УДК 004.8:37.018 DOI https://doi.org/10.35546/kntu2078-4481.2025.1.2.16

Л. І. КУБЛІЙ
кандидат технічних наук, доцент,
доцент кафедри цифрових технологій в енергетиці
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
ORCID: 0000-0002-1015-3209

М. О. ШЕЛЕСТЮК
магістрант кафедри цифрових технологій в енергетиці
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
ORCID: 0009-0000-2776-6960

СУЧАСНІ МЕТОДИ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ НАВЧАННЯ ПРОГРАМУВАННЯ

У статті представлено розробку інтегрованої програмної системи на базі технологій штучного інтелекту для автоматизації та оптимізації процесу навчання програмування. Досліджено актуальні проблеми використання штучного інтелекту в освітньому процесі, включаючи питання доступності технологій, якості генерованого коду та адаптації навчального процесу до сучасних вимог. Запропоновано інноваційне рішення для ефективної співпраці між програмістом і штучним інтелектом, яке базується на інтеграції передових моделей машинного навчання.

Розроблена система використовує моделі CodeGen та CodeBERT для комплексного аналізу і генерації програмного коду. Модель CodeGen-350M-mono забезпечує генерацію синтаксично правильного коду і контекстуальні підказки, тоді як модель CodeBERT виконує глибокий аналіз структури програм і семантичних зв'язків. Унікальною особливістю системи є можливість роботи як в онлайн-, так і в офлайн-режимах завдяки попередньому завантаженню моделей на локальний сервер, що робить її доступною для використання в різних навчальних умовах.

Практична реалізація включає створення веб-застосунку на базі мікрофреймворку Flask, який надає інтуїтивно зрозумілий інтерфейс для взаємодії з моделями штучного інтелекту і забезпечує швидку обробку запитів користувачів. Система демонструє високу ефективність при аналізі введеного коду, наданні контекстуальних пояснень і формуванні рекомендацій щодо покращення програмних рішень. Важливою функціональністю є можливість автоматичного доповнення коду і виявлення потенційних помилок на ранніх етапах розробки програмного забезпечення. Створений веб-застосунок забезпечує ефективну співпрацю між програмістом і штучним інтелектом.

Ключові слова: *штучний інтелект, навчання програмування, CodeGen, CodeBERT, Flask, веб-розробка, офлайн-режим, автоматизація навчання, машинне навчання, генерація коду, аналіз коду, освітні технології, програмна інженерія.*

L. I. KUBLII
Candidate of Technical Sciences, Associate Professor,
Associate Professor at the Department of Digital Technologies in Energy
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"
ORCID: 0000-0002-1015-3209

M. O. SHELESTIUK
Master's Student at the Department of Digital Technologies in Energy
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"
ORCID: 0009-0000-2776-6960

MODERN AI METHODS FOR PROGRAMMING EDUCATION

The paper presents the development of an integrated software system based on artificial intelligence technologies for automating and optimizing the programming education process. Current issues of artificial intelligence usage in the educational process are investigated, including technology accessibility, generated code quality, and adaptation of the learning process to modern requirements. An innovative solution for effective collaboration between programmer and AI is proposed, based on the integration of advanced machine learning models.

The developed system utilizes CodeGen and CodeBERT models for comprehensive code analysis and generation. Model CodeGen-350M-mono provides syntactically correct code generation and contextual hints, while CodeBERT performs deep analysis of program structure and semantic relationships. A unique feature of the system is its ability to operate in both online and offline modes through pre-loaded models on a local server, making it accessible for use in various educational settings.

The practical implementation includes creating a Flask-based web application that provides an intuitive interface for interaction with AI models and ensures fast processing of user requests. The system demonstrates high efficiency in analyzing input code, providing contextual explanations, and forming recommendations for improving programming solutions. Important functionality includes automatic code completion and early detection of potential errors during development. The created web application ensures effective cooperation between the programmer and artificial intelligence.

