

ISSN 2786-5460 (Print)
ISSN 2786-5479 (Online)

МІЖРЕГІОНАЛЬНА АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ
INTERREGIONAL ACADEMY OF PERSONNEL MANAGEMENT



ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СУСПІЛЬСТВО

INFORMATION TECHNOLOGY AND SOCIETY

Випуск 4 (15), 2024
Issue 4 (15), 2024



Видавничий дім
«Гельветика»
2024

*Рекомендовано до друку Вченою радою
Міжрегіональної Академії управління персоналом
(протокол № 12 від 26 грудня 2024 року)*

Інформаційні технології та суспільство / [головний редактор О. Попов]. – Київ : Міжрегіональна Академія управління персоналом, 2024. – Випуск 4 (15). – 130 с.

Журнал «Інформаційні технології та суспільство» є науковим рецензованим виданням, в якому здійснюється публікація матеріалів науковців різних рівнів у вигляді наукових статей з метою їх поширення як серед вітчизняних дослідників, так і за кордоном.

Редакційна колегія не обов'язково поділяє позицію, висловлену авторами у статтях, та не несе відповідальності за достовірність наведених даних і посилань.

Головний редактор: Попов О. О. – член-кор. НАН України, д-р техн. наук, професор, в.о. директора Центру інформаційно-аналітичного та технічного забезпечення моніторингу об'єктів атомної енергетики Національної академії наук України.

Редакційна колегія:

Василенко М. Д. – д-р фіз.-мат. наук, проф., професор кафедри кібербезпеки, Національний університет «Одеська юридична академія»; **Горбов І. В.** – канд. техн. наук, с.н.с., старший науковий співробітник, Інститут проблем реєстрації інформації НАН України; **Дуднік А. С.** – д-р техн. наук, доц., доцент кафедри мережевих та інтернет технологій, Київський національний університет імені Тараса Шевченка; **Євсєєв С. П.** – д-р техн. наук, професор кафедри кібербезпеки та інформаційних технологій, Харківський національний економічний університет імені Семена Кузнеця; **Зибін С. В.** – д-р техн. наук, доц., завідувач кафедри інженерії програмного забезпечення, Національний авіаційний університет; **Кавун С. В.** – д-р екон. наук, канд. техн. наук, проф., завідувач кафедри комп'ютерних інформаційних систем та технологій, Міжрегіональна Академія управління персоналом; **Комарова Л. О.** – д-р техн. наук, с.н.с., директор Навчально-наукового інституту інформаційної безпеки та стратегічних комунікацій, Національна академія Служби безпеки України; **Мілов О. В.** – д-р техн. наук, професор кафедри кібербезпеки та інформаційних технологій, Харківський національний економічний університет імені Семена Кузнеця; **Охріменко Т. О.** – канд. техн. наук, старший науковий співробітник науково-дослідної лабораторії протидії кіберзагрозам в авіаційній галузі, Національний авіаційний університет; **Рудніченко М. Д.** – канд. техн. наук, доц., доцент кафедри інформаційних технологій, Державний університет «Одеська політехніка»; **Скुरатовський Р. В.** – канд. фіз.-мат. наук, доц., доцент кафедри обчислювальної математики та комп'ютерного моделювання, Міжрегіональна Академія управління персоналом; **Супрун О. М.** – канд. фіз.-мат. наук, доц., доцент кафедри програмних систем і технологій, Київський національний університет імені Тараса Шевченка; **Табунщик Г. В.** – канд. техн. наук, проф., професор кафедри програмних засобів, Національний університет «Запорізька політехніка»; **Фомін О. О.** – д-р техн. наук, доц., професор кафедри комп'ютеризованих систем управління, професор кафедри прикладної математики та інформаційних технологій, Державний університет «Одеська політехніка»; **Хохлячова Ю. Є.** – канд. техн. наук, доц., доцент кафедри безпеки інформаційних технологій, Національний авіаційний університет; **Чолишкіна О. Г.** – канд. техн. наук, доц., директор Інституту комп'ютерно-інформаційних технологій та дизайну, Міжрегіональна Академія управління персоналом; **Чорний О. П.** – доктор технічних наук, професор, директор Навчально-наукового інституту електричної інженерії та інформаційних технологій, Кременчуцький національний університет імені Михайла Остроградського; **Юдін О. К.** – д-р техн. наук, проф., директор центру кібербезпеки Навчально-наукового інституту інформаційної безпеки та стратегічних комунікацій, Національна академія Служби безпеки України; **Гопєєнко Віктор** – dr. sc. ing., проф., проректор з наукової роботи, директор навчальної програми магістратури «Комп'ютерні системи», Університет прикладних наук ISMA (Латвійська Республіка); **Leszczyna Rafal** – dr hab. inż., професор кафедри комп'ютерних наук у менеджменті, Гданський технологічний університет (Республіка Польща).

