Computer Hardware Resources E-commerce Platform Based on Intelligent Recommendation and High Concurrency Processing - "Beiji Technology"

Youyi Luo, Ting Jiang*

[1,2] *School of Intelligent Science and Technology, Geely University of China, No.123, Section 2, Chengjian Avenue, East New District, Chengdu, Sichuan, China*

Article Information

Received: 21-11-2024
Revised: 28-11-2024
Published: 05-12-2024

Keywords
Intelligent recommendations; high concurrency processing; computer accessories; user experience optimization; recommender systems

****Correspondence Email:***
jiangting@guc.edu.cn

Abstract

With the rapid development of e-commerce and the diversification of the computer hardware market, consumers are facing more and more difficulties in choosing computer accessories, and traditional e-commerce platforms are difficult to meet the user's demand for accurate shopping and efficient recommendation. The purpose of this paper is to build an intelligent computer accessories shopping platform (Beiji Technology), through intelligent recommendation algorithms, to improve the user shopping experience. The platform adopts mainstream front-end technology, efficient and stable back-end architecture, supports high concurrency data processing, and at the same time realizes dynamic interaction, which enhances the responsiveness and smoothness of the user interface. Its intelligent recommendation system generates personalized accessory recommendations based on the user's historical behavior, helping users find the products they need faster. The realization of this platform not only solves the pain points of traditional e-commerce platforms in the purchase of computer accessories, but also promotes the development of the e-commerce industry in the direction of intelligence and precision, and provides valuable references for the future research and practice of intelligent e-commerce.

1. Introduction

E-commerce is becoming a key driver of transformation in the retail industry as digital transformation continues. According to research data from Statista (Statista,2023), the global e-commerce market size is expected to continue to grow in the coming years, with the market size in 2023 having exceeded USD 5 trillion. In this context, computer hardware, as a highly segmented market, poses higher shopping requirements for consumers due to its wide range of products and complex technical specifications (Lu et al. 2015:12). The computer hardware market involves a wide range of accessories such as processors, graphics cards, memory, etc., and the diversity of their brands, models, and specifications, as well as frequent technological updates, make consumers often face information overload during the shopping process. These complexities pose a great challenge to the product display and recommendation functions of traditional e-commerce platforms, and also create an urgent need for intelligent upgrading of the platforms.

In recent years, the application of big data and artificial intelligence technologies in e-commerce has gradually deepened, and intelligent recommender systems (Aggarwal 2016) have become an important tool for improving user experience and enhancing platform competitiveness. Intelligent recommender systems use user behavioral data, purchase preferences, and multi-dimensional product characterization to provide personalized product recommendations. This technology significantly reduces consumer shopping time and improves decision-making efficiency.

With the rapid development of e-commerce and computer technology, intelligent recommender systems and high concurrency processing techniques have been widely researched and applied worldwide (Davidson et al. 2010:293). For example, leading international platforms such as Netflix and Amazon enable accurate prediction of user needs and services through advanced recommendation algorithms (Smith et al. 2017:12).Lu et al. argue that recommender systems aim to provide users with personalized suggestions for online products or services in order to deal with the ever-increasing problem of online information overload and improve customer relationship management (Lu et al. 2015:12).Collaborative filtering technique proposed by Aggarwal et al. is considered to be the foundation of recommender systems (Aggarwal 2016). Collaborative filtering recommends relevant goods for users by analyzing their historical behaviors, and its core lies in constructing recommendation models using similarities among users (Sarwar et al. 2001:285). .He et al. (He et al. 2017:173) proposed a recommendation method based on neural collaborative filtering, which better captures the complex interactions between the user and the product through a deep learning model. Koren et al. (Koren et al. 2009:30) developed a matrix decomposition technique which is a classic example of collaborative filtering approach and provides theoretical support for this study.

However, the current research and practice of intelligent recommendation system for computer hardware market is still insufficient, especially in the lack of innovative exploration to meet the user's personalized needs and the integrated application of high concurrency processing capacity. To address this problem, this paper builds an intelligent computer hardware shopping platform -- "Beiji Technology". The core objective of the platform is to solve the main pain points of users in computer hardware shopping, including information overload, decision-making difficulties and shopping inefficiency. To achieve this goal, the platform introduces intelligent recommendation algorithms and high concurrency processing technology (Song et al.2023:19159), combined with mainstream back-end architectures such as Spring Boot, MySQL, and Tomcat, as well as front-end technologies such as dynamic page loading and real-time data updating, with the aim of providing users with an efficient and convenient shopping experience.

