

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Аналіз засобів та методів автоматизації тестування програмного забезпечення з використанням ШІ

Виконав(ла): студент(ка) 6 курсу, групи СНм-61
спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

	<u>Цвігун О.Р.</u> (підпис)	(прізвище та ініціали)
Керівник	<u>Боднарчук І.О.</u> (підпис)	(прізвище та ініціали)
Нормоконтроль	<u>Дуда О.М.</u> (підпис)	(прізвище та ініціали)
Завідувач кафедри	<u>Боднарчук І.О.</u> (підпис)	(прізвище та ініціали)
Рецензент	<u></u> (підпис)	(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.

(підпис)

(прізвище та ініціали)

"__" "__" 202__ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Магістр

(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки

(шифр і назва спеціальності)

студенту Цвігун Олександр Русланович

(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз засобів та методів автоматизації
тестування програмного забезпечення з використанням ШІ.

Керівник роботи к.т.н, доц. Боднарчук І.О.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від "29" листопада 2024 року № 4/7-1139

2. Термін подання студентом завершеної роботи 28 травня 2025 р.

3. Вихідні дані до роботи Літературні джерела з тематики роботи

4. Зміст роботи (перелік питань, які потрібно розробити) ВСТУП 1 ОГЛЯД ЛІТЕРАТУРНИХ
ДЖЕРЕЛ НА ТЕМУ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ 1.1 Роль технологій штучного інтелекту в автоматизованому тестуванні 1.2
Огляд стану проблеми використання технологій ШІ в автоматизованому тестуванні 1.3
Обмеження існуючих фреймворків 2 ПРОГНОЗУВАННЯ НЕСПРАВНОСТЕЙ ЗА
ДОПОМОГОЮ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ 2.1 Моделі машинного навчання 2.2
Навчання без учителя 2.3 Навчання з підкріпленням 2.4 Алгоритми прогнозування
несправностей у режимі реального часу 3 АНАЛІЗ ДАНИХ ДЛЯ ПРОГНОЗУВАННЯ ЗБОЇВ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 3.1 Прогнозне обслуговування 3.2 Механізми
самовідтворення систем 3.3 Адаптивні стратегії тестування 3.4 Приклади з тематичних
досліджень 3.5 Системи самовідновлювального автоматизованого тестування на основі
штучного інтелекту – проблеми та обмеження 3.6 Технології самовідновлення 3.7 Потреба в
подальших дослідженнях 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ
СИТУАЦІЯХ; ВИСНОВКИ; СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ; ДОДАТКИ

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сенчишин В.С, к.т.н., доц.		
Безпека в надзвичайних ситуаціях	Клепчик В.М., ст. викл.		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	14.04.25-15.04.25	Виконано
2.	Підбір наукових джерел по темі роботи	16.04.25-20.04.25	Виконано
3.	Переклад та опрацювання наукових джерел по темі кваліфікаційної роботи	20.04.25-24.04.25	Виконано
4.	Виконання дослідження щодо теми Кваліфікаційної роботи	24.04.25-10.05.25	Виконано
5.	Оформлення першого розділу	04.05.25-05.10.25	Виконано
6.	Оформлення другого розділу	05.05.25-13.05.25	Виконано
7.	Оформлення третього розділу	13.05.25-14.05.25	Виконано
8.	Виконання завдання до підрозділу "Охорона праці"	08.05.25-09.04.25	Виконано
9.	Виконання завдання до підрозділу "Безпека в надзвичайних ситуаціях"	10.05.25-05.05.25	Виконано
10.	Оформлення кваліфікаційної роботи	05.05.25-31.05.25	Виконано
11.	Нормоконтроль	14.05.25-15.05.25	Виконано
12.	Перевірка на плагіат	15.05.25	Виконано
13.	Попередній захист кваліфікаційної роботи	16.05.25	Виконано
14.	Захист кваліфікаційної роботи	29.05.2025	

Студент

(підпис)

Цвігун О.Р.

(прізвище та ініціали)

Керівник роботи

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

АНОТАЦІЯ

"Аналіз засобів та методів автоматизації тестування програмного забезпечення з використанням ШІ" // Кваліфікаційна робота освітнього рівня "Магістр" // Цвігун Олександр Русланович // Тернопільський національний технічний університет ім. І. Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2025 // с. – 58, рис. – 5, табл. – 0, джерел – 45.

Ключові слова: автоматизоване тестування, машинне навчання, штучний інтелект, прогнозування несправностей, відновлення в режимі реального часу, розробка програмного забезпечення.

У цій кваліфікаційній роботі проведено дослідження можливостей прогнозування помилок у процесі розробки програмного забезпечення та його відновлення в реальному часі з використанням технологій штучного інтелекту (ШІ) для створення системи автоматизованого тестування. Попри зростаюче значення забезпечення якості програмних продуктів, більшість сучасних систем автоматизованого тестування потребують ручного налаштування алгоритмів для ефективного виявлення та усунення збоїв. Метою цього дослідження є підвищення надійності та гнучкості таких систем шляхом застосування алгоритмів машинного навчання, що забезпечують здатність до самовідновлення.

ANNOTATION

“Analysis of Tools and Methods for Software Testing Automation Using AI” // Master’s degree qualification paper // Tsvihun Oleksandr // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Computer Science Department, group CHHM-61 // Ternopil, 2025 // p. – 58, fig. – 5, tables – 0, references – 45.

Key words: automated testing, machine learning, artificial intelligence, fault prediction, real-time recovery, software development.

This qualification thesis presents a study on the prediction of software development errors and real-time recovery using artificial intelligence (AI) technologies for the development of an automated testing system. Despite the growing importance of software quality assurance, most existing automated testing systems require manual configuration of algorithms to effectively detect and recover from failures. The aim of this research is to enhance the reliability and adaptability of such systems through the application of machine learning algorithms that enable self-recovery capabilities.

ЗМІСТ

ВСТУП.....	7
1 ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ НА ТЕМУ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	10
1.1 Роль технологій штучного інтелекту в автоматизованому тестуванні	10
1.2 Огляд стану проблеми використання технологій ШІ в автоматизованому тестуванні	11
1.3 Обмеження існуючих фреймворків.....	13
1.3.1 Штучний інтелект у тестуванні програмного забезпечення.	14
1.3.2 Роль та переваги штучного інтелекту в пришвидшенні процесу тестування	15
1.3.3 Прогнозування несправностей за допомогою моделей машинного навчання.....	18
1.3.4 Системи самовідновлення.....	19
1.3.5 Концепції та застосування в розробці програмного забезпечення	20
1.3.6 Існуючі системи та технології самовідновлення	22
2 ПРОГНОЗУВАННЯ НЕСПРАВНОСТЕЙ ЗА ДОПОМОГОЮ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ	24
2.1 Моделі машинного навчання	24
2.1.1 Навчання з учителем.....	25
2.2 Навчання без учителя	26
2.3 Навчання з підкріпленням.....	26
2.4 Алгоритми прогнозування несправностей у режимі реального часу..	27
3 АНАЛІЗ ДАНИХ ДЛЯ ПРОГНОЗУВАННЯ ЗБОЇВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	28
3.1 Прогнозне обслуговування	29
3.1.1 Моніторинг стану.....	29
3.1.2 Моделі оцінки стану системи	31

