Computer Software and Media Applications (2023) Volume 6 Issue 1
doi:10.24294/csma.v6i1.3906

ORIGINAL RESEARCH ARTICLE

Advancements in software engineering using AI

Hazem W. Marar

Computer/Electrical Engineering Department, Princess Sumaya University for Technology, P.O. Box 925819, Amman, Jordan; h.marar@psut.edu.jo

ABSTRACT

The integration of Artificial Intelligence (AI) into the space of software engineering marks a transformative period that reshapes traditional development processes and propels the industry into a new era of innovation. This exploration delves into the multifaceted impact of AI, from its roots in early symbolic AI to the contemporary dominance of machine learning and deep learning. AI's applications span various domains, but its significance in software engineering lies in its ability to enhance efficiency, improve software quality, and introduce novel approaches to problem-solving. From automating routine tasks to streamlining complex development workflows, AI acts as a virtual collaborator, allowing human developers to focus on higher-order thinking and creativity. This study introduces the application of AI in software engineering. reverse-engineering, and development environments. Moreover, ethical considerations, challenges, and future trends, including explainable AI, reinforcement learning, and human-AI collaboration, are presented.

Keywords: Artificial Intelligence; AI, software engineering; neural networks

ARTICLE INFO

Received: 29 December 2023
Accepted: 24 January 2024
Available online: 1 February 2024

1. Introduction

Artificial Intelligence includes a wide range of technologies and approaches aimed at creating various machines that can perform tasks requiring human intelligence[1]. From its early days, when AI was focused on rule-based systems and symbolic reasoning, to the current popularity and dominance of machine learning and deep learning, the field has witnessed tremendous growth. AI's applications span various sectors, from autonomous vehicles and healthcare to finance and entertainment[1,2].The contemporary field of AI is dominated by deep learning, a subset of machine learning that uses neural networks inspired by the human brain's structure[2]. Deep learning has pushed AI into new levels of capability, enabling it to excel in tasks such as image recognition, natural language processing, and complex decision-making. Technologies like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have become the foundation of the AI toolkit, enabling breakthroughs in various domains.

The influence of AI extends beyond the field of software engineering, into diverse sectors and revolutionizing conventional practices. In healthcare, AI aids in diagnostic processes, drug discovery, and personalized treatment plans[3]. Financial institutions leverage AI for fraud detection, risk assessment, and algorithmic trading[4]. Autonomous vehicles navigate complex environments using AI[5], and virtual assistants employ natural language processing to enhance human-computer interactions[6]. However, it is in the dynamic field of software engineering that AI has a vital transformative role. The combination of AI and software engineering has given rise to a collaborative relationship where intelligent systems augment and accelerate traditional development processes, pushing the industry into an era of unprecedented innovation. Early implementations focused on basic code completion suggestions[1,2], but as AI algorithms advanced, their ability to understand context, learn coding patterns, and even generate complex sections of code became increasingly sophisticated. This evolution has reshaped how developers approach their work, transforming AI from a tool that aids in specific tasks to an integral component of the developer's toolkit. This paper is organized as follows: Section 2 presents several trends and innovations that are shaping the future of AI in software engineering. Section 3 illustrates the challenges and ethical considerations of incorporating AI is the field of software engineering. The conclusion is presented in section 4.

2. Latest trends in the field of AI-driven software engineering

Since the significance of AI in software engineering lies in its ability to enhance efficiency, improve software quality, and introduce novel approaches to problem-solving, AI affects every stage of the software development flow starting from the conceptualization of projects ending by the actual deployment of software. AI-driven tools automate repetitive and time-consuming tasks, freeing up human developers to focus on higher-order thinking, creativity, and innovation. This shift in the division of labor allows for a more effective utilization of human cognitive abilities, creating an environment where developers can tackle complex challenges and push the boundaries of software innovation. The following real-world case studies illustrates how AI is making a difference in software engineering.