Реєстрація суб'єкта у сфері друкованих медіа:

Рішення Національної ради України з питань телебачення і радіомовлення № 1173 від 11.04.2024 року.

Відповідно до Наказу МОН України № 1290 від 30 листопада 2021 року (додаток 3) журнал включено до Переліку наукових фахових видань України (категорія Б) зі спеціальностей 121 – Інженерія програмного забезпечення, 122 – Комп'ютерні науки, 123 – Комп'ютерна інженерія, 124 – Системний аналіз, 125 – Кібербезпека, 126 – Інформаційні системи та технології.

Усі електронні версії статей журналу оприлюднюються на офіційній сторінці видання
<http://journals.maup.com.ua/index.php/it>

Статті у виданні перевірені на наявність плагіату за допомогою програмного забезпечення
StrikePlagiarism.com від польської компанії Plagiat.pl.

*Recommended for publication
by Interregional Academy of Personnel Management
(Minutes No. 12 dated 26 December 2024)*

Information Technology and Society / [chief editor Oleksandr Popov]. – Kyiv : Interregional Academy of Personnel Management, 2024. – Issue 4 (15). – 130 p.

Journal «Information Technology and Society» is a peer-reviewed scientific edition, which publishes materials of scientists of various levels in the form of scientific articles for the purpose of their dissemination both among domestic researchers and abroad.

Editorial board do not necessarily reflect the position expressed by the authors of articles, and are not responsible for the accuracy of the data and references.

Chief editor: Oleksandr Popov – Corresponding Member of NAS of Ukraine, Doctor of Engineering, Professor, Acting Director of the Center for Information-Analytical and Technical Support of Nuclear Power Facilities Monitoring of the National Academy of Sciences of Ukraine.

Editorial Board:

Mykola Vasylenko – Doctor of Physics and Mathematics, Professor, Professor at the Department of Cybersecurity, National University «Odesa Law Academy»; **Ivan Horbov** – PhD in Engineering, Senior Research Associate, Senior Research Fellow, Institute for Information Recording of NAS of Ukraine; **Andrii Dudnik** – Doctor of Engineering, Associate Professor, Senior Lecturer at the Department of Networking and Internet Technologies, Taras Shevchenko National University of Kyiv; **Serhii Yevseiev** – Doctor of Engineering, Professor at the Department of Cybersecurity and Information Technologies, Simon Kuznets Kharkiv National University of Economics; **Serhii Zybin** – Doctor of Engineering, Associate Professor, Head of the Department of Software Engineering, National Aviation University; **Serhii Kavun** – Doctor of Economics, PhD in Engineering, Professor, Head of the Department of Computer Information Systems and Technologies Interregional Academy of Personnel Management; **Larysa Komarova** – Doctor of Engineering, Senior Research Scientist, Laureate of State Prize, Director of Educational-Scientific Institute of Information Security and Strategic Communications, National Academy of the Security Service of Ukraine; **Oleksandr Milov** – Doctor of Engineering, Professor at the Department of Cybersecurity and Information Technologies, Simon Kuznets Kharkiv National University of Economics; **Tetiana Okhrimenko** – PhD in Engineering, Senior Research Scientist at the Scientific Research Laboratory for Countering Aviation Cyberthreats, National Aviation University; **Mykola Rudnichenko** – PhD in Engineering, Associate Professor, Senior Lecturer at the Department of Information Technologies, Odessa Polytechnic State University; **Ruslan Skuratovskiy** – PhD in Physics and Mathematics, Associate Professor, Senior Lecturer at the Department of Computational Mathematics and Computer Modeling, Interregional Academy of Personnel Management; **Olha Suprun** – PhD in Physics and Mathematics, Associate Professor, Senior Lecturer at the Department of Software Systems and Technologies, Taras Shevchenko National University of Kyiv; **Halyna Tabunshchuk** – PhD in Engineering, Professor, Professor at the Department of Software Tools, “Zaporizhzhia Polytechnic” National university; **Oleksandr Fomin** – Doctor of Engineering, Associate Professor, Professor at the Department of Computerized Control Systems, Professor at the Department of Applied Mathematics and Information Technologies, Odessa Polytechnic State University; **Yuliia Khokhlachova** – PhD in Engineering, Associate Professor, Senior Lecturer at the Department of Information Technology Security, National Aviation University; **Olha Cholyshkina** – PhD in Engineering, Associate Professor, Director of the Institute of Computer Information Technologies and Design, Interregional Academy of Personnel Management; **Oleksii Chorny** – Doctor of Technical Sciences, Professor, Director of the Educational and Scientific Institute of Electrical Engineering and Information Technologies, Kremenchuk National University named after Mykhailo Ostrogradskiy; **Oleksandr Yudin** – Doctor of Engineering, Professor, Director of the Cybersecurity Center of the Educational-Scientific Institute of Information Security and Strategic Communications, National Academy of the Security Service of Ukraine; **Hopeienko Viktor** – dr. sc. ing., Professor, Vice Rector for Research, Director of the study programme “Computer systems”, ISMA University of Applied Sciences (Republic of Latvia); **Leszczyna Rafal** – dr hab. inż., Profesor, Katedra Informatyki w Zarządzaniu, Politechnika Gdańska (Republic of Poland).