3.1.3 Аналіз ризиків	31
3.1.4 Інтеграція з Інтернетом речей.....	31
3.1.5 Системи підтримки рішень	31
3.1.6 Переваги прогнозування та підтримки.....	32
3.2 Механізми самовідтворення систем.....	32
3.3 Адаптивні стратегії тестування	35
3.4 Приклади з тематичних досліджень.....	36
3.5 Системи самовідновлювального автоматизованого тестування на основі штучного інтелекту – проблеми та обмеження.....	37
3.5.1 Технічні виклики.....	37
3.5.2 Перешкоди для впровадження.....	39
3.5.3 Майбутнє систем самовідновлювального автоматизованого тестування	40
3.6 Технології самовідновлення	41
3.7 Потреба в подальших дослідженнях	42
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	45
4.1 Методи та засоби психофізіологічного розвантаження як допоміжний процес в розробці ПЗ	45
4.2 Попередження аварій на виробництвах із застосуванням хлору.....	47
ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53
ДОДАТКИ	

ВСТУП

Актуальність задачі. Сучасний етап розвитку інформаційних технологій супроводжується стрімким зростанням складності програмного забезпечення, що, своєю чергою, висуває підвищені вимоги до його якості, надійності та стійкості до помилок. Дослідженнями в області процесів забезпечення якості програмного забезпечення займається на сьогодні немало вчених, в тому числі і в ТНТУ ім. Івана Пулюя. Зокрема, в роботах [1, 2, 3, 4, 5] міститься матеріал про управління якістю на ранніх етапах життєвого циклу програмних продуктів. В цих же роботах пропонується автоматизована система управління якістю. Проте, разом з цим питання автоматизації тестування набуває особливої актуальності, адже традиційні методи часто не справляються з обсягом та різноманіттям тестових сценаріїв, які виникають у великих і динамічних проєктах. Використання штучного інтелекту в цій галузі відкриває нові можливості для підвищення ефективності процесів тестування.

Штучний інтелект, зокрема алгоритми машинного навчання, здатен не лише автоматизувати рутинні завдання, але й адаптуватися до змін у програмному продукті без необхідності постійного втручання людини. Це дозволяє зменшити час, витрачений на налаштування тестових сценаріїв, і підвищити швидкість виявлення помилок. Крім того, системи на основі ШІ можуть самостійно аналізувати попередні збої, прогнозувати можливі проблеми в майбутньому та навіть ініціювати процеси самовідновлення, що значно покращує загальну надійність програмного забезпечення.

Застосування ШІ також сприяє зменшенню впливу людського фактору в процесі тестування, що знижує ймовірність помилок, пов'язаних із суб'єктивністю або втомою тестувальників. В умовах швидкого темпу розробки, коли вихід нового функціоналу часто відбувається у стислі терміни, можливість оперативного й точного тестування стає вирішальною перевагою. Таким чином, інтеграція технологій штучного інтелекту у процеси тестування програмного

забезпечення є не лише доцільною, а й необхідною умовою для забезпечення високої якості та конкурентоспроможності сучасних ІТ-продуктів.

Мета роботи.

Однією з цілей цього дослідження було з'ясувати, чи існують методи штучного інтелекту, які можна було б застосувати для розробки самовідновлювальних фреймворків для автоматизованого тестування. Буде проведено огляд різних алгоритмів машинного навчання та методів аналізу даних для прогнозування того, які помилки виникнуть у режимі реального часу. Таким чином, метою є розробка фреймворків, які можуть автономно виявляти, прогнозувати та усувати помилки для досягнення кращої ефективності та надійності тестування програмного забезпечення.

Це дослідження має на меті зробити внесок у розробку автоматизованих рішень для тестування, які використовують можливості штучного інтелекту для досягнення цих цілей. Таким чином, це призведе до більш гнучких і, отже, більш стійких практик розробки програмного забезпечення, що покращить якість і надійність програмних продуктів.

Об'єкт дослідження: процеси автоматизованого тестування програмного забезпечення.

Предмет дослідження: методи та засоби впровадження технологій штучного інтелекту для автоматизації тестування програмного забезпечення.

Наукова новизна отриманих результатів.

У цьому дослідженні застосовано змішаний методологічний підхід, що включає якісні та кількісні дослідницькі методи, для більш комплексного аналізу самовідновлювальних систем автоматизованого тестування на базі штучного інтелекту. Такий підхід є обґрунтуванням для поєднання найкращого з обох методологій, щоб запропонувати глибше та численніше розуміння предмета.

Цей підхід є якісним і збирає інформацію через тематичні дослідження та галузеві звіти. Якісний аналіз розглядає реальні приклади та виявляє деякі закономірності та теми, які можуть бути пропущені кількісними даними, надаючи контекст та глибину результатам. Застосування цього методу особливо

корисне для розуміння суб'єктивного досвіду практиків та складнощів впровадження технологій. Тим часом кількісний підхід використовує експерименти та опитування для отримання числових даних. Він полегшує вимірювання деяких змінних, таких як точність прогнозування несправностей та час відновлення, які важливі для оцінки ефективності методу штучного інтелекту в системах самовідновлення. Після вилучення ці аспекти кількісно оцінюють тенденції, дозволяють проводити порівняння та забезпечують кількісну основу, на якій можна робити статистично значущі висновки.

Практичне значення отриманих результатів.

В роботі здійснено огляд сучасних практик автоматизованого тестування та обмежень існуючих систем тестування. Також показано, яку роль може відігравати штучний інтелект у тестуванні програмного забезпечення, з деякими перспективами того, як машинне навчання може покращити точність виявлення та прогнозування помилок. В роботі зроблено акцент на розробці автоматизованих процесів відновлення, які динамічно коригують тести відповідно до даних у режимі реального часу. Представлено тематичні дослідження, щоб показати реальне застосування механізмів самовідновлення в різних програмних середовищах. В результаті дослідження визначено технічні проблеми, такі як інтеграція та масштабованість штучного інтелекту, та економічні та організаційні бар'єри що можуть виникнути при впровадженні цих технологій.

Це дослідження має важливі наслідки для стейкхолдерів у ІТ-галузі та містить практичні рекомендації щодо впровадження самовідновлювальних фреймворків на основі штучного інтелекту.

Апробація результатів та особистий внесок здобувача.

Основні положення роботи доповідались, розглядались та обговорювались на науковій конференції Тернопільського національного технічного університету імені Івана Пулюя. Результати кваліфікаційної роботи опубліковані у тезах студентської наукової конференції "Актуальні задачі сучасних технологій – 2024", яка проводилась у ТНТУ.

