

впровадження штучного інтелекту та технологій Edge AI в рамках концепції Industry 4.0. Його використання створює передумови для подальшого розвитку інтелектуальних систем контролю якості у харчовій промисловості.

Список використаних джерел

1. Howard A. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications / A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam // arXiv preprint. – 2017. - P. 4510-4520.
2. Bergmann P. Uninformed Students: Student-Teacher Anomaly Detection with Discriminative Latent Embeddings / P. Bergmann, M. Fauser, D. Sattlegger, C. Steger // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2020. - P. 4183–4192.

DOI 10.70286/EOSS-27.04.2026.005.131-133

HOW ARTIFICIAL INTELLECT IS CHANGING SOFTWARE DEVELOPMENT

Tsapenko L.

Candidate of Philological Sciences, PhD,
Associate Professor at Foreign Languages Department

Pylypenko D.

1st year student

Gr. AD-251

National University "Odesa Polytechnic"

Odesa, Ukraine

Artificial Intelligence (AI) is rapidly transforming the field of software development, introducing new tools, workflows, and challenges. The pace of technological change has accelerated significantly, as AI systems are now capable of generating code, detecting bugs, and offering solutions within seconds. This raises an important question for the industry: will developers remain essential, or will their roles evolve into something fundamentally different?

Understanding Artificial Intelligence in Development

Artificial Intelligence refers to the ability of machines to perform tasks that typically require human intelligence, such as learning, reasoning, and decision-making. In the context of software development, AI is primarily used as a support tool rather than a replacement for human expertise. Most modern AI systems are narrow and specialized, meaning they are designed to perform specific tasks effectively, such as code completion or bug detection.

Machine Learning and Deep Learning are key components of AI. Machine Learning allows systems to improve over time by analyzing patterns in data, while

Deep Learning relies on neural networks to process more complex information. Together, these technologies enable AI tools to assist developers in increasingly sophisticated ways.

Practical Applications of AI in Software Development

One of the most significant applications of AI is code generation. Tools such as ChatGPT and DeepSeek allow developers to describe functionality in natural language and receive working code almost instantly. This capability significantly reduces the time spent on routine programming tasks and accelerates development cycles.

AI is also transforming debugging and testing processes. Instead of manually reviewing large codebases, developers can rely on AI to scan code, identify anomalies, and suggest fixes. These tools can highlight potential issues early in the development process, reducing the cost and effort required to fix them later.

Additionally, AI contributes to performance optimization. It can recommend more efficient algorithms, detect unnecessary complexity, and help improve system architecture. As a result, applications become faster, more reliable, and easier to maintain.

Impact on Developer Productivity

AI significantly enhances developer productivity by automating repetitive tasks such as writing boilerplate code, refactoring, and scaffolding. This allows developers to focus on more complex and creative aspects of software engineering, including system design, problem-solving, and feature development.

Rather than replacing developers, AI shifts their role toward higher-level thinking and decision-making. Developers become more focused on architecture, quality assurance, and understanding business requirements, while AI handles routine operations.

Challenges and Risks

Despite its advantages, AI introduces several challenges. One major concern is the potential decline in critical thinking. If developers rely too heavily on AI-generated suggestions, they may lose the habit of analyzing problems deeply or questioning solutions. Over time, this can lead to weaker understanding and lower code quality.

Security risks are another significant issue. AI-generated code may include vulnerabilities, insecure dependencies, or poor validation practices. Without careful review, such issues can lead to serious problems in production systems, especially in sensitive areas like authentication or data handling.

Ethical concerns also play an important role. Questions about job displacement, data privacy, and responsible AI usage are becoming increasingly relevant. Organizations must carefully consider how they integrate AI tools while maintaining trust and protecting sensitive information.

Future of AI in Software Development

AI is expected to become a standard part of the developer toolkit, similar to version control systems, testing frameworks, and cloud platforms. Its capabilities will continue to improve, making it an indispensable assistant in everyday development tasks.

However, AI is unlikely to replace developers entirely. Instead, it will redefine their role. Human skills such as critical thinking, creativity, and responsibility will become even more valuable in an AI-driven environment.

The most effective approach will be a combination of human expertise and AI capabilities. AI excels at speed, pattern recognition, and automation, while humans provide context, judgement, and ethical oversight. This collaboration allows teams to work faster without compromising quality.

Conclusion

Artificial Intelligence is already reshaping software development and will continue to influence its future. While it offers significant benefits in terms of productivity and efficiency, it also introduces risks related to security, over-reliance, and ethics.

The key to success lies in balanced adoption: using AI to enhance development processes while maintaining strong human oversight, critical thinking, and responsibility. In this way, AI becomes not a replacement, but a powerful tool that supports developers in building better software.

References

1. Paiva R, Bittencourt II. Helping Teachers Help Their Students: A Human-AI Hybrid Approach. *Artificial Intelligence in Education*. 2020 Jun 9;12163:448–59. doi: 10.1007/978-3-030-52237-7_36. PMCID: PMC7334149.
2. Zeynep Ozpolat, Murat Karabatak, Özal Yildirim Artificial Intelligence-Based Tools in Software Development Processes: Application of ChatGPT. *European Journal of Technic*. 2023. Vol. 9. DOI:10.36222/ejt.1330631
3. Muhammad Azeem Akbar, Arif Ali Khan, Peng Liang Ethical Aspects of ChatGPT in Software Engineering Research. 2023.
4. E. Nascimento, A. Nguyen-Duc, I. Sundbø, and T. Conte, "Software engineering for artificial intelligence and machine learning software: A systematic literature review".

МЕТОДИ ЗАХИСТУ NEO ТА DYNAMITE

Ласкавчук Максим

здобувач вищої освіти

Кафедра прикладної математики та кібербезпеки

Донецький національний університет імені Василя Стуса, Україна

Анотація. Робота присвячена дослідженню методів захисту цифрових зображень від атак спотворення. У статті досліджуються два підходи: статичний метод NEO та динамічний метод DYNAMITE. Головною метою є оптимізація цих алгоритмів для підвищення точності виявлення маніпуляцій із зображеннями. На основі отриманих результатів проведено комплексний порівняльний аналіз обох методів.