2.1. Google's data center efficiency optimization

Google, a pioneer in AI research and applications, has used AI to improve the efficiency of its data centers. Such capabilities require advanced resource management solutions. DeepMind, an AI research lab acquired by the Google, created an AI system for controlling and optimizing data center cooling systems. The AI system utilizes reinforcement learning to adapt and optimize cooling mechanisms based on feedback from real-time data and environment conditions. This results in a significant reduction in energy consumption, making Google's data centers more efficient and environmentally friendly. This demonstrates how AI can be used to save energy and increase sustainability[7].

2.2. Microsoft's visual studio code completion

Microsoft has integrated artificial intelligence (AI) into its popular integrated development environment (IDE), Visual Studio, to improve code completion. The IntelliCode feature in Visual Studio employs machine learning models to recognize coding patterns and predict the next lines of code that developers are likely to write. This helps developers write code more efficiently by eliminating the need to manually enter repetitive or boilerplate code. Microsoft has improved developer productivity by leveraging AI to understand context and suggest relevant code snippets, while also demonstrating the potential for AI to augment the creative aspects of software development. This demonstrates how AI can be a valuable ally in developers' daily tasks, making their work more efficient and beneficial[8].

2.3. GitHub's CodeQL for security analysis

GitHub, the world's leading platform for code management and collaboration, has integrated AI into its security analysis processes. CodeQL, a semantic code analysis engine acquired by GitHub, uses advanced static analysis to detect security flaws in code. It goes beyond traditional static analysis tools by understanding the code's semantics, allowing it to detect complex vulnerabilities that other methods may miss. This demonstrates how artificial intelligence (AI), especially in the field of security, can significantly improve software quality. By automating the detection of security flaws and potential exploits, GitHub's CodeQL helps to build more robust and secure software applications. This illustrates how important AI can be in ensuring the integrity and safety of software systems[9].

2.4. OpenAI’s Codex for code generation

OpenAI’s Codex is an advanced language model that is continuously trained on a wide range of programming languages and code repositories. It demonstrates remarkable capabilities in generating human-like code snippets based on natural language descriptions. This highlights the potential of AI in code generation, simplifying the development process for programmers. Codex can be integrated into IDEs and code editors, providing developers with instant code suggestions and completions. This not only accelerates coding tasks but also serves as a valuable learning tool for developers, helping them explore and understand different programming paradigms and practices. The case of Codex illustrates the power of AI in assisting developers with code creation and problem-solving[10].

2.5. IBM’s Watson for software engineering

IBM’s Watson, known for its effectiveness in natural language processing and cognitive computing, has been applied to several software engineering challenges. Watson for Software Engineering utilizes AI to analyze vast amounts of unstructured data, including documentation, forums, and code repositories. It assists developers in understanding and navigating complex codebases, making it easier to find relevant information and solutions. This showcases how AI can be a knowledge counterpart for developers, aiding in the exploration of codebases and providing insights that contribute to better decision-making. By leveraging AI’s ability to process and understand natural language, Watson for Software Engineering enhances the efficiency of developers, particularly in scenarios involving large code repositories[11].

These cases collectively illustrate the diverse applications of AI in software engineering. From optimizing data center efficiency to improving code completion, enhancing security analysis, enabling code generation, and serving as a knowledge companion, AI is proving to be a transformative force in the industry.

3. Challenges, ethical considerations, and future trends

The integration of AI in software engineering is filled with challenges. One significant concern is bias. If the training data used to teach AI models contains biases, the resulting algorithms may perpetuate or even amplify those biases. This raises ethical considerations regarding fairness in software development. Transparency is another challenge. As AI systems become more sophisticated, they often operate as a black-box model, making it challenging to understand the decision-making processes. This lack of transparency raises questions about accountability and the potential social impact of AI-driven software. Addressing these challenges requires collective efforts from the industry to develop ethical frameworks, guidelines, and tools that promote transparency, and accountability in AI-driven software engineering.

