

Analysis of Application Strategies of Artificial Intelligence in Computer Application Software Development

Chengzhi Guo

Qinghai Minzu University, Xining 810007, Qinghai, China

Copyright: © 2025 Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), permitting distribution and reproduction in any medium, provided the original work is cited.

Abstract: The rapid development of artificial intelligence technology has provided brand-new ideas and methods for computer application software development, becoming an important driving force to promote the intelligence and efficiency of software development. Starting from the overall concepts of artificial intelligence technology and computer application software development, this paper systematically analyzes the application of artificial intelligence in computer software development, and elaborates on the functions of artificial immune technology, neural network technology, Agent technology, and expert systems in software development. It not only provides theoretical support for the application of artificial intelligence technology in software development but also offers operable strategic suggestions for practice, which is of great guiding significance for improving software development efficiency and optimizing software quality.

Keywords: Artificial intelligence; Computer; Application software development

Online publication: June 26, 2025

1. Overview of artificial intelligence technology and computer application software development

1.1. Concept of artificial intelligence technology

Artificial intelligence technology is a kind of technology that simulates, extends, and expands human intelligence. Its core goal is to enable machines to think, reason, learn, and make decisions like humans. The core of artificial intelligence technology lies in the simulation and realization of human intelligence, and its essence is to model and calculate human thinking processes through mathematical models and algorithms. The research field of artificial intelligence technology is very extensive, covering multiple levels from basic theories to practical applications. In terms of basic theories, researchers focus on how to construct more intelligent algorithm models and how to improve the generalization ability and adaptability of algorithms. At the application level, artificial intelligence technology has penetrated into all links of computer application software development, including requirement analysis, design optimization, test verification, etc. By combining artificial intelligence technology with the software development process, the efficiency and quality of software development can be significantly improved, and the development cost can be reduced.

Although artificial intelligence technology has shown great potential in computer application software development, its development also faces many challenges. For example, issues such as data privacy and security,

algorithm interpretability, and consumption of computing resources need to be gradually addressed in the process of technological development. In the future, with the continuous advancement of technology and the continuous expansion of application scenarios, artificial intelligence technology will play a more important role in computer application software development, promoting software development towards intelligence, efficiency, and personalization.

1.2. Principles for computer application software development

1.2.1. User-oriented principle

The user-oriented principle emphasizes focusing on user needs, ensuring that the software's functional design, interface layout, and operational processes can meet users' actual usage scenarios. During the software development process, the development team needs to gain an in-depth understanding of users' usage habits, functional requirements, and pain points through methods such as requirements analysis, user research, and prototype design. For example, when designing an online shopping platform, it is necessary to consider users' browsing habits, the convenience of the payment process, and the response speed of after-sales services. Adhering to the user-oriented principle can improve the software's user experience and satisfaction, thereby enhancing its market competitiveness.

1.2.2. Modular principle

The modular principle is a commonly used design method in software development, aiming to decompose a complex software system into several independent functional modules, each responsible for implementing specific functions. Through modular design, the complexity of the system can be reduced, and the maintainability and reusability of the code can be improved. When developing an enterprise management software, the system can be divided into modules such as employee management, financial management, and sales management. Each module runs independently and realizes data exchange and sharing through interfaces. Modular design not only facilitates the division of labor and collaboration within the development team but also reduces the impact on the overall system during subsequent maintenance and upgrades, thereby improving development efficiency and system stability.

1.2.3. Scalability principle

The scalability principle means that the software system, in its design and implementation, can adapt to future changes in requirements and functional expansions. With the continuous advancement of technology and the diversification of user needs, software systems need to have good scalability to support the addition of new functions, the integration of new modules, and the optimization and expansion of existing functions. By following the scalability principle, the software's lifecycle can be extended, and the development costs caused by requirements changes can be reduced.

1.2.4. Security principle

The security principle aims to protect the software system and its data from malicious attacks, virus intrusions, or other security threats. During the software development process, it is necessary to consider security requirements in all aspects from architectural design, coding implementation to testing and deployment, so as to ensure the stable operation of the software system and the security of user data, and enhance users' trust in the software^[1].

2. Research on the application of artificial intelligence technology in computer application software development

2.1. Artificial immune technology

Artificial immune technology is an important technology in the field of artificial intelligence, inspired by the principles of the biological immune system^[2]. In computer application software development, by simulating the behavior of the biological immune system, artificial immune technology can effectively improve the security, stability, and adaptability

of software systems. First, negative selection is a core mechanism in artificial immune systems, whose main function is to identify and eliminate “non-self” elements. In software development, the negative selection mechanism can be used to identify abnormal behaviors or potential threats in the system. For example, during the software testing phase, it can analyze the system’s operating status, identify behavioral patterns that do not meet expectations, and mark them as anomalies or threats. Second, clonal selection is another important mechanism in artificial immune systems. Its core idea is to enhance the system’s immune capacity by selecting and amplifying antibodies with effective characteristics. In software development, the clonal selection mechanism can be used to optimize the functional modules of software. For instance, during the requirement analysis phase, developers can use this mechanism to select the optimal solution from multiple candidates and extend it to the entire system, thereby improving the system’s adaptability and flexibility. Third, gene library is a mechanism in artificial immune systems for storing and managing the diversity of antibodies. In software development, the gene library can be used to manage functional modules or code snippets of the software system. Through this mechanism, developers can store code snippets of different functional modules in a unified library, and combine and optimize them according to needs, thus enhancing the maintainability and scalability of the system.

