

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління
проєктами

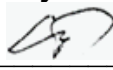
Кафедра програмного забезпечення автоматизованих систем

Спеціальність 124 «Системний аналіз»

Освітня програма «Інформаційні технології інтелектуального аналізу
даних та проєктування програмних систем»

«Допущений до захисту»

Завідувач кафедри

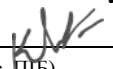
 проф. Приходько С.Б.

«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти «магістр»

на тему: Математична модель щодо прийняття рішень про якість програмних систем, які базуються на штучному інтелекті, та програма для її реалізації

Виконав: студент групи 6153м

 Корнієнко А.М.
(підпис, ПІБ)

Керівник роботи:

доцент, к.ф.-м.н.
(посада, науковий ступень вчене звання)

 Латанська Л.О.
(підпис, ПІБ)

Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра програмного забезпечення автоматизованих систем

Спеціальність 124 «Системний аналіз»

Освітня програма «Інформаційні технології інтелектуального аналізу даних та проектування програмних систем»

«ЗАТВЕРДЖУЮ»



Гарант освітньої програми

доц. Л.О.Латанська

(підпис)

«27» жовтня 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти «магістр»

Студенту Корнієнку Андрію Михайловичу

(Прізвище, ім'я, по батькові)

1. Тема роботи: Математична модель щодо прийняття рішень про якість програмних систем, які базуються на штучному інтелекті, та програма для її реалізації

Керівник роботи к.ф.-м.н., доц. Латанська Л.О.

Затверджені наказом ректора № 1222-УЧ від « 15 » 10 2025 р

2. Термін подання роботи: 08.12.2025 р.

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів)

Вступ (Актуальність теми. Зв'язок роботи з науковими програмами, планами, темами.

Мета і завдання дослідження. Об'єкт дослідження. Предмет дослідження. Методи дослідження. Наукова новизна одержаних результатів. Практичне значення одержаних результатів. Особистий внесок здобувача. Апробація результатів досліджень. Публікації.); Огляд літератури та обґрунтування необхідності проведення досліджень за обраною темою; Викладення результатів власних досліджень з висвітленням того нового, що пропонується; Розділ з розробки проєктних рішень для програмної системи; Організаційно-економічний розділ; Розділи з охорони праці та охорони навколишнього середовища; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма і методика випробувань програмної системи)

5. Перелік презентаційних матеріалів: Тема. Мета та завдання дослідження. Об'єкт та предмет дослідження. Наукова новизна та практичне значення одержаних результатів. Особистий внесок здобувача. Апробація результатів досліджень та публікації. Аналіз існуючих підходів до оцінювання якості ПС. Обґрунтування актуальності досліджень за обраною темою. Нелінійна регресійна модель для прогнозування метрики RFC для ПС, що базуються на ШІ. Математична модель для прийняття рішень про якість ПС, що базуються на ШІ. Постановка задачі на розробку програмної системи для визначення рівня якості ПС, що базуються на ШІ. Загальносистемні рішення. Рішення з інформаційного забезпечення. Рішення з програмного забезпечення. Інтерфейс користувача. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 27.10.2025 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	16.10.2025	ДП*
2.	з 01.09.2025 по 26.10.2025 р.	18.10.2025	ДП*
3.	Підготовка розділу (ів) за результатами власних досліджень	21.10.2025	ДП*
4.	Підготовка розділу з розробки проєктних рішень для програмної системи	27.11.2025	
5.	Підготовка організаційно-економічного розділу	29.11.2025	
6.	Підготовка розділу з охорони праці	02.12.2025	
7.	Підготовка розділу з охорони навколишнього середовища	04.12.2025	
8.	Підготовка розділу ВИСНОВКИ	05.12.2025	
9.	Оформлення списку використаних джерел та додатків	06.12.2025	
10.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версії роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	07.12.2025	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку)
11.	Підготовка презентації та доповіді	08.12.2025	
12.	Попередній захист роботи на засіданні кафедри ПЗАС	15.12.2025	
13.	Подання на кафедру ПЗАС електронних версії наступних документів у форматі pdf: кваліфікаційної роботи; файло-опису кваліфікаційної роботи (згідно з Додатком до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді, а також Авторського договору з додатком	19.12.2025	

* - за результатами дослідницької практики (ДП), яка була з 01.09.2025 по 26.10.2025 р.

Студент


(підпис)

Корнієнко А.М.

(ПІБ)

Керівник роботи


(підпис)

Латанська Л.О.

(ПІБ)

РЕФЕРАТ

Корнієнко Андрій Михайлович

«Математична модель щодо прийняття рішень про якість програмних систем, які базуються на штучному інтелекті, та програма для її реалізації»

Кваліфікаційна робота на здобуття освітнього рівня магістра зі спеціальності 124 – «Системний аналіз». Національний університет кораблебудування імені адмірала Макарова. Миколаїв, 2025 р.

Обсяг роботи: 107 стор., 22 табл., 8 рис., 22 використаних джерела, 6 додатків.

Актуальність теми дослідження. Оцінювання якості програмних систем (ПС), розроблених із застосуванням технологій штучного інтелекту, нині є актуальним і перспективним науковим напрямом. Це пов'язано, з одного боку, із постійною потребою підвищувати ефективність розробки та забезпечувати успішність ПС, а з іншого – зі зростанням ролі інструментів штучного інтелекту у процесі створення ПС.

Мета дослідження. Метою є підвищення достовірності оцінювання якості ПС, які базуються на штучному інтелекті, шляхом розробки спеціалізованої математичної моделі.

Об'єкт дослідження – це процес оцінювання якості ПС, які базуються на штучному інтелекті

Предмет дослідження. Предметом дослідження є математичні моделі для прийняття рішень щодо якості ПС, які базуються на штучному інтелекті, що використовують метрики RFC, CBO, WMC.

Методи дослідження. В якості методів дослідження обрано методи регресійного аналізу, математичної статистики та теорії ймовірностей, математичного програмування.

Наукова новизна одержаних результатів. Удосконалено математичну модель для прийняття рішень щодо якості ПС, які базуються на штучному інтелекті. На відміну від наявних підходів, запропонована модель враховує

інтервали значень метрик таких ПС за рахунок визначення нових параметрів моделі і застосування нормалізуючого перетворення десятковий логарифм для прогнозування метрики RFC на основі метрик CBO та WMC, що забезпечує підвищення достовірності оцінювання якості ПС відповідного класу.

Практичне значення одержаних результатів. Розроблено програмну систему для оцінювання метрики RFC та прийняття рішення про якість ПС, які базуються на штучному інтелекті.

Апробація результатів досліджень. Результати роботи оприлюднено на VIII Всеукраїнській науково-практичній інтернет конференції студентів, аспірантів та молодих вчених «СУЧАСНІ ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ» (24 листопада 2025 р., м. Херсон, м. Хмельницький).

Публікації. За результатами роботи опубліковано 1 наукову працю – тези VIII Всеукраїнської науково-практичної інтернет конференції студентів, аспірантів та молодих вчених.

Ключові слова: оцінювання метрики RFC, якість ПС, штучний інтелект, нелінійні регресійні моделі, нормалізуюче перетворення десятковий логарифм.

ABSTRACT

Andrii Korniienko

«Mathematical model for decision-making about the quality of software systems based on artificial intelligence and a software for its implementation»

Qualification work for obtaining a master's degree in specialty 124 – «Systems Analysis». Admiral Makarov National University of Shipbuilding. Mykolaiv, 2025.

Volume: 107 p., 22 tables, 8 figures, 22 references, 6 appendices.

Relevance of the topic. The quality assessment of software systems (SS) developed using artificial intelligence technologies is currently a relevant and promising scientific direction. This is due, on the one hand, to the constant need to increase the efficiency of development and ensure the success of SS, and, on the other hand, to the growing role of artificial intelligence tools in the process of creating PS.

Purpose of the research. The goal is to increase the reliability of assessing the quality of artificial intelligence-based SS by developing a specialized mathematical model.

Object of the research. The object of the study is the process of assessing the quality of artificial intelligence-based SS.

Subject of the research. The subject of the research is mathematical models for making decisions about the quality of SS based on artificial intelligence, using RFC, CBO, WMC metrics.

Methods of the research. The methods chosen for the research were regression analysis, mathematical statistics and probability theory, and mathematical programming.

Scientific novelty of the obtained results. The mathematical model for making decisions on the quality of SS based on artificial intelligence has been improved. Unlike existing approaches, the proposed model takes into account the intervals of the values of the metrics of such SS by determining new model

parameters and applying the normalizing transformation of the decimal logarithm to predict the RFC metric based on the CBO and WMC metrics, which ensures increased reliability of assessing the quality of SS of the corresponding class.

Practical significance of the obtained results. A software system has been developed for evaluating RFC metrics and making decisions about the quality of SS based on artificial intelligence.

Approbation of research results. The results of the work were presented at the VIII All-Ukrainian Scientific and Practical Internet Conference of Students, Postgraduate Students and Young Scientists «MODERN INFORMATION SYSTEMS AND TECHNOLOGIES» (November 24, 2025, Kherson, Khmelnytskyi).

Publications. Based on the results of the work, 1 scientific work was published – theses of the VIII All-Ukrainian Scientific and Practical Internet Conference of Students, Postgraduate Students and Young Scientists.

Keywords: RFC metric evaluation, SS quality, artificial intelligence, nonlinear regression models, normalizing transformation decimal logarithm.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ОЦІНЮВАННЯ ЯКОСТІ ПС ТА ОБГРУНТУВАННЯ НЕОБХІДНОСТІ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ.....	12
1.1 Аналіз існуючих підходів до оцінювання якості ПС.....	12
1.2 Обґрунтування необхідності проведення досліджень за обраною темою.....	15
1.3 Висновки за розділом 1	19
2 МАТЕМАТИЧНА МОДЕЛЬ ЩОДО ПРИЙНЯТТЯ РІШЕНЬ ПРО ЯКІСТЬ ПС, ЯКІ БАЗУЮТЬСЯ НА ШТУЧНОМУ ІНТЕЛЕКТІ.....	21
2.1 Методика побудови нелінійної регресійної моделі для прогнозування метрики RFC в залежності від метрик СВО та WMC для ПС, які базуються на штучному інтелекті	21
2.2 Метрики якості регресійних моделей	23
2.3 Побудова нелінійної регресійної моделі для прогнозування метрики RFC в залежності від метрик СВО та WMC для ПС, які базуються на штучному інтелекті	25
2.4 Прийняття рішень щодо якості ПС, які базуються на штучному інтелекті.....	27
2.5 Висновки до розділу 2	28
3 РОЗРОБКА ПРОЄКТНИХ РІШЕНЬ ДЛЯ ПРОГРАМНОЇ СИСТЕМИ ВИЗНАЧЕННЯ РІВНЯ ЯКОСТІ ПС, ЯКІ БАЗУЮТЬСЯ НА ШТУЧНОМУ ІНТЕЛЕКТІ	29
3.1 Постановка задачі на розробку програмної системи визначення рівня якості ПС, які базуються на штучному інтелекті	29
3.2 Загальносистемні рішення для програмної системи визначення рівня якості ПС, які базуються на штучному інтелекті.....	30
3.2.1 Вибір засобів моделювання програмної системи визначення рівня якості ПС.....	30
3.2.2 Розробка діаграми варіантів використання програмної системи визначення рівня якості ПС.....	32
3.2.3 Специфікації варіантів використання програмної системи визначення рівня якості ПС	36
3.2.4 Створення діаграм діяльності програмної системи визначення рівня якості ПС.....	41
3.3 Рішення з інформаційного забезпечення програмної системи визначення рівня якості ПС, які базуються на штучному інтелекті.....	43

3.4 Рішення з програмного забезпечення програмної системи визначення рівня якості ПС, які базуються на штучному інтелекті	44
3.4.1 Засоби розробки програмної системи визначення рівня якості ПС.....	44
3.4.2 Модель класів програмної системи визначення рівня якості ПС	45
3.4.3 Моделі взаємодії програмної системи визначення рівня якості ПС.....	48
3.4.4 Тестування програмної системи визначення рівня якості ПС.	49
3.4.5 Випробування програмної системи визначення рівня якості ПС.....	53
3.5 Висновки до розділу 3	56
4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОЇ СИСТЕМИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ЯКОСТІ ПС, ЯКІ БАЗУЮТЬСЯ НА ШТУЧНОМУ ІНТЕЛЕКТІ	57
4.1 Вступ	57
4.2 Розрахунок витрат розробки і впровадження програмної системи визначення рівня якості ПС, які базуються на штучному інтелекті.....	58
4.3 Економічна ефективність розробки і впровадження програмної системи визначення рівня якості ПС, які базуються на штучному інтелекті	61
4.4 Висновки до розділу 4	62
5 ОХОРОНА ПРАЦІ	63
5.1 Загальні положення та актуальність забезпечення безпеки праці	63
5.2 Характеристика основних небезпечних і шкідливих чинників при роботі із засобами обчислювальної техніки	63
5.3 Заходи щодо зменшення впливу небезпечних та шкідливих факторів.....	65
5.4 Висновки до розділу 5.....	66
6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	68
6.1. Основні положення охорони навколишнього природного середовища	68
6.2 Забруднення навколишнього середовища комп'ютерною технікою та заходи щодо його зменшення.....	69
6.3 Висновки до розділу 6.....	74
ВИСНОВКИ	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	77
ДОДАТОК А – Технічне завдання	80
ДОДАТОК Б – Інструкція користувача	86
ДОДАТОК В – Програма та методика випробувань	92
ДОДАТОК Г – Текст програми	96
ДОДАТОК Д – Опис програмної системи.....	102
ДОДАТОК Е– Повний набір вихідних даних.....	106

ВСТУП

Актуальність теми дослідження. Оцінювання якості ПС, розроблених із застосуванням технологій штучного інтелекту, нині є одним із найбільш актуальних і перспективних напрямів наукових досліджень в галузі інформаційних технологій. Така увага зумовлена комплексом факторів. З одного боку, розробники відчують постійну потребу у підвищенні ефективності процесів розробки ПС, скороченні витрат часу та ресурсів, а також у забезпеченні стабільної результативності програмних проєктів. З іншого боку, стрімке поширення та розвиток інструментів штучного інтелекту істотно змінює підхід до побудови ПС.

У таких умовах традиційні методи та метрики оцінювання якості ПС потребують адаптації або переосмислення, оскільки ПС, які базуються на штучному інтелекті, демонструють іншу поведінку та варіативність результатів. Тому дослідження підходів до оцінювання їхньої якості набуває особливого значення. Воно дозволяє не лише забезпечити високу надійність і точність роботи таких систем, а й мінімізувати ризики, пов'язані з їх використанням.

Таким чином, проблематика оцінювання якості ПС, створених із використанням ШІ, є надзвичайно актуальною та потребує подальшого наукового опрацювання як з точки зору методології, так і з позиції практичної реалізації.

Для оцінювання якості ПС та процесу їх створення розроблено різні підходи, методи та моделі [1-18]. Одним із найпоширеніших підходів до аналізу якості об'єктно-орієнтованих ПС є застосування метрик Chidamber and Kemerer (СК Metrics) [1]. Цей набір метрик був одним із перших формалізованих інструментів для оцінювання якості дизайну та складності ПС. Ряд робіт показує, що метрики CBO, WMC та RFC із набору СК є одними з найбільш значущих факторів при оцінюванні якості коду та при виявленні проблемних компонентів системи. Водночас доведено, що деякі метрики СК

не є повністю незалежними: наприклад, метрика RFC часто демонструє кореляцію з CBO та WMC, що потребує відповідного статистичного врахування при побудові моделей оцінювання [1, 2]. У науковій літературі вже запропоновано лінійні та нелінійні моделі регресії, що враховують вказану залежність [3-9].

У випадку систем, що розроблені з використанням штучного інтелекту, така залежність часто ускладнюється специфікою архітектури, використанням зовнішніх бібліотек, а також варіативністю поведінки ШІ-компонент. Проте для вказаного типу ПС подібні моделі наразі практично не сформовані.

У зв'язку з цим побудова регресійної моделі для оцінювання метрики RFC ПС, розроблених із використанням штучного інтелекту, з урахуванням значень метрик CBO та WMC, для визначення їх якості є актуальною науковою задачею. Вона забезпечує можливість більш коректного аналізу якості ПС, що містять інтелектуальні компоненти, та сприяє підвищенню надійності їх архітектури й програмної реалізації.

Мета та завдання дослідження. Метою є підвищення достовірності оцінювання якості ПС, які базуються на штучному інтелекті, шляхом розробки спеціалізованої математичної моделі.

Для досягнення мети необхідно виконати наступні задачі: аналіз існуючих підходів до оцінювання якості ПС; збір даних для побудови регресійної моделі оцінювання метрики RFC через метрики CBO та WMC ПС, які базуються на штучному інтелекті; перевірка зібраних даних на нормальність розподілу; удосконалення нелінійної регресійної моделі для оцінювання метрики RFC для вказаних ПС з метою визначення їх якості; розробка програмної системи для прийняття рішень про якість ПС, які базуються на штучному інтелекті, що реалізує отриману модель.

Об'єкт дослідження – це процес оцінювання якості ПС, які базуються на штучному інтелекті.

Предмет дослідження. Предметом дослідження є математичні моделі для прийняття рішень щодо якості ПС, які базуються на штучному інтелекті, що використовують метрики RFC, CBO, WMC.

Методи дослідження. В якості методів дослідження обрано методи регресійного аналізу, математичної статистики та теорії ймовірностей, математичного програмування.

Наукова новизна одержаних результатів. Удосконалено математичну модель для прийняття рішень щодо якості ПС, які базуються на штучному інтелекті. На відміну від наявних підходів, запропонована модель враховує інтервали значень метрик таких ПС за рахунок визначення нових параметрів моделі і застосування нормалізуючого перетворення десятковий логарифм для прогнозування метрики RFC на основі метрик CBO та WMC, що забезпечує підвищення достовірності оцінювання якості ПС відповідного класу.

Практичне значення одержаних результатів. Розроблено програмну систему для оцінювання метрики RFC та прийняття рішення про якість ПС, які базуються на штучному інтелекті.

Особистий внесок здобувача. У публікації [9], що написана у співавторстві, автору належить нелінійна регресійна модель для оцінювання метрики RFC в залежності від метрик CBO та WMC для прийняття рішень про якість ПС, які базуються на штучному інтелекті.

Апробація результатів досліджень. Результати роботи оприлюднено на VIII Всеукраїнській науково-практичній інтернет конференції студентів, аспірантів та молодих вчених «СУЧАСНІ ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ» (24 листопада 2025 р., м. Херсон, м. Хмельницький).

Публікації. За результатами роботи опубліковано 1 наукову працю – тези VIII Всеукраїнської науково-практичної інтернет конференції студентів, аспірантів та молодих вчених.

Ключові слова: оцінювання метрики RFC, якість ПС, штучний інтелект, нелінійні регресійні моделі, нормалізуюче перетворення десятковий логарифм.

1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ОЦІНЮВАННЯ ЯКОСТІ ПС ТА ОБГРУНТУВАННЯ НЕОБХІДНОСТІ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

1.1 Аналіз існуючих підходів до оцінювання якості ПС

Оцінювання якості ПС є одним із ключових етапів у процесі їх розроблення та супроводу, оскільки якість безпосередньо визначає надійність, зручність використання, продуктивність і довговічність створюваних систем. У сучасних умовах розвитку інформаційних технологій сформувалася велика кількість підходів, спрямованих на вимірювання та аналіз якості ПС. Кожен із цих підходів розглядає якість з різних точок зору: від структурних характеристик коду до поведінкових властивостей та відповідності вимогам користувачів. Основними підходами до оцінювання якості ПС є:

- Міжнародні стандарти оцінювання якості ПС

Основою для оцінювання якості ПС є сімейство стандартів SQuaRE (ISO/IEC 25000 – 25099), де ключову роль відіграють: ISO/IEC 25010 – модель якості, ISO/IEC 25002 – застосування моделі якості, ISO/IEC 25023 – метрики, ISO/IEC 25040–25041 – процеси оцінювання. Стандарти орієнтовані на сучасні програмні продукти – у тому числі системи, які базуються на штучному інтелекті [10].

