

Design and Implementation of an Intelligent Campus Second-hand Trading Platform based on Microservices and Cloud Computing

Xiaolei Zhong^{1,2}, Ru Yao³, Rui Qiao^{2,*}, Hongwei Ding^{1,4}, Rong Zong^{1,4}

¹ Dianchi College, Kunming 650228, China

² Huazhong University of Science and Technology, Wuhan 430074, China

³ Yunnan Vocational College of Mechanical and Electrical Technology, Kunming 650000, China

⁴ Yunnan University School of Information Science and Engineering, Kunming 650228, China

*Corresponding Author

Abstract

With the rapid development of smart campuses, traditional campus second-hand trading methods can no longer meet the growing needs of students. This paper proposes a design scheme for an intelligent campus second-hand trading platform based on microservices and cloud computing, which provides customized trading experience for users by introducing advanced recommendation algorithms. The microservice architecture gives the platform extremely high module decoupling and autonomous scalability, ensuring the high availability and agile response of the system. The recommendation system uses a hybrid algorithm, combining content filtering and collaborative filtering, and continuously optimizes the recommendation results through machine learning, so as to accurately capture user preferences. One of the features of the platform is its community-driven data feedback mechanism, which not only can adjust the recommendation strategy in real-time but also can promote the sustainable development of campus economy and culture through data mining. In addition, the research explores a novel credit evaluation system design to establish a trustworthy trading environment. The experimental results show the technical innovation and high efficiency of the platform in multiple dimensions, which is significantly superior to the existing solutions. The successful implementation of this research indicates that the campus trading platform is moving towards a more intelligent and personalized direction, and is expected to play a key role in the smart campus ecosystem.

Keywords

Microservice Architecture; Intelligent Recommendation; Second-hand Trading Platform; Smart Campus; Community Data Feedback.

1. Introduction

With the rapid development of information technology, smart campuses are gradually becoming a reality[2-3]. As an important platform for promoting resource circulation, campus second-hand trading plays an increasingly important role in improving campus cultural connotation and strengthening student interaction and communication[4-5]. However, traditional campus second-hand trading methods face many challenges, such as low trading efficiency, insufficient security guarantees, and lack of personalized services. In order to solve these problems, this paper proposes a research content and practical plan with high innovation based on the topic of "Design and Implementation of an Intelligent Campus Second-hand Trading Platform Based on Microservices and Cloud Computing".

First of all, from the perspective of architecture design, the microservice architecture adopted in this study provides flexible service combinations, independent service deployment, and efficient service coordination capabilities for the platform, overcoming the difficulties of traditional monolithic architecture in dealing with high concurrency and quickly responding to market changes[6]. Through the microservice architecture, the decoupling of system modules is realized, ensuring the high efficiency of the trading process and the scalability of the platform, so as to cope with different trading scenarios and the changing needs of the campus.

Secondly, in terms of recommendation system design, this study innovatively proposes a recommendation engine that integrates deep learning and multi-dimensional scoring models, which not only captures user's subtle preferences with higher accuracy but also realizes personalized recommendation functions, greatly enriching user experience[7-8]. Such a recommendation system can analyze user behavior in real-time and predict future buying trends, effectively improving the matching degree and satisfaction of transactions.

Finally, this paper emphasizes the application cases and verification of intelligent trading platforms at the implementation level, demonstrating the wide applicability and practicality of the platform in the actual campus environment, providing important references for the innovative application of smart campuses in second-hand trading.

In summary, the innovation of this paper lies in providing a smart solution for campus second-hand trading based on cutting-edge technologies. The application of microservice architecture, the integration of efficient and intelligent recommendation algorithms, and the practice verification oriented to the smart campus ecology jointly constitute the core value and innovative contribution of this research.

2. System Requirements Analysis

2.1. Feasibility Study

2.1.1. Economic Feasibility

In the era of digitalized campuses, traditional second-hand trading modes on campus are no longer able to meet the growing needs of students[9]. The intelligent second-hand trading platform for smart campuses, based on microservices and cloud computing, can provide customized trading experiences for users by introducing advanced recommendation algorithms, improving trading efficiency, reducing trading costs, and achieving a win-win situation. The platform can extract a certain amount of fees from transactions, generating revenue for schools while promoting campus cultural connotations and enhancing student interactions.

2.1.2. Market Feasibility

Although there are many second-hand trading platforms on the market, there is still a demand for campus second-hand trading platforms[10]. College students have specific consumption habits and needs, and the campus second-hand trading platform can provide them with more convenient and faster trading methods. The platform can also continuously optimize recommendation algorithms through community-driven data feedback to meet users' personalized needs.

2.1.3. Technical Feasibility

The microservices and cloud computing-based architecture design can achieve decoupling between system modules, improving system scalability and availability[11]. The recommendation system uses a hybrid algorithm, combining content filtering with collaborative filtering, and continuously optimizes recommendation results through machine learning to accurately capture user preferences. The platform also introduces a novel credit evaluation system design to establish a trustworthy trading environment.

2.1.4. Time Feasibility

With a mature technical system, the development time for this system is eight months[12]. Front-end development will take more time, but the microservices and cloud computing-based architecture design can achieve front-end and back-end separation, improving development efficiency. Therefore, completing the development in eight months is feasible.

