Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра комп'ютерно-інтегрованих технологій, автоматизації, робототехніки та безпекової інженерії

I Всеукраїнська конференція «Інтелектуальні технології цивільної безпеки та робототехнічні системи аварійно-рятувальних робіт»

I All-Ukrainian Conference “Intelligent Civil Safety Technologies and Robotic Systems for Emergency and Rescue Operations”

ICSTRO

2026

I All-Ukrainian Conference

February 12 - 13, 2026

Kharkiv

УДК: 005:004.896:62-65:338.3

Інтелектуальні технології цивільної безпеки та робототехнічні системи аварійно-рятувальних робіт 2026: матеріали І-ої Всеукраїнська конференція, Харків, 12-13 лютого 2026 р.: тези доповідей / [редкол. І.Ш. Невлюдов (відповідальний редактор)].-Харків: [електронний друк], 2026. – 192 с.

У збірник включені тези доповідей, які присвячені сучасним тенденціям розвитку технологій та засобів моделювання, прогнозування та управління ризиками у сфері цивільної безпеки; техногенна та виробнича безпека: технічні засоби, оцінка ризиків, експертиза; інтелектуальні та робототехнічні системи аварійно-рятувальних робіт; кіберфізичні системи, інформаційна безпека та цифровий захист виробництв; інформаційно-комунікаційні технології в системах управління та моніторингу надзвичайних ситуацій; сталий розвиток, екологічна безпека та соціальна відповідальність у сфері цивільної безпеки; інтелектуальні системи прийняття рішень у сфері цивільного захисту.

Intelligent Civil Safety Technologies and Robotic Systems for Emergency and Rescue Operations 2026: Proceedings of I st All-Ukrainian Conference, Kharkiv, February 12 - 13, 2026: Thesises of Reports / [Ed. I.Sh. Nevlyudov (chief editor).] .- Kharkiv .: [electronic version], 2026. - 192 p.

The collection includes the thesises of reports on devoted to current trends in the development of technologies and tools for modeling, forecasting, and risk management in the field of civil safety; industrial and technological safety, including technical means, risk assessment, and expert evaluation; intelligent and robotic systems for emergency and rescue operations; cyber-physical systems, information security, and digital protection of industrial facilities; information and communication technologies in emergency management and monitoring systems; sustainable development, environmental safety, and social responsibility in the field of civil safety; and intelligent decision-support systems in civil protection.

Харківський національний університет радіоелектроніки
Кременчуцький національний університет імені Михайла Остроградського
Національний університет «Запорізька політехніка»
Національний університет «Львівська політехніка»
Державне підприємство «Південний державний проектно-конструкторський та науково-дослідний інститут авіаційної промисловості»
Головне управління ДСНС України у Харківській області

Всеукраїнська конференція «Інтелектуальні технології цивільної безпеки та робототехнічні системи аварійно-рятувальних робіт» (ICSTRO-2026)

All-Ukrainian Conference “Intelligent Civil Safety Technologies and Robotic Systems for Emergency and Rescue Operations” (ICSTRO-2026)

ЗМІСТ

Elgun Jabrayilzade
Intelligent Control of a Collaborative Robot ……. 9
Volodymyr Makovii, Maryna Muntian
Electronic Control Systems for Bionic Prostheses Based on Microcontroller Platforms ……. 13
I. Andriukhin, S. Sotnik
The Concept of a Digital Twin as a Virtual Copy of Physical Objects, Processes, and Systems …….. 17
В. А. Вовченко, І. О. Толкунов
Управлінське рішення як елемент підвищення якості робіт з гуманітарного розмінування територій, забруднених ВНП ……. 22
M. Vorobyov, S. Sotnik
Jamstack Architecture as a Synthesis of Serverless Back-End and Dynamic Front-End ……. 25
Marina Muntian
Hybrid Seismic and Ultrasonic System for Autonomous Detection and Classification of Moving Objects ……. 30
I. Dvoinikova, S. Sotnik
Analysis of the Effectiveness and Cybersecurity Risks of the Github Copilot Tool ……. 34
I. Dvoinikova, S. Sotnik
6G Networks – A Technological Foundation for Autonomous Systems and the Internet of Everything ……. 39
Vladyslav Yevsieiev, Ihor Holod
Using Historical Data in the NNARX Model to Improve the Accuracy of Microclimate Parameter Forecasting ……... 44
K. Mandrykov, S. Sotnik
Comparative Analysis of Industrial Data Transmission Protocols (IIOT) in Automation Systems …….. 49
A. Taran, S. Sotnik
Digital Twin: A Virtual Copy of a Physical Object, Process, or System. Applications in Industry, Construction, and Cities ……. 54
R. Marunich, S. Sotnik
Security Analysis of Protocols for Integration With Access Control System ……... 59
Oleksandr Muntian
Comparative Analysis of Arduino, STM32 And ESP32 Platforms for Autonomous Sensor Systems …….. 64
A. Taran, S. Sotnik
AI as a Developer Tool: Github Copilot and Other Artificial Intelligence Assistants …….. 67
A. Fesenko, S. Sotnik
Selection of Communication Interfaces for a Microclimate Monitoring System ……. 72
Г. В. Пронюк, Геселева Н.В.
Моделювання інформаційних процесів у системах цивільної безпеки на основі DFD ... 77
A. Taran, S. Sotnik
WEB3 and Decentralized Applications. A Practical Look at Blockchain Development ……. 81