To address issues such as high database load and low efficiency, Shi et al. proposed a distributed caching system that leverages distributed caching to mitigate performance bottlenecks while resolving challenges related to disk I/O caused by large-scale and high-concurrency database access.(Shi et al. 2024:1) The design of this paper introduces Redis as a caching mechanism and combines the thread pool management function of Tomcat to ensure the stability and responsiveness of the platform in high traffic situations.

This paper aims to enhance the consumer shopping experience through intelligent recommendation and high concurrency technology, and also provides a new perspective for the digital development of the e-commerce industry by optimizing the performance and stability of the platform. With the increasing demand of users for real-time responsiveness and personalized services, the research results of this paper are of great value in promoting the intelligent transformation of e-commerce platforms, while providing a feasible path for the refined management and sales of the computer hardware market.

The research content and innovations of this paper include: (1) designing and implementing an intelligent recommendation system for computer hardware shopping, helping users to quickly find suitable accessories through accurate recommendation algorithms; (2) ensuring the stability and responsiveness of the platform under massive access requests through high concurrency processing technology; (3) optimizing the operation smoothness and experience effect of the user interface by combining with the dynamic interaction function.

1.1 Literature Review

The platform architecture is designed with user experience optimization and high concurrency processing as the core objectives, combining Spring Boot, MySQL, Tomcat and other back-end technologies, as well as JavaScript and other front-end technologies, to achieve an efficient platform for intelligent recommendation and dynamic interaction. The system adopts a layered architecture design, including front-end display layer, business logic layer and data layer, to ensure the scalability, stability and easy maintenance of the system.

1.2 Overall structure

The overall architecture of the system is based on MVC (Model-View-Controller) design pattern, which divides the system into model layer, view layer and controller layer, and its structure is shown in Fig. 1. The model layer is responsible for data processing and business logic implementation, the view layer is responsible for the front-end interface display, and the controller layer is responsible for receiving and processing user requests. The design of MVC architecture improves the clarity, scalability and maintainability of the code, which contributes to the rapid iteration of the system functional modules.

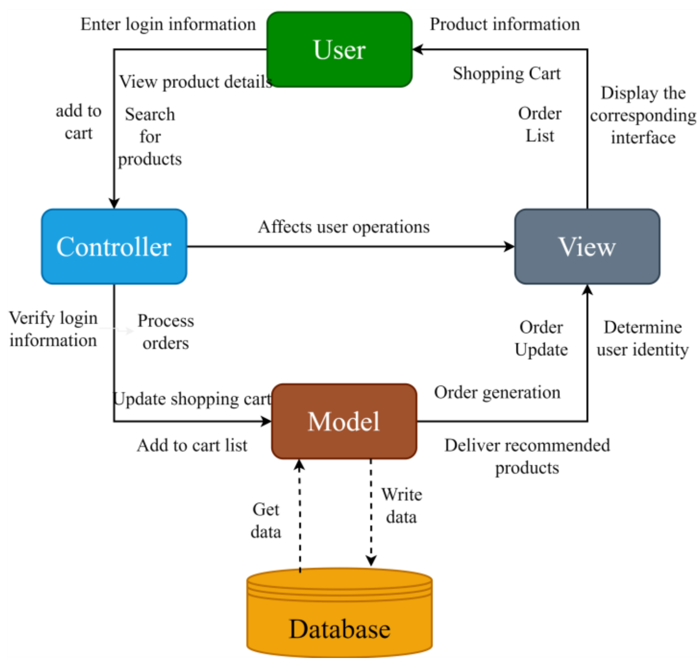

Fig. 1 System Architecture Diagram

1.3 Front-end design

The front-end design adopts mainstream technologies such as HTML, CSS and JavaScript, and its website layout is shown in Figure 2. At the same time, the combination of AJAX realizes asynchronous communication between the front and back ends to ensure that the page content is updated in real time without refreshing, so as to enhance the user experience. Firstly, the combination of JavaScript and AJAX is used to realize dynamic loading of the page and real-time data updating, providing users with instant search and recommendation functions to ensure the smoothness of the interaction; secondly, through the keyword search function, users can enter keywords to quickly find products, and the front-end sends the keywords entered by the user to the back-end to match the query and return the results in real time, effectively improving the search efficiency; in addition, the system uses dynamic particles to achieve asynchronous communication. In addition, the system adopts dynamic particle background effect to enhance the visual experience and scientific and technological atmosphere, the design creates a sense of modernity for the user at the same time, to help alleviate the visual fatigue brought about by prolonged browsing.