Registration of Print media entity:

Decision of the National Council of Television and Radio Broadcasting of Ukraine: Decision No. 1173 as of 11.04.2024.

According to the Decree of MES No. 1290 (Annex 3) dated November 30, 2021, the journal was included in the List of scientific professional publications of Ukraine (category B) in specialties 121 – Software engineering, 122 – Computer sciences, 123 – Computer engineering, 124 – Systems analysis, 125 – Cybersecurity, 126 – Information systems and technologies.

All electronic versions of articles in the collection are available on the official website edition
<http://journals.maup.com.ua/index.php/it>

The articles were checked for plagiarism using the software
StrikePlagiarism.com developed by the Polish company Plagiat.pl.

© Interregional Academy of Personnel Management, 2024
© Copyright by the contributors, 2024

ЗМІСТ

Олег БОНДАРЧУК, Сергій ЗИБІН, Світлана ВАСИЛЮК-ЗАЙЦЕВА ЕФЕКТИВНІСТЬ ВИКОРИСТАННЯ КВАНТОВИХ КОМП'ЮТЕРІВ У РОЗВ'ЯЗАННІ СКЛАДНИХ ЗАВДАНЬ	6
Олександр БОРИСОВ, Марія ТЯГУНОВА ПІДХІД ДО АВТОМАТИЗАЦІЇ ЗАМОВЛЕНЬ У РЕСТОРАНІ З ВИКОРИСТАННЯМ ІТ-ТЕХНОЛОГІЙ	14
Тетяна ВАВРИК, Ліда ГОБИР ОПТИМІЗАЦІЯ ЗАХИСТУ ДАНИХ: ПРЕВЕНТИВНІ ТА РЕАКТИВНІ СТРАТЕГІЇ.....	21
Олександр ГОРДІЄНКО, Аліна КОВАЛЬ ПРОБЛЕМИ КОНФІДЕНЦІЙНОСТІ ТА ЕТИКИ У ВИКОРИСТАННІ ВІДКРИТИХ ДАНИХ ДЛЯ РОЗРОБКИ ДОДАТКІВ	26
Олександр ГОРДІЄНКО, Аліна КОВАЛЬ КОНЦЕПЦІЯ ПІДКЛЮЧЕННЯ ФІЗИЧНИХ ОБ'ЄКТІВ У РОЗУМНОМУ БУДИНКУ: ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ МОНІТОРИНГУ ТА ПОКРАЩЕННЯ ЯКОСТІ ПОВІТРЯ.....	30
Олександр ГОРДІЄНКО, Аліна КОВАЛЬ ВИКОРИСТАННЯ КРИПТОГРАФІЇ ЯК СЕРВІСУ У ВЕБ ПРОГРАМУВАННІ	35
Олександр ГОРДІЄНКО, Аліна КОВАЛЬ МАЙБУТНЄ ПРОГРАМУВАННЯ: ЯК ШТУЧНИЙ ІНТЕЛЕКТ ЗМІНЮЄ РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	40
Андрій ДУДНІК, Олег ТИЩЕНКО, Дарина ЯРЕМЕНКО ОГЛЯД СУЧАСНИХ ТЕХНІЧНИХ ТА ПРОГРАМНИХ РІШЕНЬ ДЛЯ УПРАВЛІННЯ БПЛА.....	44
Максим ДЬЯЧЕНКО, Андрій РОСКЛАДКА ВПРОВАДЖЕННЯ ШТУЧНОГО ІНТЕЛЕКТУ В ПРОЦЕС МЕНЕДЖМЕНТУ ІНЦИДЕНТІВ.....	51
Денис ЄФІМОВ, Роман ТИМОШЕНКО, Катерина ВОЙТЕХ ВПЛИВ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ НА СУЧАСНІ БОЙОВІ СТРАТЕГІЇ.....	58
