learning System for College Physical Education Teaching Based on the BP Neural Network [J]. Wireless Communications and Mobile Computing., 2022, 2022.
9. YingHui D, Design of Online Learning System of College Sports Gymnastics Based on BP Neural Network [J]. Mobile Information Systems., 2022, 2022.
10. Laurent C, and JeanFrançois S, Discrete Event Modeling and Simulation for Reinforcement Learning System Design [J]. Information., 2022, 13 (3): 121-121.