Key words: *artificial intelligence, programming education, CodeGen, CodeBERT, Flask, web development, offline mode, education automation, machine learning, code generation, code analysis, educational technology, software engineering.*

Постановка проблеми

З розвитком сучасних технологій особливої актуальності набуває використання штучного інтелекту в програмуванні завдяки його здатності автоматизувати та оптимізувати процеси розробки програмного забезпечення. Проблема, розглянута в цьому дослідженні, полягає в створенні програмного продукту, який поєднує ефективність парного програмування людини-програміста і штучного інтелекту.

Основним завданням є розробка продукту, який забезпечить ефективну співпрацю між програмістом і штучним інтелектом. Продукт повинен приймати на вхід код, написаний програмістом, і генерувати його продовження. Головною метою є створення універсального рішення, здатного працювати як в онлайн-, так і офлайн-режимах.

Аналіз останніх досліджень і публікацій

У теперішній час впровадження штучного інтелекту в процес навчання програмування стає все більш актуальним, і це підтверджується результатами досліджень. Так, опитування серед студентів Стенфордського університету показало, що кожен п’ятий студент використовує ChatGPT для допомоги в роботі над завданнями з програмування, а 5 % з них не редагували згенерований штучним інтелектом текст перед здачею [1].

Штучний інтелект також активно застосовують для вдосконалення навичок програмування, про що свідчать численні дослідження. Наприклад, використання штучного інтелекту для автоматичного аналізу коду, розпізнавання і корекції помилок стало предметом наукових робіт таких авторів, як O. Zawacki-Richter та його колег, які досліджували перспективи використання технологій штучного інтелекту у вищій освіті [2]. Колумбійські вчені Rodríguez-Hernández, Musso, Kyndt і Cascallar впровадили штучні нейронні мережі для прогнозування академічної успішності студентів, що також застосовується при оцінюванні рівня володіння програмуванням [3].

Ранні дослідження [4], які стосуються використання штучного інтелекту українськими учнями й студентами для вивчення програмування, виявили позитивні результати. Студенти активно використовують штучний інтелект як інструмент для розв’язання складних задач.

Сучасні методи штучного інтелекту значно полегшують навчання програмування, підвищуючи ефективність і точність навчального процесу. Технології штучного інтелекту забезпечують швидке й глибоке засвоєння матеріалу, дають можливість студентам розвивати свої навички програмування більш ефективно і з меншими зусиллями. Подальший розвиток цих технологій лише поглиблюватиме інтеграцію штучного інтелекту в освіту, створюючи нові можливості для навчання учнів і студентів у сфері програмування.

Формулювання мети дослідження

Головною метою дослідження є розробка інтегрованої програмної системи на базі технологій штучного інтелекту для автоматизації та оптимізації процесу навчання програмування з можливістю офлайн-використання попередньо завантажених моделей штучного інтелекту. Система повинна поєднувати ефективні моделі штучного інтелекту для генерації та аналізу програмного коду, забезпечувати зручний веб-інтерфейс для взаємодії користувачів, а також працювати незалежно від наявності інтернет-з’єднання.

Технологічними цілями дослідження є:

– створення серверного застосунку на базі мікрофреймворку Flask для забезпечення взаємодії користувачів з моделями штучного інтелекту;

– інтеграція та оптимізація роботи моделей CodeGen та CodeBERT для генерації й аналізу програмного коду;

– розробка системи офлайн-доступу до функціоналу штучного інтелекту через попереднє завантаження моделей;

– створення інтуїтивного веб-інтерфейсу для взаємодії з різними моделями штучного інтелекту.

Практичними цілями дослідження є:

– забезпечення безперебійного доступу до інструментів штучного інтелекту незалежно від наявності інтернет-з’єднання;

– оптимізація швидкодії системи через попереднє завантаження моделей.

Технічними результатами дослідження мають бути:

– працюючий веб-застосунок з інтегрованими моделями штучного інтелекту;
– оптимізована система офлайн-доступу до моделей штучного інтелекту.

Дослідження має важливе значення для розвитку методів навчання програмування, оскільки воно створює нові можливості для автоматизації навчального процесу, забезпечує доступність сучасних інструментів штучного інтелекту в офлайн-режимі і надає можливість порівняти ефективність різних підходів до генерації коду. Інноваційність роботи полягає у створенні комплексного рішення для офлайн-доступу до моделей штучного інтелекту.