Олексій КЛИМЕНКО СТВОРЕННЯ SELF-HEALING МЕРЕЖІ	63
Владислав КОЗУБ ІНФОРМАЦІЙНА ПІДТРИМКА ПРИЙНЯТТЯ РІШЕНЬ ІЗ ЗАСТОСУВАННЯМ ТЕХНОЛОГІЙ РОЗПОДІЛЕННОГО ШТУЧНОГО ІНТЕЛЕКТУ	71
Богдан КОРНІЄНКО, Леся ЛАДІЄВА, Ксенія УЛЬЯНИЦЬКА, Лілія ГАЛАТА Андрій НЕСТЕРУК ЗАХИСТ КРИТИЧНИХ РЕСУРСІВ ВЕБ-ЗАСТОСУНКУ З ОРЕНДИ НЕРУХОМОСТІ.....	80
Максим КУНДОС, Людмила СОЛОВЕЙ ВЕБ ДОДАТКИ У ЕКОСИСТЕМІ ІОТ.....	88
Сергій ЛУК'ЯНЕНКО, Павло ВДОВІН, Валентин ГРОМИКО, Роман ТИМОШЕНКО РОЛЬ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В СУЧАСНІЙ УКРАЇНСЬКІЙ ВІЙСЬКОВІЙ СПРАВІ.....	94
Геннадій МОГИЛЬНИЙ, Володимир ДОНЧЕНКО, Світлана ДОНЧЕНКО ОГЛЯД ТА АНАЛІЗ ІНСТРУМЕНТІВ СТВОРЕННЯ КОРПОРАТИВНОГО СЕРЕДОВИЩА.....	99
Олександр ПСАРЬОВ, Євген ДРУЖИНІН AGILE-ФРЕЙМВОРК ЯК КАТАЛІЗАТОР ЕФЕКТИВНОГО ВПРОВАДЖЕННЯ СИСТЕМ УПРАВЛІННЯ ІНФОРМАЦІЄЮ.....	108
Денис РЕДЬКО, Альона ДЕСЯТКО, Байтума БІСАРІНОВ, Айгуль БІСАРІАНОВА ОГЛЯД МЕТОДІВ АНАЛІЗУ ТРАФІКУ КОМПАНІЇ НА ОСНОВІ АНСАМБЛЕВОЇ КЛАСТЕРИЗАЦІЇ	115
Олександр СТОРОЖУК, Квітослава-Ольга ЯЦИНА ОСОБЛИВОСТІ ЗАСТОСУВАННЯ AR У ВІЗУАЛІЗАЦІЇ ТА АНАЛІЗІ АНАТОМІЧНИХ ОБ'ЄКТІВ З ВИКОРИСТАННЯМ ЕВКЛІДОВОЇ МЕТРИКИ В НАВЧАЛЬНОМУ ПРОЦЕСІ ПІДГОТОВКИ МЕДИЧНИХ ФАХІВЦІВ	125

УДК 004.8:004.42

DOI <https://doi.org/10.32689/maup.it.2024.4.7>

Олександр ГОРДІЄНКО

кандидат технічних наук, доцент кафедри комп'ютерних інформаційних систем та технологій,
Інститут комп'ютерно-інформаційних технологій та дизайну

ПрАТ «ВНЗ «Міжрегіональна Академія управління персоналом»,

oleksandr.m.hordienko@gmail.com

ORCID: 0009-0002-7764-8668

Аліна КОВАЛЬ

викладач кафедри комп'ютерних інформаційних систем та технологій,

Інститут комп'ютерно-інформаційних технологій та дизайну

ПрАТ «ВНЗ «Міжрегіональна Академія управління персоналом», sora9393@gmail.com

ORCID: 0009-0001-7379-5065

МАЙБУТНЄ ПРОГРАМУВАННЯ: ЯК ШТУЧНИЙ ІНТЕЛЕКТ ЗМІНЮЄ РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Анотація. Мета роботи. Дослідити вплив штучного інтелекту (ШІ) на процеси розробки програмного забезпечення, виявити основні напрямки змін, які спричиняє інтеграція ШІ у програмування, а також оцінити перспективи використання ШІ для оптимізації та автоматизації розробницьких процесів.