1 ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ НА ТЕМУ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Роль технологій штучного інтелекту в автоматизованому тестуванні

У сучасній розробці програмного забезпечення автоматизоване тестування є необхідним, оскільки воно дозволяє командам швидше та ефективніше виконувати перевірку функціональності, надійності та продуктивності програми. Організації можуть значно зменшити ручну роботу та пришвидшити цикли випуску тестів, автоматизуючи повторювані завдання тестування. Однак у традиційних фреймворках автоматизації більшість з них потребують втручання з боку людини для обробки змін або помилок, які виникають несподівано під час тестування.

Завдяки інтеграції штучного інтелекту (ШІ) в автоматизоване тестування, тестування відбувається абсолютно по-новому. Саме тому ШІ може допомогти створити кращі інтелектуальні, більш адаптивні фреймворки тестування, які навчаються на минулому досвіді та вдосконалюються з часом. Методи машинного навчання можуть передбачати можливі помилки до їх виникнення та пропонувати найкращі стратегії відновлення, коли вони трапляються, щоб мінімізувати час простою та підвищити загальну якість програмного забезпечення. У цьому контексті самовідновлювальні фреймворки є особливо важливими. Відновлення відбувається автоматично та без втручання людини. Ця можливість є критично важливою в швидкозмінних середовищах розробки, які використовують концепції безперервної інтеграції та безперервного розгортання (CI/CD). Процеси тестування є стійкими та мають механізми самовідновлення, які допомагають процесу тестування швидко адаптуватися та створювати більш надійні програмні продукти.

Зокрема, хоча автоматизоване тестування значно просунулося протягом останніх десятиліть, фреймворкам бракує надійної підтримки для прогнозування

помилки і відновлення. Сучасні системи, як правило, стикаються із затримками у виявленні та усуненні помилок. У цій ситуації можливе збільшення часу простою, зниження ефективності та вищі витрати на обслуговування. Крім того, відновлення після збоїв у більшості існуючих фреймворків є ручним та трудомістким. Ці проблеми підкреслюють необхідність більш інтелектуальних та автономних рішень для тестування для прогнозування та автоматичного вирішення проблем. Підключення штучного інтелекту в фреймворках для автоматизованого тестування відкриває перспективні можливості для подолання цих обмежень та забезпечення комплексного процесу тестування.

1.2 Огляд стану проблеми використання технологій ШІ в автоматизованому тестуванні

Розробка програмного забезпечення – це дуже важливий процес, який в сучасних умовах часто здійснюється з залученням автоматизованого тестування. Ми використовуємо спеціальні програмні інструменти для проведення попередньо написаних тестів програмного застосунку перед його випуском у виробництво. Основна причина його існування полягає в перевірці того, чи поводить ся програмне забезпечення належним чином, і чи не було допущено жодних дефектів чи помилок, які можуть вплинути на функціональність. Це дозволяє командам ефективніше проходити процес тестування та мати набагато вищий рівень гарантії якості.

Автоматизоване тестування відрізняється від ручного тестування, яке є виснажливим і схильним до людських помилок; ви можете запускати та виконувати повторювані тестові випадки за лічені хвилини. Така ефективність необхідна для забезпечення безперервної інтеграції та безперервного розгортання (CI/CD), як це визначено для сучасних практик розробки програмного забезпечення. Автоматизоване тестування в середовищах CI/CD, де зміни коду інтегруються та розгортаються багато разів на день, гарантує, що нові

функції та зміни не будуть впроваджені, що призведе до марнування грошей та часу. Основні фреймворки автоматизованого тестування показані на рисунку 1.1.

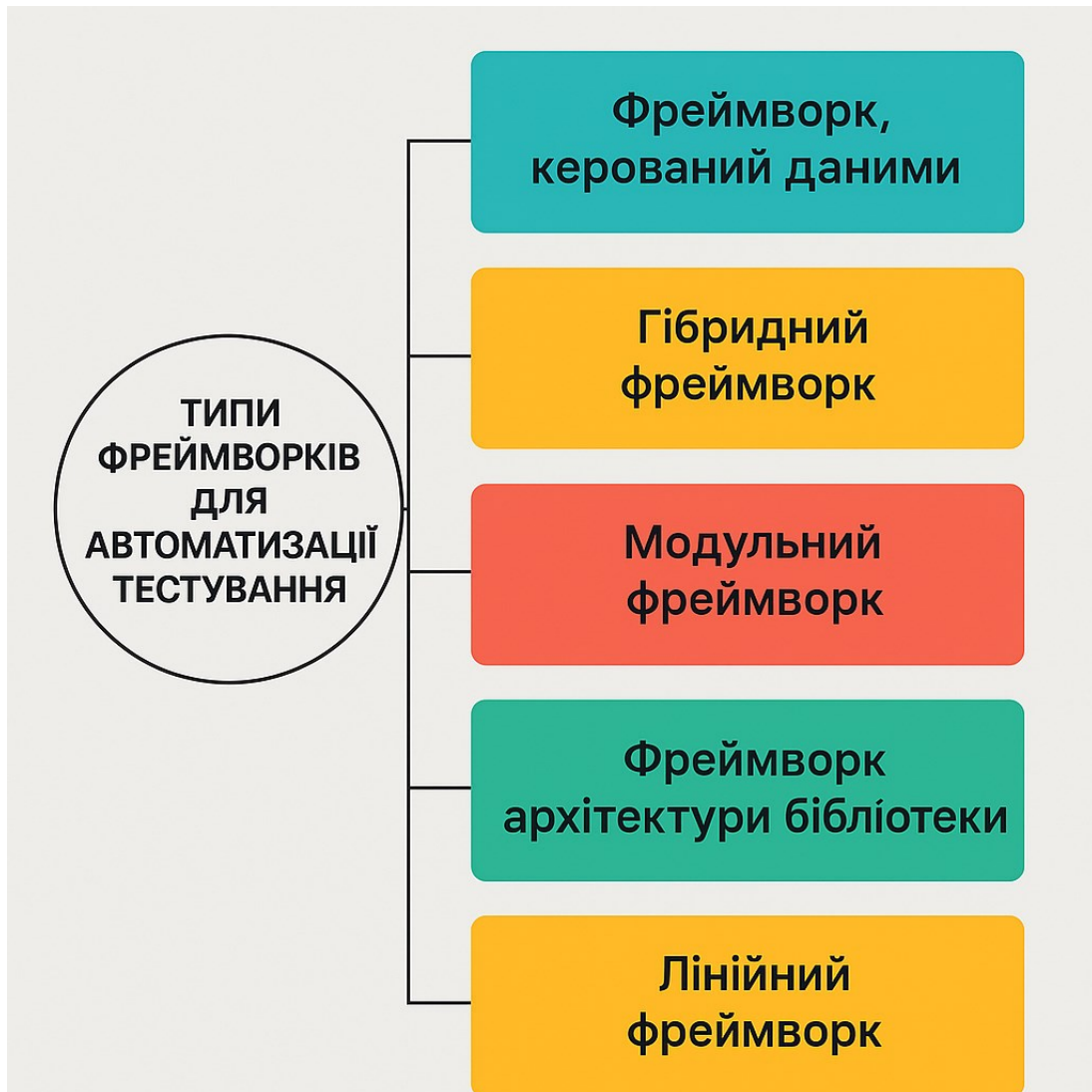


Рисунок 1.1 – Типи фреймворків для автоматизації тестування

Автоматизоване тестування використовується в таких сценаріях, як Selenium, JUnit, TestNg та Appium. Ці фреймворки значною мірою підтримують різні типи тестів, такі як модульне тестування, інтеграційне тестування, функціональне тестування та регресійне тестування. Ці інструменти допомагають тестувальникам писати скрипти, що імітують дії користувачів, вивчають результати та публікують повні результати тестування. Через важливість швидкого зворотного зв'язку та швидких ітерацій у середовищах

Agile та DevOps з жорсткими термінами та високоякісним програмним забезпеченням, автоматизоване тестування є там поширеним явищем.