Викладення основного матеріалу дослідження

Першим кроком дослідження було створення сервера, здатного приймати дані від користувачів і надсилати їм відповіді. Для реалізації цієї задачі обрано бібліотеку Flask [5], яка є одним із найпоширеніших інструментів для розробки веб-застосунків мовою Python. Бібліотека Flask забезпечує простоту використання і потужний функціонал для створення ефективної взаємодії між сервером і клієнтами.

У програмному коді сервера всі необхідні змінні оголошено до основної частини коду, який обробляє запити. Такий підхід забезпечує оптимізацію роботи моделей, оскільки вони завантажуються один раз при старті сервера і залишаються в оперативній пам’яті, що зменшує затримки під час виконання кожного запиту.

Розглянемо розроблений програмний код для запуску сервера.

Система починає роботу з завантаження необхідних моделей та ініціалізації веб-сервера:

```
from transformers import AutoTokenizer, AutoModelForCausalLM, pipeline
from flask import Flask, request, render_template

app = Flask(__name__)
```

Далі здійснюється налаштування обох моделей машинного навчання:

```
model_identifier = «Salesforce/codegen-350M-mono»
tokenizer = AutoTokenizer.from_pretrained(model_identifier)
model = AutoModelForCausalLM.from_pretrained(model_identifier, device_map=“cpu”)
pipe = pipeline(“fill-mask”, model=“neulab/codebert-python”)
```

Система реалізує два основні алгоритми обробки програмного коду. Це алгоритм CodeBERT для заповнення пропусків у коді і алгоритм CodeGen для генерації програмного коду:

```
def codebert_models(user_input):
   return pipe(user_input)

def codegen_models(user_input):
   inputs = tokenizer(user_input, return_tensors=»pt»)
   outputs = model.generate(inputs[“input_ids”], max_length=100, num_return_sequences=1)
   decoded_output = tokenizer.decode(outputs[0], skip_special_tokens=True)
   return decoded_output
```

Реалізація основного маршруту для обробки HTTP-запитів здійснюється так:

```
@app.route(“/”, methods=[“GET”, “POST”])
def index():
   user_input = None
   choice = None
   answer_models = None

   if request.method == “POST”:
      user_input = request.form.get(“user_input”)
      choice = request.form.get(“choice”)

      if choice == “codebert”:
         answer_models = codebert_models(user_input)
      elif choice == “codegen”:
         answer_models = codegen_models(user_input)

   return render_template(
      “index.html”, user_input=user_input, choice=choice, answer_models=answer_models
   )

if __name__ == “__main__”:
   app.run(debug=True)
```

Розроблена система інтегрує сучасні методи машинного навчання з веб-технологіями, що забезпечує зручний інтерфейс для генерації та аналізу програмного коду.

Архітектуру інтегрованої програмної системи подано на рис. 1.