Методологія. Програмування вже давно стало основою сучасного світу, визначаючи розвиток технологій, бізнесу та суспільства. Однак із появою штучного інтелекту (ШІ) ця галузь переживає революційні зміни. ШІ не лише полегшує роботу розробників, автоматизуючи рутинні задачі, але й відкриває нові горизонти для творчості та інновацій. Від генерації коду до прогнозування поведінки систем, інструменти на основі ШІ змінюють саму суть програмування. Завдяки таким технологіям, як GitHub Copilot, TabNine чи ChatGPT, розробники отримали можливість працювати швидше, якісніше та ефективніше. ШІ вже зараз допомагає виявляти помилки, покращувати код і навіть створювати нові програмні рішення. Але як ці зміни вплинуть на майбутнє професії? Чи залишиться місце для людської творчості? І які виклики стоять перед програмістами в умовах стрімкого розвитку ШІ?

Наукова новизна. Вперше систематизовано основні підходи до застосування ШІ у розробці програмного забезпечення, такі як автоматизація кодування, виявлення помилок, оптимізація продуктивності програм та створення моделей генеративного дизайну. Представлено аналіз впливу генеративних мовних моделей (GPT, Codex тощо) на спрощення розробки та зміну ролі програмістів. Запропоновано новий погляд на майбутню співпрацю між розробниками і ШІ як «інтерактивну симбіотичну систему», де обидва учасники доповнюють сильні сторони один одного.

Висновок. Штучний інтелект уже змінює парадигму програмування, скорочуючи час на розробку, зменшуючи кількість помилок та підвищуючи продуктивність команд. У майбутньому роль розробників трансформуватиметься з написання коду на більш стратегічну та аналітичну діяльність, спрямовану на розв'язання складних задач, що потребують творчого підходу та критичного мислення. Програмування стає більш доступним для широкого кола людей, відкриваючи нові можливості для інновацій у різних галузях.

Ця стаття досліджує, як штучний інтелект трансформує процес розробки програмного забезпечення, які переваги він пропонує, і з якими ризиками доведеться стикнутися в майбутньому.

Ключові слова: штучний інтелект, обробка природної мови, машинне навчання, розробка програмного забезпечення, автоматична генерація коду.

Oleksandr HORDIENKO, Alina KOVAL. THE FUTURE OF PROGRAMMING: HOW ARTIFICIAL INTELLIGENCE IS TRANSFORMING SOFTWARE DEVELOPMENT

Abstract. Purpose of the work: To investigate the impact of artificial intelligence (AI) on software development processes, to identify the main areas of change caused by the integration of AI into programming, and to assess the prospects for using AI to optimize and automate development processes.

Methodology. Programming has long been a cornerstone of the modern world, shaping the development of technology, business, and society. However, with the advent of artificial intelligence (AI), this industry is undergoing revolutionary changes. AI not only makes developers' work easier by automating routine tasks, but also opens up new horizons for creativity and innovation. From code generation to predicting system behavior, AI-based tools are changing the very essence of programming.

Thanks to technologies such as GitHub Copilot, TabNine, and ChatGPT, developers have the opportunity to work faster, better, and more efficiently. AI is already helping to detect errors, improve code, and even create new software solutions. But how will these changes affect the future of the profession? Will there be room for human creativity? And what challenges do programmers face in the face of the rapid development of AI?

This article explores how artificial intelligence is transforming the software development process, what benefits it offers, and what risks it will face in the future.

Scientific novelty. For the first time, the main approaches to the application of AI in software development, such as coding automation, error detection, program performance optimization and the creation of generative design models, are systematized. An analysis of the impact of generative language models (GPT, Codex, etc.) on simplifying development and changing the role of programmers is presented. A new view of the future cooperation between developers and AI is proposed as an «interactive symbiotic system», where both participants complement each other's strengths.

Conclusion. Artificial intelligence is already changing the programming paradigm, reducing development time, reducing the number of errors and increasing team productivity. In the future, the role of developers will transform from writing code to a more strategic and analytical activity aimed at solving complex problems that require a creative approach and critical thinking. Programming is becoming more accessible to a wide range of people, opening up new opportunities for innovation in various industries.

This article explores how artificial intelligence is transforming the software development process, what benefits it offers, and what risks it will face in the future.

Key words: artificial intelligence, natural language processing, machine learning, software development, automatic code generation.

Вступ. Один із найважливіших аспектів впливу ШІ – можливість автоматичної генерації коду. Завдяки алгоритмам обробки природної мови (NLP), розробники можуть описувати бажаний функціонал у текстовій формі, а ШІ-системи генерують код на основі цих описів [5]. Це відкриває двері до більш інклюзивного програмування, де навіть користувачі без глибоких технічних знань можуть створювати прості додатки.