Meanwhile, looking ahead, several trends and innovations are shaping the future of AI in software engineering. Explainable AI (XAI)[12] is gaining prominence as a response to the transparency challenge. XAI aims to make AI systems more interpretable, providing insights into how decisions are made. This not only addresses ethical concerns, but also builds trust among developers and end-users. In addition, reinforcement learning and unsupervised learning are areas of active research, promising more autonomous and adaptive software development processes[13]. These approaches allow systems to learn from their environments and experiences, leading to intelligent decision-making and problem-solving. Furthermore, the integration of AI with DevOps practices is another trend[14], creating a more efficient software development lifecycle. AI-driven analytics and monitoring tools enhance the continuous integration and continuous delivery processes, ensuring faster and more reliable software releases.

3.1. Explainable AI (XAI)

Explainable AI (XAI) is a famous field that addresses the challenge of understanding and interpreting the decisions made by AI models. As AI systems become more complex, there is a growing need for transparency in their decision-making processes. XAI aims to provide insights into how AI algorithms reach specific conclusions, making them more interpretable for developers, stakeholders, and end-users. In the context of software engineering, XAI is crucial for building trust in AI-driven systems. Developers need to comprehend why an AI model made a particular suggestion or decision, especially in critical areas such as code generation, bug detection, and security analysis. As XAI evolves, it is likely to become an integral part of AI development workflows, enforcing accountability and transparency[12].

3.2. Reinforcement learning and unsupervised learning

Reinforcement learning and unsupervised learning are gaining popularity as AI paradigms that can revolutionize the software development processes. Reinforcement learning, where agents learn by interacting with an environment and receiving feedback, has the potential to create more autonomous and adaptive systems. In software engineering, this could show how AI-driven tools that learn from the development process, adapt to changing requirements, and optimize workflows. Unsupervised learning, which involves training models on unlabeled data without explicit guidance, opens new possibilities for understanding complex relationships within codebases. This can lead to more complex code analysis, improved bug detection, and a deeper understanding of software architectures. Both reinforcement learning and unsupervised learning are anticipated to play a significant role in shaping the next generation of intelligent software development tools[13].

3.3. Integration of AI with DevOps practices

The integration of AI with DevOps practices is a trend that aligns with the growing demand on continuous integration and continuous delivery (CI/CD) in software development. AI-powered analytics and monitoring tools enhance the CI/CD pipeline by providing real-time insights into code quality, performance, and potential issues. AI-driven DevOps tools can automatically detect patterns in the development process, predict potential bottlenecks, and optimize resource allocation. This integration not only accelerates the release cycle but also ensures the delivery of more reliable and high-quality software. As organizations increasingly adopt DevOps principles, the coordination between AI and DevOps is expected to become a driving force behind efficient and automated software delivery[14].

3.4. AI for code generation and AutoML

The future holds exciting possibilities for AI in code generation, enabling developers to leverage advanced language models to automate portions of the coding process. OpenAI's Codex, for instance, has demonstrated the potential of AI in generating human-like code based on natural language descriptions. As these models continue to evolve, they are expected to become more proficient in understanding complex requirements and producing high-quality code. AutoML (Automated Machine Learning) is another trend that showcases the automation of the machine learning pipeline. AI-driven tools can automate tasks such as feature engineering, model selection, and hyper-parameter tuning, democratizing machine learning and making it more accessible to developers with varying levels of expertise. This trend aligns with the present movement towards making AI more user-friendly and integrated into everyday software development workflows[15].

In summary, the future trends of AI in software engineering is expected to make the field become more transparent, autonomous, collaborative, and user-friendly. These trends signify a shift towards a future where AI is an integral and seamless part of the software development lifecycle, contributing to increased efficiency, innovation, and the democratization of advanced technologies despite several challenges and ethical concerns.