1.3 Обмеження існуючих фреймворків

Незважаючи на численні переваги автоматизованого тестування, існуючі фреймворки стикаються з кількома суттєвими обмеженнями, які можуть перешкоджати їхній ефективності:

1. Витрати на обслуговування. Обслуговування скриптів автоматизації є однією з найголовніших проблем автоматизованого тестування. З розробкою програми, щоразу, коли існуючий тестовий скрипт потрібно оновити через будь-які зміни в інтерфейсі користувача та функціональності, це стає обов'язковим. Таке обслуговування, з одного боку, може бути дуже ресурсо- та трудомістким, особливо якщо воно відбувається в динамічному середовищі, де оновлення здійснюється регулярно. Однак постійна потреба в оновленнях скриптів може призвести до низької продуктивності та морального духу в командах тестування.

2. Відсутність адаптивності. Класичні фреймворки автоматизації не містять вбудованої здатності до адаптації. Вони значною мірою залежать від попередньо визначених скриптів, які не будуть масштабуватися самостійно на основі динамічних умов або неочікуваних збоїв. Якщо в тестованому застосунку відбуваються зміни, ці фреймворки можуть зіткнутися з невдалим тестуванням, що призведе до значних зусиль з усунення несправностей, що може ще більше уповільнити процес тестування.

3. Обмежене прогнозування та відновлення після збоїв. Більшість сучасних систем автоматизованого тестування повинні включати механізми для прогнозування збоїв заздалегідь або автоматичного відновлення після збоїв. Відсутність передбачення та стійкості може подовжити цикли тестування, і багато команд витратять багато зусиль на діагностику та вирішення проблем вручну. Це ускладнює прогнозування збоїв та збільшує ризик того, що

збої у продакшені міститимуть невідстежені проблеми, що означає значне розчарування користувачів.

4. Високі початкові витрати на налаштування. Створення системи автоматизованого тестування є дорогим на початку; воно потребує часу та ресурсів. Скрипти складні та можуть вимагати спеціалізованих навичок, які тестувальники повинні розробити та налаштувати. Ця перша перешкода може бути квитком, який лише деякі великі команди чи компанії використовують для виправдання або уникнення автоматизованого тестування.

5. Проблеми масштабованості. Масштабування автоматизованих тестів для врахування наступних нових функцій та інтеграцій стає неймовірно складним зі зростанням складності програмних застосунків. Однак такі тестові випадки можуть призвести до того, що існуючі фреймворки не зможуть ефективно керувати та виконувати багато тестових випадків, що призведе до вузьких місць у процесі тестування. Така проблема масштабованості може обмежити можливості організацій підтримувати високий рівень охоплення тестами та належний контроль якості в міру розробки їхніх застосунків.

Штучний інтелект може надавати прогностні висновки, дозволяючи проактивно виявляти помилки та використовувати ефективні стратегії адаптивного тестування. Здатність фреймворків автоматично адаптуватися до змін або обходити збої спростить самовідновлення. Оскільки автоматизоване тестування стає більш надійним, адаптивним та стійким, воно забезпечить кращу якість програмних продуктів та кращу ефективність циклів розробки.

1.3.1 Штучний інтелект у тестуванні програмного забезпечення

Тестування програмного забезпечення – це найважливіша фаза життєвого циклу розробки програмного забезпечення (SDLC), під час якої програмне забезпечення оцінюється, щоб гарантувати його відповідність вимогам та відсутність дефектів. Основна причина тестування полягає у виявленні помилок або проблем програмного забезпечення перед його випуском у виробництво.

Загалом, ми можемо розділити типи тестування на модульне тестування, інтеграційне тестування, функціональне тестування та тестування продуктивності. Однак кожне з них використовується для різних цілей та різних частин програми.

Тестування програмного забезпечення необхідне для зниження ризику збоїв у продакшені, оскільки якщо помилку не виявити під час тестування, це може призвести до значних втрат грошей, репутації та задоволеності користувачів. Зі зростанням складності програмного застосунку та зростанням очікувань користувачів, традиційні методології ручного тестування повинні наздоганяти. Отже, цикл випуску та витрати зростають. Інтеграція штучного інтелекту (ШІ) має вирішальне значення, впроваджуючи концепцію штучного інтелекту в процес тестування та змінюючи підхід до тестування та його проведення.

1.3.2 Роль та переваги штучного інтелекту в пришвидшенні процесу тестування

Революція штучного інтелекту торкається сфери тестування програмного забезпечення за допомогою інтелектуальної автоматизації, роблячи процес тестування ефективнішим та точнішим. Технології штучного інтелекту можуть допомогти тестовим системам навчатися на основі історичних даних, адаптуватися до нових сценаріїв та покращувати процес тестування. Розвиваючись таким чином, тестування програмного забезпечення отримує кілька фундаментальних переваг, які значно підвищують його ефективність (див. рис. 1.2).



Рисунок 1.2 – Переваги штучного інтелекту в тестуванні

1. Автоматизована генерація тестів. На думку багатьох експертів у галузі тестування програмного забезпечення, однією з найпереконливіших переваг штучного інтелекту в тестуванні програмного забезпечення є його автономна генерація тестових випадків з використанням поведінки та взаємодії користувачів із програмою. ШІ може створювати всеохоплюючі набори тестів, аналізуючи, як користувачі взаємодіють із програмним забезпеченням, та визначаючи різні сценарії. Ця автоматизація позбавляє команди від ручної роботи в розробці тестів, і стає очевидним, як команда підвищує продуктивність, витрачаючи більше часу на стратегічну сторону.

2. Покращене покриття тестами. ШІ спеціально покращує покриття тестами, аналізуючи код та шаблони поведінки користувачів, і на основі цього знаходить області, які потребують додаткового тестування. Інші традиційні методи тестування можуть тестувати лише обмежену кількість шляхів або програму в цілому, тоді як ШІ знаходитиме прогалини в покритті. Застосування цього комплексного підходу не тільки може допомогти знайти приховані дефекти, які інакше могли б залишитися непоміченими, але й дозволяє нам створювати програмне забезпечення вищої якості.