- Метрики об'єктно-орієнтованого програмування

Для оцінювання внутрішніх характеристик ПС широко застосовуються метрики об'єктно-орієнтованого дизайну. Найвідомішим є набір метрик Chidamber & Kemerer (СК), який включає [1]:

- WMC (Weighted Methods per Class) – кількість і складність методів класу;
- CBO (Coupling Between Objects) – рівень зв'язності між класами;

- RFC (Response For a Class) – кількість методів, які можуть бути викликані у відповідь на повідомлення;

- LCOM (Lack of Cohesion of Methods) – оцінка зв'язності всередині класу;

- NOC (Number of Children) та DIT (Depth of Inheritance Tree) – метрики, що характеризують наслідування.

Метрики СК дають змогу кількісно оцінювати складність, зв'язність, згуртованість та інші властивості коду, що безпосередньо впливають на якість, супроводжуваність та модульність ПС. Дослідження показують, що не всі метрики є незалежними, зокрема RFC залежить від CBO та WMC [2, 3]. СК метрики широко застосовуються для аналізу систем як класичного програмування, так і сучасних систем зі штучним інтелектом, де високий рівень складності модулів потребує кількісного вимірювання.

– Підходи на основі моделей прогнозування якості

Окрім традиційних метрик, активно застосовуються методи математичного моделювання та прогнозування:

- Регресійні моделі [3-9]

Застосовуються для встановлення залежностей між різними метриками. Наприклад, модель залежності RFC від CBO та WMC дозволяє прогнозувати поведінкові характеристики компонента за структурними показниками.

- Машинне навчання [11-15]

Для прогнозування дефектності та якості коду використовують:

- дерева рішень,
- випадкові ліси,
- нейронні мережі,
- метод опорних векторів.

Підхід є особливо актуальним для складних ПС, зокрема ШІ-систем, де звичайні статистичні моделі часто не відображають нелінійні залежності між характеристиками систем.

- Баєсівські моделі [16]

Застосовуються для оцінювання ризиків дефектності, надійності та невизначеності у поведінці ПС.

- Підходи на основі моделювання процесів

Важливими є також моделі, які оцінюють якість процесу розроблення ПС:

- CMMI (Capability Maturity Model Integration)

Оцінює зрілість процесів організації та здатність стабільно створювати якісну ПС.

- ISO/IEC 33000

Регламентує процеси розроблення та оцінює їх якість, що впливає на кінцеву якість системи.

Ці моделі корисні для організаційного рівня оцінювання, але не дають кількісних показників для конкретних програмних модулів.

Аналіз існуючих підходів показує, що вони охоплюють різні рівні:

- процесний рівень (CMMI, ISO/IEC 33000),
- структурний рівень (метрики СК, інші ООП-метрики),
- модельно-орієнтований рівень (метрики, моделі).

Незважаючи на це, існують проблеми:

- відсутність універсального підходу для комплексного оцінювання якості;
- нелінійний взаємозв'язок між окремими метриками та характеристиками;
- специфічність систем зі штучним інтелектом;
- проблема інтерпретованості складних моделей прогнозування.

Проведений аналіз показує, що сучасні підходи до оцінювання якості ПС різноманітні й орієнтовані на різні аспекти якості – від структурних характеристик до моделей машинного навчання. Це створює передумови для розроблення нових моделей та удосконалення існуючих підходів оцінювання якості ПС, особливо тих, що містять компоненти штучного інтелекту.

1.2 Обґрунтування необхідності проведення досліджень за обраною темою

Як уже було зазначено, для прийняття рішень щодо якості ПС, серед інших, використовуються регресійні моделі, які дають можливість встановити залежність між різними метриками. Також було підкреслено про використання СК метрик для оцінювання якості ПС та про залежність метрики RFC від CBO та WMC.

Ряд сучасних досліджень пов'язаний з розробкою регресійних моделей для оцінювання якості ПС. У роботі [3] було запропоновано модель лінійної регресії для оцінювання середнього значення RFC на основі метрик CBO та WMC. Незважаючи на широке застосування лінійної регресії, її використання не завжди має теоретичне підґрунтя, зокрема у випадках, коли залишки моделі відхиляються від нормального розподілу, що може впливати на достовірність оцінок і точність прогнозів. У роботах [4-9] побудовано нелінійні регресійні моделі для оцінювання середнього RFC в залежності від CBO та WMC для програм з відкритим кодом на PHP, на Java, на Kotlin, для застосунків для онлайн-покупок з відкритим кодом на Java, для застосунків, що створюються на платформі Flutter.

Для оцінювання метрики RFC ПС, які базуються на штучному інтелекті, було прийнято рішення побудувати власну нелінійну регресійну модель з наступних причин:

- зазвичай дані, зібрані з метрик RFC, CBO та WMC, не мають нормального розподілу (характер зв'язку між даними нелінійний);
- існує ряд сучасних публікацій, присвячених розробці нелінійних регресійних моделей оцінювання метрики RFC для інших типів ПС [4-9].
- зазвичай існують відмінності в діапазонах значень метрик RFC, CBO та WMC для різного типу застосунків;
- відомо, що якість регресійних моделей залежить від типу застосунків, на якому модель була розроблена та випробувана.

Для аргументування ряду сформульованих тверджень, проаналізуємо зібрані для побудови моделі дані. Емпіричні дані зібрано з 47 проєктів, які базуються на штучному інтелекті та розміщені на сайті GitHub (<https://github.com>), за допомогою власного ПЗ. Фрагмент вихідних даних разом з обчисленим квадратом відстані Махаланобіса (SMD) наведено в таблиці 1.1.

Таблиця 1.1 – Фрагмент вихідних даних разом з обчисленим квадратом відстані Махаланобіса (SMD)

№	Назва проєкту	CBO (X_1)	WMC (X_2)	RFC (Y)	SMD (D^2)
1	transformers	4,98	9,22	14,11	1,5104
2	pytorch	4,27	11,64	13,89	0,4185
3	d2l-zh	1,42	4,25	10,27	3,0399
4	scikit-learn	2,43	14,72	19,4	2,6143
5	keras	2,38	10,31	17,68	2,1896
6	annotated_deep_learning_paper_implementations	3,16	4,63	11,97	3,6026
7	yolov5	2,92	11,73	18,8	1,6684
8	faceswap	4,66	18,2	24,99	3,3527
9	OpenBB	2,54	5,66	6,89	1,6748
10	ultralytics	4,7	15,39	22,89	2,3135
11	airflow	4,76	9,14	12	1,1926
12	streamlit	4,35	8,35	11,6	0,9528
13	DeepSpeed	3,81	16,98	18,82	2,7758
14	gradio	4,44	12,4	11,2	3,1090
15	ray	5,69	11,82	14,2	1,9093
16	Open-Assistant	1,91	5,11	6,35	2,0338
17	supervision	5,59	15,94	18,47	1,8690
18	paperless-ngx	2,3	4,21	5,25	2,4699
19	spaCy	2,84	8,16	7,41	2,3925
20	qlib	4,58	10,9	17,28	1,1190

Повний набір вихідних даних наведено в Додатку Е.

Описова статистика набору даних наведена в таблиці 1.2.

Таблиця 1.2 – Описова статистика набору даних

Метрика	Мінімальне значення	Максимальне значення	Середнє значення	СКВ
RFC	1,71	29,14	14,46	5,93
CBO	0,29	10,60	3,61	1,79
WMC	1,16	23,64	10,73	4,81

Для кожної i -ї ($i = \overline{1, N}$) багатовимірної точки вихідних даних розраховується SMD згідно формули [19]:

$$d_i^2 = (X_i - \bar{X})^T S_N^{-1} (X_i - \bar{X}),$$

де

X_i – i -ий елемент вибірки ($X_i = \{X_{1i}, X_{2i}, \dots, X_{mi}\}^T$, де m – кількість компонент багатовимірної точки),

N – кількість елементів у вибірці,

$\bar{X}$ – вектор середніх значень компонент вибірки,

S_N – коваріаційна матриця вибірки:

$$S_N = \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X}) (X_i - \bar{X})^T.$$

Спочатку вихідні дані перевіряємо на нормальність. Дані не мають нормального розподілу тому, що для 31 та 39 рядків даних d_i^2 перевищує квантіль розподілу χ^2 для рівня значущості 0,005, який дорівнює 12,84.

Для побудови нелінійної моделі використаємо метод нормалізуючих перетворень з десятковим логарифмом у якості нормалізуючого перетворення.

Таким чином, проаналізували зібрані емпіричні дані дійшли до висновку, що для відображення закономірностей зв'язку між ними необхідно будувати нелінійну регресійну модель. В якості методу побудови нелінійної моделі обрали метод нормалізуючих перетворень.

Для підтвердження необхідності побудови власної моделі, проаналізуємо діапазони значень метрик існуючих моделей оцінювання середнього RFC в залежності від CBO та WMC для різного типу застосунків: для програм з відкритим кодом на Java [5], на Kotlin [6], для застосунків, що створюються на платформі Flutter [8], для застосунків для онлайн-покупок з відкритим кодом на Java [7] та для ПС, які базуються на штучному інтелекті (таблиця 1.3).

Таблиця 1.3 – Діапазони значень метрик RFC, CBO та WMC для різного типу застосунків

	CBO	WMC	RFC
Java	[5,071; 22,417]	[8,042; 71,837]	[7,306; 37,278]
Kotlin	[0,995; 4,724]	[2,167; 13,020]	[2,118; 9,487]
Flutter	[0,125; 1,46]	[2,5; 13,66]	[0,28; 9,92]
Online shopping applications in Java	[0,4; 6,39]	[1,6; 20,85]	[1,6; 13,95]
III	[1,13; 10,60]	[3,89; 23,64]	[5,10; 29,14]

Проведений аналіз засвідчив, що наявні моделі не придатні для застосування для частини рядків даних ПС, які базуються на штучному інтелекті. Основною причиною є невідповідність діапазонів значень метрик. Це обумовлює потребу в розробленні нової моделі для оцінювання метрики RFC ПС, які базуються на штучному інтелекті.

Таким чином проведення дослідження за обраною темою є актуальним.

1.3 Висновки за розділом 1

У цьому розділі представлено аналіз існуючих підходів до оцінювання якості ПС, які базуються на штучному інтелекті, а також наведено обґрунтування необхідності проведення дослідження за обраною темою.

У розділі:

- Підкреслено, що сучасні підходи до оцінювання якості ПС різноманітні й орієнтовані на різні аспекти якості – від структурних характеристик до моделей машинного навчання. Це створює передумови для розроблення нових моделей та удосконалення існуючих підходів оцінювання якості ПС, особливо тих, що містять компоненти штучного інтелекту.

- Визначено, що при прийнятті рішень про якість ПС використовуються, серед інших, метрики RFC, CBO та WMC.

- Відомо, що метрика RFC залежить від метрик CBO та WMC.

- Визначено, що при оцінюванні якості ПС з використанням метрик RFC, CBO та WMC застосовуються, в тому числі, лінійні та нелінійні регресійні моделі. Регресійні моделі розроблено для програм з відкритим кодом на PHP, на Java, на Kotlin та інших, проте для ПС, що містять компоненти штучного інтелекту, таких моделей не існує.

- Відомо, що для обґрунтування правомірності використання лінійної регресійної теорії необхідне виконання певних умов, серед яких нормальний розподіл залишків, що справедливо лише в окремих випадках. Тому зазвичай будують нелінійні регресійні моделі. Для побудови моделей нелінійної регресії використовують, серед інших, і метод нормалізуючих перетворень.

- Відомо, що якість регресійних моделей залежить від типу ПС, на яких модель була побудована й навчена.

- Визначено, що наявні моделі не придатні для застосування до частини даних ПС, які базуються на штучному інтелекті. Основною причиною є невідповідність діапазонів значень метрик

– Відомо, що стрімке поширення та розвиток інструментів штучного інтелекту істотно змінює підхід до побудови ПС. У таких умовах існуючі методи та моделі оцінювання якості ПС потребують адаптації або переосмислення, оскільки програмні системи з компонентами штучного інтелекту демонструють іншу поведінку та варіативність результатів. Тому проблематика оцінювання якості ПС, створених із використанням ШІ, є актуальною та потребує подальшого наукового опрацювання як з точки зору методології, так і з позиції практичної реалізації.

– Прийнято рішення побудувати математичної моделі для оцінювання метрики RFC через метрики CBO та WMC ПС, створених із використанням ШІ, здійснювати у вигляді нелінійної регресійної моделі з використанням нормалізуючих перетворень. Ця модель разом з інтервалами прогнозування слугуватиме інструментом для прийняття рішень про якість вказаних ПС.

2 МАТЕМАТИЧНА МОДЕЛЬ ЩОДО ПРИЙНЯТТЯ РІШЕНЬ ПРО ЯКІСТЬ ПС, ЯКІ БАЗУЮТЬСЯ НА ШТУЧНОМУ ІНТЕЛЕКТІ

2.1 Методика побудови нелінійної регресійної моделі для прогнозування метрики RFC в залежності від метрик CBO та WMC для ПС, які базуються на штучному інтелекті

Для побудови нелінійної регресійної моделі для прогнозування метрики RFC в залежності від метрик CBO та WMC для ПС, які базуються на штучному інтелекті, будемо використовувати методику, основними кроками якої є:

1) Нормалізація вихідних даних.

У якості нормалізуючого перетворення в даній роботі використаємо десятковий логарифм.

Позначимо нормалізуюче перетворення вихідних величин Y, X_1, X_2 у нормалізовані дані Z_Y, Z_1, Z_2 через ψ . Отримаємо [19]

$$T = \psi(P), \quad (2.1)$$

де $P = \{Y, X_1, X_2\}^T$ – вихідні дані,

$T = \{Z_Y, Z_1, Z_2\}^T$ – нормалізовані дані,

$\psi = \{\psi_Y, \psi_1, \psi_2\}^T$ – нормалізуюче перетворення.

2) Перевірка тривимірних нормалізованих даних на наявність викидів з використанням SMD та квантілю розподілу $\chi^2_{m,\alpha}$ ($m=3$; $\alpha=0,005$). Якщо викиди знайдено, то вилучаємо максимальний викид і повертаємося на крок 1, інакше переходимо на крок 3.

3) Побудова лінійної регресійної моделі залежності нормалізованого RFC (Z_Y) від нормалізованих CBO (Z_1) та WMC(Z_2) [19]:

$$Z_Y = \hat{Z}_Y + \varepsilon = b_0 + b_1 Z_1 + b_2 Z_2 + \varepsilon, \quad (2.2)$$

де

ε – залишки регресії,

$\hat{Z}_Y$ – розраховане за регресією значення Z_Y .

Параметри моделі визначаємо методом найменших квадратів.

4) Перевірка залишків регресії на нормальність розподілу. У разі виявлення відхилення від нормальності, значення з найбільшим за модулем залишком вилючається з вибірки, після чого процес повертається до кроку 1, інакше переходимо на крок 5.

5) Побудова довірчих та прогнозних інтервалів лінійної регресії.

Формула для розрахунку довірчого інтервалу має вигляд [19]:

$$\hat{Z}_Y \pm t_{\frac{\alpha}{2}, v} S_{Z_Y} \left\{ \frac{1}{N} + (z_X^+)^T [(Z_X^+)^T Z_X^+]^{-1} (z_X^+) \right\}^{1/2}, \quad (2.3)$$

а прогнозний інтервал розраховується за формулою [19]:

$$\hat{Z}_Y \pm t_{\frac{\alpha}{2}, v} S_{Z_Y} \left\{ 1 + \frac{1}{N} + (z_X^+)^T [(Z_X^+)^T Z_X^+]^{-1} (z_X^+) \right\}^{1/2}, \quad (2.4)$$

де

$$S_{Z_Y}^2 = \frac{1}{v} \sum_{i=1}^N (Z_{Y_i} - \hat{Z}_{Y_i})^2,$$

v – кількість ступенів вільності ($v = N - 3$),

$t_{\frac{\alpha}{2}, v}$ – квантиль розподілу Стюдента з рівнем значущості $\frac{\alpha}{2}$ та v ступенями

вільності,

$(Z_X^+)^T Z_X^+ - 2 \times 2$ матриця

$$(Z_X^+)^T Z_X^+ = \begin{pmatrix} S_{Z_1 Z_1} & S_{Z_1 Z_2} \\ S_{Z_1 Z_2} & S_{Z_2 Z_2} \end{pmatrix},$$

де $S_{Z_q Z_r} = \sum_{i=1}^N [Z_{qi} - \bar{Z}_q][Z_{ri} - \bar{Z}_r]$, $q, r = \overline{1, 2}$.

6) Застосування зворотного до нормалізуючого перетворення для переходу від лінійної до нелінійної регресії [19]:

$$P = \psi^{-1}(T). \quad (2.5)$$

7) Отримання інтервалів довіри та прогнозу нелінійної регресії, через зворотне перетворення.

8) Перевірка належності фактичних значень RFC до прогнозного інтервалу побудованої нелінійної регресії. У випадку, коли окремі значення RFC не потрапляють у межі прогнозного інтервалу, вони класифікуються як викиди; такі рядки даних вилучаються з вихідного набору і повертаємося на крок 1. Інакше модель побудована.

2.2 Метрики якості регресійних моделей

Для оцінювання якості побудованої регресійної моделі традиційно застосовують такі показники: коефіцієнт детермінації R^2 , середню величину відносної похибки $MMRE$ та рівень прогнозування $PRED(l)$ [19].

Значення коефіцієнта детермінації R^2 визначається за формулою [19]:

$$R^2 = 1 - \left(\sum_{i=1}^N (Y_i - \hat{Y}_i)^2 / \sum_{i=1}^N (Y_i - \bar{Y})^2 \right), \quad (2.6)$$

де Y_i – вихідні значення Y ;

$\hat{Y}_i$ – розраховані за регресією значення Y ;

$\bar{Y} = \frac{1}{N} \sum_{i=1}^N Y_i$ – середнє значення Y .

Коефіцієнт детермінації R^2 відображає частку варіації результативної ознаки Y , яка пояснюється включеними до моделі факторами. Його значення лежать у межах від 0 до 1. Модель вважається прийнятною за умови, що R^2 має значення не нижче 0,5; при $R^2 \geq 0,75$ модель характеризується достатньою точністю та ефективністю. Значення $R^2 = 1$ свідчить про повну адекватність та достовірність моделі.

Для обчислення середньої величини відносної похибки $MMRE$ та рівня прогнозування $PRED(l)$ використовується величина відносної похибки MRE_i , яка характеризує відносне відхилення прогнозованих значень від фактичних спостережень. Цей показник дозволяє оцінити точність моделі та якість її прогнозів у відсотковому або безрозмірному вигляді і розраховується наступним чином [19]:

$$MRE_i = \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right|.$$

Тоді середня величина відносної похибки $MMRE$ обчислюється за формулою [19]:

$$MMRE = \frac{1}{N} \sum_{i=1}^N MRE_i, \quad (2.7)$$

а показник рівня прогнозування $PRED(l)$ визначається за такою залежністю [19]:

$$PRED(l) = \frac{k}{N}, \quad (2.8)$$

де k – кількість тих MRE_i , які не перевищують l .

Прийнятними вважаються значення: $MMRE \leq 0,25$; $PRED(l) \geq 0,75$ (при $l = 0,25$).

2.3 Побудова нелінійної регресійної моделі для прогнозування метрики RFC в залежності від метрик CBO та WMC для ПС, які базуються на штучному інтелекті

Побудову нелінійної регресійної моделі для прогнозування метрики RFC в залежності від метрик CBO та WMC для ПС, які базуються на штучному інтелекті, будемо проводити згідно методики, наведеної в 2.1. Фрагмент вихідних даних для побудови моделі представлений в таблиці 1.1, а описова статистика набору даних – в таблиці 1.2. Згідно аналізу, наведеному в підрозділі 1.2, дані мають негаусівський розподіл. Тому було виконано нормалізацію вихідних даних з використанням нормалізуючого перетворення десятковий логарифм. При перевірці нормалізованих даних на наявність багатовимірних викидів з допомогою SMD та квантілю розподілу $\chi^2_{m,\alpha}$ ($m=3$; $\alpha = 0,005$) було знайдено та вилучено один викид з номером рядка даних 47. На наступній ітерації викидів не знайдено, тобто всі значення SMD не перевищують $\chi^2_{3;0,005} = 12,84$.

На наступному кроці будуємо лінійну регресійну модель згідно (2.2). Параметри моделі b_0 , b_1 , b_2 розраховані методом найменших квадратів і відповідно дорівнюють -0,00024; -0,04205; 1,15118. Залишки регресії не мають нормального розподілу. Вилучаємо рядок 6 з найбільшим по модулю залишком. Залишилося 45 рядків даних. Повертаємося на крок 1.