4. Conclusion

In conclusion, the integration of AI into software engineering is a transformative journey that brings both opportunities and challenges. The impact on development processes, the enhancement of software quality, and the exploration of ethical considerations collectively shape the trajectory of this evolving relationship. As technology advances, it is crucial to hold a balance between innovation and ethical responsibility. The future promises continued evolution, with emerging trends and innovations intended to redefine the use of AI in software engineering. By ensuring a collaborative and responsible approach, the industry can harness the full potential of AI while addressing challenges and ensuring a positive impact on the field of software development.

Conflict of interest

The author declares no conflict of interest.

References

1. Makridakis S. The forthcoming Artificial Intelligence (AI) revolution: Its impact on society and firms. *Futures* 2017; 90: 46–60. doi: 10.1016/j.futures.2017.03.006
2. Smith RG, Eckroth J. Robert S. Building AI applications: Yesterday, today, and tomorrow. *AI Magazine* 2017; 38(1): 6–22. doi: 10.1609/aimag.v38i1.2709
3. Jiang F, Jiang Y, Zhi H, et al. Artificial intelligence in healthcare: Past, present and future. *Stroke and Vascular Neurology* 2017; 2(4): 230–243. doi: 10.1136/svn-2017-000101
4. Cao L. AI in finance: Challenges, techniques, and opportunities. *ACM Computing Surveys* 2022; 55(3): 1–38. doi: 10.1145/3502289
5. Ma Y, Wang Z, Yang H, et al. Artificial intelligence applications in the development of autonomous vehicles: A survey. *IEEE/CAA Journal of AutomaticaSinica*2020; 7(2): 315–329. doi: 10.1109/jas.2020.1003021
6. Mattas PS. ChatGPT: A study of AI language processing and its implications. *International Journal of Research Publication and Reviews* 2023; 4(2): 435–440.doi: 10.55248/gengpi.2023.4218
7. Isaev EA, Kornilov VV, Grigoriev AA. Data center efficiency model: A new approach and the role of Artificial Intelligence. *Mathematical Biology and Bioinformatics* 2023; 18(1): 215–227. doi: 10.17537/2023.18.215
8. Salvaris M, Dean D, Tok WH. Microsoft AI platform. In: *Deep Learning with Azure*. Apress; 2018. pp. 79–98. doi: 10.1007/978-1-4842-3679-6_4
9. Yetistiren B, Ozsoy I, Tuzun E. Assessing the quality of GitHub copilot's code generation. In: Proceedings of the 18th International Conference on Predictive Models and Data Analytics in Software Engineering; 17 November 2022; Singapore, Singapore. doi: 10.1145/3558489.3559072
10. Finnie-Ansley J, Denny P, Becker BA, et al. The robots are coming: Exploring the implications of OpenAI Codex on introductory programming. In: Proceedings of the 24th Australasian Computing Education Conference; 14–18 February 2022; Online conference. doi: 10.1145/3511861.3511863
11. Magistretti S, Dell'Era C, Messeni Petruzzelli A. How intelligent is Watson? Enabling digital transformation through artificial intelligence. *Business Horizons* 2019; 62(6): 819–829. doi: 10.1016/j.bushor.2019.08.004
12. Barredo Arrieta A, Díaz-Rodríguez N, Del Ser J, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 2020; 58: 82–115. doi: 10.1016/j.inffus.2019.12.012
13. Morales EF, Escalante HJ. A brief introduction to supervised, unsupervised, and reinforcement learning. In: *Biosignal Processing and Classification Using Computational Learning and Intelligence*. Academic Press; 2022. pp. 111–129. doi: 10.1016/b978-0-12-820125-1.00017-8
14. Lwakatare LE, Crnkovic I, Bosch J. DevOps for AI—Challenges in development of AI-enabled applications. In: Proceedings of 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM); 17–19 September 2020; Online conference. doi: 10.23919/softcom50211.2020.9238323
15. Sabharwal N, Agrawal A. *Up and Running Google AutoML and AI Platform: Building Machine Learning and NLP Models Using AutoML and AI Platform for Production Environment*. BPB Publications; 2021.