3. Швидше виконання та зворотний зв'язок. Тестові випадки можуть бути пріоритетизовані шляхом оцінки ризиків за допомогою інструментів тестування на базі штучного інтелекту та виконуватися ефективніше, ніж за допомогою попередніх методів. Це надає пріоритет цим тестам та дозволяє розробникам швидко отримувати зворотний зв'язок для швидшого виправлення своїх тестів. Ця можливість пришвидшує процес розробки, випускаючи релізи швидше без шкоди для якості в гнучких середовищах, де необхідні швидкі ітерації.

4. Прогнозна аналітика. Завдяки прогнозній аналітиці штучний інтелект може аналізувати історичні дані про дефекти та поточні результати тестування, щоб передбачити можливі майбутні збої, перш ніж вони спричинять будь-які дефекти. Команди можуть проактивно вирішувати проблеми та зменшувати ймовірність того, що дефекти потраплять у виробництво, виявляючи закономірності та тенденції. Передбачення таким чином сприяє своєчасному втручанню та більш обґрунтованому прийняттю рішень протягом циклів розробки.

5. Можливості самовідновлення. Можливості самовідновлення є однією з найінноваційніших функцій ШІ в тестуванні. Як приклад завдань, які ШІ може виконувати автоматично без втручання людини, пошкоджені тестові скрипти можна виявляти та виправляти автоматично, коли змінюється інтерфейс користувача програми. Ця функція самовідновлення значно знижує витрати на

обслуговування, водночас забезпечуючи ефективність процесів тестування навіть за швидких змін середовища.

1.3.3 Прогнозування несправностей за допомогою моделей машинного навчання

Моделі машинного навчання (ML) відіграють важливу роль у тестуванні програмного забезпечення, особливо з точки зору прогнозування помилок. Ці моделі використовують історичні дані для аналізу закономірностей і тенденцій, а також прогнозування потенційних проблем, які можна вирішити, перш ніж вони проявляться. Кілька ключових моделей ML зазвичай використовуються для прогнозування помилок:

1. Регресійний аналіз. Цей статистичний підхід прогнозує ймовірність дефектів залежно від історичних даних. Команди співвідносять різні фактори, щоб визначити, які області мають високий ризик і де потрібно зосередити зусилля тестування, щоб покращити управління дефектами.

2. Алгоритми класифікації. Зміни коду класифікуються на основі ймовірності того, що їх впровадження призведе до дефектів, за допомогою таких методів, як дерева рішень та методи опорних векторів. Використання цієї класифікації допомагає командам вирішувати, які зміни тестувати, тим самим підвищуючи ефективність процесу тестування.

3. Кластеризація. Подібні шаблони дефектів групуються за допомогою методів кластеризації, щоб команди могли знайти основні спільні причини. Розуміння цих шаблонів дозволяє командам удосконалювати свою стратегію тестування та стратегію забезпечення якості в цілому.

4. Нейронні мережі. Моделі глибокого навчання – це аспект нейронних мереж, які розуміють складні форми, шаблони та великі дані. Вони точно прогнозують несправності, оскільки можуть навчатися на основі великої кількості даних.

5. Аналіз часових рядів. У цьому аналітичному методі ми намагаємося передбачити, до якого часу мають виникнути дефекти, враховуючи певні тенденції. Команди можуть навчатися на історичних закономірностях, щоб проактивно керувати ризиками, розуміючи історичні закономірності та вчасно вживаючи заходів.

Оскільки ландшафт програмного забезпечення продовжує розширюватися, а впровадження штучного інтелекту продовжує зростати, використання штучного інтелекту в тестуванні буде важливим для визначення майбутнього розробки програмного забезпечення, щоб переконатися, що додатки працюватимуть належним чином, відповідатимуть високим очікуванням користувачів щодо продуктивності та надійності.

1.3.4 Системи самовідновлення

Хоча термін «самовідновлювальні системи» охоплює широкий спектр застосувань, оскільки він включає концепції виявлення та діагностики несправностей (FDD) або ізоляції та відновлення несправностей (FIR), у цьому дослідженні відмовостійкість (FT) вважається більш доцільною сферою для такої системи. Основною метою самовідновлювальних систем є підтримка або відновлення функціональності для безперервної роботи та мінімізація часу простою. У розробці програмного забезпечення ці функції дозволяють застосунку відновлюватися після помилок та змін таким чином, що зменшує проблеми з надійністю, одночасно покращуючи взаємодію з користувачем.

Аналогічно, самовідновлення – це здатність біологічних систем відновлюватися після фізичного пошкодження. На програмному рівні це означає системи, які автоматично виправляють себе у разі виникнення порушень (наприклад, програмних помилок, апаратних проблем, збоїв у мережі тощо). Це стосується систем самовідновлення, які зменшують потребу в ручному усуненні несправностей та втручанні, а також забезпечують більш плавний

користувацький досвід, оскільки програми продовжують реагувати та працювати за наявності проблем.

1.3.5 Концепції та застосування в розробці програмного забезпечення

Системи самовідновлення використовують різноманітні концепції та технології для досягнення стійкості та надійності.

1. Моніторинг та виявлення. Як і у випадку із системами самовідновлення, постійний моніторинг продуктивності та стану системи є фундаментальним. Дані з інтенсивним обсягом даних у режимі реального часу аналізуються за допомогою передових алгоритмів штучного інтелекту та машинного навчання, які виявляють ранні аномалії та потенційні проблеми. Такий проактивний моніторинг виявляє проблеми на ранній стадії, перш ніж вони перетворяться на проблемні збої, і є час для вжиття коригувальних заходів.

2. Діагностика та аналіз. Коли виникає несправність, немає часу на пошук її кореня. Все робиться за допомогою складних методів аналізу, де моделі машинного навчання функціонують для виявлення закономірностей та кореляцій у даних, що призводить до основної проблеми. Системи самовідновлення можуть використовувати історичні дані та можливості прогнозу аналітики, щоб автоматично вирішувати, чи є проблема повторюваною чи одноразовою, тим самим забезпечуючи більш доцільну реакцію.

3. Автоматизоване відновлення. Коли помилку діагностовано, системи самовідновлення вживають коригувальних заходів для усунення дефекту. Це робиться шляхом перезапуску служб, відкату до попереднього стабільного стану або автоматичного застосування виправлень. Метою є якомога швидше відновлення функціональності з мінімальним впливом на користувачів. Наприклад, якщо доступна резервна копія, система самовідновлення може автоматично перезапустити службу після збою веб-служби або перенаправити трафік на інший екземпляр.

4. Петлі зворотного зв'язку. Впровадження петель зворотного зв'язку є важливою частиною систем самовідновлення. Ці петлі дозволяють системі адаптувати стратегії виявлення та відновлення несправностей, щоб їх можна було вдосконалити, враховуючи попередні інциденти. Порівняння ефективності відповідей дозволяє системі коригувати та калібрувати свої алгоритми, щоб забезпечити більш стійку інфраструктуру. Цикл постійного вдосконалення має вирішальне значення для підтримки високої доступності та продуктивності в складних програмних середовищах.