Викиди по Махаланобісу відсутні, залишки лінійної регресії нормально розподілені. Будуємо довірчі та прогнозні інтервали лінійної регресії. Переходимо до нелінійної регресії та її інтервалів та перевіряємо належність фактичних значень RFC до прогнозного інтервалу нелінійної регресії. По інтервалу вилучаємо 3 та 25 рядок даних. Залишилося 43 рядки даних. Повертаємося до кроку 1.

Викиди по Махаланобісу відсутні, залишки лінійної регресії нормально розподілені. Будуємо довірчі та прогнозні інтервали лінійної регресії. Переходимо до нелінійної регресії та її інтервалів та перевіряємо належність

фактичних значень RFC до прогнозного інтервалу. По інтервалу вилучаємо 33 рядок даних. Залишилося 42 рядки даних. Повертаємося до кроку 1.

Викиди по Махаланобісу відсутні, залишки лінійної регресії нормально розподілені. Будуємо довірчі та прогнозні інтервали лінійної регресії. Переходимо до нелінійної регресії та її інтервалів та перевіряємо належність фактичних значень RFC до прогнозного інтервалу. Усі фактичні значення RFC належать до прогнозного інтервалу нелінійної регресії.

Модель побудовано. Параметри моделі b_0, b_1, b_2 дорівнюють відповідно -0,00016; -0,01632; 1,11990. Остаточна лінійна регресійна модель згідно (2.2) має вигляд:

$$Z_Y = \hat{Z}_Y + \varepsilon = 0,00016 - 0,01632Z_1 + 1,11990Z_2 + \varepsilon. \quad (2.9)$$

Довірчі та прогнозні інтервали визначаються згідно (2.3) та (2.4). Матриця для розрахунку інтервалів дорівнює:

$$\begin{pmatrix} 1,61419 & 0,70776 \\ 0,70776 & 1,33113 \end{pmatrix}.$$

Нелінійна регресійна модель, використовуючи зворотнє перетворення, має вигляд:

$$Y = 10^{\varepsilon+0,00016} X_1^{-0,01632} X_2^{1,11990}. \quad (2.10)$$

Довірчі та прогнозні інтервали нелінійної регресії розраховуються аналогічно через зворотні перетворення (2.3), (2.4).

При оцінюванні якості побудованої нелінійної регресійної моделі для прийняття рішень про якість ПС, які базуються на штучному інтелекті, обчислено R^2 , $MMRE$ та $PRED(0,25)$ згідно (2.6) – (2.8). Отримали відповідно 0,64457; 0,16160; 0,78571. Таким чином R^2 має прийнятне значення, а $MMRE$ та $PRED(0,25)$ – задовільні. Для покращення якості моделі можна спробувати інші нормалізуючі перетворення.

2.4 Прийняття рішень щодо якості ПС, які базуються на штучному інтелекті

Математична модель щодо прийняття рішень про якість ПС, розроблених з використанням штучного інтелекту, базується на інтервалах довіри та прогнозу нелінійної регресії для метрики RFC з використанням метрик CBO та WMC на рівні застосунку [5]. Автори пропонують методику визначення рівня якості ПС, яка ґрунтується на тому, де розташоване фактичне значення метрики RFC відносно:

- довірчого інтервалу нелінійної регресії,
- прогнозного інтервалу нелінійної регресії.

Запропонований в [5] підхід дає змогу класифікувати якість ПС як високу, середню або низьку залежно від положення значення RFC відносно меж довірчого та прогнозного інтервалів:

- якщо значення RFC потрапляє всередину довірчого інтервалу, це свідчить про середній рівень якості ПС.
- розташування RFC між довірчим і прогнозними інтервалами вказує на високу якість.
- перевищення верхньої межі прогнозного інтервалу може сигналізувати про низьку якість програмної системи.

Цей підхід дає змогу отримати не лише точкове значення метрики RFC, а також оцінити діапазон її можливих варіацій із заданою ймовірністю, що підвищує достовірність отриманих результатів.

Така методика дозволяє:

- виявляти застосунки з аномальною або нетиповою складністю коду;
- урахувати статистичну невизначеність під час визначення якості програмних рішень;
- ухвалювати обґрунтовані рішення щодо потреби в рефакторингу.

2.5 Висновки до розділу 2

У даному розділі виконано наступне:

- Наведена методика побудови нелінійної регресійної моделі для прогнозування метрики RFC в залежності від метрик CBO та WMC для ПС, які базуються на штучному інтелекті, а також довірчих та прогнозних інтервалів.
- Побудована нелінійна регресійна модель для прогнозування метрики RFC в залежності від метрик CBO та WMC для вказаних ПС.
- Побудовані довірчі та прогнозні інтервали нелінійної регресії для прогнозування метрики RFC в залежності від метрик CBO та WMC для ПС, які базуються на штучному інтелекті.
- Розглянута методика прийняття рішення щодо якості ПС, що розроблені з використанням штучного інтелекту, яка базується на довірчих та прогнозних інтервалах нелінійної регресійної моделі для прогнозування метрики RFC в залежності від метрик CBO та WMC.

3 РОЗРОБКА ПРОЄКТНИХ РІШЕНЬ ДЛЯ ПРОГРАМНОЇ СИСТЕМИ ВИЗНАЧЕННЯ РІВНЯ ЯКОСТІ ПС, ЯКІ БАЗУЮТЬСЯ НА ШТУЧНОМУ ІНТЕЛЕКТІ

3.1 Постановка задачі на розробку програмної системи визначення рівня якості ПС, що базуються на штучному інтелекті

З огляду на викладені вище положення, сформулюємо постановку задачі на створення програмної системи для прийняття рішень щодо якості ПС, які базуються на штучному інтелекті. У кваліфікаційній роботі необхідно розробити ПС, що відповідає таким основним вимогам:

Вимоги до складу виконуваних функцій

Програмна система повинна надавати наступні функціональні можливості:

- Ввести значення метрики CBO для ПС, що базується на штучному інтелекті.
- Ввести значення метрики WMC для ПС, що базується на штучному інтелекті.
- Ввести фактичне значення метрики RFC для ПС, що базується на штучному інтелекті.
- Ввести значення довірчої ймовірності для розрахунку інтервалів.
- Перевірити вихідні дані на коректність.
- Оцінити значення метрики RFC для ПС, що базується на штучному інтелекті.
- Обчислити довірчий інтервал оцінки значення метрики RFC для ПС, що базується на штучному інтелекті.
- Обчислити прогнозний інтервал оцінки значення метрики RFC для ПС, що базується на штучному інтелекті.

– Визначити рівень якості ПС, що базується на штучному інтелекті, за результатами порівняння фактичного значення метрики RFC з отриманими інтервалами.

Вимоги до технічного забезпечення

Для коректної роботи програмної системи персональний комп'ютер (або інший пристрій) користувача повинен відповідати наступним мінімальним вимогам:

- Центральний процесор (CPU): Будь-який сучасний процесор із тактовою частотою від 1 ГГц (наприклад, Intel Core i3, AMD Ryzen 3 або аналогічні).
- Оперативна пам'ять (RAM): Мінімум 2 ГБ (рекомендовано 4 ГБ або більше для стабільної роботи операційної системи та веб-браузера).
- Накопичувач на жорсткому диску (HDD/SSD): Для розміщення файлів програми потрібно менше 1 МБ вільного простору.
- Пристрої вводу-виводу: Клавіатура, комп'ютерна миша (або тачпад), монітор із роздільною здатністю екрана не менше 1280x720 пікселів.
- Мережеве підключення: Не є обов'язковим (система працює локально без доступу до мережі Інтернет).

Деталізована постановка задачі наведена в технічному завданні, яке представлено в Додатку А.

3.2 Загальносистемні рішення програмної системи визначення рівня якості ПС, які базуються на штучному інтелекті

3.2.1 Вибір засобів моделювання програмної системи визначення рівня якості ПС

У процесі розроблення програмної системи визначення рівня якості ПС, які базуються на штучному інтелекті, важливим етапом є вибір засобів моделювання, що забезпечують формалізований і наочний опис

функціональних можливостей та поведінки системи. Застосування методів об'єктно-орієнтованого моделювання дозволяє забезпечити узгодженість між вимогами, проєктними рішеннями та програмною реалізацією.

У якості основного засобу моделювання було обрано мову UML – уніфіковану мову візуального моделювання, яка є загальноприйнятим стандартом у галузі розробки програмного забезпечення. UML надає набір типових діаграм, що дозволяють описати систему з різних точок зору та на різних рівнях абстракції.

Для відображення функціональних вимог до програмної системи у роботі буде використано діаграму варіантів використання, яка дозволяє визначити основних акторів, перелік функцій системи та межі її відповідальності. Така діаграма забезпечить чітке розуміння сценаріїв взаємодії користувача з програмною системою під час виконання операцій визначення якості ПС, що базуються на штучному інтелекті.

З метою опису логіки виконання окремих процесів та послідовності дій у системі буде застосовано діаграму діяльності, яка відображає алгоритмічні аспекти функціонування програмної системи, зокрема процеси введення вихідних даних, оцінювання значень метрики RFC, визначення довірчих і прогнозних інтервалів та визначення рівня якості ПС, що базуються на штучному інтелекті.

Крім того, для опису статичної структури програмної системи та взаємозв'язків між її компонентами планується розробити діаграму класів, а для моделювання динамічної взаємодії між об'єктами під час реалізації окремих варіантів використання – діаграми послідовності. Сукупне використання зазначених UML-діаграм дозволить комплексно описати програмну систему, підвищити якість проєктних рішень і спростити подальші етапи реалізації, тестування та супроводу.

3.2.2 Розробка діаграми варіантів використання програмної системи визначення рівня якості ПС

Перед проведенням побудови діаграми варіантів використання та діаграм сценаріїв взаємодії, необхідно визначити ключових учасників процесу роботи з програмною системою. У контексті програмної системи для оцінювання якості ПС, що базуються на штучному інтелекті, важливо ідентифікувати ролі, які виконують основні дії, ініціюють операції та отримують результати аналізу.

Нижче наведено результати виявлення акторів системи, їхні основні функції та взаємодії з програмною системою.

Таблиця 3.1 – Ідентифікація акторів програмної системи для оцінювання якості ПС, які базуються на штучному інтелекті

Ідентифіковані актори	Основні дії
Користувач	Особа, яка з допомогою програмної системи буде оцінювати якість ПС, які базуються на штучному інтелекті

Розроблювана програмна система має реалізовувати визначені функціональні можливості, які в UML представляються як варіанти використання.

Сформований перелік ідентифікованих варіантів використання, що відображають взаємодію користувача з системою, представлено в таблиці 3.2.

Таблиця 3.2 – Визначення та опис варіантів використання

Ідентифіковані актори	Ідентифіковані варіанти використання	Опис варіантів використання
1	2	3
Користувач	Ввести СВО для ПС із ШІ	Цей варіант використання реалізує можливість внесення користувачем числового значення метрики СВО ПС, яка базується на штучному інтелекті
Користувач	Ввести WMC для ПС із ШІ	Вказаний варіант використання реалізує можливість внесення користувачем числового значення метрики WMC ПС, яка базується на штучному інтелекті
Користувач	Ввести RFC для ПС із ШІ	Вказаний варіант використання реалізує можливість внесення користувачем числового значення метрики RFC ПС, яка базується на штучному інтелекті
Користувач	Ввести довірчу ймовірність	Вказаний варіант використання реалізує можливість внесення користувачем числового значення довірчої ймовірності, яка необхідна для обчислення меж довірчого та прогнозного інтервалів
Користувач	Оцінити	Вказаний варіант використання реалізує можливість переходу до перевірки даних та проведення оцінювання

Кінець табл. 3.2

1	2	3
Користувач	Перевірити вихідні дані	Вказаний варіант використання реалізує можливість перевірки вихідних даних, внесених користувачем
Користувач	Оцінити RFC для ПС із ІІІ	Вказаний варіант використання реалізує можливість оцінювання значення метрики RFC ПС яка базується на штучному інтелекті, через задані значення метрик CBO та WMC згідно розробленої моделі
Користувач	Обчислити довірчий інтервал оцінки RFC для ПС із ІІІ	Вказаний варіант використання реалізує можливість обчислення користувачем меж довірчого інтервалу для отриманої оцінки значення RFC ПС, яка базується на штучному інтелекті
Користувач	Обчислити прогнозний інтервал оцінки RFC для ПС із ІІІ	Вказаний варіант використання реалізує можливість обчислення користувачем меж прогнозного інтервалу для отриманої оцінки значення RFC ПС, яка базується на штучному інтелекті
Користувач	Визначити рівень якості ПС із ІІІ	Вказаний варіант використання реалізує можливість визначення рівня якості ПС, яка базується на штучному інтелекті, використовуючи розраховані довірчі та прогнозні інтервали

Після проведеного аналізу варіантів використання та визначення акторів було створено діаграму варіантів використання програмної системи для оцінювання якості ПС, що базуються на штучному інтелекті (рисунок 3.1). Розроблена діаграма сприяє більш точному визначенню вимог до програмної системи, полегшує процес проєктування та тестування, а також забезпечує основу для подальшого детального опису прецедентів і розробки діаграми діяльності.

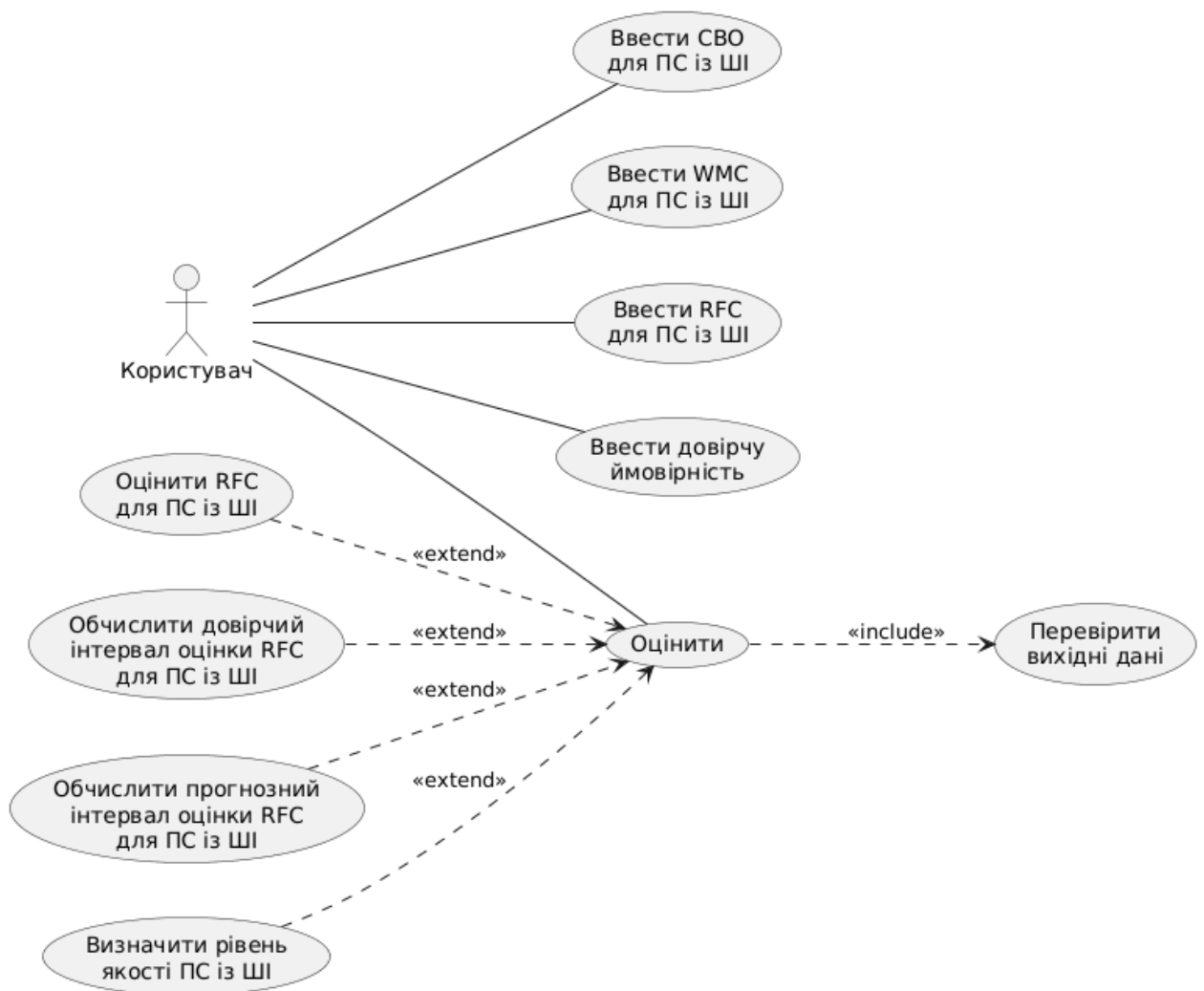


Рисунок 3.1 – Діаграма варіантів використання програмної системи для оцінювання якості ПС, які базуються на штучному інтелекті

На діаграмі зв'язки між актором та варіантами використання представлені у вигляді асоціацій, а між варіантами використання – включення та розширення.

3.2.3 Специфікації варіантів використання програмної системи визначення рівня якості ПС

На наступному кроці розроблено специфікації варіантів використання, які представлено в таблицях 3.3-3.12.

Специфікації варіантів використання створюються після побудови діаграми варіантів використання і призначені для: точного та формального опису роботи програмної системи в кожному сценарії взаємодії, забезпечення узгодженості вимог, спрощення розробки, тестування та документування.

Таблиця 3.3 – Специфікація варіанту використання «Ввести СВО для ПС із ІІІ»

Назва	Опис
Назва варіанту використання	Ввести СВО для ПС із ІІІ
Призначення	Реалізує можливість внесення користувачем числового значення метрики СВО ПС, яка базується на штучному інтелекті
Актор	Користувач
Передумови	Програма запущена
Основний потік	Користувач вводить у форму числове значення метрики СВО ПС
Альтернативні потоки	Не визначені
Постумови	Не визначені.

Таблиця 3.4 – Специфікація варіанту використання «Ввести WMC для ПС із ІІІ»

Назва	Опис
Назва варіанту використання	Ввести WMC для ПС із ІІІ
Призначення	Реалізує можливість внесення користувачем числового значення метрики WMC ПС, яка базується на штучному інтелекті
Актор	Користувач
Передумови	Програма запущена
Основний потік	Користувач вводить у форму числове значення метрики WMC ПС
Альтернативні потоки	Не визначені
Постумови	Не визначені.

Таблиця 3.5 – Специфікація варіанту використання «Обчислити прогнозний інтервал оцінки RFC для ПС із ІІІ»

Назва	Опис
Назва варіанту використання	Обчислити прогнозний інтервал оцінки RFC для ПС із ІІІ
Призначення	Реалізує можливість обчислення меж прогнозного інтервалу для отриманої оцінки значення RFC ПС
Актор	Користувач
Передумови	Варіант використання «Оцінити RFC для ПС із ІІІ»
Основний потік	1 Програма виконує обчислення меж прогнозного інтервалу для отриманої оцінки значення RFC ПС 2 Результат обчислення меж прогнозного інтервалу виводиться на екран
Альтернативні потоки	Не визначені
Постумови	Не визначені.

Таблиця 3.6 – Специфікація варіанту використання «Ввести довірчу ймовірність»

Назва	Опис
Назва варіанту використання	Ввести довірчу ймовірність
Призначення	Реалізує можливість внесення користувачем числового значення довірчої ймовірності для розрахунку довірчих та прогнозних інтервалів
Актор	Користувач
Передумови	Програма запущена
Основний потік	Користувач вводить у форму числове значення довірчої ймовірності
Альтернативні потоки	Не визначені
Постумови	Не визначені.

Таблиця 3.7 – Специфікація варіанту використання «Оцінити»

Назва	Опис
Назва варіанту використання	Оцінити
Призначення	Реалізує можливість переходу до перевірки даних та проведення оцінювання
Актор	Користувач
Передумови	Програма запущена
Основний потік	1. Актор натиснув кнопку «Оцінити». 2. Програма ініціює перевірку введених даних. 3. У випадку коректних даних реалізується перехід до оцінювання
Альтернативні потоки	У випадку некоректних даних відбувається вихід
Постумови	Не визначені.

