

УДК 004.8:004.4:004.9

Assol Mezentseva¹, Nataliia Zhukova²

¹ student of group CST-124 NU “Zaporizhzhia Polytechnic”

² PhD (Philology), Associate Professor NU “Zaporizhzhia Polytechnic”

AI-POWERED CODING: THE FUTURE OF SOFTWARE DEVELOPMENT

AI-powered coding significantly accelerates software development by generating code snippets, debugging errors, and optimizing performance. Traditional coding, on the other hand, relies entirely on human effort, requiring more time for writing, testing, and debugging. Studies indicate that AI-driven tools like GitHub Copilot increase coding speed by up to 50 %. AI-driven coding tools provide optimized code suggestions that reduce redundancy and improve efficiency. In contrast, traditional coding depends on a developer's experience and skill level, potentially leading to suboptimal solutions. AI can also refactor existing codebases to enhance performance and readability. Traditional debugging requires manual review and extensive testing, which can be time-consuming and error-prone. AI-powered coding assistants detect syntax and logical errors in real time, offering quick fixes and recommendations, significantly reducing debugging time. While AI-generated code improves efficiency, it can introduce security vulnerabilities due to the lack of context awareness. Traditional coding allows for better security control, as developers manually assess potential risks. AI-generated code requires thorough review to ensure compliance with security standards.

Microsoft's GitHub Copilot, powered by OpenAI's Codex, is a prime example of AI-powered coding integration. It assists developers by generating entire code blocks, reducing the need for repetitive manual input. However, Copilot's limitations include potentially generating incorrect or biased code, necessitating human oversight to ensure accuracy.

Several AI-powered tools and language models have transformed coding efficiency. OpenAI Codex enables tools like GitHub Copilot to help developers generate, complete, and debug code based on natural language instructions. Tabnine is an AI-powered autocomplete tool that enhances developer productivity by suggesting entire lines of code, significantly improving workflow in languages like Python, JavaScript, and C++. DeepCode is a cloud-based AI-driven static code analysis tool that scans repositories for vulnerabilities and optimization

opportunities in multiple programming languages. Amazon CodeWhisperer is Amazon's AI-powered coding assistant designed to provide contextual suggestions and automate code generation within integrated development environments (IDEs). ChatGPT (GPT-4) is widely used for AI-assisted coding, offering explanations, debugging assistance, and generating complex functions in various programming languages.

Vibe Coding is an emerging trend in AI-powered development, where artificial intelligence converts human instructions into working code, making programming more intuitive and accessible. This approach is gaining traction in Silicon Valley, with companies striving to bridge the gap between human creativity and machine execution.

Another promising advancement is using Spiking Neural Networks (SNNs) in hybrid neural coding. This technique integrates deep learning with spike-based processing, improving pattern recognition efficiency and energy consumption. AI models leveraging SNNs are particularly effective in image processing and real-time decision-making.

Neural Network Coding (NNC) represents a novel combination of neural networks and data compression techniques. This approach enhances information transfer efficiency, optimizing software performance while reducing resource consumption. It is widely used in applications requiring high-speed data processing, such as autonomous systems and smart devices.

AI is also evolving beyond code generation into autonomous scientific discovery. AI-driven systems, sometimes called "AI Scientists," can autonomously design and execute experiments without human intervention. These advancements could revolutionize software engineering by automating algorithm selection, hyperparameter tuning, and even code refactoring.

Furthermore, AI-driven coding is increasingly accessible to beginners through Python-based tutorials. Educational platforms like Real Python provide step-by-step guidance on building neural networks, covering fundamental AI concepts like activation functions and learning algorithms. These resources help developers integrate AI capabilities into their projects more efficiently.

Despite its advantages, AI-powered coding has notable challenges. AI lacks deep comprehension of project-specific requirements, leading to inaccurate suggestions. AI models generate code based on existing datasets, which may include outdated or biased information. AI-generated code may unintentionally replicate copyrighted material, raising legal concerns.

AI-powered coding enhances speed, efficiency, and debugging while presenting challenges such as security risks and contextual limitations. Traditional coding remains essential for critical decision-making, security assurance, and customization. The future of software development lies in a hybrid approach, where AI acts as an assistant rather than a replacement for human programmers.