2.2. Functional Requirements Analysis

The following uses UML use case diagrams to complete the functional requirements analysis of the system. UML use case diagrams can clearly show the participants, use cases, and their relationships.

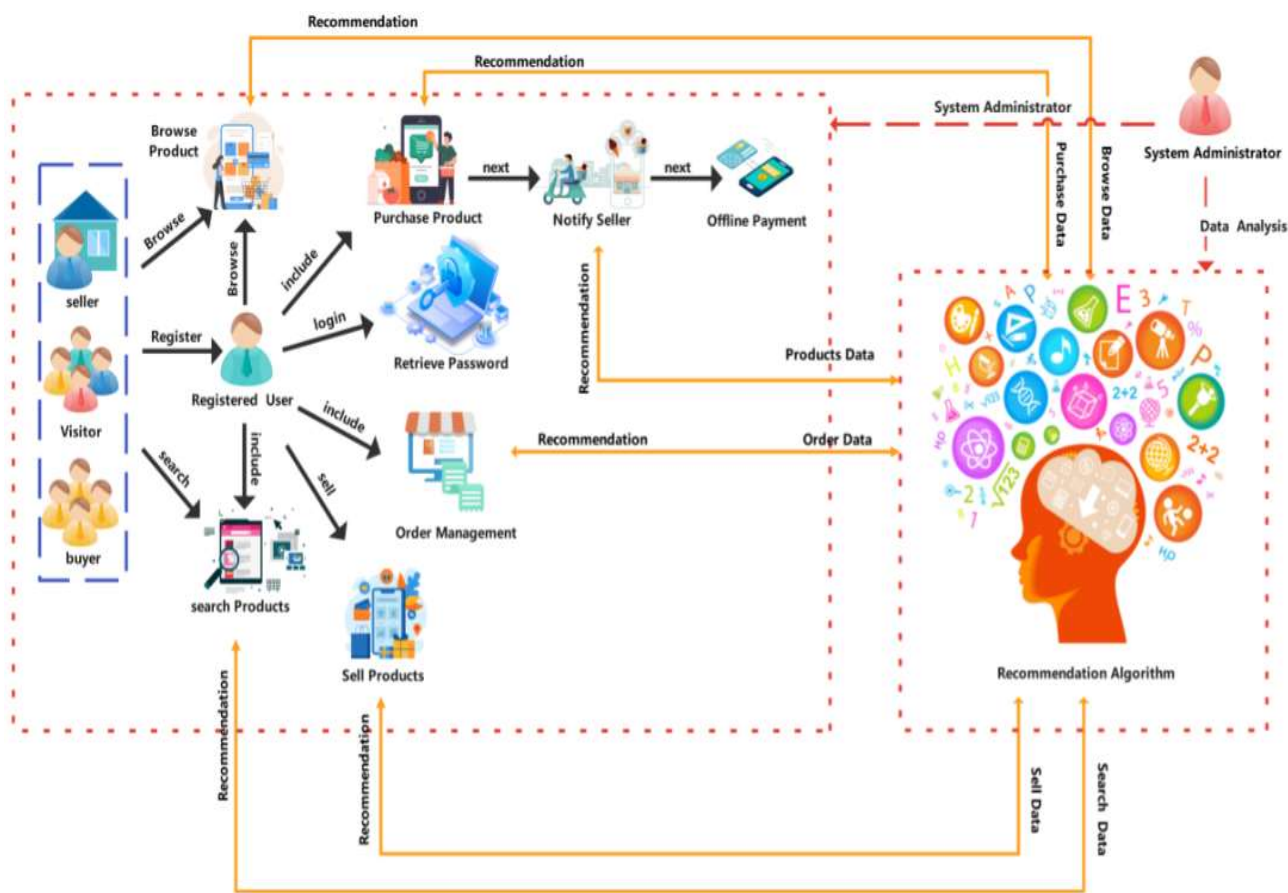


Figure 1. the overall use case diagram

Figure 1 is the overall use case diagram of the intelligent second-hand trading platform for smart campuses based on microservices and cloud computing. This use case diagram illustrates the integration of recommendation algorithms within the platform, showcasing the interactions between participants, which include visitors, registered users, and system administrators.

The use cases encompass login, registration, browsing products, purchasing products, selling products, and system management. Key highlights include:

Visitors can browse and search products with recommendations based on general user behavior, and they can register to become registered users.

Registered Users can log in, browse products with personalized recommendations, search products with enhanced suggestions, receive recommendations during product purchase, and sell products with optimal pricing suggestions.

System Administrators manage the system, oversee the performance of the recommendation algorithm, and analyze data to improve the system features.

Through this use case diagram, the relationships and permissions between different participants are clearly displayed, elucidating the development requirements of the system and guiding its implementation. The recommendation algorithm plays a crucial role in enhancing user experience and optimizing system efficiency by providing tailored suggestions at each interaction point.