Самовідновлювальні системи знаходять широке застосування в різних галузях розробки програмного забезпечення, включаючи наступні.

1. Хмарні сервіси. Для хмарних обчислювальних середовищ можливості самовідновлення є критично важливими для високої доступності та надійності. Ці системи автоматично піклуються про всі ресурси. Крім того, вони можуть витримувати багато збоїв, завдяки чому хмарні додатки працюють так само, як локально, з набагато меншими зусиллями з боку менеджера ресурсів. Наприклад, якщо віртуальна машина виходить з ладу, хмарна інфраструктура негайно запустить новий екземпляр, тому користувачеві не доведеться стикатися з простоями.

2. Архітектура мікросервісів. З розвитком мікросервісної архітектури з'явилася ще одна причина, що підсилює потребу в системах самовідновлення. У цій архітектурі програми розбиваються на менші, незалежні компоненти, які взаємодіють між собою. Механізми самовідновлення допомагають забезпечити безперервність обслуговування, ізолюючи та вирішуючи проблеми окремих мікросервісів, не спричиняючи випадкових збоїв у роботі всієї програми. Система, що складається з кількох мікросервісів, може перенаправляти запити у разі збою екземпляра одного з них або запускати новий екземпляр, щоб забезпечити належну роботу інших сервісів.

3. Конвеєри DevOps та CI/CD. Механізми самовідновлення, що використовуються в середовищах DevOps, використовують автоматизацію виявлення та виправлення проблем, що виникають у конвеєрі розробки. Системи

самовідновлення можуть автоматично запускати процедури відкату для виправлення процесів збірки або етапів розгортання, повідомляти розробників або вживати коригувальних заходів для відновлення функціональності. Перевагою цієї автоматизації є те, що вона зменшує час, витрачений на затримки, і допомагає зробити розгортання ефективнішим, щоб високоякісне програмне забезпечення можна було доставляти швидше.

1.3.6 Існуючі системи та технології самовідновлення

З'явилося кілька фреймворків і технологій для підтримки впровадження систем самовідновлення, надаючи розробникам потужні інструменти для підвищення стійкості та надійності.

1. Kubernetes. Будучи платформою оркестрації контейнерів із вбудованими функціями самовідновлення, автоматичним перезапуском, заміною та масштабуванням контейнерів за заздалегідь визначеними умовами, серед іншого, Kubernetes є провідною системою. Коли контейнер виходить з ладу, Kubernetes може автоматично замінити його без ручного втручання, щоб забезпечити постійну підтримку бажаного стану програми.

2. Автоматичне масштабування AWS. Можливості автоматичного масштабування в Amazon Web Services (AWS) допомагають підтримувати продуктивність програм, автоматично регулюючи ресурси. AWS відстежує закономірності використання своїх програм і регулює потужність на льоту, щоб ваші програми залишалися стійкими та добре працювали за різних профілів навантаження.

3. Microsoft Azure Monitor. Ви отримуєте комплексний моніторинг та оповіщення, які запускають автоматичні реакції на виявлені проблеми. Коли перевищено певні порогові значення, цю службу можна використовувати для запуску дій самовідновлення (для захисту продуктивності та надійності програм).

4. Chaos Monkey від Netflix. Chaos Monkey є частиною Simian Army, набору інструментів від Netflix, який навмисно викликає збої в системах для проведення стрес-тестів на їхню здатність до самовідновлення або відновлення після збоїв. Організації можуть аналізувати реакцію та відновлення своїх систем на реальні збої та підвищувати їхню стійкість та надійність.

5. Автономні обчислення. Ширша структура включає самовідновлення, критичну вимогу цієї ширшої структури, що вимагає мінімального втручання людини, що забезпечує самокерування. Автономні обчислювальні системи можуть динамічно змінювати свою поведінку для досягнення оптимальної продуктивності та автоматично відновлюватися після збоїв.

Вони забезпечують засоби для розробки надійних, стійких застосунків, здатних до самокерування та самовідновлення після збоїв. Оскільки сучасні програмні ландшафти різноманітні та складні, самовідновлювальні системи стають незамінними: надійними та стабільно продуктивними.

2 ПРОГНОЗУВАННЯ НЕСПРАВНОСТЕЙ ЗА ДОПОМОГОЮ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ

Прогнозування несправностей є важливим фактором підтримки надійності та ефективності системи в багатьох галузях, таких як виробництво, охорона здоров'я, транспорт або енергетика. Оскільки системи стають складнішими, а потреба в безперебійному постачанні стає критичною, для прогнозування та вирішення передбачуваних проблем використовуються передові методи. За допомогою методів штучного інтелекту (ШІ) організації можуть краще прогнозувати несправності та своєчасно вживати коригувальних заходів, що призводить до скорочення простоїв та покращення загальної продуктивності. Зосереджуючись на різних методах ШІ, що використовуються в прогнозуванні несправностей, це дослідження містить детальний огляд моделей машинного навчання, методологій аналізу даних та методів прогнозного обслуговування.

2.1 Моделі машинного навчання

Моделі машинного навчання на основі штучного інтелекту значною мірою підтримують системи прогнозування несправностей на основі штучного інтелекту. Ці моделі створені для навчання на основі історичних даних, щоб формувати закономірності та визначати ймовірну ситуацію найближчим часом (див. рис. 2.1).



Рисунок 2.1 – Набір алгоритмів машинного навчання

Застосування машинного навчання в прогнозуванні несправностей можна розділити на три основні типи: навчання з учителем, навчання без учителя та навчання з підкріпленням.

2.1.1 Навчання з учителем

Прикладом навчання з учителем є моделі, які навчаються на розмічених наборах даних; ці дані містять пари вхід-вихід. Перевагою цього підходу є навчання зв'язку між заданими вхідними ознаками та відповідними вихідними мітками. Навчання з учителем особливо корисне для прогнозування несправностей, оскільки воно дозволяє робити точні прогнози (завдяки історичним випадкам несправностей). Загальні алгоритми включають:

- **Дерева рішень:** ці моделі розділяють дані на гілки на основі значень ознак; вони прогнозують виникнення помилок. Інтуїтивно зрозуміла структура ваших даних полегшує їх інтерпретацію та реалізацію.

- Випадкові ліси: метод ансамблю пов'язаний з побудовою моделі прогнозування з використанням кількох дерев рішень для підвищення точності прогнозування. Випадкові ліси зменшують ризик перенавчання шляхом усереднення результатів (отже, стійкі до зашумлених даних).

- Методи опорних векторів (SVM): вони дуже добре працюють у просторах високої вимірності та добре підходять для класифікації. Оскільки SVM можуть вивчати гіперплощини для розділення різних класів у складних наборах даних, вони є гарною практикою для виявлення помилок.

2.2 Навчання без учителя

Навчання без учителя стосується випадків, коли дані не мають позначок, і метою є вивчення структури даних, тобто виявлення різних закономірностей або внутрішніх структур даних. У випадках, коли доступ до історичних даних про несправності обмежений або відсутній, цей підхід корисний для прогнозування несправностей. Ключові методи включають:

- Алгоритми кластеризації: методи K-Means та DBSCAN групують точки даних на основі їхньої подібності, і виявляють незвичайні закономірності, які є можливими помилками та повинні бути перевірені.