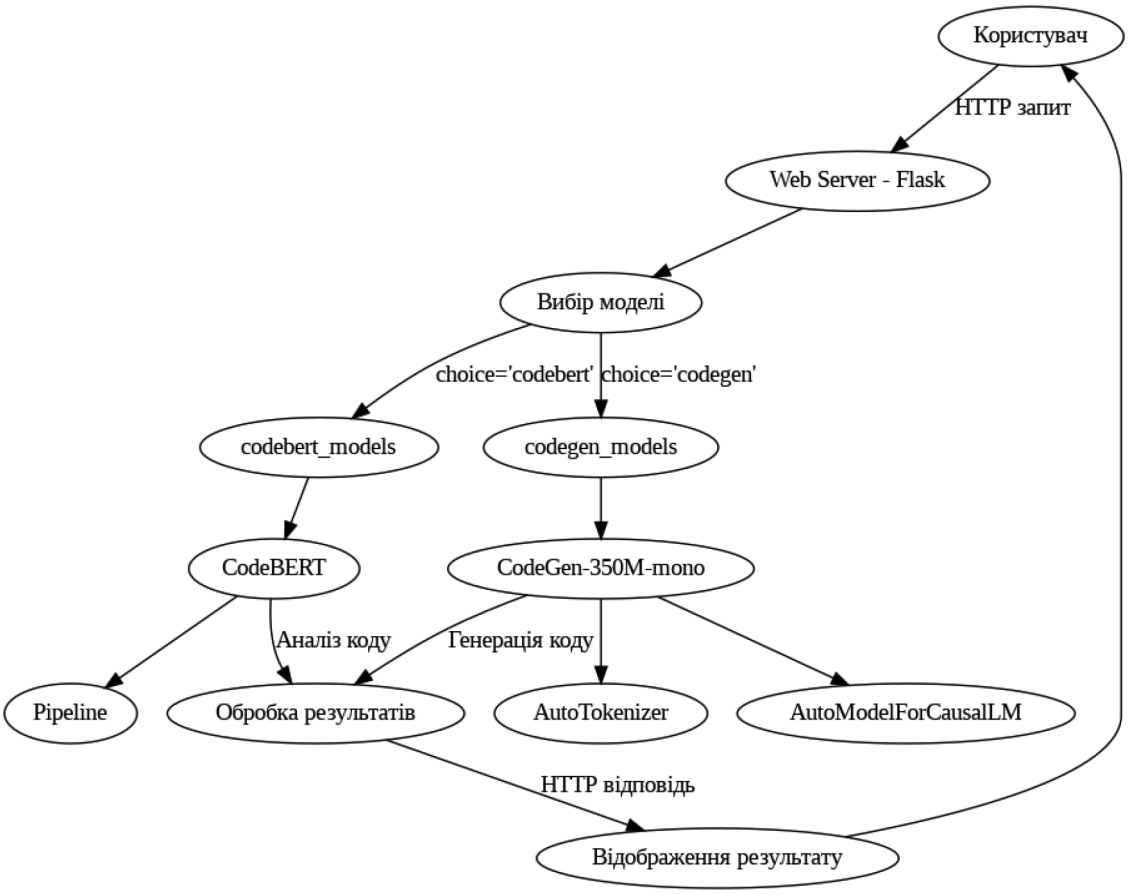

Рис. 1. Архітектура інтегрованої програмної системи

Використані в коді змінні та їхнє призначення подано в таблиці 1.

Таблиця 1

Змінні, які використовуються в коді, та їхнє призначення

Змінна	Призначення
Model_identifier	Завантажує модель codegen-350M-mono для роботи з нею
tokenizer	Токенізатор
model	Сама модель, яка буде надавати відповідь за запит
pipe	Завантаження моделі codebert-python

Використані методи передбачають інтеграцію двох моделей штучного інтелекту – Codegen-350M-mono і CodeBERT.

Модель Codegen-350M-mono має 350 мільйонів параметрів, які можна навчити [6]. Вона навчена на монолінгвальному наборі даних коду і спеціалізується на генерації нового коду, оптимізована для створення послідовного синтаксично правильного коду, добре розуміє контекст запиту користувача і пропонує релевантні відповіді. Модель підтримує автодоповнення коду і може пропонувати варіанти завершення програмних конструкцій. Через свій помірний розмір вона ефективна для розв'язання відносно простих завдань програмування.

Модель CodeBERT [7], натомість, базується на архітектурі BERT і адаптована для роботи з програмними кодами різними мовами програмування. Вона здатна розуміти семантичні зв'язки між частинами коду, виконувати глибокий аналіз його структури, має широку функціональність. Модель CodeBERT використовують для пошуку схожого коду, аналізу функціональності, виявлення помилок і вразливостей, генерації документації, класифікації типів коду.

Основна відмінність між цими моделями полягає в їхньому призначенні: модель Codegen-350M-mono фокусується на генерації нового коду, тоді як модель CodeBERT спеціалізується на аналізі та розумінні наявного коду.

Обидві моделі було попередньо завантажено на локальний сервер, що надало можливість їхнього використання навіть без підключення до Інтернету.

Після запуску застосунку відображається його головна сторінка (рис. 2).