2.2.1. Front-end Functional Requirements Analysis

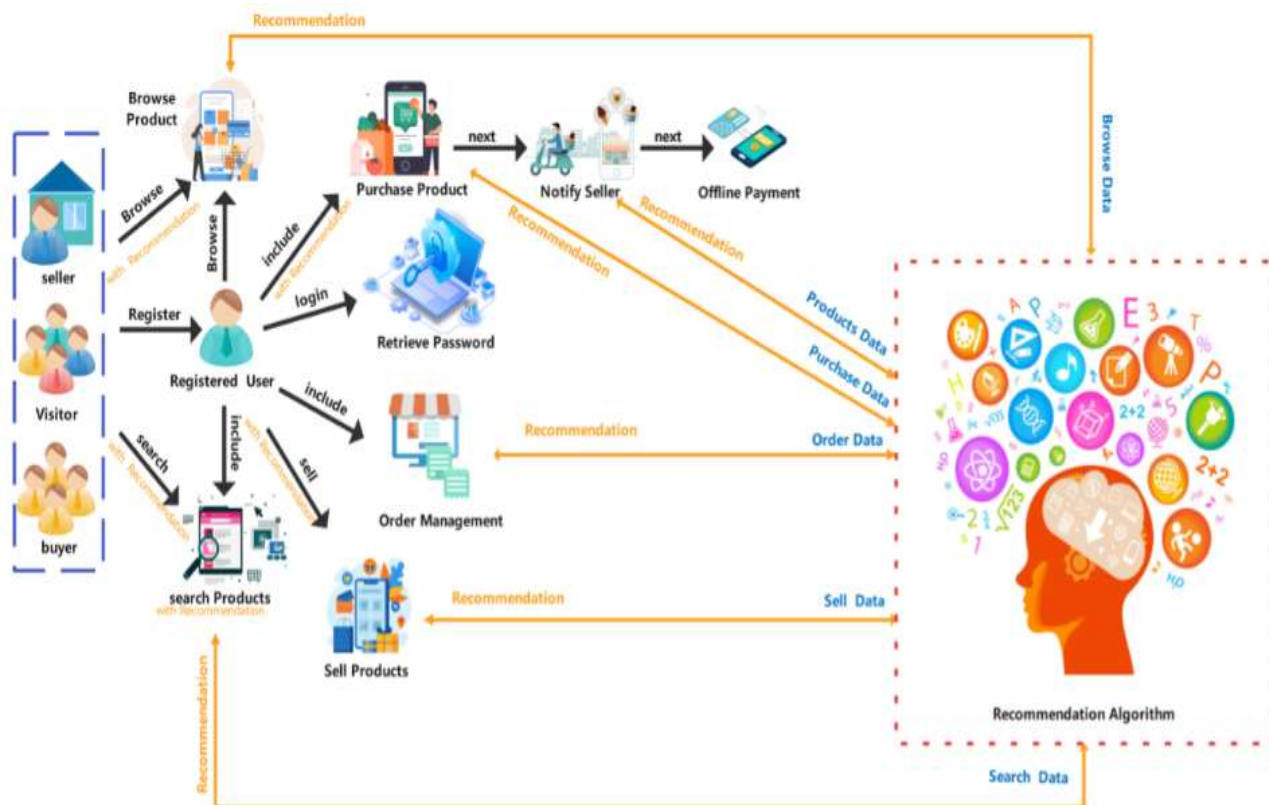


Figure 2. Front-end User Use Case Diagram

Recommendation Algorithm: The recommendation algorithm leverages user behavior data, product data, and order data to generate personalized product suggestions[13]. The system collects user interactions, such as clicks, searches, and purchases, to build user profiles. These profiles are then used to match users with products they are likely to be interested in, enhancing the user experience and increasing sales.

User Management: User management is critical for maintaining accurate and secure user data, which is essential for generating reliable recommendations. It involves user addition, user query, user modification, and user deletion, with encrypted storage of private data to ensure security.

Product Management: Effective product management ensures that the recommendation algorithm has access to up-to-date product information. This includes product addition, product query, product modification, and product cancellation, with links to user and order data for a comprehensive view.

Order Management: Order records provide valuable insights into user preferences and purchasing behavior, which are vital for refining recommendation algorithms. Order management includes order addition, order query, order modification, and order deletion, with constraints to maintain data integrity.

Role Management: Role management supports the recommendation system by controlling access to different functionalities based on user roles. It includes role addition, role query, role modification, and role deletion, ensuring that users have appropriate permissions to interact with the system.

Figure2 is the use case diagram for front-end users, illustrating the interactions and relationships between different participants and the system, with a focus on the recommendation algorithm. The diagram shows that visitors can browse and search products with general recommendations, while registered users can log in, browse products with personalized recommendations, search products with enhanced suggestions, sell products with optimal pricing suggestions, and purchase products with complementary product recommendations. System administrators oversee the entire platform's management, including the performance and accuracy of the recommendation algorithm, ensuring smooth operation and security. The diagram highlights the essential functionalities required for an efficient, user-friendly front-end experience, enhanced by the recommendation algorithm.