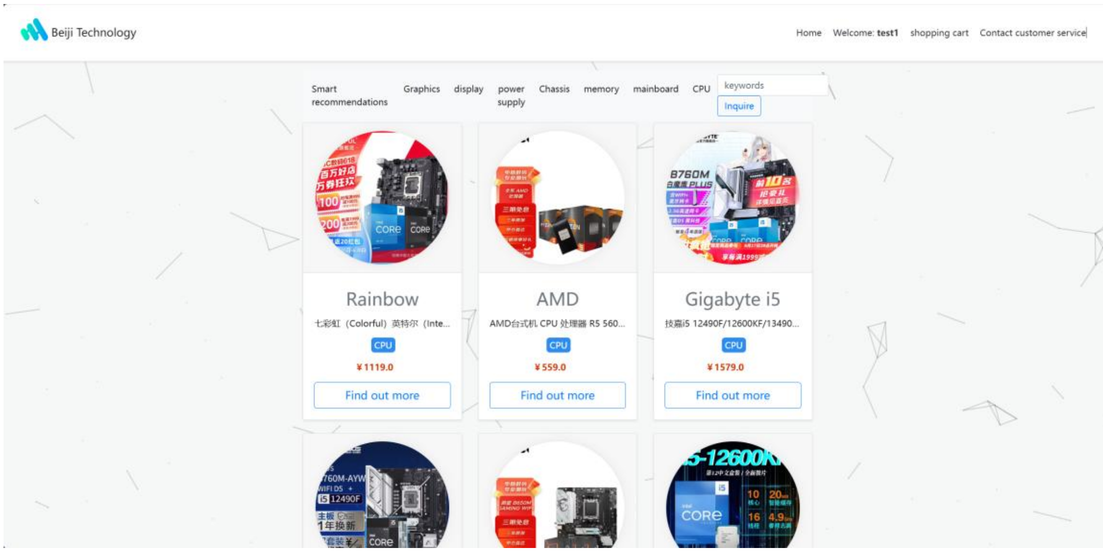

Fig. 2 Front-end style

1.4 Back-end design

The back-end system uses Spring Boot framework, combined with Spring MVC, Spring Data JPA and Spring Security to realize the core business logic, data processing and security management functions. Spring Boot accelerates the development process by simplifying the configuration and automation of the deployment and embedding the Tomcat server, which is suitable for highly concurrent environment; Spring MVC is responsible for processing user requests, invoking business logic and returning view pages; Spring Data JPA combined with MyBatis simplifies the development process of the data access layer. environment; Spring MVC is responsible for handling user requests, invoking business logic and returning view pages to achieve high efficiency of front-end and back-end data interaction; Spring Data JPA combined with MyBatis simplifies the development process of the data access layer, supports the database operation of addition, deletion, modification and checking, and ensures persistent management of data; Spring Security is used for authentication and privilege control to ensure system data security. Spring Security is used for authentication and access control to ensure system data security.

2. Research Methods

2.1 Dynamic particle algorithm

The algorithm generates a set of particles by randomly generating a set of particles, and each particle moves dynamically according to a preset motion trajectory. The algorithm calculates the distance between particles in real time during operation, and when the distance between two particles is less than a set threshold, it will dynamically generate a connecting line to connect the two; when the distance between particles exceeds the threshold, the connecting line will automatically disappear. At the same time, when the user moves the mouse, the mouse will produce a gravitational effect on the surrounding particles, causing the particles to gather towards the mouse position, forming a dynamic interaction effect. This design makes the visual expression of the background richer and increases the interaction between the user and the interface. The algorithm can enhance the dynamic expressiveness of the page and the user's interactive experience. On the one hand, by utilizing the random movement of particles and dynamically changing interactive effects, it breaks the monotony of the traditional static background and effectively relieves the visual fatigue that may occur when users browse the static page for a long period of time; on the other hand, it injects a sense of science and technology and dynamic aesthetics into the interface by combining the interactive effects between the mouse and the particles. This innovative design not only makes the system reflect a distinctive sense of modernity in addition to functionality, but also enhances the attractiveness and interest of the page, thus better retaining users and improving their satisfaction.