- Виявлення аномалій: в цьому методі головна ідея полягає у виявленні викидів у даних, які не відповідають нормі. Системи попереджають про ці аномалії, щоб запобігти їм на етапі, перш ніж вони перетворяться на проблеми.

2.3 Навчання з підкріпленням

Це динамічний підхід до навчання з підкріпленням, де моделі навчаються прийнятним діям через помилку дій. Хоча він використовується рідше для прогнозування несправностей, він може бути корисним для прогнозування несправностей систем, які за своєю суттю змінюються (адаптуються) для запобігання несправностям. Наприклад, навчання з підкріпленням

використовується для пошуку оптимальної роботи машин з урахуванням зворотного зв'язку в режимі реального часу в промислових умовах.

2.4 Алгоритми прогнозування несправностей у режимі реального часу

У реальному часі необхідні алгоритми, здатні обробляти дані в режимі реального часу, швидко та ефективно. Деякі ключові алгоритми включають:

- Аналіз часових рядів: Тип аналізу, який можна виконати на історичних даних для виявлення майбутніх помилок, полягає в використанні таких методів, як мережі ARIMA або LSTM. Вони особливо добре підходять для роботи з часовими даними та тенденціями з плином часу.
- Нейронні мережі: Моделі глибокого навчання, такі як згорткові нейронні мережі (ЗНМ) та рекурентні нейронні мережі (РНМ), чудово справляються з обробкою великих та складних наборів даних. ЗНМ особливо добре підходять для вилучення ознак із просторових даних, тоді як РНМ розроблені для послідовного аналізу даних, що робить їх ідеальними для прогнозування несправностей, що залежить від часу.
- Алгоритми онлайн-навчання: ці алгоритми постійно адаптуються до надходження нових даних, що робить їх надзвичайно важливими в середовищах, де дані постійно змінюються. Вони допомагають підтримувати актуальність і точність моделі з часом.

3 АНАЛІЗ ДАНИХ ДЛЯ ПРОГНОЗУВАННЯ ЗБОЇВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Аналіз даних є наріжним каменем ефективного прогнозування несправностей, що охоплює вивчення наборів даних для отримання змістовної інформації та виявлення закономірностей, що вказують на потенційні несправності. Цей процес включає кілька ключових методів, які покращують загальну прогностичну здатність системи.

Вилучення ознак – один із перших кроків аналізу даних, що зосереджений на визначенні найважливіших змінних, що впливають на виникнення несправностей. Вибираючи ключові ознаки, організації можуть покращити продуктивність своїх прогностичних моделей. Такі методи, як аналіз головних компонентів (РСА), допомагають зменшити розмірність, перетворюючи складні набори даних у більш керовані форми, зберігаючи при цьому важливу інформацію.

Попередня обробка даних – це ще один критичний аспект, який забезпечує їхню чистоту та готовність до аналізу. Це включає обробку відсутніх значень, нормалізацію даних та кодування категоріальних змінних, щоб зробити набір даних придатним для алгоритмів машинного навчання. Ефективна попередня обробка життєво важлива для підтримки цілісності даних та забезпечення точних прогнозів.

Розпізнавання образів відіграє значну роль у виявленні повторюваних тенденцій, які можуть передувати несправностям. Статистичний аналіз може бути використаний для оцінки розподілу даних та виявлення аномалій. Водночас, методи обробки сигналів, такі як перетворення Фур'є та вейвлет-аналіз, корисні для виявлення нерівностей у даних часових рядів, особливо в моніторингу машин.

Інструменти візуалізації відіграють важливу роль у передачі інформації, отриманої в результаті аналізу даних. Графіки, діаграми та інтерактивні інформаційні панелі дозволяють зацікавленим сторонам швидко виявляти

тенденції, аномалії та кореляції даних. Ці візуальні представлення сприяють прийняттю обґрунтованих рішень та дозволяють швидше реагувати на потенційні проблеми.

Виявлення закономірностей та аномалій є критично важливим у процесі прогнозування несправностей. Організації можуть вживати проактивних заходів для запобігання несправностям, розпізнаючи відхилення від нормальної поведінки. Методи виявлення аномалій, включаючи статистичні методи та спеціалізовані нейронні мережі, допомагають виявляти незвичайні закономірності, які можуть сигналізувати про неминучі збої. Прогнозна аналітика додатково покращує прогнозування несправностей, використовуючи історичні дані для оцінки ймовірності майбутніх подій, забезпечуючи міцну основу для прийняття рішень.

3.1 Прогнозне обслуговування

Прогнозне обслуговування – це проактивний підхід, заснований на використанні штучного інтелекту для передбачення та потенційного запобігання виникненню збоїв до того, як вони виникнуть. Його все частіше розглядають як критичний компонент стратегій технічного обслуговування сучасних систем, особливо в галузях, де безвідмовна робота обладнання має першочергове значення. Ключові стратегії прогнозного обслуговування включають (див. рис. 3.1):

3.1.1 Моніторинг стану

Безперервний контроль стану обладнання за допомогою датчиків та пристроїв Інтернету речей відомий як моніторинг стану. Ці пристрої збирають дані в режимі реального часу про різні параметри, включаючи температуру, вібрації та тиск, і дозволяють організаціям легко збирати швидкі дані про стан обладнання. Ці дані дозволяють командам виявляти, коли знос або інші ознаки можуть спричинити поломку.