AI AS A DEVELOPER TOOL: GITHUB COPILOT AND OTHER ARTIFICIAL INTELLIGENCE ASSISTANTS

A. Taran, S. Sotnik
Kharkiv National University of Radio Electronics
Ukraine, 61166, Kharkiv, Nauky av., 14
E-mail: artem.taran1@nure.ua

***Annotation*:** The study provides a comprehensive analysis of the use of artificial intelligence as a tool to support software developers. The research systematizes the functional capabilities and limitations of modern AI assistants, analyzes their impact on the development process, and establishes the boundaries of the technology's applicability in complex and critical domains. The work substantiates the view that AI is transforming the role of the programmer, shifting the focus from writing code to architectural work and quality control, making the study of this symbiosis critically important for the effective future of the software industry.

Key words: artificial intelligence, AI assistants, programming, software development, GitHub Copilot.

ШІ ЯК ІНСТРУМЕНТ РОЗРОБНИКА: GITHUB COPILOT ТА ІНШІ АСИСТЕНТИ ЗІ ШТУЧНИМ ІНТЕЛЕКТОМ

А. Ю. Таран, С. В. Сотник
Харківський національний університет радіоелектроніки,
Україна, 61166, Харків, пр. Науки 14
E-mail: artem.taran1@nure.ua

***Анотація*:** У роботі проведено комплексний аналіз використання штучного інтелекту як інструмента підтримки розробника програмного забезпечення. Дослідження систематизує функціональні можливості та обмеження сучасних ШІ-асистентів, аналізує їхній вплив на процес розробки, а також встановлює межі застосовності технології у складних та критичних доменах. Робота обґрунтовує положення про те, що ШІ трансформує роль програміста, зміщуючи акцент з написання коду на архітектурну діяльність та контроль якості, що робить вивчення цього симбіозу критично важливим для ефективного майбутнього індустрії програмного забезпечення.

***Ключові слова*:** штучний інтелект, ШІ-асистенти, програмування, розробка програмного забезпечення, GitHub Copilot.

Artificial intelligence (AI) today is radically transforming the role of the developer, functioning not merely as an auxiliary tool but as a full-fledged intellectual partner in the code creation process [1-9]. Replacing traditional programming, which relied solely on manual work and experience, is a human-AI symbiosis, where intelligent systems analyze context, suggest entire blocks of code, automate routine tasks, and significantly increase productivity. The history of these changes begins with the development of automation and code analysis tools, which evolved from simple autocompletions to complex developer support systems [10-15].

Specific tools like GitHub Copilot, Amazon CodeWhisperer, Tabnine, or local models such as Code Llama bring this transition to practice. They integrate directly into the development environment (IDE) and, based on deep analysis of millions of lines of code, offer relevant autocomplete suggestions, create functions based on descriptions in comments, fill in template code, and help identify errors. This transforms programming into an interactive dialogue where the

developer defines high-level tasks and architectural decisions, and the AI assistant handles the technical implementation, significantly reducing cognitive load and speeding up development.

Thus, a modern AI development tool is not just a «coding assistant», but a systemic component that improves product quality through template standardization, reduces technical debt, and opens up opportunities for innovation, allowing programmers to focus on the complex, creative aspects of a project. Let's start by comparing traditional programming with AI-assisted programming (Table 1).