Таблиця 3.8 – Специфікація варіанту використання «Перевірити вихідні дані»

Назва	Опис
Назва варіанту використання	Перевірити вихідні дані
Призначення	Реалізує перевірку вихідних даних на коректність
Актор	Користувач
Передумови	Програма запущена. Варіант використання «Оцінити»
Основний потік	Програма перевіряє дані на коректність
Альтернативні потоки	У випадку некоректних даних видається повідомлення про помилку
Постумови	Не визначені.

Таблиця 3.9 – Специфікація варіанту використання «Оцінити RFC для ПС із ІІІ»

Назва	Опис
Назва варіанту використання	Оцінити RFC для ПС із ІІІ
Призначення	Реалізує можливість оцінювання значення метрики RFC ПС, яка базується на штучному інтелекті, через задані значення метрик СВО та WMC згідно розробленої моделі
Актор	Користувач
Передумови	Варіант використання «Оцінити», варіант використання «Перевірити вихідні дані»
Основний потік	1.Програма виконує оцінювання значення метрики RFC згідно розробленої моделі з використанням метрик СВО та WMC 2 Результат оцінювання виводиться на екран
Альтернативні потоки	Не визначені
Постумови	Не визначені.

Таблиця 3.10 – Специфікація варіанту використання «Обчислити довірчий інтервал оцінки RFC для ПС із ІІІ»

Назва	Опис
Назва варіанту використання	Обчислити довірчий інтервал оцінки RFC для ПС із ІІІ
Призначення	Реалізує можливість обчислення меж довірчого інтервалу для отриманої оцінки значення RFC ПС
Актор	Користувач
Передумови	Варіант використання «Оцінити RFC для ПС із ІІІ»
Основний потік	1.Програма виконує обчислення меж довірчого інтервалу для отриманої оцінки значення RFC ПС 2 Результат обчислення меж довірчого інтервалу виводиться на екран
Альтернативні потоки	Не визначені
Постумови	Не визначені.

Таблиця 3.11 – Специфікація варіанту використання «Ввести RFC для ПС із ІІІ»

Назва	Опис
Назва варіанту використання	Ввести RFC для ПС із ІІІ
Призначення	Реалізує можливість внесення користувачем числового значення метрики RFC ПС, яка базується на штучному інтелекті
Актор	Користувач
Передумови	Програма запущена
Основний потік	Користувач вводить у форму числове значення метрики RFC ПС
Альтернативні потоки	Не визначені
Постумови	Не визначені.

Таблиця 3.12 – Специфікація варіанту використання «Визначити рівень якості ПС із ІІІ»

Назва	Опис
Назва варіанту використання	Визначити рівень якості ПС із ІІІ
Призначення	Реалізує можливість визначення рівня якості ПС, використовуючи розраховані довірчі та прогнозні інтервали згідно описаного підходу
Актор	Користувач
Передумови	«Обчислити довірчий інтервал оцінки RFC для ПС із ІІІ», «Обчислити прогнозний інтервал оцінки RFC для ПС із ІІІ»
Основний потік	1.Програма виконує визначення рівня якості ПС, використовуючи розраховані довірчі та прогнозні інтервали згідно описаного підходу 2 Результат визначення рівня якості ПС виводиться на екран
Альтернативні потоки	Не визначені
Постумови	Не визначені.

3.2.4 Створення діаграм діяльності програмної системи визначення рівня якості ПС

Діаграма діяльності призначена для моделювання послідовності дій, що виконуються в процесі роботи системи, бізнес-процесу або окремого варіанту використання.

Рисунок 3.2 ілюструє діаграму діяльності програмної системи, розробленої для прийняття рішень про якість ПС, які базуються на штучному інтелекті.

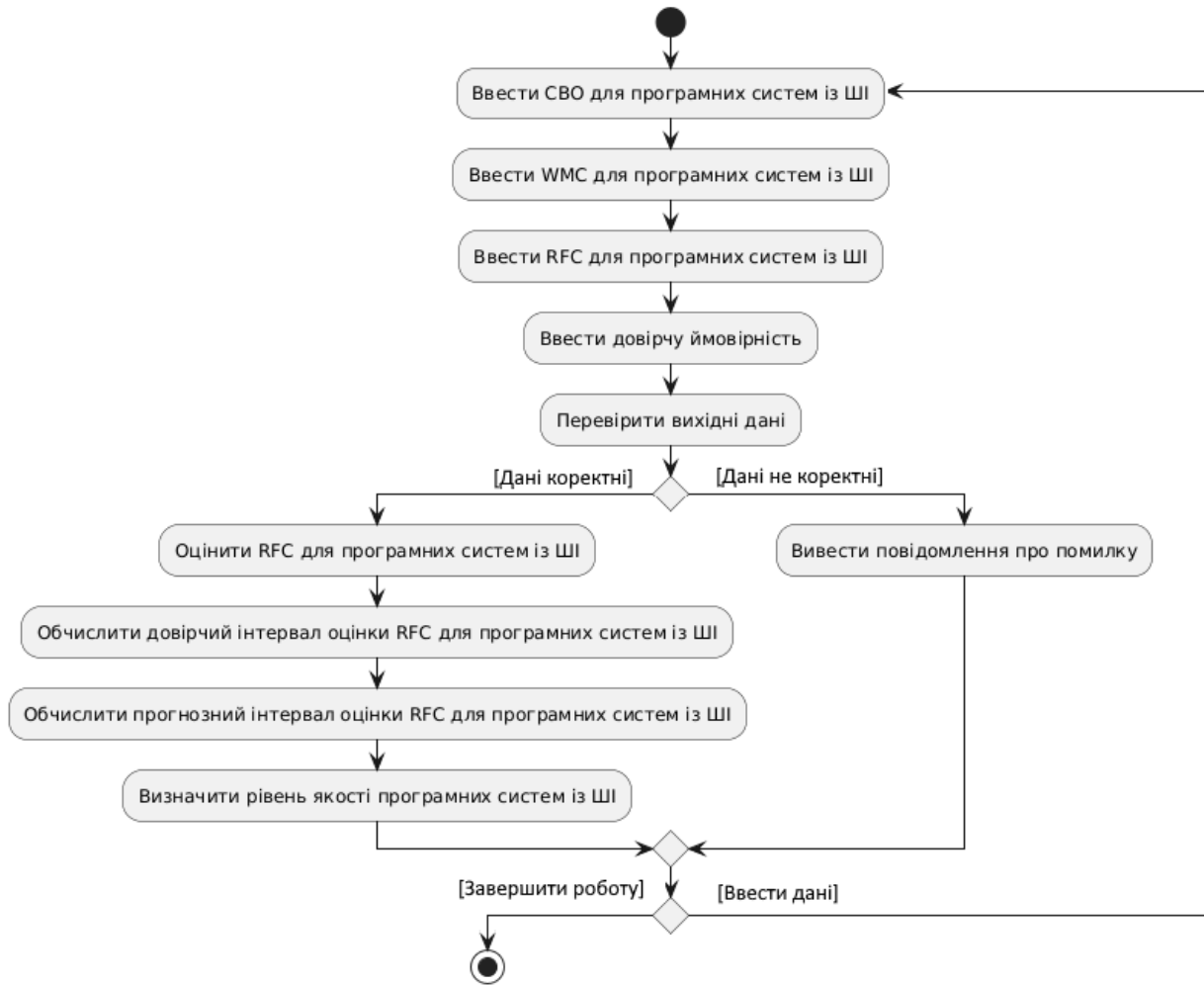


Рисунок 3.2 – Діаграма діяльності програмної системи для прийняття рішень про якість ПС, які базуються на штучному інтелекті

Побудована діаграма наочно відображає послідовність і взаємозв'язок основних дій, починаючи з введення метричних показників та довірчої ймовірності і завершуючи визначенням рівня якості ПС із ШІ.

Це забезпечує цілісне розуміння алгоритму роботи системи та створює підґрунтя для подальшого проєктування, реалізації та проведення тестування.

3.3 Рішення з інформаційного забезпечення програмної системи визначення рівня якості ПС, які базуються на штучному інтелекті

Інформаційне забезпечення програмної системи визначення рівня якості ПС, які базуються на штучному інтелекті, спрямоване на забезпечення коректного збирання, зберігання, оброблення та подання даних, необхідних для реалізації процесу оцінювання якості ПС. Воно визначає склад вихідної та результуючої інформації, структуру даних, а також правила їх використання у межах системи.

На вході програмної системи використовуються значення об'єктно-орієнтованих метрик, що характеризують структурні та поведінкові властивості ПС, а також довірна ймовірність. До таких вихідних даних належать:

- WMC (Weighted Methods per Class) – показник, що відображає кількість методів класу з урахуванням їх складності та використовується для оцінювання внутрішньої складності класу;
- CBO (Coupling Between Objects) – метрика, яка характеризує рівень зв'язності між класами та визначає ступінь їх взаємозалежності;
- RFC (Response For a Class) – кількість методів, які можуть бути викликані у відповідь на повідомлення, надіслане об'єкту класу, що відображає потенційну поведінкову складність класу;
- довірна ймовірність, яка задає рівень статистичної надійності результатів оцінювання та використовується під час формування довірчих і прогнозних інтервалів.

Усі вихідні дані вводяться користувачем через відповідні екранні форми програмної системи та проходять перевірку на коректність. Це дозволяє забезпечити цілісність інформації та уникнути помилок під час подальших обчислень.

Інформацією на виході програмної системи є:

- оцінене значення метрики RFC, отримане на основі нелінійної регресійної моделі з використанням метрик WMC та CBO;
- значення верхньої та нижньої меж довірчого інтервалу для оцінки RFC;
- значення верхньої та нижньої меж інтервалу прогнозування для оцінки RFC, що характеризують очікувану варіативність результату;
- інформація про рівень якості ПС, які базуються на штучному інтелекті.

Таким чином, інформаційне забезпечення програмної системи забезпечує формалізований опис вихідних і результуючих даних, їх взаємозв'язок та правила оброблення, що є необхідною умовою коректного функціонування програмної системи визначення рівня якості ПС, які базуються на штучному інтелекті.

3.4 Рішення з програмного забезпечення програмної системи визначення рівня якості ПС, які базуються на штучному інтелекті

3.4.1 Засоби розробки програмної системи визначення рівня якості ПС

Для розроблення програмної системи було використано сучасні вебтехнології, які забезпечують гнучкість, масштабованість та зручність у подальшій експлуатації системи, а саме: HTML5, CSS3, JavaScript (ES6+).

Основою реалізації користувацького інтерфейсу є мова гіпертекстової розмітки HTML5, яка застосовується для створення семантично коректної структури вебзастосунку. Використання стандартів HTML5 сприяє підвищенню доступності, зручності навігації та сумісності з різними браузерами.

Для оформлення зовнішнього вигляду та адаптації інтерфейсу під різні розміри екранів використано CSS3. Застосування сучасних можливостей цієї

технології, зокрема Flexbox та змінних CSS, дозволяє реалізувати адаптивний дизайн і забезпечити коректне відображення елементів інтерфейсу на різних пристроях.

Реалізація бізнес-логіки програмної системи, обробка подій користувача, а також виконання математичних обчислень здійснюються за допомогою мови програмування JavaScript (ES6+). Використання сучасних стандартів JavaScript забезпечує підвищену продуктивність, читабельність коду та спрощує подальшу підтримку і розширення функціональних можливостей системи.

3.4.2 Модель класів програмної системи визначення рівня якості ПС

Діаграма класів відображає основні класи програмної системи, їх властивості та методи, а також взаємозв'язки між об'єктами. Вона слугує основою для розуміння структури системи та для подальшої реалізації і тестування. Діаграма класів програмної системи для прийняття рішень про якість ПС, які базуються на штучному інтелекті, показана на рисунку 3.3.

Наведена діаграма класів відображає статичну структуру програмної системи для оцінювання метрики RFC та прийняття рішень щодо якості ПС, які базуються на штучному інтелекті. Архітектура побудована за принципом розділення відповідальностей між інтерфейсом користувача, обчислювальною логікою та допоміжними математичними функціями.

Клас UIController

Клас UIController відповідає за взаємодію з користувачем та керування графічним інтерфейсом. Він містить посилання на елементи HTML-інтерфейсу (форми та поля введення значень метрик СВО, WMC, RFC, рівня довіри), а також об'єкт класу RFCEvaluator, який використовується для виконання розрахунків.

Метод calculate() ініціює процес обчислення: зчитує вхідні дані з форми, передає їх до обчислювального модуля та відображає отримані результати

користувачеві. Зв'язок між `UIController` та `RFCEvaluator` реалізовано у вигляді композиції, що означає жорстку залежність контролера від обчислювального компонента.

Клас `RFCEvaluator`

Клас `RFCEvaluator` реалізує основну бізнес-логіку програмної системи та відповідає за оцінювання метрики RFC. У класі зберігаються параметри нелінійної регресійної моделі, зокрема коефіцієнти, допоміжні вектори та статистичні характеристики, необхідні для розрахунків.

Метод `predictRFC()` використовується для прогнозування значення метрики RFC на основі вхідного вектора метрик. Метод `getSEZ()` обчислює стандартну похибку оцінки, тоді як методи `calcPredInterval()` та `calcConfInterval()` визначають межі прогнозного та довірчого інтервалів відповідно. Таким чином, клас забезпечує як розрахунок оцінки метрики, так і статистичну оцінку достовірності результатів.

Клас `MathUtils`

Клас `MathUtils` є допоміжним (utility) класом і містить статичні математичні методи, що використовуються в процесі обчислень. Зокрема, метод `t_ppf()` забезпечує визначення квантиля розподілу Стюдента, а метод `matmul()` реалізує операції множення матриць і векторів. Клас `RFCEvaluator` використовує `MathUtils` для виконання низькорівневих математичних операцій, що сприяє підвищенню модульності та повторного використання коду.

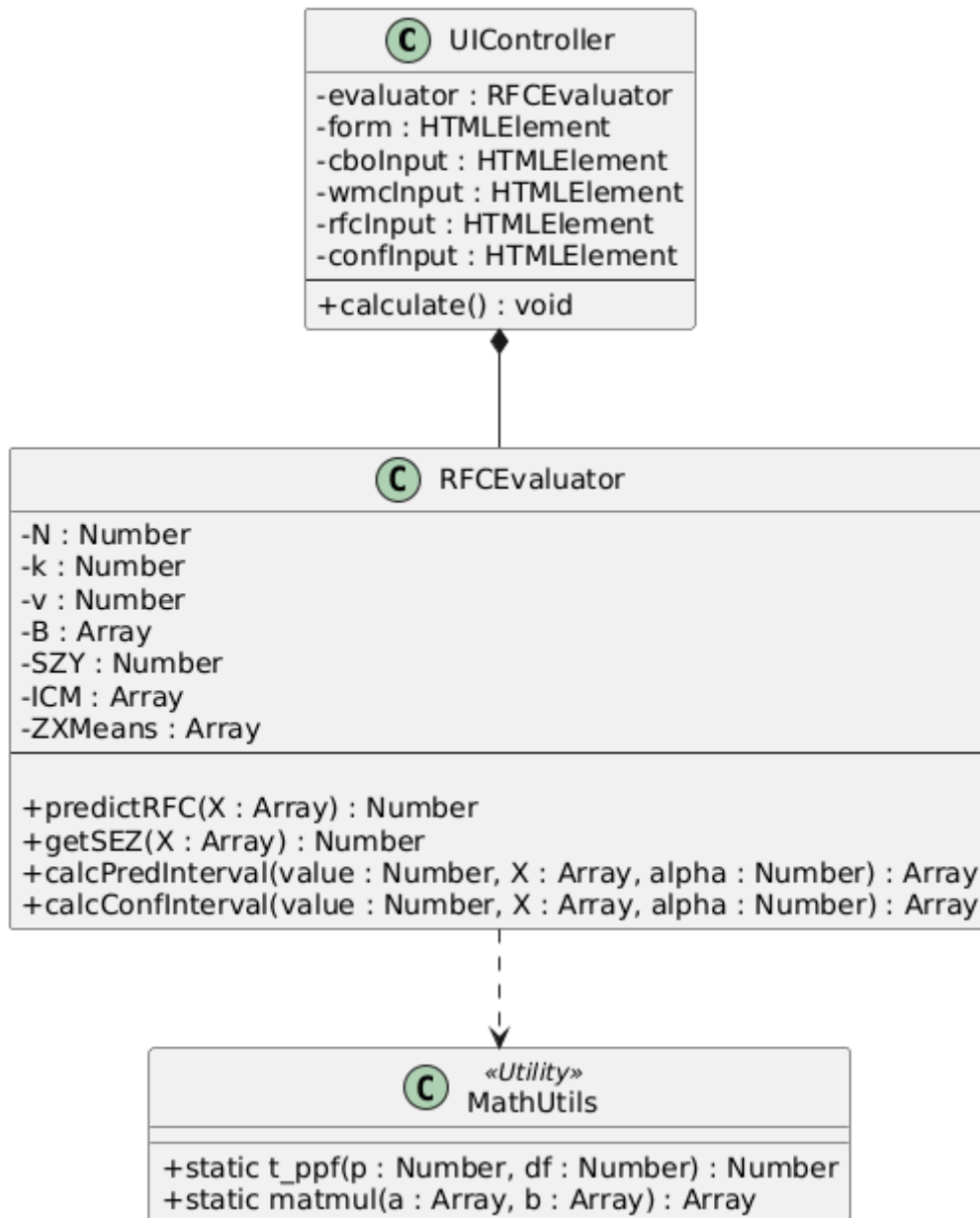


Рисунок 3.3 – Діаграма класів програмної системи для прийняття рішень про якість ПС, які базуються на штучному інтелекті

У цілому діаграма класів демонструє чітке розмежування між рівнем представлення, обчислювальною логікою та допоміжними сервісами, що відповідає принципам об'єктно-орієнтованого проектування та забезпечує розширюваність і підтримуваність програмної системи.

3.4.3 Моделі взаємодії програмної системи визначення рівня якості ПС

З метою детального відображення поведінки програмної системи та аналізу взаємодії між її компонентами у процесі виконання функціональних сценаріїв побудовано діаграми послідовності для варіантів використання «Введення значення метрики RFC» та «Введення довірчої ймовірності» програмної системи (рисунки 3.4, 3.5).

Діаграми послідовності описують часову послідовність обміну повідомленнями між об'єктами, виклики методів і реакції системи на дії користувача. Використання діаграм послідовності дозволяє уточнити логіку роботи системи, виявити потенційні суперечності у взаємодії компонентів та забезпечити коректну реалізацію функціональних вимог. На рисунку 3.4 представлена діаграма послідовності варіанту використання «Введення значення метрики RFC».

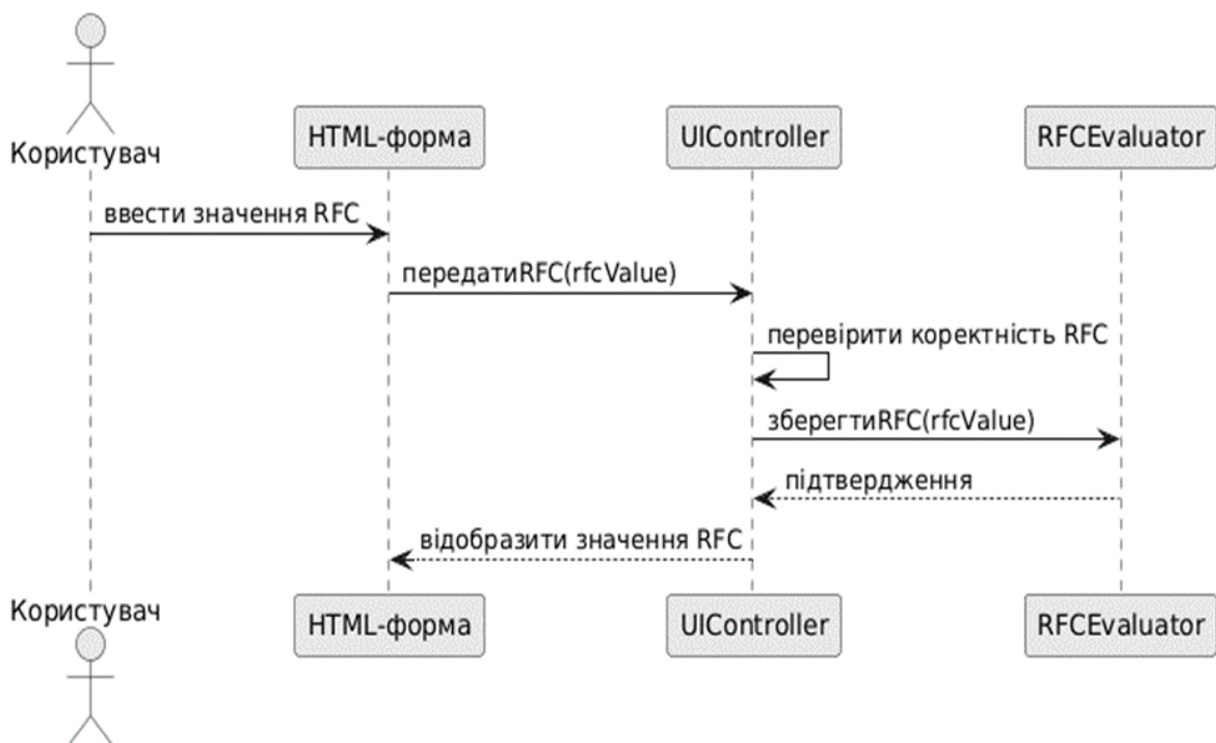


Рисунок 3.4 – Діаграма послідовності варіанту використання «Введення значення метрики RFC»

На рисунку 3.5 представлена діаграма послідовності варіанту використання «Введення довірчої ймовірності».