2.2 Hybrid Collaborative Filtering Algorithm

The algorithm combines user behavior analysis and cosine similarity computation to model user preferences through weighted behavioral data for more accurate personalized recommendation (Xie et al. 2021). First, the algorithm collects multiple behavioral data of users, including product browsing, adding shopping cart, purchasing, etc., and assigns weights to each behavior. These weights reflect the different degree of influence of the behaviors on the user's interest, e.g., the purchase behavior often represents the user's real demand more than the browsing behavior. Through weighting, the user's behavioral data is vectorized to ensure the scientific nature and differentiation of the data in subsequent calculations, and the process is shown in Figure 3.

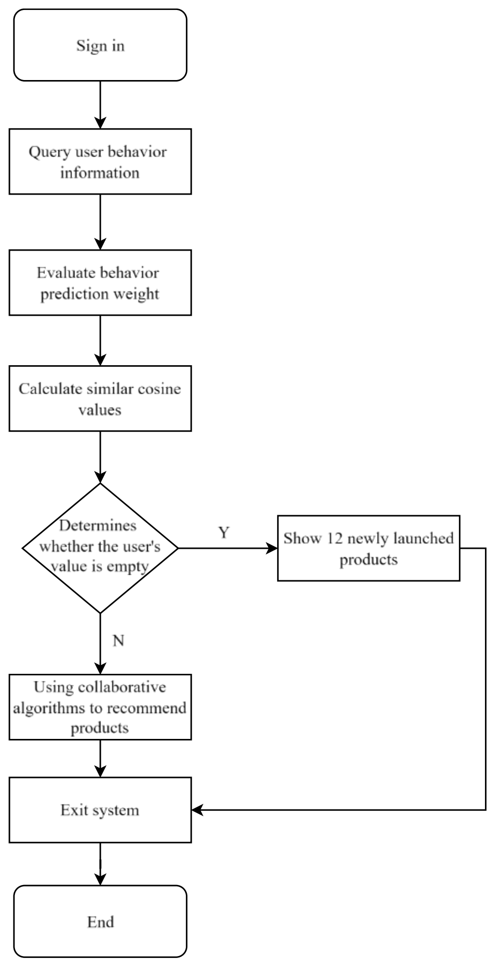

Fig. 3 Graphical representation of the hybrid collaborative filtering algorithm

After vectorizing the user behavior data, the algorithm uses cosine similarity as a measure to calculate the similarity between users. Specifically, cosine similarity yields a similarity score ranging between [-1, 1] by calculating the cosine of the angle between two vectors as shown in equation (1). The closer the value is to 1, the more similar the users' behaviors are; the closer the value is to -1, the behaviors show opposite trends. Based on the similarity score, the system finds a set of high similarity users for the current user, and extracts the products that these users have purchased or browsed, and generates a list of products according to the recommendation priority.

$$Similarity(A,B) = \frac{\sum(A_i \times B_i)}{\sqrt{\sum A_i^2} \times \sqrt{\sum B_i^2}} \tag{1}$$

In the formula, **i** represents the index of an item, while A and B are the behavior vectors of two users, with each dimension corresponding to a specific item. The variable **i** indicates the *i*-th item, reflecting the behavioral data of both users for that item. Specifically, A_i :represents user A's behavioral data (e.g., ratings, clicks, or purchases) on the *i*-th item, and $B_i$represents user B's behavioral data on the same item.

2.3 Intranet Penetration

Intranet Penetration is a technology that maps intranet resources to the public network and establishes a communication channel between the intranet and extranet through a third-party transit server or service to break through the restrictions of NAT (Network Address Translation) and firewalls (Kumar at al 2019). The core principle lies in the fact that the intranet service takes the initiative to establish a stable connection with the relay server, and the external access is forwarded to the intranet service through the relay server to realize resource sharing. And Tunnel technology (Tunnel) is the key to realize the intranet penetration, through the encrypted bi-directional communication channel to ensure the security and reliability of data transmission, its process and design is shown in Figure 4.

Cloudflare Tunnel in Figure 4 is an intranet penetration scheme based on reverse tunneling, which enables public access to intranet services without exposing the public IP of the server through Cloudflare's globally distributed network. The intranet server actively connects to the Cloudflare server by running the Cloudflare tool, forming an encrypted tunnel. Cloudflare assigns a unique public domain name, and the access request initiated by the user through the domain name is forwarded to the intranet service by the Cloudflare network, realizing a seamless connection. This process not only circumvents the complexity of traditional NAT configuration, but also improves the security and performance of data access.