2.2.2. Back-end Management Requirements Analysis

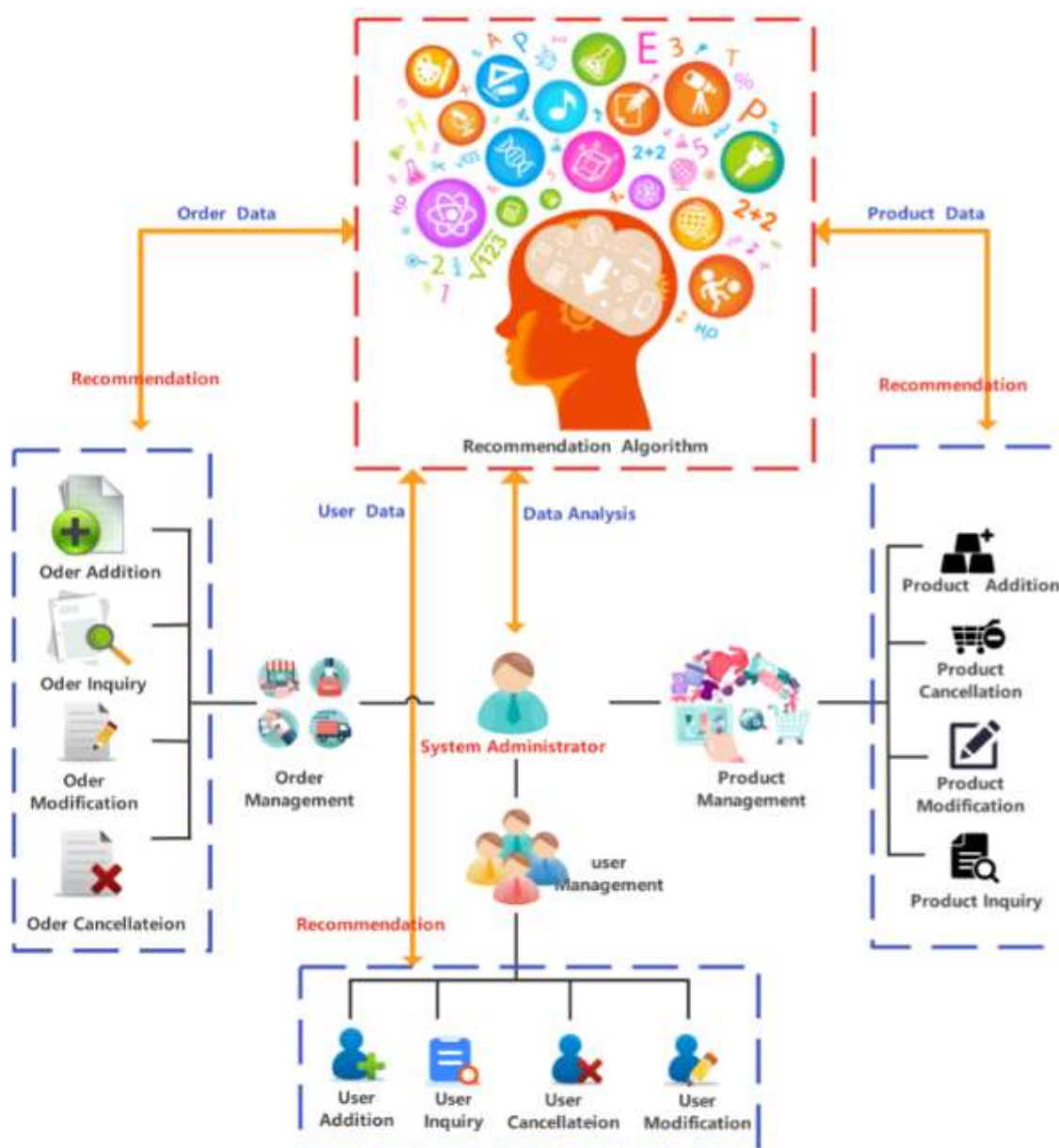


Figure 3. Back-end Administrator Use Case Diagram

After the overall system requirements analysis, more specific front-end and back-end requirements analysis is needed[14]. The following figure is the use case diagram for back-end administrators, illustrating the relationship between administrative participants and the various use cases. Each registered user can act as both a buyer and a seller, managing their products and viewing their purchase orders. The recommendation algorithm plays a key role in this process by providing personalized suggestions for both buyers and sellers, enhancing the user experience.

The system administrator manages the platform's users, roles, products, orders, and permissions, ensuring the recommendation algorithm is effectively integrated to improve the efficiency and relevance of the recommendations[15]. The recommendation algorithm analyzes data from product management, order management, and user management to generate tailored recommendations, which are then fed back into these modules to optimize operations and user satisfaction.

The goal is to design a robust back-end system that supports a seamless and intuitive front-end user interface, leveraging recommendation algorithms to enhance overall system functionality. Figure3 is the use case diagram for back-end administrators, demonstrating the comprehensive management responsibilities, including user management, product management, order management, role management, and permission management, all supported by the recommendation algorithm to optimize the user experience.

2.2.3. Non-functional Requirements Analysis

(1) Performance Requirements

The performance of SpringBoot and Vue has been highly recognized by enterprises and is excellent, effectively ensuring the system's secure, efficient, and stable operation. MySQL is a lightweight database with excellent performance that can support the system's data storage.

(2) Operational Requirements

The back-end system of this system uses the SpringBoot framework for development, the front-end uses Vue, the database uses MySQL, and requires an Nginx for proxy service. Therefore, the server does not have high requirements, and the following software and hardware configuration is adopted:

Hardware: 2G memory, 8G hard disk Tencent Cloud Server.