Генерація коду – це процес автоматичного створення програмного коду на основі заданих умов, специфікацій або вхідних даних [1]. Це може включати генерацію простих скриптів, функцій, класів або навіть складних програм, що виконують певні завдання.

Зазвичай, процес генерації коду включає кілька етапів:

1. *Визначення вимог.* Перед початком генерації коду необхідно чітко визначити, які функції, логіка або алгоритми мають бути реалізовані. Це можуть бути, наприклад, специфікації для роботи з базою даних, веб-сервером чи обробки даних.

2. *Перетворення вимог у структуровану модель.* На цьому етапі створюється модель, яка відображає архітектуру програми чи структуру даних. Це можуть бути діаграми класів, алгоритмічні блок-схеми або UML-діаграми.

3. *Генерація коду.* Використовуються інструменти або програми для створення коду. Це може бути спеціалізоване ПЗ або скрипти, які беруть на вхід структуру або шаблон і генерують код на певній мові програмування. Наприклад, генератори коду можуть створювати API клієнтів або сервіси за заданими параметрами.

4. *Автоматичні шаблони.* Генерація може базуватися на заздалегідь заданих шаблонах, що дозволяє швидко генерувати частини коду (наприклад, шаблони для роботи з базою даних, REST API тощо).

5. *Постобробка коду.* Після генерації код може потребувати додаткової обробки або оптимізації, щоб відповідати вимогам або специфікаціям проекту. Це може включати очищення коду, рефакторинг або інтеграцію з іншими частинами програми.

6. *Тестування згенерованого коду.* Важливий етап перевірки, щоб переконатися, що згенерований код працює правильно і відповідає вимогам. Для цього можуть використовуватись автоматичні тести чи інші методи перевірки.

Важливо зазначити, що генерація коду може бути частиною більш широкого процесу автоматизації розробки програмного забезпечення, зокрема в контексті таких інструментів, як IDE (Integrated Development Environment) або CI/CD (Continuous Integration/Continuous Delivery) [8].

Виявлення та виправлення помилок. ШІ-системи також досягли успіху у виявленні помилок у коді та їх виправленні. Наприклад, інструменти на основі машинного навчання аналізують великі обсяги коду, ідентифікуючи проблеми, які можуть бути неочевидними для людини. Вони також пропонують способи оптимізації та підвищення продуктивності коду.

Виявлення та виправлення помилок у програмному забезпеченні є важливими етапами в процесі розробки програм. У контексті ШІ-систем ці процеси можуть бути значно складнішими [15], оскільки помилки можуть бути неочевидними, і їх складно передбачити за допомогою традиційних методів тестування. Однак штучний інтелект може значно покращити ці етапи [3], надаючи нові підходи до виявлення та виправлення помилок.

1. *Виявлення помилок в ШІ-системах.* Існує кілька способів виявлення помилок в програмному коді та в самих моделях ШІ:

– Традиційні методи (дебагінг) – основний інструмент дебагінгу, дозволяють розробникам відстежувати виконання коду, крок за кроком, перевіряти змінні та стани програми. Дебагінг може допомогти виявити логічні помилки або невірну взаємодію між компонентами системи.

– Юніт-тести – тести які пишуться для кожної частини коду, що дозволяє перевіряти правильність роботи маленьких модулів програми. Цей метод може бути адаптований для перевірки окремих компонентів ШІ-моделей.

2. Методи для ШІ та машинного навчання:

- Логічне тестування моделей – виявлення помилок в ШІ-моделях може включати перевірку на логічні або статистичні аномалії, такі як невірна категоризація, класифікація або прогноз [10].
- Перевірка якості даних – ШІ-моделі дуже залежні від даних, на яких вони тренуються. Помилки можуть бути пов'язані з відсутніми, неповними або некоректними даними [2].
- Контроль перенавчання (overfitting) – виявлення того, чи модель надто пристосована до тренувальних даних, що може призвести до поганої роботи на нових або невідомих даних [14].
- Аналіз впливу параметрів (sensitivity analysis) – це дозволяє перевірити, як зміна вхідних даних або параметрів моделі впливає на результат, допомагаючи виявити потенційні помилки [6].

3. Інструменти на основі ШІ для виявлення помилок:

- Автоматичне тестування на основі ШІ – використання нейронних мереж або інших методів машинного навчання для аналізу великих обсягів коду або логів та автоматичного виявлення аномалій.
- Системи на основі ШІ для статичного аналізу коду – ШІ може вивчати структуру коду, знаходити непотрібні або недоопрацьовані частини, а також пропонувати виправлення на основі патернів помилок.