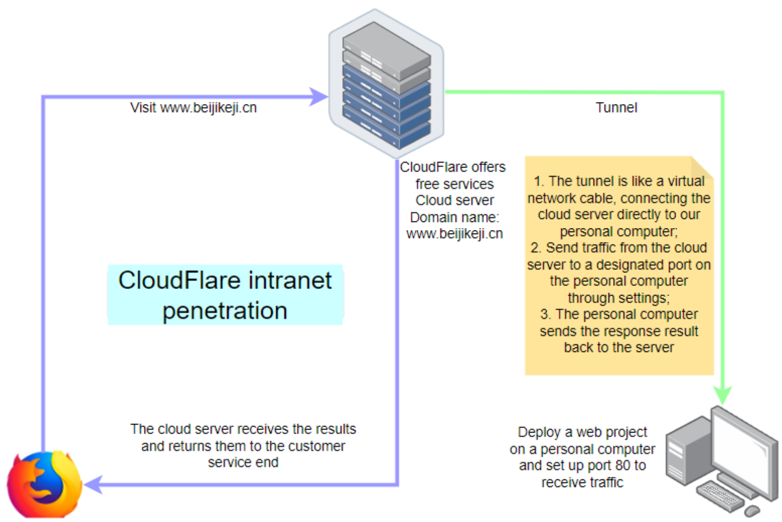

Fig 4 Intranet penetration diagram

In this paper, we publish the intranet service to the public network (with the domain name www.beijikeji.cn) through Cloudflare Tunnel, which not only avoids the risk of public IP exposure, but also enhances the access performance by using the network optimization function of Cloudflare. Specifically, through Cloudflare's secure network encrypted transmission, combined with DDoS protection and caching mechanisms, potential attack risks and latency issues can be effectively reduced. Fine-grained control of access rights is further realized through the combination of CLI tools and Cloudflare Access.

2.4 High Concurrency Handling and Performance Optimization

First, Redis caching technology is used to cache high-frequency accessed data such as popular products, user login status and frequently used query results into memory, which can reduce the direct access to the database, significantly improve the response speed of data queries and reduce the database load. At the same time, a reasonable cache expiration time is set to ensure the timeliness of data.

Second, Nginx is used for load balancing to distribute user requests to multiple back-end servers for processing. By configuring Nginx's load balancing policies (e.g., polling, weighted polling, etc.), the system is able to balance the workload of each server under high concurrency, ensuring uniformity of request distribution and stability of processing, and avoiding single-point overload.

Finally, resources are monitored and dynamically scaled. Use monitoring tools (e.g. Prometheus and Grafana) to monitor system resource usage (CPU, memory, disk, network, etc.) in real time to detect bottlenecks and anomalies in a timely manner. Combined with containerization technologies (e.g., Docker and Kubernetes), dynamic scaling of the system is achieved, automatically increasing or decreasing service instances to cope with traffic fluctuations, ensuring that the system is always in the best performance state.

3. Result and Discussion

3.1 Experimentation and Testing

In order to verify the accuracy of the intelligent recommendation algorithm of this platform and the performance of the system under high concurrency conditions, this paper conducts comprehensive tests in terms of algorithmic effect, system response time, resource consumption and so on. Its test results are shown in Table 1.

Table 1. Results of testing the inaccessible types

Test Category	descriptive	Indicators/Results
Algorithmic accuracy	Evaluate the accuracy, reach, and diversity of recommendation algorithms by analyzing user browsing history and purchasing patterns.	Accuracy: 82%; Coverage: 91%; Diversity: High
System response time	Monitor system CPU, memory, and network usage under sustained high loads to ensure stability and identify bottlenecks.	Average response time: 1000 users - 0.85 seconds, 5000 users - 1.3 seconds, 10000 users - 2.5 seconds; Success rate: 98%
Depletion of resources	Monitor system CPU, memory, and network usage under sustained high loads to ensure stability and identify bottlenecks.	CPU utilization <80%. memory utilization <75%. Stable network bandwidth

3.2 Algorithmic accuracy

The accuracy, coverage and diversity of the recommendation algorithms were tested by analyzing the browsing history and purchasing patterns of users. In that test, the platform's recommendation algorithm demonstrated high accuracy and broad coverage, as shown in the following: recommendation accuracy of 82%, coverage of 91%, and high diversity score. This shows that the algorithm is able to effectively meet the personalized needs of users, while recommending a rich variety of goods, avoiding the repeated recommendation of a single type of goods, and improving the diversity of the user experience.