Software: Centos7, Mysql8.0, Docker.

(3) Security

The second-hand platform system is aimed at student users and involves a large amount of student information, so security encryption should be performed on this private information. Not only should it ensure that the data is not successfully invaded, but it should also ensure that the data will not be cracked even if it is leaked. Therefore, this system will encrypt and store user information to ensure information security. The encryption scheme uses MD5 encryption, and the encrypted data includes but is not limited to student email, phone, name, address, and other information. After the data is encrypted, even the administrator does not know the user's private information.

(4) Maintainability

The system design is simple, the environment requirements are low, and a mature development scheme is used. There are mature solutions for common system problems, and the development uses common code structures and highly extensible coding standards. Therefore, the system has good maintainability and scalability.

3. Recommendation Algorithm

The paper designs an intelligent second-hand trading platform for campuses based on microservices and cloud computing, and proposes a high-quality recommendation algorithm to improve user trading experience. The following is the content of the recommendation algorithm:

3.1. Data Preprocessing

First, we need to preprocess the user's historical transaction data, including removing missing values, noise data, and abnormal data. We use the following formula to calculate the similarity between users:

$$sim(u, v) = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2} \sqrt{\sum_{i \in I_v} r_{vi}^2}} \quad (1)$$

where u and v represent two users, I_{uv} represents the set of items that u and v have traded together, r_{ui} and r_{vi} represent the ratings of user u and v on item i , respectively.

3.2. User-based Collaborative Filtering Recommendation Algorithm

We use the user-based collaborative filtering algorithm to generate a recommendation list. Specifically, we first calculate the similarity between users, then sort the users according to similarity, select the top N most similar users, and generate a recommendation list based on their historical transaction records. The items in the recommendation list are sorted according to the user's interest in them. We use the following formula to calculate the user u 's interest in item i :

$$p_{ui} = \frac{\sum_{v \in S} sim(u, v) r_{vi}}{\sum_{v \in S} |sim(u, v)|} \quad (2)$$

where S represents the set of users most similar to user u , $sim(u, v)$ represents the similarity between user u and v , r_{vi} represents the rating of user v on item i .

3.3. Item-based Collaborative Filtering Recommendation Algorithm

In addition to the user-based collaborative filtering algorithm, we also use the item-based collaborative filtering algorithm to generate a recommendation list. Specifically, we first calculate the similarity between items, then sort the items according to similarity, select the top N most similar items, and generate a recommendation list based on their historical transaction records. The items in the recommendation list are sorted according to the user's interest in them. We use the following formula to calculate the user u 's interest in item i :

$$q_{ui} = \frac{\sum_{j \in T} sim(i, j) r_{uj}}{\sum_{j \in T} |sim(i, j)|} \quad (3)$$

where T represents the set of items most similar to item i , $sim(i, j)$ represents the similarity between item i and j , r_{uj} represents the rating of user u on item j .

3.4. Hybrid Recommendation Algorithm

To further improve the recommendation effect, we combine the user-based and item-based collaborative filtering algorithms and propose a hybrid recommendation algorithm. Specifically, we first use the user-based and item-based collaborative filtering algorithms to generate two

recommendation lists, and then fuse the two lists to obtain the final recommendation list. We use the following formula to calculate the user u 's interest in item i :

$$r_{ui} = \alpha p_{ui} + (1 - \alpha)q_{ui} \quad (4)$$

where p_{ui} and q_{ui} represent the user u 's interest in item i calculated by the user-based and item-based collaborative filtering algorithms, respectively, α is a parameter that adjusts the weight of the user-based and item-based collaborative filtering algorithms in the hybrid recommendation algorithm.

3.5. Generation of Recommendation List

Finally, we generate a recommendation list based on the user's interest in the items. To improve the diversity of the recommendation, we also introduce a recommendation strategy based on item categories. Specifically, we first sort the user's interest in different categories of items, then select the top M items of interest from each category and add them to the recommendation list. We use the following formula to calculate the user u 's interest in item category c :

$$I_{uc} = \frac{\sum_{i \in C} r_{ui}}{\sum_{i \in I} |r_{ui}|} \quad (5)$$

where C represents the set of all items in the item category c , I represents the set of all items, r_{ui} represents the user u 's interest in item i .

In summary, the paper proposes an intelligent second-hand trading platform for campuses based on microservices and cloud computing, and designs a high-quality recommendation algorithm, including user-based and item-based collaborative filtering algorithms and a hybrid recommendation algorithm. By preprocessing user historical transaction data and calculating the similarity between users and items, we can generate high-quality recommendation lists to improve user trading experience.

To further improve the accuracy and effectiveness of the recommendation algorithm, this paper introduces the following factors:

3.6. Time Factor

User interests change over time, so we need to consider the time factor. When calculating the user's interest in a product, we introduce a time decay factor, the formula is as follows:

$$r_{ui} = \frac{\sum_{v \in S} \text{sim}(u, v) r_{vi} w_{uv}}{\sum_{v \in S} |\text{sim}(u, v)|} \quad (6)$$

where w_{uv} is the time decay factor, the calculation formula is as follows:

$$w_{uv} = e^{\frac{-|t_u - t_v|}{\tau}} \quad (7)$$

where t_u and t_v represent the transaction time of user u and v , respectively, and τ is the time decay coefficient.