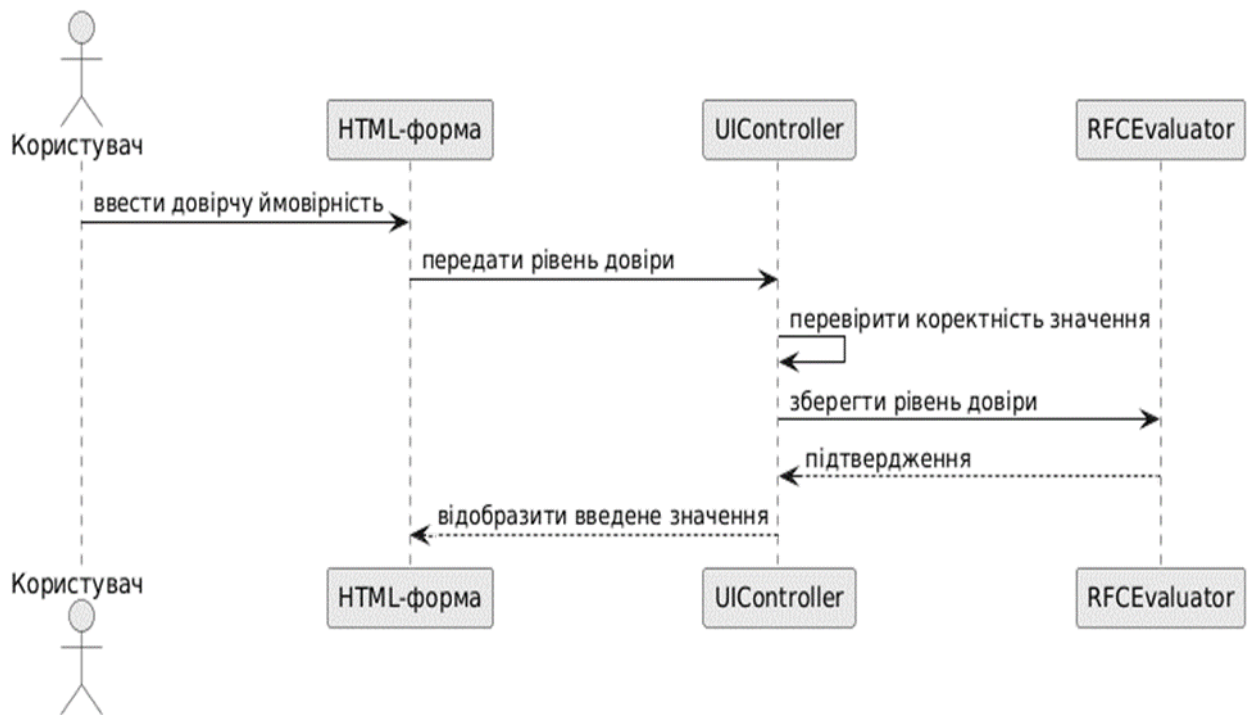


Рисунок 3.5 – Діаграма послідовності варіанту використання «Введення довірчої ймовірності»

Подані діаграми послідовності наочно демонструють логіку виконання сценарію, часову послідовність викликів методів та взаємодію між об'єктами системи, що дозволяє уточнити поведінку програмної системи та перевірити коректність реалізації алгоритму [18, 19].

3.4.4 Тестування програмної системи визначення рівня якості ПС

Тестування програмної системи є невід'ємною складовою процесу її розроблення та спрямоване на виявлення помилок, перевірку коректності функціонування системи та оцінювання відповідності продукту визначеним вимогам. Залежно від підходу до аналізу внутрішньої будови системи розрізняють функціональне та структурне тестування.

Функціональне тестування (чорна скринька) спрямоване на перевірку зовнішньої поведінки системи, без аналізу її внутрішньої реалізації. Основна мета – підтвердити, що програмна система виконує всі передбачені функції відповідно до вимог користувача. До найбільш поширених методів функціонального тестування належать:

- тестування за еквівалентними класами – розбиття вихідних даних на групи, що очікувано поведуться однаково, та перевірка представників кожної групи;
- аналіз граничних значень – перевірка поведінки системи на крайніх або критичних значеннях вхідних даних;
- тестування сценаріїв використання – моделювання реальних послідовностей дій користувача;
- регресійне тестування – повторна перевірка функцій після внесення змін, щоб переконатися у відсутності нових помилок.

Структурне тестування (біла скринька) орієнтоване на аналіз внутрішньої будови системи та перевірку логіки її роботи. Основна мета – забезпечити повноту покриття коду та виявити приховані помилки в реалізації алгоритмів. До основних методів структурного тестування належать:

- тестування покриття операторів – перевірка виконання всіх операторів у програмі;
- тестування покриття гілок – перевірка проходження всіх умовних переходів та розгалужень;
- тестування покриття шляхів – перевірка всіх можливих шляхів виконання системи;
- тестування покриття умов – перевірка всіх логічних виразів у програмі.

У роботі тестування виконувалось за еквівалентними класами.

Розглянемо тестування функції «Ввести фактичне значення метрики RFC для ПС, що базується на штучному інтелекті» програмної системи

прийняття рішень про якість відповідних ПС. Еквівалентні класи представлені в таблиці 3.13.

Таблиця 3.13 – Еквівалентні класи функції «Ввести фактичне значення метрики RFC для ПС, що базується на штучному інтелекті»

Вхідні дані	Правильні еквівалентні класи	Неправильні еквівалентні класи
фактичне значення RFC для ПС, що базується на штучному інтелекті	1. невід’ємне число	2. від’ємне число 3. не число 4. пусте поле

Результати тестування функції «Ввести фактичне значення RFC для ПС, що базується на штучному інтелекті» наведено в таблиці 3.14.

Таблиця 3.14 – Тести для еквівалентних класів функції «Ввести фактичне значення RFC для ПС, що базується на штучному інтелекті»

№ тесту	Вхідні дані	Покритий клас	Результат тестування
1	20,00	1	можливість виконати оцінювання
2	-15,49	2	повідомлення про недопустиме значення RFC
3	Дек1?%	3	повідомлення про недопустиме значення RFC
4	відсутні	4	повідомлення не відсутнє значення

Тестування функції «Ввести значення метрики СВО для ПС, що базується на штучному інтелекті» програмної системи для прийняття рішень про якість відповідних ПС. Еквівалентні класи представлені в таблиці 3.15.

Таблиця 3.15 – Еквівалентні класи функції «Ввести значення метрики СВО для ПС, що базується на штучному інтелекті»

Вхідні дані	Правильні еквівалентні класи	Неправильні еквівалентні класи
значення метрики СВО для ПС, що базується на штучному інтелекті	1. невід’ємне число	2. від’ємне число 3. не число 4. пусте поле

Результати тестування функції «Ввести фактичне значення СВО для ПС, що базується на штучному інтелекті» наведено в таблиці 3.16.

Таблиця 3.16 – Тести для еквівалентних класів функції «Ввести фактичне значення СВО для ПС, що базується на штучному інтелекті»

№ п/п	Вхідні дані	Покритий клас	Результат
1	5,00	1	можливість виконати оцінювання
2	-0,35	2	повідомлення про недопустиме значення СВО
3	p%ов*9)	3	повідомлення про недопустиме значення СВО
4	відсутні	4	повідомлення не відсутнє значення

3.4.5 Випробування програмної системи визначення рівня якості ПС

Після завершення етапу розробки та реалізації програмної системи, проведено її випробування з метою оцінювання функціональності, надійності та відповідності вимогам технічного завдання. Результати випробувань дозволяють проаналізувати фактичні показники якості ПС, виявити можливі недоліки та перевірити коректність роботи основних функцій.

Документ «Програма та методика випробувань програмної системи для прийняття рішень щодо якості ПС, які базуються на штучному інтелекті», наведений в Додатку В. В роботі випробовувалися наступні функції:

- Ввести значення СВО для ПС, що базується на штучному інтелекті.
- Ввести значення WMC для ПС, що базується на штучному інтелекті.
- Ввести фактичне значення RFC для ПС, що базується на штучному інтелекті.
- Ввести значення довірчої ймовірності для розрахунку інтервалів.
- Оцінювання значення RFC для ПС, що базується на штучному інтелекті.
- Обчислення меж довірчого інтервалу для оцінки значення RFC.
- Обчислення меж прогнозного інтервалу для оцінки значення RFC.
- Визначення рівня якості ПС, що базується на штучному інтелекті, за результатами порівняння фактичного значення RFC з отриманими інтервалами.

У процесі випробувань було завантажено програмну систему та здійснене введення всіх необхідних даних у відповідну форму. Інтерфейс на етапі введення даних для прийняття рішень про якість ПС, що базуються на штучному інтелекті, наведений на рисунках 3.6, 3.7.

Прийняття рішень щодо якості ПС, що базуються на штучному інтелекті

Вхідні параметри

Ввести СВО для ПС із ШІ:

Наприклад: 4,7

Ввести WMC для ПС із ШІ:

Наприклад: 15,39

Ввести RFC для ПС із ШІ:

Наприклад: 22,89

Ввести довірчу ймовірність:

0.95

Оцінити

Рисунок 3.6 – Інтерфейс перед введенням даних для прийняття рішення щодо якості ПС, що базуються на штучному інтелекті

Прийняття рішень щодо якості ПС, що базуються на штучному інтелекті

Вхідні параметри

Ввести СВО для ПС із ШІ:

4.7

Ввести WMC для ПС із ШІ:

15.39

Ввести RFC для ПС із ШІ:

22.89

Ввести довірчу ймовірність:

0.95

Оцінити

Рисунок 3.7 – Інтерфейс після введення даних для прийняття рішення щодо якості ПС, що базуються на штучному інтелекті

Після введення даних та натискання кнопки «Оцінити» отримуємо результати оцінювання.

Інтерфейс на етапі перегляду результатів оцінювання ПС, що базуються на штучному інтелекті, наведений на рисунку 3.8.

The screenshot displays a web interface titled "Результати розрахунків" (Results of calculations). It contains four rows of data, each with a label and a corresponding value in a light blue box:

- Результат оцінювання RFC (Прогноз): 20.8189
- Рівень якості ПС із ШІ: висока
- Довірчий інтервал оцінки RFC: 19.0739 - 22.7235
- Прогнозний інтервал оцінки RFC: 13.6421 - 31.7712

Рисунок 3.8 – Інтерфейс на етапі перегляду результатів оцінювання ПС, що базуються на штучному інтелекті

За результатами проведених випробувань встановлено, що розроблена програмна система коректно реалізує передбачений функціонал та забезпечує виконання основних сценаріїв використання відповідно до вимог технічного завдання. Усі перевірені функції працюють стабільно, а результати обчислень відповідають очікуваним значенням.

Таким чином, результати випробувань дозволяють зробити висновок про відповідність розробленої програмної системи поставленим вимогам, можливість її практичного використання та доцільність впровадження для визначення рівня якості ПС, що базуються на штучному інтелекті.

3.5 Висновки до розділу 3

У розділі 3 розроблено проєктні рішення для програмної системи визначення рівня якості ПС, що базуються на штучному інтелекті.

Сформульовано постановку задачі, визначено функціональні можливості та вимоги до програмної системи. Обґрунтовано загальносистемні рішення із застосуванням UML-нотації, розроблено діаграми варіантів використання, їх специфікації та діаграми діяльності, що відображають основні сценарії роботи системи.

Опрацьовано рішення з інформаційного забезпечення, визначено склад вихідних та результуючих даних. Обґрунтовано рішення з програмного забезпечення, побудовано модель класів і моделі взаємодії, а також описано процеси тестування та випробування системи.

У результаті сформовано цілісну структуру програмної системи, яка забезпечує автоматизоване та достовірне визначення рівня якості ПС, що базуються на штучному інтелекті.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОЇ СИТЕМИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ЯКОСТІ ПС, ЯКІ БАЗУЮТЬСЯ НА ШТУЧНОМУ ІНТЕЛЕКТІ

4.1 Вступ

У цій роботі розглядається процес створення програмної системи, призначеної для оцінювання якості ПС, які базуються на штучному інтелекті.

Економічний аналіз розробки програмної системи є необхідною складовою обґрунтування доцільності її створення та впровадження. Такий аналіз передбачає визначення повної собівартості розробки, яка включає витрати на оплату праці розробників, використання технічних ресурсів, амортизацію обладнання, супутні адміністративні витрати та інші складові, пов'язані з процесом створення продукту.

Поряд із визначенням витрат проводиться оцінювання потенційних доходів, що можуть бути отримані від реалізації або використання програмної системи. До таких доходів належать прямі прибутки від продажу ліцензій, опосередковані вигоди від підвищення продуктивності праці, зменшення витрат на виконання окремих бізнес-процесів та підвищення якості управлінських рішень.

Результатом економічного аналізу є розрахунок річного економічного ефекту, який виступає основним критерієм визначення доцільності інвестицій у розробку програмної системи. Цей показник дає змогу оцінити співвідношення між отриманими результатами та понесеними витратами, а також визначити строк окупності проєкту та його фінансову перспективність.

Таким чином, комплексний економічний підхід забезпечує обґрунтованість управлінських рішень щодо розробки та впровадження програмної системи, дозволяє мінімізувати ризики та забезпечує ефективне використання фінансових ресурсів підприємства.

4.2 Розрахунок витрат розробки і впровадження програмної системи визначення рівня якості ПС, які базуються на штучному інтелекті

Витрати на створення програмної системи формуються з таких основних складових: амортизаційних та експлуатаційних відрахувань на ПК, що використовується у процесі розробки; оплати праці розробника; вартості програмних засобів, необхідних для розроблення; витрат на матеріали та комплектуючі.

З огляду на те, що середній розмір місячної заробітної плати розробника становить 20000 грн, орієнтовна вартість сучасного ПК – 30000 грн, а тариф на електроенергію дорівнює 4,32 грн за 1 кВт·год, можна здійснити розрахунок сукупної вартості розробки системи.

Розрахунок заробітної плати:

$$З_{зп} = ЗП_{роз} \cdot T_{роз},$$

де $ЗП_{роз}$ – зарплатня розробника за місяць; $T_{роз}$ – тривалість розробки.

Для створення даної системи передбачено тримісячний термін розробки, що визначено на підставі експертної оцінки тривалості виконання аналогічних проєктів. Роботи здійснює один розробник.

Відтак загальний обсяг витрат на оплату його праці становитиме:

$$З_{зп} = ЗП_{роз} \cdot T_{роз} = 20000 \cdot 3 = 60000 \text{ грн.}$$

Амортизаційні витрати на ПК, що використовується у процесі розробки, визначаються за такою формулою:

$$З_{амор} = Ц_{варт} \cdot A \cdot T_{роз},$$

де $C_{\text{варт}} = 10000$ грн. – балансова вартість ПК; $A = 50\%$ – річна амортизація; строк експлуатації ПК – 5 років; $T_{\text{роз}} = 0,25$ року – термін створення програмної системи.

$$Z_{\text{амор}} = 10000 \cdot 0,5 \cdot 0,25 = 1250 \text{ грн.}$$

Під час розробки основною статтею експлуатаційних витрат на ПК є оплата електроенергії, що розраховується за заданою формулою:

$$Z_{\text{експ}} = P_{\text{ЕОМ}} \cdot T_{\text{розр}} \cdot N_{\text{рд}} \cdot N_{\text{рч}} \cdot E_{\text{ел}},$$

де $P_{\text{ЕОМ}} = 0,4$ кВт./год. – потужність ПК; $T_{\text{розр}} = 3$ місяці – строк розроблення; $N_{\text{рд}} = 22$ дні – норма робочих днів на місяць; $N_{\text{рч}} = 8$ годин – тривалість робочого дня (у годинах); $E_{\text{ел}} = 4,32$ грн. – вартість однієї кіловат-години електроенергії:

$$Z_{\text{експ}} = 0,4 \cdot 3 \cdot 22 \cdot 8 \cdot 4,32 = 912,38 \text{ грн.}$$

До складу витрат на матеріали та комплектуючі входять витрати на папір для принтерів, картриджі та тонери, а також кошти на непередбачені потреби. Детальний розрахунок за поточними ринковими та договірними цінами наведено в таблиці 4.1.

Таблиця 4.1 – Розрахунок допоміжних витрат

№ з/п	Складові витрат	Ціна, грн.
1	Папір для принтера	150,00
2	Картридж та тонер для принтера	300,00
3	Доступ до Internet	160,00
4	Непередбачені витрати	40,00
Всього		650,00

У таблиці 4.2 наведено загальний кошторис проєкту, що включає всі розглянуті види витрат.

Таблиця 4.2 – Загальний кошторис проекту

№ з/п	Перелік витрат	Ціна, грн.
1	Зар. плата основна	60000,00
2	Зар. плата додаткова (20% від п.1)	12000,00
3	Соц. страхув. (38% від пп. 1 та 2)	27360,00
4	Амортизація ПК	1250,00
5	Експлуатація ПК	912,38
6	Доп. витрати	650,00
7	Адмін. витрати (50% від п. 1)	30 000,00
Всього		132172,38

Кошторис впровадження системи включає такі основні статті витрат: закупівлю нового технічного та програмного забезпечення, оплату послуг з налаштування мережі та навчання персоналу.

Вартість наявного обладнання, характеристики якого відповідають вимогам проекту, до розрахунку не включена. Витрати на супроводження проекту приймаються на рівні 1 % його вартості та становлять становлять 1321,72 грн. На підставі досвіду розроблення аналогічних проектів визначено, що витрати на підготовку персоналу складають 1000,00 грн.

Інвестиційні витрати на впровадження системи з урахуванням вартості навчання кадрів і витрати на супровід представлені в таблиці 4.3.

Таблиця 4.3 – Витрати на впровадження

№	Призначення витрат	Вартість, грн.
3	Підготовка кадрів	1000,00
4	Супровід	1321,72
Всього		2321,72

Разом сума витрат на розробку і впровадження дорівнює 134494,1 грн.

4.3 Економічна ефективність розробки і впровадження програмної системи визначення рівня якості ПС, які базуються на штучному інтелекті

Пряму економічну ефективність визначимо, виходячи з того, що впровадження системи дозволяє зменшити потребу в трудових ресурсах на 1 працівника.

Зарплата 1 робітника у рік дорівнює $\Delta C = 20000 \cdot 12 \cdot 1 = 240000$ грн. Економічний ефект за рік обраховується за формулою:

$$E_{\text{рік}} = \Delta C - C_{\text{супр}} - E_n \cdot k,$$

де $C_{\text{супр}}$ – витрати на супроводження; E_n – нормативний показник економічної ефективності капітальних інвестицій у відповідну галузь (для засобів обчислювальної техніки) встановлюється на рівні $= 0,5$; k – додаткові капітальні вкладення, що включають затрати на проектування, розроблення та подальшу експлуатацію системи.

$$E_{\text{рік}} = 240000 - 1321,72 - 0,5 \cdot 175658,06 = 150849,25 \text{ (грн.)}$$

Термін окупності розробки обчислюється згідно:

$$I = \frac{k}{\Delta C - C_{\text{супр}}}.$$

Звідси

$$I = (175658,06 / (240000 - 1321,72)) = 0,74 \text{ (року).}$$

Тобто, система окупиться через 0,74 року, що приблизно дорівнює 8,9 місяців.