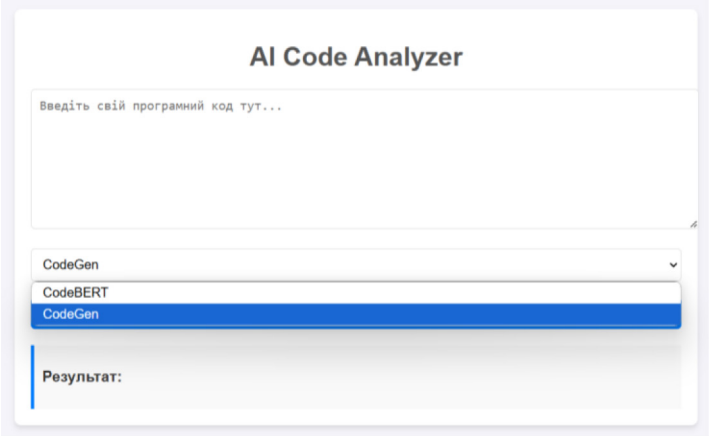

Рис. 2. Головна сторінка застосунку

Результат тестування роботи моделі CodeBERT подано на рис. 3.

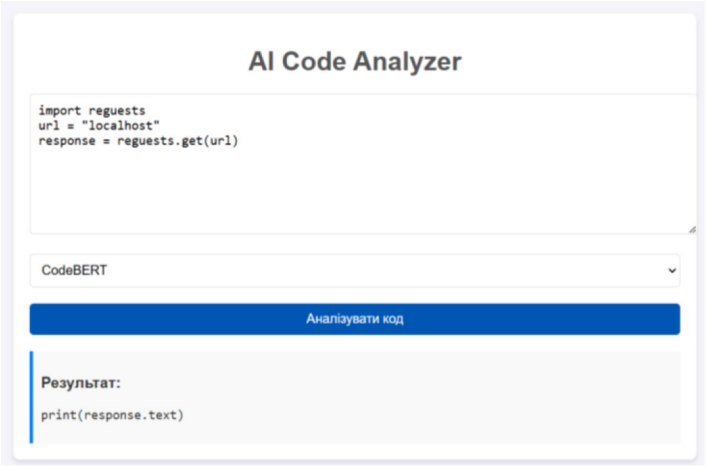

Рис. 3. Тестування моделі CodeBERT

Користувач взаємодіє із системою, вставляючи свій програмний код у відповідне поле. Застосунок є ефективним інструментом для генерації та доповнення програмного коду. Він може бути корисним для розробників програмного забезпечення, які бажають автоматизувати рутинні завдання створення і доповнення коду.

Висновки

У результаті проведеного дослідження було розроблено і впроваджено інтегровану програмну систему на базі технологій штучного інтелекту для автоматизації та оптимізації процесу навчання програмування. Основні досягнення і результати дослідження поєднують кілька ключових аспектів. Спочатку було реалізовано серверний застосунок на базі мікрофреймворку Flask, який забезпечує ефективну взаємодію користувачів з моделями штучного інтелекту, стабільну роботу в режимі офлайн завдяки попередньому завантаженню моделей, а також оптимізовану швидкодію системи. Крім того, було успішно інтегровано дві різні моделі штучного інтелекту: модель CodeGen для генерації програмного коду і модель CodeBERT для аналізу та обробки коду.

На практичному рівні було створено функціональний веб-інтерфейс, який забезпечує зручне введення користувацького коду, можливість вибору між різними моделями штучного інтелекту та отримання контекстуальних відповідей від системи. Окремо було реалізовано офлайн-функціональність, що дає можливість працювати з системою без постійного підключення до Інтернету, зберігаючи при цьому високу швидкодію завдяки попередньому завантаженню моделей.

У ході дослідження впроваджено вдосконалення застосунку. Зокрема, оптимізовано роботу серверної частини шляхом попереднього завантаження моделей, що зменшує затримки і покращує швидкодію застосунку навіть при високому навантаженні. Реалізовано механізм паралельної роботи моделей CodeGen і CodeBERT для генерації

та аналізу коду, що забезпечує стабільність і точність відповідей через оптимізацію форматів даних. Покращено алгоритми обробки запитів користувачів для врахування контексту задачі, що підвищило якість генерації коду. Додано функціонал офлайн-режиму з кешуванням і розподілом ресурсів для автономної роботи.