4. Виправлення помилок в ШІ-системах [4, 11]. Виправлення помилок в ШІ-системах може вимагати спеціальних підходів через складність моделей і залежність від великої кількості параметрів та даних.

- Адаптивне навчання – якщо помилка виникає через некоректні прогнози або рішення моделі, можна використати техніки адаптивного навчання, щоб модель змогла «навчитися» на нових, коректних даних або відкоригувати свої рішення.
- Техніки перенавчання (fine-tuning) – це може допомогти усунути помилки, які виникають при перенавченні моделі або неправильній генералізації. Для цього модель може бути додатково натренована на іншому наборі даних.
- Генерація контрприкладів – один із підходів для виправлення помилок в ШІ – це створення контрприкладів, які є спеціально сконструйованими прикладами, що демонструють, як модель може помилитися. Це допомагає моделі навчитися краще справлятися з різними варіантами ситуацій.
- Регуляризація та оптимізація – помилки в моделі можуть бути пов'язані з переоснащенням або недооснащенням. Використання технік регуляризації (наприклад, L1/L2 регуляризація) дозволяє моделі краще узагальнювати та уникати помилок.

- Оптимізація гіперпараметрів – пошук кращих параметрів для моделі за допомогою методів, таких як Grid Search, Random Search або баєсівська оптимізація.

5. Автоматичне виправлення коду за допомогою ШІ. ШІ може також допомогти в автоматичному виправленні помилок у коді. Наприклад, GitHub Copilot використовує GPT для генерування коду і виправлення помилок у програмному коді.

Використання ШІ для покращення процесів тестування та виявлення помилок. ШІ-системи можуть значно покращити процес тестування, оскільки вони можуть:

- автоматично генерувати тести для різних сценаріїв.
- визначати ненавмисні помилки на основі статистичних методів та аномалій.
- аналізувати великі набори даних, щоб знайти рідкісні або складні для виявлення помилки.

В цілому, ШІ може не тільки допомогти виявити та виправити помилки в програмному забезпеченні, а й значно покращити ефективність тестування та адаптації моделей в реальних умовах.

Як ШІ змінює підхід до розробки програмного забезпечення [7, 9, 16]

1. Прискорення циклу розробки. Інтеграція ШІ в процеси розробки дозволяє значно скоротити час, необхідний для створення програмного забезпечення [13]. Завдяки автоматизації та швидкій генерації коду, команди можуть зосередитися на вирішенні складних задач та впровадженні нових функцій, замість витрачання часу на рутинні завдання.

2. Підвищення якості продукту. Алгоритми ШІ здатні аналізувати велику кількість даних і знаходити закономірності, які складно виявити людині. Це допомагає у створенні більш стабільних і якісних продуктів, мінімізуючи ризик появи критичних помилок.

3. Інтеграція самонавчаючих систем. Завдяки ШІ можливим стало впровадження самонавчаючих систем у програмне забезпечення. Наприклад, програми можуть автоматично адаптуватися до потреб користувачів або змінювати свій функціонал залежно від змін у середовищі використання.

4. Демократизація програмування. Із розвитком ШІ програмування стає доступним для ширшої аудиторії. Люди, які не володіють глибокими технічними знаннями, тепер можуть створювати прості програми або навіть складні системи за допомогою інтуїтивно зрозумілих інструментів. Це стимулює інновації в різних сферах – від бізнесу до освіти.

Виклики та ризики використання ШІ в програмуванні.

1. *Етичні питання.* Зростання залежності від ШІ ставить нові етичні виклики. Наприклад, як забезпечити, щоб автоматично згенерований код не містив упереджень чи неетичних рішень? Крім того, автоматизація може призвести до втрати робочих місць серед програмістів початкового рівня.

2. *Надмірна залежність від технологій.* Залежність від ШІ може створити ризики, коли розробники покладаються на автоматизацію настільки, що втрачають здатність до самостійного вирішення проблем. Це може призвести до втрати критичного мислення та професійних навичок.

3. *Безпека та конфіденційність.* Алгоритми ШІ часто базуються на великих обсягах даних, що можуть включати конфіденційну інформацію. Захист цих даних є критично важливим, оскільки витоки або зловживання можуть завдати значної шкоди.

4. *Якість та контроль.* Хоча ШІ може автоматизувати велику частину процесу розробки, зберігається ризик генерації коду, який важко перевірити або оптимізувати вручну. Розробники мають бути готовими до ретельного аналізу автоматично створених рішень.