Table 1 – Comparison of traditional programming and AI-based programming

Aspect	Traditional approach to programming	Approach using AI assistants
Coding	The programmer writes each line of code manually	AI suggests ready-made code snippets based on context
Development speed	Relatively slow, depends on the programmer's experience	Significantly higher due to code autogeneration
Routine operations	Executed manually	Automated by AI assistants
Number of errors	Higher likelihood of syntax errors	Reduced thanks to prompts and checks if controlled
Learning new technologies	Requires additional time and documentation	AI helps explain code and provide examples
Developer role	Direct performer	Analyst, controller, and solution architect
Code quality	Depends on the individual experience of the programmer	Increases if properly controlled
Risks	Human factor	Incorrect code generation, dependency on AI

AI assistants for programmers are based on large language models trained on substantial amounts of open-source code, documentation, and technical texts. They are capable of understanding the syntax and semantics of programming languages, analyzing code context, and generating meaningful code snippets. Such assistants perform a wide range of functions: code autocompletion, generation of functions and classes, explanation of algorithms, suggestions for fixing errors, as well as assistance in writing comments and documentation. An important feature is that the AI assistant works directly in the development environment or in a dialogue format, making it accessible and convenient for a programmer's daily work. In addition to practical benefits, AI assistants also serve an educational purpose, facilitating faster learning of new technologies and the improvement of developers' skills.

GitHub Copilot is a prime example of a modern AI assistant actively used in professional software development. Its main feature is tight integration with popular development environments and the ability to generate code in real time. Copilot analyzes the current file context, project structure, and programmer comments, offering relevant code snippets. Beyond Copilot, there is a whole ecosystem of AI tools, including ChatGPT, Amazon CodeWhisperer, Tabnine, and others. They differ in approaches to code generation, level of integration, and focus on different user categories. A common feature of such tools is the aim to reduce the programmer's routine work and increase development efficiency. Examples of AI programming assistants are provided in Table 2.

Table 2 – Examples of AI assistants for programming

Tool	Main Purpose
GitHub Copilot	Code autocompletion and generation
ChatGPT	Code suggestions with security in mind
Amazon CodeWhisperer	Code explanation, algorithm generation
Tabnine	Intelligent autocompletion

The use of AI assistants leads to profound changes in the programming process itself. Instead of manually writing every line of code, the programmer increasingly takes on the role of a task setter and quality controller of the results. The main focus is on logic, system architecture, and the correctness of requirement implementation. The developer's role gradually evolves from an executor to a design engineer who makes strategic decisions and is responsible for the final outcome. Thus, AI does not replace the programmer but enhances their capabilities.

The use of AI assistants in programming has a number of advantages, including increased productivity, reduced development time, and fewer common errors. They are particularly effective when working with template code, standard algorithms, and typical structures. At the same time, there are certain risks associated with the use of AI. In particular, there is the possibility of generating incorrect or unsafe code, insufficient consideration of the specifics of a particular project, and the risk of excessive developer dependence on automated solutions. This requires mandatory human oversight and a responsible approach to the use of AI. In the future, AI tools for programming will become even more intelligent, capable of analyzing entire codebases, automatically detecting errors, suggesting optimal architectural solutions, and supporting the software product throughout all stages of its lifecycle.

AI assistants demonstrate high efficiency when working with standard constructs and typical algorithms; however, their performance significantly decreases when solving complex engineering tasks. In particular, when working with optimization algorithms, AI often offers basic implementations that do not take asymptotic complexity into account and may propose solutions with high computational cost. Complex data structures, such as self-balancing trees, priority graph algorithms, or concurrent lock-free structures, are generated with a high probability of errors and require significant manual refinement. Architectural solutions at the system level, including complex microservices patterns, Domain-Driven Design, and scaling strategies, require a deep understanding of the business context and infrastructure, which AI assistants do not possess.

AI limitations are particularly critical in specific domains where a coding error can lead to catastrophic consequences. In medical systems, AI does not take into account industry protocols and standards; in financial applications, regulatory requirements and calculation accuracy may be violated; in aviation and automotive software, safety standards and real-time system requirements are ignored. For scientific computations, AI often proposes unstable numerical methods without considering the accumulation of errors, and optimization for specific hardware for high-performance computing is implemented suboptimally. Embedded systems require special attention to memory constraints, energy efficiency, and precise timing when working with peripherals, which AI assistants cannot guarantee. In addition, AI only analyzes the local context and does not have access to business requirements, the history of architectural decisions, or corporate standards, which can lead to violations of encapsulation, the creation of cyclical dependencies, or the generation of code with typical security vulnerabilities. The optimal strategy is to use AI to generate initial solution options, followed by mandatory analysis and refactoring by an experienced developer, especially for critical business algorithms, systems with high security requirements, and code subject to certification.

This paper presents a comprehensive study on the use of AI assistants as tools for supporting software developers. It has been established that the use of systems such as GitHub Copilot, ChatGPT, Amazon CodeWhisperer, and Tabnine transforms the role of the developer from a direct executor to an analyst and solution architect. A comparison of traditional and AI-supported approaches revealed qualitative changes in development speed, automation of routine operations, and reduction of syntax errors, provided proper oversight is maintained.