4.4 Висновки до розділу 4

У цьому розділі проведено розрахунок витрат, пов'язаних із розробленням та експлуатацією програмної системи для оцінювання якості ПС, що ґрунтуються на технологіях штучного інтелекту, а також визначено показники економічної ефективності та строку його окупності. За результатами виконаних розрахунків встановлено, що впровадження програмної системи забезпечує повну окупність інвестицій протягом 8,9 місяців. Це свідчить про економічну доцільність і раціональність розроблення даної програмної системи.

5 ОХОРОНА ПРАЦІ

5.1 Загальні положення та актуальність забезпечення безпеки праці

Охорона праці є невід’ємною складовою сучасного виробничого середовища та ключовим елементом організації безпечних умов діяльності працівників. У контексті інформаційних технологій, де значна частина робочого часу припадає на взаємодію з персональними комп’ютерами, офісним обладнанням та програмними засобами, питання створення комфортних, безпечних і здорових умов праці набуває особливого значення.

Актуальність проблеми посилюється тим, що робота з комп’ютером має низку специфічних факторів впливу: тривале статичне напруження, високе навантаження на зоровий аналізатор, вплив електромагнітних полів, можливість психоемоційного перевантаження та ризику, пов’язані з організацією робочого середовища. Вчасне виявлення таких небезпечних і шкідливих факторів та впровадження системи превентивних заходів дозволяє мінімізувати негативний вплив на здоров’я працівника, підвищити продуктивність праці та забезпечити відповідність вимогам нормативно-правових актів України у сфері охорони праці, серед яких Закон України “Про охорону праці” [21], Кодекс законів про працю України (КЗпП) [22] та інші.

5.2 Характеристика основних небезпечних і шкідливих чинників при роботі із засобами обчислювальної техніки

Працівники, що здійснюють професійну діяльність за персональним комп’ютером, зазнають впливу ряду небезпечних та шкідливих чинників. Нижче наведено їх класифікацію та опис:

– Фізичні фактори

Електромагнітне випромінювання (ЕМВ)

Монітори, системні блоки та периферійні пристрої створюють електромагнітні поля низьких частот. Хоча їх рівень, як правило, не перевищує нормативних значень, тривале перебування поряд з джерелами ЕМВ може спричиняти головний біль, втому та погіршення загального самопочуття.

Шум та вібрація

Джерелами шуму виступають вентилятори системного блоку, жорсткі диски, принтери. Довготривалий шум навіть низької інтенсивності сприяє зниженню концентрації уваги та втомі.

Мікроклімат приміщення

Температурний режим, вологість та швидкість руху повітря прямо впливають на комфорт працівника. Порушення норм призводить до швидкої втомлюваності, зниження працездатності та захворювань.

– Ергономічні фактори

Статичні навантаження

Тривале перебування у незручній позі сприяє викривленню хребта, захворюванням опорно-рухового апарату та порушенням кровообігу.

Невдале розміщення робочого місця

Невідповідність висоти столу та стільця антропометричним параметрам працівника призводить до перенапруження м'язів плечового поясу та спини.

Перенапруження зорового апарату

Тривала робота за монітором спричиняє синдром зорової втоми, сухість очей, погіршення гостроти зору та головний біль.

– Психофізіологічні фактори

Нервово-емоційне перевантаження

Робота у сфері ІТ часто пов'язана з високою інтенсивністю інформаційних потоків, дедлайнами, необхідністю багатозадачності.

Ризик професійного вигорання

Монотонність, надлишкові вимоги та високий темп роботи сприяють появі хронічної втоми та зниженню мотивації.

5.3 Заходи щодо зменшення впливу небезпечних та шкідливих факторів

Забезпечення безпечних умов праці потребує впровадження комплексу технічних, організаційних та санітарно-гігієнічних заходів.

– Організаційні заходи

Раціональний режим праці та відпочинку

Відповідно до ДСанПіН 3.3.2-007-98 рекомендовано:

- робити перерви 10–15 хвилин після кожних 45–60 хвилин роботи за ПК;
- змінювати види діяльності для розвантаження нервової та зорової систем.

Проведення інструктажів з охорони праці

Первинний, повторний та позаплановий інструктажі сприяють підвищенню рівня обізнаності працівників.

Організація психологічної підтримки

За необхідності – тренінги з управління стресом, формування емоційної стійкості.

– Технічні заходи

Використання ергономічного обладнання

- Столи та крісла з регулюванням висоти.
- Монітори з матовим покриттям, частотою оновлення не менше 75 Гц.
- Клавіатури та миші ергономічного типу.

Оптимізація розміщення робочого місця

- Відстань від очей до монітора 50–70 см.
- Верхня межа екрана повинна бути на рівні очей або трохи нижче.
- Освітлення не повинно спричиняти відблиски на екрані.

Використання засобів захисту та контролю

- Захисні екрани або фільтри на монітори (за потреби).
 - Стабілізатори напруги.
 - Вимірювання рівня ЕМВ, шуму, освітлення.
- Санітарно-гігієнічні заходи

Параметри мікроклімату

Відповідно до ДСН 3.3.6.042-99:

- температура: 18–24°C,
- вологість: 40–60 %,
- швидкість руху повітря: до 0,1–0,2 м/с.

Раціональне освітлення приміщення

- Освітленість робочої зони 300–500 лк.
- Комбіноване освітлення (природне + штучне).

Профілактика зорового перенапруження

Виконувати вправи:

- фокусування на дальніх об'єктах,
- кругові рухи очей,
- чергування роботи на близькій та далекій відстані.

Профілактика захворювань опорно-рухового апарату

- Розминка кожні 1–2 години.
- Вправи на розтягування та зміцнення м'язів спини, шиї, плечового пояса.

5.4 Висновки до розділу 5

У розділі проаналізовано основні небезпечні й шкідливі фактори, що виникають під час роботи з персональним комп'ютером, а також розглянуто комплекс заходів для їх мінімізації. Показано, що дотримання норм ергономіки, санітарно-гігієнічних вимог, технічне забезпечення робочого

місця та організація раціонального режиму праці є необхідними умовами створення безпечного робочого середовища.

Запропоновані заходи спрямовані на підвищення рівня безпеки, зниження ризиків професійних захворювань, підвищення продуктивності праці та забезпечення відповідності чинним нормативним документам України в галузі охорони праці.

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1. Основні положення охорони навколишнього природного середовища

Охорона навколишнього природного середовища охоплює сукупність міжнародних, державних та регіональних нормативно-правових актів, інструкцій і стандартів, що встановлюють обов'язкові вимоги для всіх суб'єктів господарювання, діяльність яких може спричинити екологічне забруднення. Зазначені норми забезпечують належний рівень відповідальності та зацікавленості сторін у дотриманні екологічних правил і впровадженні природоохоронних заходів.

Ефективне функціонування системи охорони навколишнього природного середовища можливе лише за умови узгодженості та взаємозалежності її складових, а також єдності підходів до їх подальшого розвитку.

Оскільки проблема захисту природних ресурсів від негативного впливу різноманітної діяльності залишається актуальною, особливої ваги набуває забезпечення безпеки людини в умовах зміненого природного середовища. У зв'язку з цим підходи до охорони природи та охорони здоров'я людини інтегруються в єдину концепцію – охорону навколишнього (природного) середовища.

Основні напрями охорони навколишнього природного середовища включають:

- правовий напрям, який передбачає формування системи законодавчих норм на основі екологічних принципів;
- економічне стимулювання, що спрямоване на забезпечення вигідності та доцільності впровадження природоохоронних заходів для підприємств;

- інженерно-технічний напрям, пов'язаний із розробленням і застосуванням екологічно безпечних технологій та обладнання.

До основних принципів охорони навколишнього природного середовища належать:

- пріоритет створення умов, що забезпечують безпечне та здорове життя, працю і відпочинок населення;
- науково обґрунтоване поєднання екологічних вимог з економічними потребами суспільства;
- урахування природних закономірностей, здатності екосистем до самовідновлення та самоочищення;
- недопущення необоротних негативних змін у довкіллі та загроз для здоров'я людей;
- забезпечення відкритості та доступності інформації про стан навколишнього середовища і вплив промислових об'єктів на здоров'я населення;
- невідворотність відповідальності за порушення екологічного законодавства.

6.2 Забруднення навколишнього середовища комп'ютерною технікою та заходи щодо його зменшення

Проблематика екологічного впливу комп'ютерних технологій упродовж останніх років набула особливої уваги у сфері охорони довкілля. Стрімкий розвиток ІТ-індустрії, зростання масштабів виробництва електронного обладнання, збільшення кількості комп'ютерних систем у побуті, бізнесі та промисловості зумовили появу нових викликів, пов'язаних із забрудненням навколишнього середовища. Протягом останніх 2–3 років ці питання активно обговорюються у засобах масової інформації, на наукових форумах, спеціалізованих виставках та в професійних спільнотах.

Сучасні тенденції розвитку комп'ютерної галузі свідчать, що питання екологічної відповідальності дедалі більше інтегрується в корпоративні стратегії провідних виробників. Це обумовлено не лише нормативними вимогами, а й соціальними очікуваннями щодо зменшення шкідливого впливу технологій на довкілля та здоров'я людини.

Основні джерела екологічного навантаження комп'ютерної техніки

Екологічний вплив комп'ютерних систем проявляється на всіх етапах їх життєвого циклу:

- виготовлення компонентів, що супроводжується значним споживанням енергії та ресурсів;
- експлуатація, де головним чинником є високе енергоспоживання;
- утилізація та переробка, що пов'язані з накопиченням електронних відходів.

Виробництво комп'ютерів потребує значних обсягів рідкоземельних металів, водних ресурсів, хімічних реагентів та енергії. Експлуатація потужних центрів з обробки інформації супроводжується великим обсягом тепловиділення, що потребує систем охолодження, які в свою чергу також споживають ресурси. Утилізація електронних компонентів створює особливу небезпеку через вміст токсичних речовин, які за відсутності належної переробки потрапляють у повітря, ґрунт і воду.

Поширення та значення «зелених» технологій (Green IT)

У відповідь на зростання екологічних ризиків у комп'ютерній галузі формується концепція Green IT, яка охоплює комплекс технічних та організаційних заходів, спрямованих на зменшення негативного впливу комп'ютерних технологій на довкілля. Основними характеристиками «зелених» комп'ютерних технологій є:

- зменшення енергоспоживання обладнання;
- зниження електромагнітного випромінювання;
- зменшення рівня шуму;
- відсутність або мінімізація токсичних компонентів;

- використання матеріалів, придатних до повторної переробки;
- створення конструкцій, що спрощують утилізацію;
- застосування енергоефективних алгоритмів та програмних рішень.

Впровадження Green IT охоплює виробників апаратного забезпечення, керівників дата-центрів, розробників програмного забезпечення, корпоративних користувачів та кінцевих споживачів. На сьогодні ця сфера розглядається як один із ключових напрямів сталого розвитку IT-індустрії.

Екологічні виклики, пов'язані з роботою дата-центрів

Проблема енергоспоживання інформаційних систем є однією з найгостріших. Сучасні центри обробки даних (ЦОД) споживають величезні обсяги електроенергії, що супроводжується значним рівнем теплових викидів. Підтримання стабільного температурного режиму потребує складних та енергоємних систем кондиціонування, що збільшує навантаження на енергосистему та генерує додаткові витрати.

У відповідь на ці виклики провідні технологічні компанії впроваджують інноваційні рішення. Одним із таких є розміщення дата-центрів у нестандартних локаціях – зокрема, на спеціальних морських платформах, що розміщуються на значній відстані від узбережжя. Такий підхід дозволяє використовувати природне охолодження морською водою, енергію хвиль та вітру, що істотно зменшує навантаження на традиційні енергетичні системи та знижує загальний рівень викидів.

Швидке старіння техніки та проблема електронних відходів

Одним із найвагоміших чинників негативного впливу комп'ютерних технологій є швидка деградація та моральне старіння пристроїв. Досі актуальним залишається закон Мура, згідно з яким кількість транзисторів у мікросхемах подвоюється приблизно кожні два роки. Це спричиняє постійну зміну поколінь техніки, підвищення вимог до продуктивності, а отже – зростання обсягів електронних відходів.

За результатами досліджень, середній житель Великої Британії протягом життя генерує близько трьох тон електронного сміття. У глобальному масштабі

це становить мільйони тон відходів, значна частина яких утилізується без дотримання екологічних норм, що створює значні ризики для довкілля та здоров'я людини.

Токсичність компонентів комп'ютерної техніки

Багато комп'ютерних пристроїв містять токсичні елементи:

- ртуть;
- свинець;
- кадмій;
- хром;
- бромовані антипірени;
- різноманітні пластики тривалого розкладу.

Монітори на електронно-променевих трубках, старі системні блоки, акумулятори та друковані плати є джерелом небезпечних речовин, які здатні вимиватися в ґрунт, потрапляти у воду або випаровуватися в атмосферу. Час розкладання пластикового корпусу комп'ютера може сягати 150–200 років, а токсичні метали залишаються активними в навколишньому середовищі ще довше.

Вирішення проблем токсичності потребує розробки нових матеріалів, впровадження екологічних стандартів виробництва та створення ефективних систем переробки електронних відходів.

Green IT як відповідь на виклики техногенного розвитку

Попри складність зазначених проблем, екологічно орієнтовані підходи в сфері Green IT демонструють значний потенціал для зменшення негативного впливу комп'ютерних технологій. На рівні підприємств застосовуються такі технічні рішення, як:

- оптимізація енергоспоживання серверних систем;
- застосування модульних систем охолодження;
- перехід до енергоефективних обчислювальних архітектур;
- збільшення строку служби обладнання;
- запровадження політик раціональної утилізації.

Важливо також враховувати соціальну складову: популяризацію екологічно відповідальної поведінки серед користувачів, підвищення рівня поінформованості про ризики електронних відходів та необхідність відповідального споживання технологій.

Технологічні ініціативи для зниження енергоспоживання у центрах обробки даних

Три ключові технологічні напрями, що забезпечують зниження енергоспоживання центрів обробки даних:

– *Віртуалізація та консолідація*

Забезпечує скорочення кількості фізичних серверів шляхом об'єднання їх функціональності в одному обладнанні. Це дозволяє суттєво зменшити споживання електроенергії та витрати на охолодження.

– *Управління життєвим циклом інформації*

Передбачає розподіл даних за рівнями важливості та їх автоматичне переміщення між типами сховищ залежно від етапу життєвого циклу. Це дає змогу використовувати менш енергоємні ресурси для даних, що втратили актуальність.

– *Дедуплікація даних*

Дозволяє зменшити обсяг збережених даних до 200–300 разів шляхом усунення дубльованих фрагментів інформації. Це оптимізує використання простору на серверах, прискорює резервне копіювання та знижує навантаження на мережеву інфраструктуру.

Міжнародна співпраця та шляхи мінімізації екологічного впливу

Створення екологічно стійкої системи виробництва та споживання комп'ютерних технологій потребує широкої міжнародної кооперації. До розв'язання екологічних проблем мають бути залучені:

- міжнародні організації (ООН, ЮНЕП тощо);
- урядові структури;
- приватні технологічні корпорації;
- університети та науково-дослідні установи;

- громадські організації;
- кінцеві користувачі.

Необхідне формування інтегрованого підходу, який враховує взаємозв'язок між виробництвом енергії, її споживанням, змінами клімату та сталим розвитком. Такий підхід має включати розробку міжнародних стандартів, економічні стимули, державні програми підтримки інновацій, інвестиції у наукові дослідження та екологічну просвіту населення.

6.3 Висновки до розділу 6

У розділі розглянуто основні положення охорони навколишнього природного середовища, а також проаналізовано характер і масштаби впливу комп'ютерної техніки на екологічний стан довкілля. Встановлено, що система екологічної безпеки ґрунтується на принципах раціонального природокористування, запобігання забрудненню та забезпечення сталого розвитку, а також регулюється чинним законодавством України у сфері охорони довкілля.

Показано, що хоча сучасні інформаційні технології не належать до найбільш небезпечних галузей, їх використання супроводжується низкою екологічних ризиків. Серед ключових негативних чинників визначено утворення електронних відходів, споживання електроенергії, генерацію теплових викидів, використання хімічних компонентів у виробництві та утилізації обладнання. За умови неналежного поводження електронні відходи можуть становити значну загрозу для ґрунтів, водних ресурсів та здоров'я населення.

Проаналізовані технічні, організаційні та технологічні заходи засвідчили, що зменшення екологічного навантаження від комп'ютерної техніки є досяжним за рахунок упровадження енергоефективних пристроїв, оптимізації режимів роботи обладнання, забезпечення правильної утилізації

електронних відходів, а також переходу до сучасних екологічних стандартів виробництва та експлуатації техніки.

Загалом розділ демонструє, що забезпечення екологічної безпеки у сфері використання комп'ютерної техніки потребує комплексного підходу, який поєднує законодавче регулювання, технологічні рішення, належну організацію процесів та відповідальне ставлення користувачів. Реалізація таких заходів сприяє зниженню негативного впливу на довкілля, підтриманню стійкого екологічного балансу та формуванню безпечних умов життєдіяльності.

ВИСНОВКИ

Мета, яка ставилася в кваліфікаційній роботі, а саме, підвищення достовірності оцінювання якості ПС, які базуються на штучному інтелекті, шляхом розробки спеціалізованої математичної моделі, досягнута. Для досягнення мети були вирішені наступні завдання:

- Проведено аналіз існуючих підходів до оцінювання якості ПС, що дозволило систематизувати основні методи та виявити обмеження традиційних підходів при застосуванні до ПС, які базуються на штучному інтелекті.

- Виконано збір емпіричних даних для побудови нелінійної регресійної моделі оцінювання метрики RFC через метрики CBO та WMC ПС, які базуються на штучному інтелекті.

- Здійснено перевірку зібраних даних на відповідність нормальному розподілу, що забезпечило коректність подальшого статистичного аналізу та моделювання.

- Удосконалено нелінійну регресійну модель оцінювання метрики RFC ПС, які базуються на штучному інтелекті. Запропонована модель враховує взаємозв'язок між метриками CBO та WMC та забезпечує підвищення точності й обґрунтованості результатів оцінювання.

- Розроблено програмну систему для підтримки прийняття рішень щодо якості ПС, що базуються на штучному інтелекті, яка автоматизує процеси введення вихідних даних, оцінювання метрики RFC, формування довірчих та прогнозних інтервалів оцінки RFC, визначення рівня якості ПС, які базуються на штучному інтелекті, з використанням отриманих інтервалів.

- Розроблено розділи з економіки, охорони праці, охорони навколишнього середовища.

Отримані результати можуть бути використані в практиці оцінювання якості ПС, які базуються на штучному інтелекті, та слугувати основою для подальших досліджень у напрямі вдосконалення методів оцінювання якості ПС.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Chidamber S.R. Towards a Metrics Suite for Object Oriented Design / S.R. Chidamber, C.F. Kemerer // Proc. of the 6th ACM Conference on Object Oriented Programming, Systems, Languages and Applications (OOPSLA), 1991, Phoenix, AZ, pp. 197-211.
2. Basili V.R. A Validation of Object-Oriented Design Metrics as Quality Indicators / V.R. Basili, L.C. Briand, W.L. Melo // IEEE Transactions on Software
3. Laing V. Principal Components of Orthogonal Object-Oriented Metrics / V. Laing, C. Coleman // White Paper Analyzing Results of NASA Object-Oriented Data (Task 323-08-14). – Greenbelt, Maryland (US): NASA Goddard Space Flight Center, the Software Assurance Technology Center (SATC), 2001.
4. Prykhodko S. A modified technique for constructing nonlinear regression models based on the multivariate normalizing transformations / S. Prykhodko, N. Prykhodko // Selected Papers of the VIII International Scientific Conference “Information Technology and Implementation” (IT&I-2021). Workshop Proceedings (IT&I-WS-2021), Kyiv, Ukraine, December 1-3, 2021. CEUR Workshop Proceedings, 3179, 156-166.
5. Prykhodko S.A. Technique for Detecting Software Quality Based on the Confidence and Prediction Intervals of Nonlinear Regression for RFC Metric / S. Prykhodko, N. Prykhodko // Proceedings of the 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), 2022. – pp. 499-502.
6. Приходько С.Б. Нелінійна регресійна модель для оцінювання метрики RFC застосунків з відкритим кодом на Kotlin / С.Б.Приходько, А.В. Кольцов, Є.В. Грабовський // Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2023): Матеріали IV-ої Всеукраїнської науково-практичної інтернет конференції (30-31 жовтня 2023 р.). – Миколаїв: НУК імені адмірала Макарова, 2023. – С. 20-21.