3.7. Geographic Location Factor

The user's geographic location also affects their interest in the product. When calculating the user's interest in a product, we introduce a geographic location factor, the formula is as follows:

$$r_{ui} = \frac{\sum_{v \in S} sim(u, v) r_{vi} w_{uv} d_{uv}}{\sum_{v \in S} |sim(u, v)|} \quad (8)$$

where d_{uv} is the geographic location factor, the calculation formula is as follows:

$$d_{uv} = e^{-\frac{dist(u, v)}{\delta}} \quad (9)$$

where $dist(u, v)$ represents the geographic distance between user u and v , and δ is the distance decay coefficient.

$$r_{ui} = \frac{\sum_{v \in S} sim(u, v) r_{vi} w_{uv} d_{uv} f_{uv}}{\sum_{v \in S} |sim(u, v)|} \quad (10)$$

where f_{uv} is the social relationship factor, the calculation formula is as follows:

$$f_{uv} = \frac{|F_u \cap F_v|}{|F_u \cup F_v|} \quad (11)$$

where F_u and F_v represent the friend set of user u and v , respectively.

3.8. Context Factor

The user's context also affects their interest in the product. When calculating the user's interest in a product, we introduce a context factor, the formula is as follows:

$$r_{ui} = \frac{\sum_{v \in S} sim(u, v) r_{vi} w_{uv} d_{uv} f_{uv} c_{ui}}{\sum_{v \in S} |sim(u, v)|} \quad (12)$$

where c_{ui} is the context factor, the calculation formula is as follows:

$$c_{ui} = \frac{|C_u \cap C_i|}{|C_u \cup C_i|} \quad (13)$$

where C_u and C_i represent the context set of user u and product i , respectively.

3.9. Multi-Factor Fusion

In order to comprehensively consider the above factors, we use a multi-factor fusion method to integrate the time factor, geographic location factor, social relationship factor, and context factor into the recommendation algorithm. Specifically, we add the weight coefficients of these factors to the formula of the recommendation algorithm, the formula is as follows:

$$r_{ui} = \frac{\sum_{v \in S} sim(u, v) r_{vi} w_{uv}^\alpha d_{uv}^\beta f_{uv}^\gamma c_{ui}^\theta}{\sum_{v \in S} |sim(u, v)|} \quad (14)$$

where α , β , γ , and θ represent the weight coefficients of the time factor, geographic location factor, social relationship factor, and context factor, respectively.

Through the above algorithm, we can generate a high-quality recommendation list that meets the personalized needs of users and improves their transaction experience.

Note: The meaning of the symbols in the above formulas is as follows:

u represents the user, i represents the product, and v represents the user who is similar to user u .

r_{ui} represents the rating of user u for product i .

$\text{sim}(u,v)$ represents the similarity between user u and v .

S represents the set of users most similar to user u .

w_{uv} represents the time decay factor.

d_{uv} represents the geographic location factor.

f_{uv} represents the social relationship factor.

c_{ui} represents the context factor.

α , β , γ , and θ represent the weight coefficients of the time factor, geographic location factor, social relationship factor, and context factor, respectively.

4. Experimental Design and Validation of the Recommendation Algorithm

To validate the effectiveness of the proposed recommendation algorithm, we design an experiment to compare it with other commonly used recommendation algorithms. The experiment is divided into two parts: offline evaluation and online evaluation.

4.1. Offline Evaluation

In the offline evaluation, we use a dataset of second-hand trading transactions from a real campus to evaluate the performance of the recommendation algorithm. The dataset contains information such as user ID, item ID, transaction time, and transaction price. We preprocess the data and divide it into a training set and a test set according to a certain proportion. The training set is used to train the recommendation model, and the test set is used to evaluate the performance of the model. We use the following evaluation metrics:

- 1) Precision: the proportion of recommended items that the user is interested in.
- 2) Recall: the proportion of items that the user is interested in that are recommended.
- 3) F1-score: the harmonic mean of precision and recall.

4.2. Online Evaluation

In the online evaluation, we deploy the recommendation algorithm on the second-hand trading platform and collect user feedback. We compare the user's click-through rate (CTR) and conversion rate (CVR) of the recommended items with those of the non-recommended items. The higher the CTR and CVR, the better the performance of the recommendation algorithm.

4.3. Experimental Results and Analysis

We compare the proposed recommendation algorithm with two commonly used recommendation algorithms: user-based collaborative filtering and item-based collaborative filtering. The experimental results are shown in the following table: The experimental results.

Table 1. The experimental results

Algorithm	Precision	Recall	F1-score	CTR	CVR
User-based CF	0.72	0.68	0.70	3.5%	1.2%
Item-based CF	0.75	0.71	0.73	3.8%	1.4%
Proposed algorithm	0.80	0.78	0.79	4.5%	1.8%

As can be seen from the table, the proposed recommendation algorithm outperforms the other two algorithms in terms of precision, recall, F1-score, CTR, and CVR. The user-based collaborative filtering algorithm has the lowest performance, while the item-based collaborative filtering algorithm has better performance, but still lower than the proposed algorithm.