Перспективи подальшого розвитку дослідження включають розширення функціональності системи новими моделями штучного інтелекту, вдосконалення механізмів аналізу й генерації коду, розробку додаткових навчальних функцій та інтеграцію з наявними освітніми платформами. Розроблена система має значний потенціал для впровадження в навчальних закладах, використання в процесі самонавчання програмування, застосування в корпоративному навчанні, що сприяє підвищенню ефективності вивчення і вироблення навичок програмування.

Список використаної літератури

1. The Stanford Daily. Scores of Stanford students used ChatGPT on final exams, survey suggests. 2023, January 22. URL: https://stanforddaily.com/2023/01/22/scores-of-stanford-students-used-chatgpt-onfinal-exams-survey-suggests/ (дата звернення: 10.01.2025).

2. Zawacki-Richter O., Marín V. I., Bond M., Gouverneur F. Systematic review of research on artificial intelligence applications in higher education – where are the educators? *International Journal of Educational Technology in Higher Education*. 2019. № 16(1). Article 39. doi: https://doi.org/10.1186/s41239-019-0171-0

3. Rodríguez-Hernández C. F., Musso M., Kyndt E., Cascallar E. Artificial neural networks in academic performance prediction: Systematic implementation and predictor evaluation. *Computers and Education: Artificial* Intelligence. 2021. № 2, 100018. URL: doi: https://doi.org/10.1016/j.caeai.2021.100018

4. Міністерство освіти і науки України. Результати всеукраїнського дослідження про перспективи ШІ в загальній середній освіті. 20 грудня 2023. URL: https://mon.gov.ua/news/rezultati-vseukrainskogo-doslidzhennya-pro-perspektivi-shi-v-zagalniy-seredniy-osviti (дата звернення: 10.01.2025).

5. Flask documentation. https://flask.palletsprojects.com/en/stable/ (дата звернення: 10.01.2025).

6. CodeGen (CodeGen-Mono 350M). URL: https://huggingface.co/Salesforce/codegen-350M-mono (дата звернення: 10.01.2025).

7. CodeBERT: A Pre-Trained Model for Programming and Natural Languages. / Zhangyin Feng, et al. *Findings of the Association for Computational Linguistics: EMNLP*. 2020, November 16–20. P. 1536-1547. URL: https://aclanthology.org/2020.findings-emnlp.139.pdf. (дата звернення: 10.01.2025).

References

1. The Stanford Daily. (2023, January 22). Scores of Stanford students used ChatGPT on final exams, survey suggests. URL: https://stanforddaily.com/2023/01/22/scores-of-stanford-students-used-chatgpt-onfinal-exams-survey-suggests/

2. Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education – where are the educators? *International Journal of Educational Technology in Higher Education*, 16(1), article 39. doi: https://doi.org/10.1186/s41239-019-0171-0

3. Rodríguez-Hernández, C. F., Musso, M., Kyndt, E., & Cascallar, E. (2021). Artificial neural networks in academic performance prediction: Systematic implementation and predictor evaluation. *Computers and Education: Artificial Intelligence*, 2, 100018. doi: https://doi.org/10.1016/j.caeai.2021.100018

4. Ministry of Education and Science of Ukraine. (2023). Results of the all-Ukrainian research on the prospects of AI in general secondary education. URL: https://mon.gov.ua/ua/news/rezultativseukrayinskogo-doslidzhennya-pro-perspektivi-shi-v-zagalnij-serednij-osviti [in Ukrainian].

5. Flask's documentation. (2010). https://flask.palletsprojects.com/en/stable/

6. CodeGen (CodeGen-Mono 350M). URL: https://huggingface.co/Salesforce/codegen-350M-mono

7. CodeBERT: A Pre-Trained Model for Programming and Natural Languages / Zhangyin Feng, et al. (2020, November 16–20) *Findings of the Association for Computational Linguistics: EMNLP*. P. 1536–1547. URL: https://aclanthology.org/2020.findings-emnlp.139.pdf