3.3 System response time

The average response time and success rate of the platform under different loads are tested by simulating highly concurrent access scenarios with 1000, 5000 and 10000 users. The results show that in the case of 1000 concurrent users, the average response time is 0.85 seconds; in the case of 5000 concurrent users, the average response time is 1.3 seconds; and in the case of 10000 concurrent users, the response time is 2.5 seconds. In addition, the system achieved a request success rate of 98%. This result shows that the platform has good response speed and stability in the case of high concurrency, and can support large-scale user access requirements.

3.4 Resource consumption

The stability and resource management efficiency of the platform is assessed by monitoring the platform's CPU, memory and network bandwidth usage under sustained high load. Its results show that the CPU utilization rate stays below 80%, the memory utilization rate below 75%, and the network bandwidth is stable under high load environment. This shows that the platform is able to effectively control resource consumption and ensure stable operation under high load scenarios, providing reliable support for the platform's scalability and high concurrent processing capability.

3.5 Results and analysis

Through the above tests on the accuracy of the recommendation algorithm, system response time and resource consumption of the platform, it can be seen that the platform excels in the accuracy and diversity of the recommendation algorithm, the response time of the system under the condition of high concurrency and resource management. The recommendation algorithm can accurately identify users' needs and provide diversified product choices, which enhances users' willingness to buy and the platform's user stickiness; the system responds quickly and has a high success rate under high concurrency access, providing users with a smooth access experience; the optimization of resource management enables the system to maintain its stability and efficiency under high load and ensures the platform's sustainable development and scalability. In conclusion, the technical architecture and optimization strategy of this platform have reached the expected goal, and it has the ability to support high-traffic e-commerce applications. In the future, further optimization of algorithms and resource management strategies will help the platform continue to maintain high-quality performance in the face of higher user demands.

4. Conclusions

Based on the e-commerce platform "Beiji Technology", this paper designs and builds an intelligent computer accessory shopping platform, focusing on optimizing the two key modules of intelligent recommendation and high concurrency processing. Through a series of experimental tests, the performance of the platform's recommendation algorithm and the system's processing capacity are verified, indicating its feasibility and effectiveness in practical applications. The system adopts a hybrid collaborative filtering recommendation algorithm, which performs better in terms of recommendation accuracy, and at the same time better meets the user's personalized needs in terms of the diversity and coverage of recommended content. Especially when dealing with new users or new products, the recommendation algorithm is still able to provide more accurate recommendation results, which significantly improves the user's shopping experience and the practicality of the system. In addition, performance optimization is carried out for the high concurrency demand of the platform. By introducing caching technology, load balancing and asynchronous processing strategies, the system can still maintain a short response time and a high success rate under a high load environment, and shows good scalability and resource management capabilities. These technical optimizations not only effectively improve data processing efficiency and reduce system resource consumption, but also ensure the stability of the platform in high traffic access scenarios. With the rapid development of artificial intelligence and large-scale pre-training models, future research will further explore deep learning algorithms or other intelligent technologies to further improve the accuracy and reasonableness of recommendations.

5. References

Aggarwal, C. (2016). Recommender systems: the textbook. Springer Publishing Company, Incorporated.

Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., ... & Sampath, D. (2010, September). The YouTube video recommendation system. In Proceedings of the fourth ACM conference on Recommender systems (pp. 293–296).

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017, April). Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web (pp. 173–182).

Kumar, R., & Tlhagadikgora, K. (2019). Internal network penetration testing using free/open source tools: network and system administration approach.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30–37.

Li, S. S., & Karahanna, E. (2015). Online recommendation systems in a B2C E-commerce context: a review and future directions. Journal of the Association for Information Systems, 16(2), 2.

Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: a survey. Decision Support Systems, 74(June), 12–32.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285–295).

Shi, L., Qiao, H., & Chen, Y. C. (2024). Research and application of distributed cache based on redis. Journal of Software, 19(1), 1–8.

Smith, B., & Linden, G. (2017). Two decades of recommender systems at Amazon.com. IEEE Internet Computing, 21(3), 12–18.

Song, Y., & He, Y. (2023). Toward an intelligent tourism recommendation system based on artificial intelligence and IoT using Apriori algorithm. Soft Computing, 27(24), 19159–19177.

Statista Research Department. (2023). "Global eCommerce Market Size from 2014 to 2023." Statista.

Xie, W., & Sun, X. (2021). A hybrid recommendation algorithm based on latent factor model and collaborative filtering. Springer, Singapore.