7. Приходько С.Б. Нелінійна регресійна модель для оцінювання метрики RFC Застосунків для онлайн-покупок з відкритим кодом на Java / С.Б. Приходько, Г.О. Скакунов // Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2024): Матеріали V-ої Всеукраїнської науково-практичної інтернет конференції (30-31 жовтня 2024 р.). – Миколаїв: НУК імені адмірала Макарова, 2024. – С. 59-60.
8. Приходько С.Б. Нелінійна регресійна модель для оцінювання метрики RFC застосунків, що створюються на платформі Flutter / С.Б. Приходько, Н.О. Шебалін // Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2024): Матеріали V-ої Всеукраїнської науково-практичної інтернет конференції (30-31 жовтня 2024 р.). – Миколаїв: НУК імені адмірала Макарова, 2024. – С. 43-44.
9. Латанська Л.О. Оцінювання якості програмних систем, що створюються з використанням штучного інтелекту / Л.О. Латанська, А.М. Корнієнко, Д.О. Шалапко // Матеріали VIII Всеукраїнської науково-практичної інтернет-конференції студентів, аспірантів та молодих вчених за тематикою «Сучасні комп'ютерні системи та мережі в управлінні» (24 листопада 2025 року, м. Херсон, м. Хмельницький) / за ред. А.А. Григорової. – Херсон: Видавництво ФОП Вишемирський В.С., 2025. – С.156-157.
<https://kntu.net.ua/ukr/content/download/119322/668093/file/!CICT%202024.pdf>
10. ISO/IEC/IEEE 25000:2014 SQuaRE -- Guide to SQuaRE. URL: https://www.omgwiki.org/dido/doku.php?id=dido%3Apublic%3Ara%3Axapend%3Axapend.b_stds%3Atech%3Aiso%3Asquare_guide&utm_source=chatgpt.com
 (дата звернення: 05.12.2025)
11. Хіль О.С. Аналіз проблеми застосування методів машинного навчання для оцінювання та прогнозування дефектів програмного забезпечення / О.С. Хіль, В.С. Яковина // Scientific Bulletin of UNFU. 2022. Т. 32, № 3. С. 106-116. DOI: <https://doi.org/10.36930/40330316>
12. Sagarna R. Software Defect Prediction Based on Machine Learning and Deep Learning Techniques: An Empirical Approach / R. Sagarna, M. Ndiaye, E. Abdelwahed // AI. 2024. Vol. 3, No. 4. Article 86. <https://doi.org/10.3390/ai5040086>

13. Яковина В. С. Прогнозування дефектів програмного забезпечення ансамблем нейронних мереж / В.С. Яковина, І. І. Симець // Scientific Bulletin of UNFU. 2021. Т. 31, № 6. С. 238-247. DOI: <https://doi.org/10.36930/40310616>
14. Dam H. K. A Deep Tree-Based Model for Software Defect Prediction / H. K. Dam, T. Pham, S. W. Ng, T. Tran, J. Grundy, A. Ghose et al. // arXiv preprint. 2018. arXiv:1802.00921. URL: <https://arxiv.org/abs/1802.00921> (дата звернення: 05.12.2025)
15. Abu Alhija H. Software Defect Prediction Using Support Vector Machine / Alhija H. Abu, M. Azzeh, F. Almasalha //arXiv preprint. 2022. arXiv:2209.14299. URL: <https://arxiv.org/abs/2209.14299> (дата звернення: 05.12.2025)
16. Jensen F. Bayesian Networks and Decision Graphs. New York: Springer, 2001. 268 p. URL: https://link.springer.com/book/10.1007/978-0-387-68282-2?utm_source=chatgpt.com) (дата звернення: 06.12.2025)
17. CMMI Product Team. CMMI for Development, Version 1.3. CMU/SEI-2010-TR-033. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2010. 482 p. URL: https://se.inf.ethz.ch/courses/2013b_fall/dsl/slides/reading_material/cmmi-for-development_13.pdf (дата звернення: 06.12.2025)
18. Family ISO/IEC 33000. URL: <https://www.iso33000.es/index.php/familia-iso-33000/2-iso-iec-33000> (дата звернення: 05.12.2025)
19. Приходько С.Б. Методичні вказівки та завдання до виконання лабораторних робіт з дисципліни «Емпіричні методи програмної інженерії» / С.Б. Приходько, Л.М. Макарова, Н.В. Приходько, А.В. Пухалевич. Миколаїв: НУК, 2023. 56 с.
20. Mardia K. V. Measures of multivariate skewness and kurtosis with applications. Biometrika. 1970. Vol. 57. P. 519–530. <https://doi.org/10.1093/biomet/57.3.519>.
21. Закон України “Про охорону праці”. – Київ: Верховна Рада України, 1992. – № 2694-XII. URL: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення: 07.12.2025).
22. Кодекс законів про працю України (КЗпП). – Київ: ВРУ, 1971. URL: <https://zakon.rada.gov.ua/laws/show/322-08> (дата звернення: 07.12.2025).

ДОДАТОК А

Технічне завдання

Вступ

Найменування програмної системи – програмна система для прийняття рішень про якість ПС, які базуються на штучному інтелекті

1 Підстави для розробки

Підставою для розробки програмної системи є завдання на кваліфікаційну (магістерську) роботу «Математична модель щодо прийняття рішень про якість ПС, які базуються на штучному інтелекті, та програма для її реалізації», видане кафедрою програмного забезпечення автоматизованих систем навчально-наукового інституту комп'ютерних наук та управління проєктами Національного університету кораблебудування імені адмірала Макарова.

2 Призначення розробки

2.1 Функціональне призначення

Функціональним призначенням системи є надання користувачу можливості визначити рівень якості ПС, які базуються на штучному інтелекті.

2.2 Експлуатаційне призначення

Програмна система розробляється для спрощення прийняття рішень про якість ПС, які базуються на штучному інтелекті, за рахунок реалізації розробленої математичної моделі.

3 Вимоги до програмної системи

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

Програмна система повинна надавати наступні функціональні можливості:

- Ввести значення метрики СВО для ПС, що базується на штучному інтелекті.
- Ввести значення метрики WMC для ПС, що базується на штучному інтелекті.
- Ввести фактичне значення метрики RFC для ПС, що базується на штучному інтелекті.
- Ввести значення довірчої ймовірності для розрахунку інтервалів.
- Перевірити вихідні дані на коректність.
- Оцінити значення метрики RFC для ПС, що базується на штучному інтелекті.
- Обчислити довірчий інтервал оцінки значення метрики RFC для ПС, що базується на штучному інтелекті.
- Обчислити прогнозний інтервал оцінки значення метрики RFC для ПС, що базується на штучному інтелекті.
- Визначити рівень якості ПС, що базується на штучному інтелекті, за результатами порівняння фактичного значення метрики RFC з отриманими інтервалами.

3.1.2 Вимоги для організації вихідних та результуючих даних

Задання вихідних даних здійснюється шляхом їх внесення у відповідну інтерфейсну форму програми.

Вихідною інформацією є введені значення метрик СВО, WMC, RFC ПС, що базуються на штучному інтелекті, та довірча ймовірність.

Виведення результатів повинно бути реалізовано у спеціально відведених місцях інтерфейсу програми.

Результуючими даними програмної системи є: оцінене значення метрики RFC для ПС, що базується на штучному інтелекті; довірчий інтервал оцінки значення метрики RFC для ПС, що базується на штучному інтелекті; прогнозний інтервал оцінки значення метрики RFC для ПС, що базується на штучному інтелекті; рівень якості ПС, що базується на штучному інтелекті.

3.2 Вимоги до надійності

Програмна система повинна нормально функціонувати при безперебійній роботі ПК. При виникненні збоїв в роботі, відновлення нормальної роботи повинне проводитися після перезавантаження програми.

3.3 Вимоги до складу та параметрів технічних засобів

Для коректної роботи програмної системи персональний комп'ютер (або інший пристрій) користувача повинен відповідати наступним мінімальним вимогам:

- Центральний процесор (CPU): Будь-який сучасний процесор із тактовою частотою від 1 ГГц (наприклад, Intel Core i3, AMD Ryzen 3 або аналогічні).
- Оперативна пам'ять (RAM): Мінімум 2 ГБ (рекомендовано 4 ГБ або більше для стабільної роботи операційної системи та веб-браузера).
- Накопичувач на жорсткому диску (HDD/SSD): Для розміщення файлів системи потрібно менше 1 МБ вільного простору.
- Пристрої вводу-виводу: Клавіатура, комп'ютерна миша (або тачпад), монітор із роздільною здатністю екрана не менше 1280x720 пікселів.
- Мережеве підключення: Не є обов'язковим (система працює локально без доступу до мережі Інтернет).

3.4 Умови експлуатації

Кліматичні умови експлуатації, за яких забезпечується дотримання заданих характеристик, мають відповідати нормативним вимогам, встановленим для технічних засобів з урахуванням умов їх застосування.

3.5 Вимоги до інформаційної та програмної сумісності

Програмна система реалізована з використанням стандартних веб-технологій (HTML5, CSS3, JavaScript ES6+), що забезпечує її кросплатформеність.

Для запуску та експлуатації необхідні наступні програмні засоби:

– Операційна система:

- Windows (7, 8, 10, 11);
- macOS (версії 10.12 Sierra і новіше);
- Linux (будь-який дистрибутив із графічним інтерфейсом);
- Android або iOS (за умови наявності сумісного браузера).

– Веб-браузер: Необхідний будь-який сучасний веб-браузер з підтримкою стандарту ECMAScript 2015 (ES6). Рекомендовані варіанти:

- Google Chrome (версії 60+);
- Mozilla Firefox (версії 60+);
- Microsoft Edge (на базі Chromium);
- Safari (версії 11+);
- Opera.

Додаткові бібліотеки: Встановлення Node.js, Python або інших середовищ виконання не вимагається. Програмна система працює автономно.

3.6 Вимоги до захисту інформації та програм

Відсутні.

3.7 Вимоги до маркування та пакування

Відсутні.

3.8 Вимоги до транспортування та зберігання

Відсутні.

4 Вимоги до програмної документації

Програмна документація має містити наступні документи:

- технічне завдання;
- опис програмної системи;
- текст програми;
- інструкцію користувача;
- програму та методику випробувань програмної системи.

5 Техніко-економічні показники

Розрахунок економічної ефективності впровадження програмної системи наведено у розділі 4. Отримані результати свідчать про доцільність впровадження розробленої програмної системи, при цьому орієнтовний строк її окупності становить близько 8,9 місяців.

6 Стадії та етапи розробки системи

Стадії та етапи виконання розробки програмної системи можна представити наступним чином (табл. А.1):

Таблиця А.1 – Стадії та етапи розробки програмної системи

Стадії розробки	Етапи робіт	Термін виконання робіт	
		Початок етапу	Кінець етапу
1	2	3	4
1 Технічне завдання	1.1 Обґрунтування необхідності розробки програмної системи	6.09.2025	11.09.2025
	1.2 Розробка технічного завдання	11.09.2025	16.09.2025
	1.3 Затвердження технічного завдання	16.09.2025	28.09.2025

Кінець табл. А.2

1	2	3	4
2 Загальносистемні рішення	2.1 Розробка загальносистемних рішень	28.09.2025	08.10.2025
	2.2 Затвердження загальносистемних рішень	08.10.2025	15.10.2025
3 Рішення з інформаційного забезпечення	2.1 Розробка рішень з інформаційного забезпечення	15.10.2025	21.10.2025
	2.2 Затвердження рішення з інформаційного забезпечення	21.10.2025	01.11.2025
4 Рішення з програмної системи	4.1 Розробка рішень з програмної системи	01.11.2025	07.11.2025
	4.2 Затвердження рішення з програмної системи	07.11.2025	15.11.2025
	4.3 Розробка програмної системи	15.11.2025	4.12.2025

7 Порядок контролю та приймання

Контроль виконання робіт з аналізу та проектування окремих частин програмної системи для підтримки прийняття рішень щодо якості ПС, що базуються на штучному інтелекті, здійснюється на всіх етапах життєвого циклу розробки відповідно до вимог, визначених у технічному завданні. Кожна стадія розробки підлягає виконанню у встановлені терміни та обов'язковому погодженню із замовником.

Приймання готової системи документують за допомогою протоколу проведення випробувань. У разі виявлення невідповідностей або помилок у процесі приймання, складається акт про виявлені недоліки, який підписується уповноваженими представниками замовника та розробника і затверджується керівниками відповідних організацій. Розробник зобов'язаний усунути зазначені недоліки у строк, що не перевищує двох тижнів з моменту затвердження акта, та повідомити замовника про готовність програмної системи до повторного проведення перевірки.

ДОДАТОК Б

Інструкція користувача

1 Призначення програми

1.1 Функціональне призначення

Функціональним призначенням програмної системи є надання користувачу можливості визначити рівень якості ПС, які базуються на штучному інтелекті.

1.2 Експлуатаційне призначення

Програмна система розробляється для спрощення прийняття рішень про якість ПС, які базуються на штучному інтелекті, за рахунок реалізації розробленої математичної моделі.

1.3 Склад функцій

Програмна система повинна надавати наступні функціональні можливості:

- Ввести значення метрики СВО для ПС, що базується на штучному інтелекті.
- Ввести значення метрики WMC для ПС, що базується на штучному інтелекті.
- Ввести фактичне значення метрики RFC для ПС, що базується на штучному інтелекті.
- Ввести значення довірчої ймовірності для розрахунку інтервалів.
- Перевірити вихідні дані на коректність.
- Оцінити значення метрики RFC для ПС, що базується на штучному інтелекті.

- Обчислити довірчий інтервал оцінки значення метрики RFC для ПС, що базується на штучному інтелекті.
- Обчислити прогнозний інтервал оцінки значення метрики RFC для ПС, що базується на штучному інтелекті.
- Визначити рівень якості ПС, що базується на штучному інтелекті, за результатами порівняння фактичного значення метрики RFC з отриманими інтервалами.

2 Умови виконання програми

2.1 Мінімальний склад апаратних засобів

Для коректної роботи програмної системи персональний комп'ютер (або інший пристрій) користувача повинен відповідати наступним мінімальним вимогам:

- Центральний процесор (CPU): Будь-який сучасний процесор із тактовою частотою від 1 ГГц (наприклад, Intel Core i3, AMD Ryzen 3 або аналогічні).
- Оперативна пам'ять (RAM): Мінімум 2 ГБ (рекомендовано 4 ГБ або більше для стабільної роботи операційної системи та веб-браузера).
- Накопичувач на жорсткому диску (HDD/SSD): Для розміщення файлів програми потрібно менше 1 МБ вільного простору.
- Пристрої вводу-виводу: Клавіатура, комп'ютерна миша (або тачпад), монітор із роздільною здатністю екрана не менше 1280x720 пікселів.
- Мережеве підключення: Не є обов'язковим (система працює локально без доступу до мережі Інтернет).

2.2 Мінімальний склад програмних засобів

Програмну систему реалізовано з використанням стандартних веб-технологій (HTML5, CSS3, JavaScript ES6+), що забезпечує її кросплатформеність.

Для запуску та експлуатації необхідні наступні програмні засоби:

– Операційна система:

- Windows (7, 8, 10, 11);
- macOS (версії 10.12 Sierra і новіше);
- Linux (будь-який дистрибутив із графічним інтерфейсом);
- Android або iOS (за умови наявності сумісного браузера).

– Веб-браузер: Необхідний будь-який сучасний веб-браузер з підтримкою стандарту ECMAScript 2015 (ES6). Рекомендовані варіанти:

- Google Chrome (версії 60+);
- Mozilla Firefox (версії 60+);
- Microsoft Edge (на базі Chromium);
- Safari (версії 11+);
- Opera.

Додаткові бібліотеки: Встановлення Node.js, Python або інших середовищ виконання не вимагається. Програмна система працює автономно.

2.3 Виконання програмної системи

Для початку роботи з програмною системою необхідно виконати такі дії:

– Здійснити запуск програмної системи для прийняття рішень про якість ПС, які базуються на штучному інтелекті. Після чого на екрані відобразиться відповідна форма інтерфейсу, наведена на рисунку В1.

**Прийняття рішень щодо якості ПС, що базуються
на штучному інтелекті**

Вхідні параметри

Ввести СВО для ПС із ШІ:

Наприклад: 4,7

Ввести WMC для ПС із ШІ:

Наприклад: 15,39

Ввести RFC для ПС із ШІ:

Наприклад: 22,89

Ввести довірчу ймовірність:

0.95

Оцінити

Рисунок В.1 – Інтерфейс після запуску програмної системи для прийняття рішень про якість ПС, які базуються на штучному інтелекті

- Ввести значення СВО для ПС, що базується на штучному інтелекті.
- Ввести значення WMC для ПС, що базується на штучному інтелекті.
- Ввести фактичне значення RFC для ПС, що базується на штучному інтелекті.
- Ввести значення довірчої ймовірності для розрахунку інтервалів.

Інтерфейс після введення вихідних даних для прийняття рішення щодо якості ПС, що базуються на штучному інтелекті, наведений на рисунку В2.

**Прийняття рішень щодо якості ПС, що базуються
на штучному інтелекті**

Вхідні параметри

Ввести СВО для ПС із ШІ:

Ввести WMC для ПС із ШІ:

Ввести RFC для ПС із ШІ:

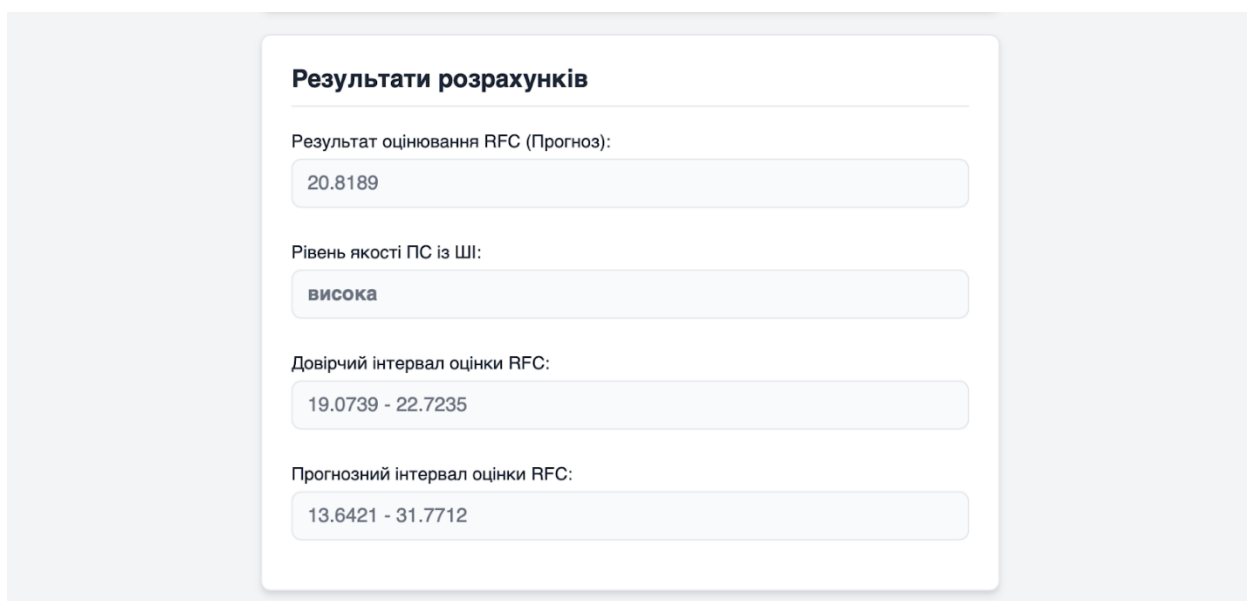
Ввести довірчу ймовірність:

Оцінити

Рисунок В2 – Інтерфейс після введення даних для прийняття рішення щодо якості ПС, що базуються на штучному інтелекті

– Після введення вихідних даних натиснути кнопку «Оцінити» для отримання: оцінки значення метрики RFC, меж довірчого інтервалу для оцінки значення метрики RFC, меж прогнозного інтервалу для оцінки значення метрики RFC, рівня якості ПС, що базується на штучному інтелекті.