The functional capabilities analysis revealed high efficiency of AI tools when working with template code, standard algorithms, and typical data structures. At the same time, critical limitations of AI assistants in complex engineering tasks were identified. AI performance significantly decreases when dealing with optimization algorithms, complex data structures, and system-level architectural

solutions, which require substantial manual refinement of the generated code. AI limitations are particularly critical in specialized domains where errors can lead to catastrophic consequences. In medical systems, AI does not take into account industry protocols and standards; in financial applications, regulatory requirements may be violated; and in aviation and automotive software, safety standards for real-time systems are ignored.

The practical significance of the study lies in forming a clear understanding of the capabilities and limitations of AI assistants. An optimal usage strategy is justified, where AI generates initial solutions for routine tasks, and an experienced developer performs mandatory review and refactoring, especially for critical business algorithms and systems with high security requirements. The results can be used to develop corporate standards for AI use in programming and to improve software quality.

REFERENCES

1. Nevludov, I. Sh. & et al.. Application of artificial intelligence in additive manufacturing (3D printing). Information Technologies and Automation – 2025 / Proceedings of the XVIII International Scientific and Practical Conference. Odessa, October 30-31, 2025. – Odessa, ONUT Publishing House, 2025.
2. Marunich, R.V. & et al.. Features of IoT application in the security sector. «Computer-integrated technologies, automation and robotics» CITAR-2025, 2025.
3. Polikanov, К. А. & et al.. Overview of modern technologies for residential automation. «Computer-integrated technologies, automation and robotics» CITAR-2025, 2025.
4. Yechevskyi A., & et al.. Methods Of Identification Of Objects On Industrial Lines. International Journal of Academic Engineering Research (IJAER), 2024.
5. Marunich, R., & et al.. Approaches to ensuring the effective implementation of IoT technologies in various industries. International Conference «DIGITAL INNOVATION & SUSTAINABLE DEVELOPMENT 2024», 2024.
6. Polikanov, К., & et al.. Smart home with house module: overview of automation technologies. International Conference «DIGITAL INNOVATION & SUSTAINABLE DEVELOPMENT 2024», 2024.
7. Sotnik, S. Integration of IoT into security systems: opportunities and risks. International Journal of Academic Engineering Research (IJAER), 2024.
8. Khalimonov, Y. I. et al.. Overview of computer vision areas application for inspection and quality control. Комп’ютерно-інтегровані технології автоматизації технологічних процесів на транспорті та у виробництві : матеріали всеукр. наук.-практ. конф. здобувачів вищ. освіти і молодих учених, 20 листоп. 2024 р. / Харків. нац. автомоб.-дор. ун-т. – Харків, 2024.
9. Lykho, T.A., Sotnik, S.V. Pattern recognition and computer vision technologies in decision support systems of robotic systems. Proceedings of the XVII International scientific and practical conference «Information technologies and automation – 2024», 2024.
10. Cherednichenko, T. & et al.. Features of automatic working time control systems. Manufacturing & Mechatronic Systems 2025: Proceedings of IX st International Conference, Kharkiv, October 25-26, 2025: Thesises of Reports, 2025.
11. Сотник, С. Розробка автоматизованої інформаційно-пошукової системи вибору маніпулятора промислових роботів. Електромеханічні і енергозберігаючі системи, 2025. Article ID 2072-2052.2025.1.68.6, https://doi.org/10.32782/2072-2052.2025.1.68.6.
12. Sotnik, S. Development of a range measurement module on an ultrasonic sensor with a GSM module. Radio Electronics, Computer Science, Control, 2025. Article ID 1607-3274-2025-2-3, https://doi.org/10.15588/1607-3274-2025-2-3

13. Sotnik, S. V. Development of automated control system and registration of metal in continuous casting. Radio Electronics, Computer Science, Control, 2024. Article ID 1607-3274-2024-3-17, https://doi.org/10.15588/1607-3274-2024-3-17

14. Fesenko, A. & et al.. Review and selection of optimal sensors for building a production facility microclimate monitoring system. Manufacturing & Mechatronic Systems 2025: Proceedings of IX st International Conference, Kharkiv, October 25-26, 2025: Thesises of Reports, 2025.

15. Andreiev, A. & et al.. “Web application security: protection against modern cyber threats” Overview of key vulnerabilities (XSS, CSRF, SQL injections), protection methods, use of HTTPS, authentication, and authorization. Manufacturing & Mechatronic Systems 2025: Proceedings of IX st International Conference, Kharkiv, October 25-26, 2025: Thesises of Reports, 2025.