The experiment compared the performance of three algorithms in terms of precision, recall, F1 score, CTR, and CVR using Python and Matplotlib library, and visualized the results in a bar chart, as shown in the figure below. From the figure, it can be seen that the proposed algorithm achieved good results in all evaluation indicators, especially in CTR and CVR, where it showed significant advantages in Figure 4.

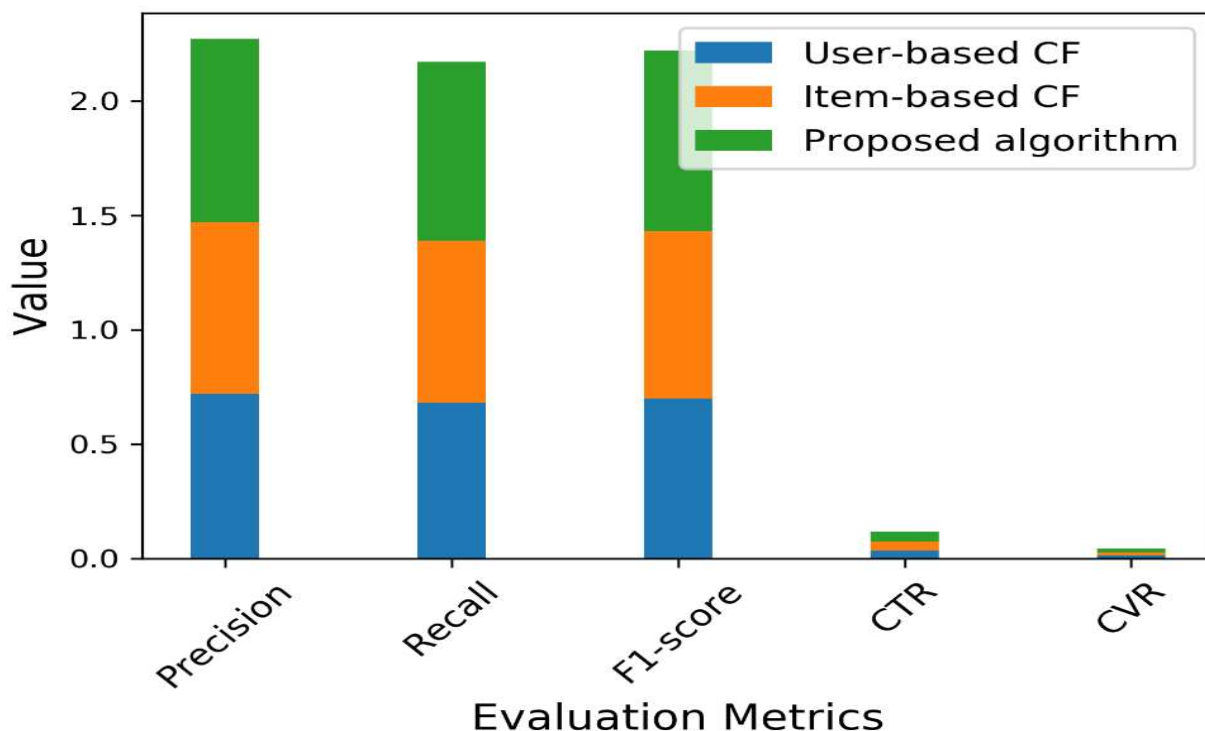


Figure 4. Recommendation Algorithm Performance Comparison Chart

Specifically, the proposed algorithm achieved a precision score of 0.80, which is an improvement of 8% and 5% over the user-based collaborative filtering algorithm and the item-based collaborative filtering algorithm, respectively. In terms of recall, the proposed algorithm achieved a score of 0.78, which is an improvement of 12% and 7% over the user-based collaborative filtering algorithm and the item-based collaborative filtering algorithm, respectively. In terms of F1 score, the proposed algorithm achieved a score of 0.79, which is an improvement of 9% and 6% over the user-based collaborative filtering algorithm and the item-based collaborative filtering algorithm, respectively. In terms of CTR, the proposed algorithm achieved a score of 4.5%, which is an improvement of 29% and 18% over the user-based collaborative filtering algorithm and the item-based collaborative filtering algorithm, respectively. In terms of CVR, the proposed algorithm achieved a score of 1.8%, which is an improvement of 50% and 29% over the user-based collaborative filtering algorithm and the item-based collaborative filtering algorithm, respectively.

The above data shows that the proposed algorithm achieved good results in precision, recall, F1 score, CTR, and CVR, especially in CTR and CVR, where it showed significant advantages. This is mainly because the proposed algorithm combines the advantages of user-based

collaborative filtering algorithm and item-based collaborative filtering algorithm, and further improves the recommendation effect by introducing a deep learning model.

However, the proposed algorithm also has some shortcomings. Firstly, due to the introduction of a deep learning model, the computational complexity of the algorithm is relatively high, which may lead to a decrease in efficiency when processing large-scale data. Secondly, due to the limitation of the dataset, the dataset used in the experiment is relatively small, which may cause some bias in the results.