Інтерфейс програмної системи на етапі перегляду результатів оцінювання ПС, що базуються на штучному інтелекті, наведений на рисунку В3.



The image shows a software interface titled "Результати розрахунків" (Results of calculations). It displays four rows of data, each with a label and a corresponding value in a light gray box:

Label	Value
Результат оцінювання RFC (Прогноз):	20.8189
Рівень якості ПС із ШІ:	висока
Довірчий інтервал оцінки RFC:	19.0739 - 22.7235
Прогнозний інтервал оцінки RFC:	13.6421 - 31.7712

Рисунок В.3 – Інтерфейс програмної системи на етапі перегляду результатів оцінювання ПС, що базуються на штучному інтелекті

ДОДАТОК В

Програма та методика випробувань

1 Об'єкт випробування

Об'єктом випробування є програмна система для прийняття рішень про якість ПС, які базуються на штучному інтелекті.

2 Мета випробувань

Метою проведення випробувань є оцінка працездатності програмної системи та перевірка відповідності її характеристик і функціональних можливостей вимогам, визначеним у технічному завданні.

3 Вимоги до програмної системи

Під час проведення випробувань здійснюється перевірка функціональних можливостей програмної системи на відповідність вимогам, визначеним у розділі «Вимоги до функціональних характеристик» технічного завдання.

Програмна система повинна надавати наступні функціональні можливості:

- Ввести значення метрики СВО для ПС, що базується на штучному інтелекті.
- Ввести значення метрики WMC для ПС, що базується на штучному інтелекті.
- Ввести фактичне значення метрики RFC для ПС, що базується на штучному інтелекті.
- Ввести значення довірчої ймовірності для розрахунку інтервалів.
- Перевірити вихідні дані на коректність.
- Оцінити значення метрики RFC для ПС, що базується на штучному інтелекті.

- Обчислити довірчий інтервал оцінки значення метрики RFC для ПС, що базується на штучному інтелекті.
- Обчислити прогнозний інтервал оцінки значення метрики RFC для ПС, що базується на штучному інтелекті.
- Визначити рівень якості ПС, що базується на штучному інтелекті, за результатами порівняння фактичного значення метрики RFC з отриманими інтервалами.

4 Вимоги до програмної документації

Програмна документація має містити наступні документи:

- технічне завдання;
- опис програмної системи;
- текст програмної системи;
- інструкцію користувача;
- програму та методику випробувань програмної системи.

5 Засоби та порядок випробувань

Склад технічних засобів, що використовувалися під час випробувань

Для коректної роботи програмної системи персональний комп'ютер (або інший пристрій) користувача повинен відповідати наступним мінімальним вимогам:

- Центральний процесор (CPU): Будь-який сучасний процесор із тактовою частотою від 1 ГГц (наприклад, Intel Core i3, AMD Ryzen 3 або аналогічні).
- Оперативна пам'ять (RAM): Мінімум 2 ГБ (рекомендовано 4 ГБ або більше для стабільної роботи операційної системи та веб-браузера).
- Накопичувач на жорсткому диску (HDD/SSD): Для розміщення файлів програми потрібно менше 1 МБ вільного простору.
- Пристрої вводу-виводу: Клавіатура, комп'ютерна миша (або тачпад), монітор із роздільною здатністю екрана не менше 1280x720 пікселів.

– Мережеве підключення: Не є обов'язковим (програма працює локально без доступу до мережі Інтернет).

Склад програмних засобів, що використовувалися під час випробувань:

Програмну систему реалізовано з використанням стандартних веб-технологій (HTML5, CSS3, JavaScript ES6+), що забезпечує його кросплатформеність.

Для запуску та експлуатації необхідні наступні програмні засоби:

– Операційна система:

- Windows (7, 8, 10, 11);
- macOS (версії 10.12 Sierra і новіше);
- Linux (будь-який дистрибутив із графічним інтерфейсом);
- Android або iOS (за умови наявності сумісного браузера).

– Веб-браузер: Необхідний будь-який сучасний веб-браузер з підтримкою стандарту ECMAScript 2015 (ES6). Рекомендовані варіанти:

- Google Chrome (версії 60+);
- Mozilla Firefox (версії 60+);
- Microsoft Edge (на базі Chromium);
- Safari (версії 11+);
- Opera.

Додаткові бібліотеки: Встановлення Node.js, Python або інших середовищ виконання не вимагається. Програма працює автономно

Порядок проведення випробувань

Порядок проведення випробувань складається з перевірки вимог до функціональних характеристик програмної системи та перевірки вимог до програмної документації.

6 Методи випробувань

Випробування програмної системи проводяться за методикою «чорної скриньки». Перевіряються всі функціональні можливості програмної системи

на відповідність вимогам, зазначеним у розділі «Вимоги до функціональних характеристик» технічного завдання. Випробування вважаються успішно завершеними у разі повної відповідності виконуваних функцій встановленим вимогам.

Порядок проведення випробувань включає наступні етапи:

- Перевірка на коректне введення даних, а саме: CBO, WMC, RFC, довірчої ймовірності.
- Перевірка реакції на введення некоректних даних.
- Перевірка можливості виконання оцінювання метрики RFC на основі введених значень метрик CBO та WMC із застосуванням побудованої нелінійної регресійної моделі.
- Перевірка можливості визначення верхньої та нижньої меж довірчого інтервалу для оцінювання значення RFC.
- Перевірка можливості визначення верхньої та нижньої меж прогнозного інтервалу для оцінювання значення RFC.
- Перевірка можливості визначення рівня якості ПС, що базується на штучному інтелекті, за результатами порівняння фактичного значення метрики RFC з отриманими інтервалами.

ДОДАТОК Г

Текст програми

```

class MathUtils {
  static t_ppf(p, df) {
    const a1 = -39.6968302866538,
      a2 = 220.946098424521,
      a3 = -275.928510446969;
    const a4 = 138.357751867269,
      a5 = -30.6647980661472,
      a6 = 2.50662827745924;
    const b1 = -54.4760987982241,
      b2 = 161.585836858041,
      b3 = -155.698979859887;
    const b4 = 66.8013118877197,
      b5 = -13.2806815528857;
    const c1 = -7.78489400243029e-3,
      c2 = -3.22396458041136e-1,
      c3 = -2.40075827716184;
    const c4 = -2.54973253934373,
      c5 = 4.37466414146497,
      c6 = 2.93816398269878;
    const d1 = 7.78469570904146e-3,
      d2 = 3.2246712907004e-1,
      d3 = 2.445134137143;
    const d4 = 3.75440866190742;
    const p_low = 0.02425,
      p_high = 1 - p_low;

    let q, r, z;
    if (p < p_low) {
      q = Math.sqrt(-2 * Math.log(p));
      z =
        (((((c1 * q + c2) * q + c3) * q + c4) * q + c5) * q + c6) /
        (((d1 * q + d2) * q + d3) * q + d4) * q + 1);
    }
  }
}

```

```

} else if (p <= p_high) {
  q = p - 0.5;
  r = q * q;
  z =
    ((((((a1 * r + a2) * r + a3) * r + a4) * r + a5) * r + a6) * q) /
    (((((b1 * r + b2) * r + b3) * r + b4) * r + b5) * r + 1);
} else {
  q = Math.sqrt(-2 * Math.log(1 - p));
  z =
    -((((((c1 * q + c2) * q + c3) * q + c4) * q + c5) * q + c6) /
    (((((d1 * q + d2) * q + d3) * q + d4) * q + 1);
}

return (
  z +
  (Math.pow(z, 3) + z) / (4 * df) +
  (5 * Math.pow(z, 5) + 16 * Math.pow(z, 3) + 3 * z) / (96 * df * df)
);
}

static matmul(a, b) {
  if (Array.isArray(a) && Array.isArray(b) && Array.isArray(b[0])) {
    const result = new Array(b[0].length).fill(0);
    for (let j = 0; j < b[0].length; j++) {
      for (let i = 0; i < a.length; i++) {
        result[j] += a[i] * b[i][j];
      }
    }
    return result;
  }
  if (Array.isArray(a) && Array.isArray(b) && !Array.isArray(b[0])) {
    let sum = 0;
    for (let i = 0; i < a.length; i++) {
      sum += a[i] * b[i];
    }
    return sum;
  }
}

```

```

    throw new Error("Непідтримувані типи для matmul");
  }
}

class RFCEvaluator {
  constructor() {
    this.N = 42;
    this.k = 2;
    this.V = this.N - this.k - 1; // 39

    this.B = [-0.000161884, -0.016315048, 1.119897052];

    this.SZY = 0.088792859;

    this.ICM = [
      [0.807835294, -0.429525623],
      [-0.429525623, 0.979619057],
    ];

    this.ZXMeans = [0.540000629, 1.027326384];
  }

  predictRFC(X) {
    return 10 ** this.B[0] * X[0] ** this.B[1] * X[1] ** this.B[2];
  }

  getSEZ(X) {
    const zx = [
      Math.log10(X[0]) - this.ZXMeans[0],
      Math.log10(X[1]) - this.ZXMeans[1],
    ];
    const tempVector = MathUtils.matmul(zx, this.ICM);
    return MathUtils.matmul(tempVector, zx);
  }

  calcPredInterval(value, X, alpha) {
    const Y = Math.log10(value);

```

```

const sez = this.getSEZ(X);
const tVal = MathUtils.t_ppf(1 - alpha / 2, this.V);
const predHalfIntervalValue =
  tVal * this.SZY * Math.sqrt(1 + 1.0 / this.N + sez);
return [
  10 ** (Y - predHalfIntervalValue),
  10 ** (Y + predHalfIntervalValue),
];
}

```

```

calcConfInterval(value, X, alpha) {
  const Y = Math.log10(value);
  const sez = this.getSEZ(X);
  const tVal = MathUtils.t_ppf(1 - alpha / 2, this.V);
  const confHalfIntervalValue =
    tVal * this.SZY * Math.sqrt(1.0 / this.N + sez);
  return [
    10 ** (Y - confHalfIntervalValue),
    10 ** (Y + confHalfIntervalValue),
  ];
}
}

```

```

class UIController {
  constructor() {
    this.evaluator = new RFCEvaluator();

    // Inputs
    this.form = document.getElementById("calc-form");
    this.cboInput = document.getElementById("cbo-input");
    this.wmcInput = document.getElementById("wmc-input");
    this.rfcInput = document.getElementById("rfc-input"); // Нове поле
    this.confInput = document.getElementById("confidence-input");

    // Outputs
    this.rfcOutput = document.getElementById("rfc-result");
    this.qualityOutput = document.getElementById("quality-result");
  }
}

```

```
this.confOutput = document.getElementById("conf-interval-result");
this.predOutput = document.getElementById("pred-interval-result");
```

```
this.form.addEventListener("submit", (e) => {
  e.preventDefault();
  this.calculate();
});
}
```

```
calculate() {
  // Parsing
  const cbo = parseFloat(this.cboInput.value.replace(",", "."));
  const wmc = parseFloat(this.wmcInput.value.replace(",", "."));
  const inputRFC = parseFloat(this.rfcInput.value.replace(",", "."));
  const conf = parseFloat(this.confInput.value.replace(",", "."));

  if (isNaN(cbo) || isNaN(wmc) || isNaN(inputRFC) || isNaN(conf)) {
    alert("Будь ласка, перевірте правильність введених даних.");
    return;
  }
}
```

```
const X = [cbo, wmc];
const alpha = 1 - conf;
```

```
// Оцінюємо значення RFC на основі CBO та WMC
const predictedValue = this.evaluator.predictRFC(X);
```

```
// Знаходимо інтервали навколо оцінки RFC
const predInterval = this.evaluator.calcPredInterval(
  predictedValue,
  X,
  alpha
);
const confInterval = this.evaluator.calcConfInterval(
  predictedValue,
  X,
  alpha
);
```

```

);

// Визначаємо якість
let quality = "невизначена";

if (inputRFC >= confInterval[0] && inputRFC <= confInterval[1]) {
  quality = "середня";
} else if (
  (inputRFC >= predInterval[0] && inputRFC <= confInterval[0]) ||
  (inputRFC >= confInterval[1] && inputRFC <= predInterval[1])
) {
  quality = "висока";
} else if (inputRFC >= predInterval[1]) {
  quality = "низька";
}

// Rendering
this.rfcOutput.value = predictedValue.toFixed(4);
this.qualityOutput.value = quality;
this.confOutput.value = `${confInterval[0].toFixed(
  4
)} - ${confInterval[1].toFixed(4)}`;
this.predOutput.value = `${predInterval[0].toFixed(
  4
)} - ${predInterval[1].toFixed(4)}`;
}
}

// Start
document.addEventListener("DOMContentLoaded", () => {
  new UIController();
});

```

ДОДАТОК Д

Опис програмної системи

1 Загальні відомості

1.1 Позначення і найменування програмної системи

Найменування: програмна система для прийняття рішень про якість ПС, які базуються на штучному інтелекті

1.2 Мови програмування

Для розроблення програмної системи було використано сучасні вебтехнології, які забезпечують гнучкість, масштабованість та зручність у подальшій експлуатації системи, а саме: HTML5, CSS3, JavaScript (ES6+).

Основою реалізації користувацького інтерфейсу є мова гіпертекстової розмітки HTML5, яка застосовується для створення семантично коректної структури вебзастосунку. Використання стандартів HTML5 сприяє підвищенню доступності, зручності навігації та сумісності з різними браузерами.

Для оформлення зовнішнього вигляду та адаптації інтерфейсу під різні розміри екранів використано CSS3. Застосування сучасних можливостей цієї технології, зокрема Flexbox та змінних CSS, дозволяє реалізувати адаптивний дизайн і забезпечити коректне відображення елементів інтерфейсу на різних пристроях.

Реалізація бізнес-логіки програмної системи, обробка подій користувача, а також виконання математичних обчислень здійснюються за допомогою мови програмування JavaScript (ES6+). Використання сучасних стандартів JavaScript забезпечує підвищену продуктивність, читабельність коду та спрощує подальшу підтримку і розширення функціональних можливостей системи.

2 Функціональне призначення

2.1 Класи розв'язуваних завдань та призначення програмної системи

Програмна система для прийняття рішень про якість ПС, які базуються на штучному інтелекті, повинна забезпечувати виконання наступних функцій:

- Ввести значення метрики СВО для ПС, що базується на штучному інтелекті.
- Ввести значення метрики WMC для ПС, що базується на штучному інтелекті.
- Ввести фактичне значення метрики RFC для ПС, що базується на штучному інтелекті.
- Ввести значення довірчої ймовірності для розрахунку інтервалів.
- Перевірити вихідні дані на коректність.
- Оцінити значення метрики RFC для ПС, що базується на штучному інтелекті.
- Обчислити довірчий інтервал оцінки значення метрики RFC для ПС, що базується на штучному інтелекті.
- Обчислити прогнозний інтервал оцінки значення метрики RFC для ПС, що базується на штучному інтелекті.
- Визначити рівень якості ПС, що базується на штучному інтелекті, за результатами порівняння фактичного значення метрики RFC з отриманими інтервалами.

2.2 Функціональні обмеження

Функціональні обмеження відсутні.

3 Технічні засоби, що використовуються

Для коректної роботи програмної системи персональний комп'ютер (або інший пристрій) користувача повинен відповідати наступним мінімальним вимогам:

- Центральний процесор (CPU): Будь-який сучасний процесор із тактовою частотою від 1 ГГц (наприклад, Intel Core i3, AMD Ryzen 3 або аналогічні).
- Оперативна пам'ять (RAM): Мінімум 2 ГБ (рекомендовано 4 ГБ або більше для стабільної роботи операційної системи та веб-браузера).
- Накопичувач на жорсткому диску (HDD/SSD): Для розміщення файлів програми потрібно менше 1 МБ вільного простору.
- Пристрої вводу-виводу: Клавіатура, комп'ютерна миша (або тачпад), монітор із роздільною здатністю екрана не менше 1280x720 пікселів.
- Мережеве підключення: Не є обов'язковим (програма працює локально без доступу до мережі Інтернет).

4 Виклик та завантаження

Програмна система постачається у вигляді набору файлів веб-ресурсу. Процес розгортання та запуску включає наступні кроки:

Крок 1. Підготовка файлів. Переконайтеся, що всі компоненти програмної системи знаходяться в одній директорії (папці):

- index.html – головний файл інтерфейсу;
- style.css – файл стилів оформлення;
- script.js – файл з логікою розрахунків.

Крок 2. Запуск програмної системи. Для початку роботи необхідно відкрити файл index.html. Це можна зробити двома способами:

- Двічі клацнути лівою кнопкою миші по файлу index.html (відкриється у браузері за замовчуванням).
- Натиснути правою кнопкою миші по файлу index.html, обрати пункт «Відкрити за допомогою» («Open with») та обрати бажаний веб-браузер.

Крок 3. Використання. Після завантаження інтерфейсу:

- У відповідні поля введіть метрики системи (CBO, WMC, RFC).

- Вкажіть рівень довірчої ймовірності (за замовчуванням встановлено 0.95).
- Натисніть кнопку «Оцінити».
- Результати розрахунків (оцінка значення метрики RFC, межі довірчого інтервалу для оцінки значення метрики RFC, межі прогнозного інтервалу для оцінки значення метрики RFC, рівень якості програмної системи, що базується на штучному інтелекті) з'являться у формі.

5 Вихідні та результуючі дані

Вихідною інформацією є введені значення метрик CBO, WMC, RFC ПС, що базуються на штучному інтелекті, та довірна ймовірність.

Результуючими даними програмної системи є: оцінене значення метрики RFC для ПС, що базується на штучному інтелекті; довірчий інтервал оцінки значення метрики RFC для ПС, що базується на штучному інтелекті; прогнозний інтервал оцінки значення метрики RFC для ПС, що базується на штучному інтелекті; рівень якості ПС, що базується на штучному інтелекті.

ДОДАТОК Е

Повний набір вихідних даних

Повний набір вихідних даних представлений в таблиці Е.1.

Таблиця Е.1 – Повний набір вихідних даних

№	Назва проекту	СВО (X_1)	WMC (X_2)	RFC (Y)
1	2	3	4	5
1	transformers	4,98	9,22	14,11
2	pytorch	4,27	11,64	13,89
3	d2l-zh	1,42	4,25	10,27
4	scikit-learn	2,43	14,72	19,4
5	keras	2,38	10,31	17,68
6	annotated_deep_learning_ paper_implementations	3,16	4,63	11,97
7	yolov5	2,92	11,73	18,8
8	faceswap	4,66	18,2	24,99
9	OpenBB	2,54	5,66	6,89
10	ultralitics	4,7	15,39	22,89
11	airflow	4,76	9,14	12
12	streamlit	4,35	8,35	11,6
13	DeepSpeed	3,81	16,98	18,82
14	gradio	4,44	12,4	11,2
15	ray	5,69	11,82	14,2
16	Open-Assistant	1,91	5,11	6,35
17	supervision	5,59	15,94	18,47
18	paperless-ngx	2,3	4,21	5,25
19	spaCy	2,84	8,16	7,41
20	qlib	4,58	10,9	17,28
21	pytorch-lightning	6,76	20,28	22,32

Кінець таблиці Е.1

1	2	3	4	5
22	ML-From-Scratch	1,13	7,3	13,1
23	data-science-ipython-notebooks	4,61	11,68	16,35
24	EasyOCR	3,64	10,31	18,52
25	d2l-en	1,72	5,81	13,3
26	Resume-Matcher	2,49	3,89	5,1
27	haystack	5,46	15,26	16,5
28	mlflow	3,98	10,79	12,62
29	serve	4,34	7,96	9,75
30	recommenders	4,33	15,3	26,59
31	rasa	10,6	16,75	19,64
32	datasets	4,76	15,42	16,4
33	deepface	2,62	4,5	8,43
34	onnx	1,6	5,81	6,3
35	HivisionIDPhotos	3	10,38	14,15
36	vision	4,23	6,69	12,57
37	DocsGPT	4,1	13,8	15,75
38	marimo	3,98	7,68	9,19
39	gensim	2,5	23,64	24,15
40	numpy-ml	2,17	15,5	19,44
41	LaTeX-OCR	6,29	15,71	29,14
42	ml-engineering	1,77	8,5	12,73
43	dvc	2,74	10,28	11,4
44	abu	3,53	9,7	12,38
45	imgaug	1,54	13	15,5
46	horovod	1,92	8,22	13,28
47	ChatterBot	0,29	1,16	1,71