5. *Перспективи розвитку, нові інструменти та платформи.* Очікується, що у майбутньому з'являться ще більш потужні інструменти, які будуть поєднувати можливості ШІ з традиційними підходами до розробки. Вони надаватимуть більш гнучкі можливості для команд будь-якого рівня кваліфікації.

6. *Співпраця людини та ШІ.* Майбутнє програмування полягає у тісній співпраці між людиною та ШІ. Люди забезпечуватимуть творчий і стратегічний підхід, тоді як ШІ автоматизуватиме рутинні процеси та пропонуватиме оптимальні рішення.

7. *Підготовка нових поколінь розробників.* Системи освіти мають адаптуватися до змін, які вносять ШІ в програмування. Навчання повинно зосереджуватися не лише на традиційних методах кодування, але й на використанні інструментів ШІ, управлінні їхніми ризиками та створенні етичних рішень.

Висновки. Штучний інтелект уже змінює програмування, роблячи його більш доступним, ефективним і якісним. Попри численні виклики, які супроводжують ці зміни, можливості, які відкриває ШІ, значно переважають ризики. Майбутнє програмування буде визначатися синергією між людськими навичками та потужністю ШІ, що дозволить створювати інноваційні продукти, які змінюють світ.

Список використаних джерел:

- Ahmad W., Chakraborty S., Ray B., Chang K. W. Unified Pre-training for Program Understanding and Generation. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021. 2655–2668. <https://doi.org/10.18653/v1/2021.naacl-main.210>
- Chen H., Li Z. "Data Quality in Machine Learning: Challenges and Methods for Error Detection." *International Journal of Data Science and Analytics*, 2020. 9(2), 121–137.
- Chen M., Tworek J., Jun H., Yuan Q., de Oliveira Pinto H. P., Kaplan J., Schulman J. Evaluating Large Language Models Trained on Code. arXiv preprint arXiv:2107.03374. 2021.
- Davis K., Lee R. Fine-tuning models: Techniques and challenges in AI error correction. *International Journal of AI and Software Engineering*, 2020. 12(2), 101–115.
- Feng Y., Guo D., Tang D., Duan N., Wei Z., Zhou M., Yin J. CodeBERT: A Pre-Trained Model for Programming and Natural Languages. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020. 1536–1547. <https://doi.org/10.18653/v1/2020.emnlp-main.154>
- Jiang Z., Sun D. "Sensitivity Analysis in AI Models: Evaluating the Impact of Parameter Changes." *Artificial Intelligence Review*, 2020. 53(3), 159–179.
- Kumar R., Singh P. "Enhancing Software Quality with AI-Based Data Analytics." *Software Engineering Review*, 2020. 8(1), 45–59. <https://doi.org/10.1109/ser.2020.015007>.
- Li C., Li H., Sun J. Deep Learning for Automatic Code Generation and Completion: A Survey. *ACM Computing Surveys (CSUR)*, 2021. 54(6), 1–29. <https://doi.org/10.1145/3469023>.
- Li Z., Chen L. "AI-Powered Tools for Code Generation and Testing Automation." *International Journal of Computer Science and Information Technology*, 2020. 12(2), 187–202. <https://doi.org/10.1093/ijcsit/ijcsit.2020.012030>.
- Liu Z., Zhang C., Wang Z. "Logical Testing of Machine Learning Models: A Comprehensive Review." *Journal of Machine Learning Research*, 2021. 22(4), 153–172.
- Smith J., Brown A. Adaptive learning techniques for error correction in AI systems. *Journal of Machine Learning and Artificial Intelligence*, 2020. 18(4), 234–249.
- Tiwari S., Chaturvedi A., Mishra R. Artificial Intelligence in Software Engineering: Benefits, Challenges, and Future Prospects. *International Journal of Advanced Research in Computer Science*, 2021. 12(2), 45–51. <https://doi.org/10.26483/ijarcs.v12i2>.
- Vaithilingam P., Chen J., Alvarado C. Expectations vs. Reality: How Software Developers Use Generative AI Tools for Code. Proceedings of the 2022 ACM CHI Conference on Human Factors in Computing Systems, 2022. 1–15. <https://doi.org/10.1145/3491102.3517489>.
- Wang Y., Xu L. "Overfitting in Machine Learning Models: Approaches to Detection and Mitigation." *Journal of Artificial Intelligence Research*, 2022. 68(1), 45–63.
- Zaremba W., Sutskever I. Recurrent Neural Network Models for Code Generation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 201–213.
- Zhang Y., Wang X. "Artificial Intelligence and Its Impact on Software Development." *Journal of Software Engineering and Applications*, 2020. 13(4), 233–245. <https://doi.org/10.4236/jsea.2020.134014>.