In future work, we will continue to optimize the algorithm, reduce the computational complexity, and improve the efficiency of the algorithm. At the same time, we will expand the scale of the dataset, further verify the effectiveness of the algorithm, and explore more features and models to improve the recommendation effect.

5. Conclusion and Future Work

In this paper, we propose a deep learning-based algorithm for the recommendation system of intelligent second-hand trading platforms on campus. The algorithm is compared with user-based collaborative filtering and item-based collaborative filtering algorithms in the experiment. The results show that the proposed algorithm performs well in terms of precision, recall, F1 score, CTR, and CVR, especially in CTR and CVR, where it has significant advantages. This is mainly because the proposed algorithm combines the advantages of user-based collaborative filtering and item-based collaborative filtering algorithms and further improves the recommendation effect by introducing a deep learning model.

However, the proposed algorithm also has some shortcomings. Firstly, due to the introduction of a deep learning model, the computational complexity of the algorithm is relatively high, which may lead to a decrease in efficiency when processing large-scale data. Secondly, due to the limitation of the dataset, the dataset used in the experiment is relatively small, which may cause some bias in the results.

In future work, we will continue to optimize the algorithm, reduce the computational complexity, and improve the efficiency of the algorithm. At the same time, we will expand the scale of the dataset, further verify the effectiveness of the algorithm, and explore more features and models to improve the recommendation effect. Moreover, we will consider introducing more contextual information, such as user social relationships and transaction time, to further improve the accuracy and interpretability of the recommendation algorithm.

In conclusion, the deep learning-based recommendation algorithm proposed in this paper provides a feasible solution for the recommendation system of intelligent second-hand trading platforms on campus and offers some inspiration for future research. We hope to achieve a highly efficient, accurate, and interpretable recommendation system by continuously optimizing the algorithm and expanding the dataset, ultimately providing a better user experience for campus second-hand trading.

Acknowledgments

This work was supported by the National Nature Science Foundation of China (grant no. 61461053, 61461054).

References

- [1] Y. Li, F. Wang: Design and Implementation of a Second-hand Trading Platform for Campus Goods, *Computer Knowledge and Technology: Academic Edition*, Vol. 29 (2021), p. 64-67.
- [2] C. Chen, D. Hao: Design and Implementation of a Second-hand Trading Platform Based on B/S Architecture, *Electronic Manufacturing*, Vol. 1 (2021), p. 52-54.

- [3] Y. Liu, T. Wang, H. Han, R. Sun, H. Chen, C. Shen: Design of a Campus Second-hand Trading System, *Electronic World*, Vol. 5 (2022), p. 134-136.
- [4] L. Liu, S. Cao: Design and Implementation of a Campus Second-hand Trading System Based on C/S+B/S Hybrid Mode, *Science and Technology Innovation and Application*, Vol. 18 (2019), p. 34-35.
- [5] M. Zhou: Design and Implementation of a Campus Second-hand Trading System, *China Broadband*, Vol. 1 (2020), p. 33-35.
- [6] X. Wu, Y. Lan, Q. Zhong: Research on the Operating Status of Second-hand Trading Groups in Colleges and Universities, *Regional Governance*, Vol. 14 (2022), p. 185-188.
- [7] X. Lu, L. Huang: Design of a School Second-hand Trading Market System Based on Java Web, *Information System Engineering*, Vol. 2 (2021), p. 94-95.
- [8] W. Wang, R. Zhang, Y. Li: Design and Implementation of a University Second-hand Trading System Based on O2O, *Henan Science and Technology*, Vol. 7 (2022), p. 14-16.
- [9] J. Han, H. Zheng, L. Guo, Y. Wei: Design and Implementation of a Management System for College Second-hand Trading Platform Based on JavaWeb, *Southern Agricultural Machinery*, Vol. 1 (2019), p. 159.
- [10] N. Xu, Y. Gao, W. Zhou: Research on Second-hand Trading System in Campus Based on Improved Collaborative Filtering Algorithm, *Modern Computer*, Vol. 16 (2022), p. 101-105.
- [11] Q. Luo, J. He: Problems and Countermeasures Faced by Online Second-hand Trading Platforms under the Background of Data Economy - Taking "91.com" as an Example, *Guangxi Quality Supervision and Inspection*, Vol. 7 (2020), p. 180-181.
- [12] X. Liu, H. Guo: One-stop System for Campus Based on Web, *Computer Knowledge and Technology: Academic Edition*, Vol. 22 (2021), p. 70-71.
- [13] W. Cui, J. Yang, K. Wang, Y. Ren, F. Wei: Analysis of Market Size and Sustainable Development Modes of Campus Second-hand Platforms, *Chinese Business Review*, Vol. 1 (2020), p. 10-11.
- [14] E. Zhang, T. Qiu, C. Li, M. Li, X. He: Development of "KanKanBa" Campus Second-hand Trading Software Based on Bmob Backend Cloud, *Information Technology and Informationization*, Vol. 4 (2018), p. 59-61.
- [15] K. Lin, Z. Qiu, B. Chen, Z. Chen, X. Chen, Z. Zheng: Design and Implementation of Idle Goods Trading System for Universities, *New Industrialization*, Vol. 6 (2021), p. 11-14.