2.2. Neural network technology

Neural network technology is a computational model based on the principles of biological neural networks. It realizes the learning and reasoning of complex problems by simulating the connections and information transmission between neurons in the human brain. In defect prediction, neural network technology can identify key factors that may lead to software defects by learning historical data in the software development process^[3]. For example, by analyzing factors such as code quality, test coverage, and developers’ experience, neural networks can build a defect prediction model to predict potential defects in the software development process, improving the reliability and stability of the software. In performance optimization, neural network technology can be used to analyze performance bottlenecks in software operation and put forward optimization suggestions. For example, by collecting performance data during software operation—such as CPU utilization, memory consumption, and response time—neural networks can model and analyze these data to identify key factors affecting software performance. Based on these analysis results, the development team can take targeted optimization measures, such as optimizing algorithms, adjusting code structures, or improving resource management strategies, to enhance the operating efficiency and performance level of the software. In addition, neural network technology can also be used to improve user experience. For example, in software interface design, function layout, and user behavior analysis, neural networks can learn from user behavior data to identify user needs and preferences, thereby optimizing the software’s interactive design. By analyzing data such as user click behavior and operation paths, neural networks can generate personalized interface layouts or function recommendations, improving user operation efficiency and satisfaction^[4].

2.3. Agent technology

In the development of computer application software, Agent technology, as an important artificial intelligence technology, provides a new approach to the design and implementation of software systems by simulating intelligent entities capable of autonomous decision-making and behavior. Firstly, Agent technology enables modular design of complex systems through task decomposition and multi-Agent collaboration. For example, in large-scale software systems, multiple Agents can collaborate to perform complex functions such as data processing, task scheduling, and resource management, thereby improving the overall efficiency of the system. Secondly, Agent technology exhibits a high degree of adaptability, allowing dynamic adjustment of behavior in response to changes in the external environment. This feature enhances the robustness of software systems when facing uncertainties. For instance, in intelligent recommendation systems, Agents can dynamically adjust recommendation strategies based on users’ usage habits and real-time behaviors, providing more personalized services. Furthermore, Agent technology realizes intelligent management of software systems through autonomous decision-making and behavior planning. In automated testing

tools, for example, Agents can automatically design test cases, execute test tasks, and dynamically adjust testing strategies based on results. This intelligent management not only improves testing efficiency but also reduces the cost of manual intervention. In terms of security, Agent technology can identify potential threats in real-time through autonomous monitoring and anomaly detection, and take corresponding protective measures. In network security, for example, Agents can learn normal patterns of network traffic, identify abnormal traffic, and issue timely alerts, thereby enhancing system security. In practical applications, Agent technology also enhances the flexibility and practicality of software systems. In intelligent office software, for instance, Agents can interact with users through natural language processing, understand user intentions, and provide corresponding services. This intelligent human-computer interaction improves software usability and meets users' demands for personalized services.

2.4. Expert systems

The application of expert systems in computer application software development is primarily reflected in their ability to analyze and make decisions on complex issues.

In real-time monitoring of the operational status of computer application software, expert systems can collect and analyze various parameters during software operation through pre-set knowledge bases and reasoning mechanisms. For example, they can monitor software performance indicators, resource usage, and the stability of the operating environment, providing timely feedback to developers. In intrusion detection and judgment, expert systems can integrate multiple data sources, such as network traffic, user behavior logs, and system logs, for comprehensive analysis. Using pre-defined intrusion detection rules and machine learning models, they can identify potential intrusion behaviors, such as abnormal login attempts, unauthorized access, or malicious code injection. Compared with traditional intrusion detection systems, expert systems have the advantage of making judgments based on contextual information, thereby reducing false positive rates. When detecting a login attempt from an unknown IP address, for example, an expert system will further analyze the IP address's historical behavior, login time, and frequency to determine if it is a genuine intrusion. In response to intrusion behaviors, expert systems can take corresponding defensive measures according to pre-set strategies. For instance, upon detecting a potential intrusion, they can trigger firewall rules to block access from suspicious IP addresses; or when malicious code is detected, automatically isolate the infected module and generate detailed logs for subsequent analysis. Additionally, expert systems can interact with developers to provide specific response suggestions, such as vulnerability patching, security policy updates, or enhanced access control.

Through real-time monitoring, intelligent analysis, and automated responses, expert systems provide reliable decision support in complex and dynamic software development environments, safeguarding the stable operation and security of computer application software.

3. Conclusion

In summary, the application of artificial intelligence technology in computer application software development presents both opportunities and challenges. In the future, comprehensive layouts in technology, applications, and talent cultivation will be required to fully leverage the potential of artificial intelligence and drive the intelligent development of the software development field.

Disclosure statement

The author declares no conflict of interest.

References

- [1] Liang Y, 2024, A Brief Analysis of the Application of Artificial Intelligence Technology in Computer Application Software Development. *Information and Computers (Theoretical Edition)*, 2024(8): 59-61.
- [2] Li D, 2023, Analysis of Computer Application Software Development Technology Based on Artificial Intelligence. *Software*, 2023(9): 124-126.
- [3] Duan T, Liu H, 2023, Research on Computer Application Software Development Technology of Artificial Intelligence. *Electronic Components and Information Technology*, 2023(1): 102-105.
- [4] Yu K, 2022, Research and Analysis of Computer Game Software Development Technology Based on Artificial Intelligence. *Software*, 2022(10): 39-41, 45.

Publisher's note

Whioce Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.