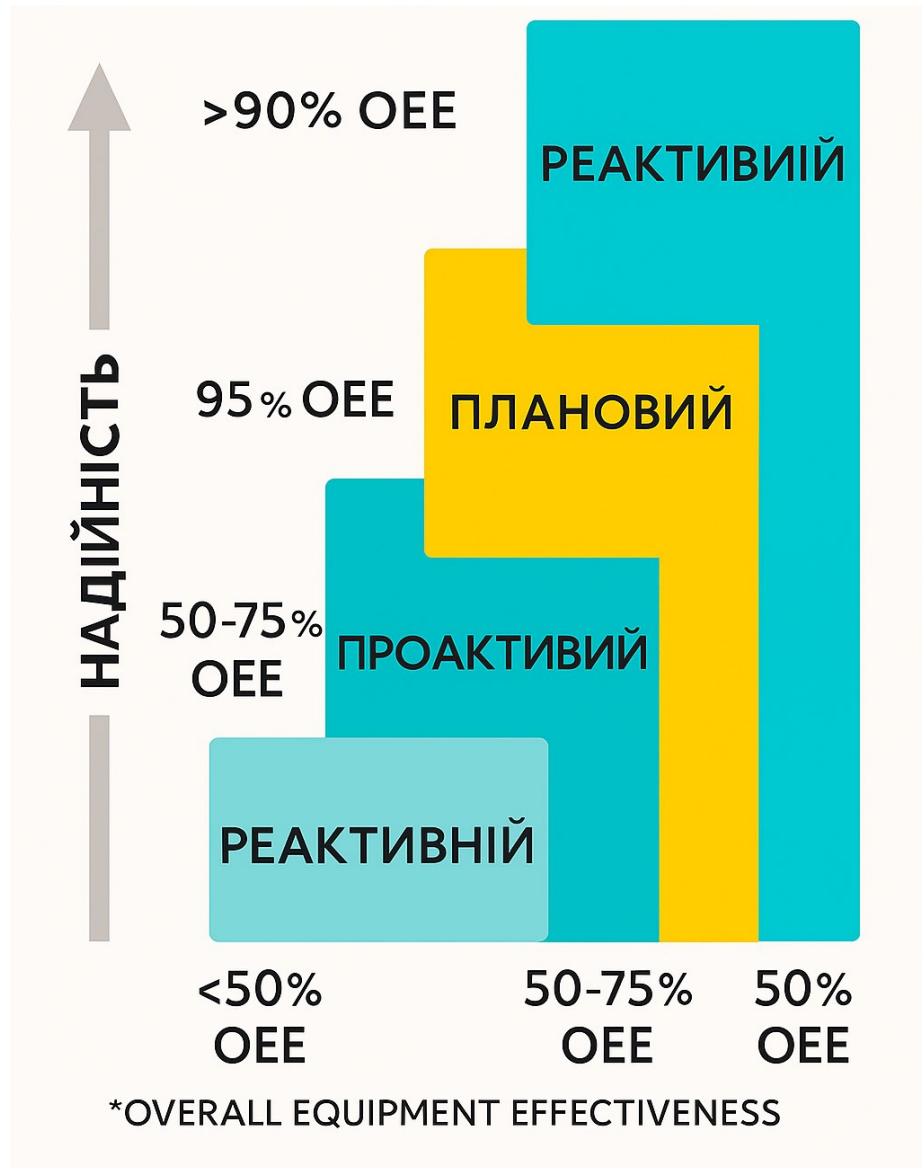


Рисунок 3.1 – Ефективність прогнозного технічного обслуговування

3.1.2 Моделі оцінки стану системи

Для оцінки поточного стану обладнання за допомогою моделей оцінки стану використовувалися підходи, засновані на даних. Моделі можуть прогнозувати термін служби обладнання, що залишився (RUL), і організації можуть планувати технічне обслуговування на основі фактичних потреб, а не фіксованих інтервалів. Завдяки цій стратегії графіки технічного обслуговування оптимізуються, а термін служби обладнання продовжується.

3.1.3 Аналіз ризиків

Визначення пріоритетів дій з технічного обслуговування критично залежить від аналізу ризиків, який аналізує розподіл ймовірності та впливу можливих збоїв. Організації можуть використовувати аналіз дерева відмов (FTA) або аналіз режиму та наслідків відмов (FMEA) для виявлення критичних точок відмови та розподілу ресурсів для подолання найбільших ризиків.

3.1.4 Інтеграція з Інтернетом речей

Інтеграція технологій Інтернету речей допомагає вивести функції прогнозного обслуговування на рівень, де підключені пристрої надають дані для збору та аналізу. Ця взаємопов'язаність означає повну картину продуктивності системи та дії, які можна вжити на основі аналізу в режимі реального часу .

3.1.5 Системи підтримки рішень

Рекомендації щодо технічного обслуговування, що надаються системами підтримки прийняття рішень на основі штучного інтелекту, отримуються з аналізу даних та моделей прогнозування. Використовуючи дані, синтезовані з різних джерел, такі системи допомагають командам технічного обслуговування

приймати рішення щодо того, коли і як виконувати завдання з технічного обслуговування, тим самим підвищуючи ефективність операцій з технічного обслуговування.

3.1.6 Переваги прогнозування та підтримки

Переваги прогнозного обслуговування численні, зокрема:

- Зменшення витрат: організації мінімізують незаплановані простої, усувають непотрібні ремонти та, в свою чергу, значно зменшують витрати на технічне обслуговування.
- Підвищена надійність: система проактивно вирішує потенційні проблеми, що виникають, для підвищення надійності та терміну служби обладнання, а також зменшення кількості перерв у роботі.
- Підвищена безпека: Завдяки прогнозованому технічному обслуговуванню можна мінімізувати ризик нещасних випадків, оскільки обладнання використовується в безпечних умовах, як для персоналу, так і для активів.

Тенденція до інтеграції штучного інтелекту в прогнозування несправностей виникає з розвитком технологій, і будуть відкриті більш складні методи для підтримки систематичної цілісності цих складних систем. У майбутньому ці методи штучного інтелекту продовжуватимуть розвиватися та вдосконалюватися, оскільки бізнес і світ навколо нас стають все більш складними та взаємопов'язаними, а здатність галузей промисловості прогнозувати несправності стане реальністю.

3.2 Механізми самовідтворення систем

Зі швидкістю, з якою люди працюють у сучасному цифровому світі, надійність та ефективність систем є надзвичайно важливими. Потреба в механізмах самовідновлення, які вирішують проблему управління несправностями у складних системах, стала критично важливим рішенням. Ці

механізми дозволяють системам виявляти, діагностувати та виправляти несправності, значно зменшуючи час простою та втручання людини. У цьому документі задокументовано автоматизовані процеси відновлення, адаптовані стратегії відновлення та випадки, коли системи самовідновлення продемонстрували успіх.

Автоматизовані процеси відновлення.

Автоматизовані процеси відновлення є критично важливими, оскільки системи повинні автоматично відновлюватися після збоїв. Вони охоплюються кількома методами та стратегіями, які працюють разом для підвищення стійкості системи (див. рис. 3.2).

Виявлення та діагностика помилок є одним з основних складових автоматизованого відновлення. Робота починається з інструментів моніторингу, що використовують датчики та програмне забезпечення для безперервного відстеження продуктивності системи та аномалій у режимі реального часу. Використовуючи діагностичні алгоритми, засновані на машинному навчанні, системи можуть швидко виявляти першопричини несправностей та швидше реагувати, вживаючи заходів для їх усунення.

У парадигмі самовідновлення методи відновлення є однаково необхідними. Коли виявляється несправність, механізми відкату запобігають знищенню ресурсів внаслідок збою, повертаючись до попереднього стабільного стану. Одним з важливих аспектів резервування є те, що можна створити кілька повторюваних компонентів, які можуть зайняти місце, коли основні системи виходять з ладу, щоб забезпечити переривання обслуговування. Контрольні точки також зберігають стан системи через регулярні проміжки часу з точкою відновлення, з якої вони можуть відновлюватися після збоїв.

Можливості самовідновлення ще більше посилюються завдяки автономним обчисленням. Це передбачає самоконфігурацію, за якої системи адаптують свої конфігурації для покращення продуктивності та мінімізації ризику помилок. Крім того, розподіл та використання ресурсів постійно самооптимізуються для усунення проблем до їх виникнення.

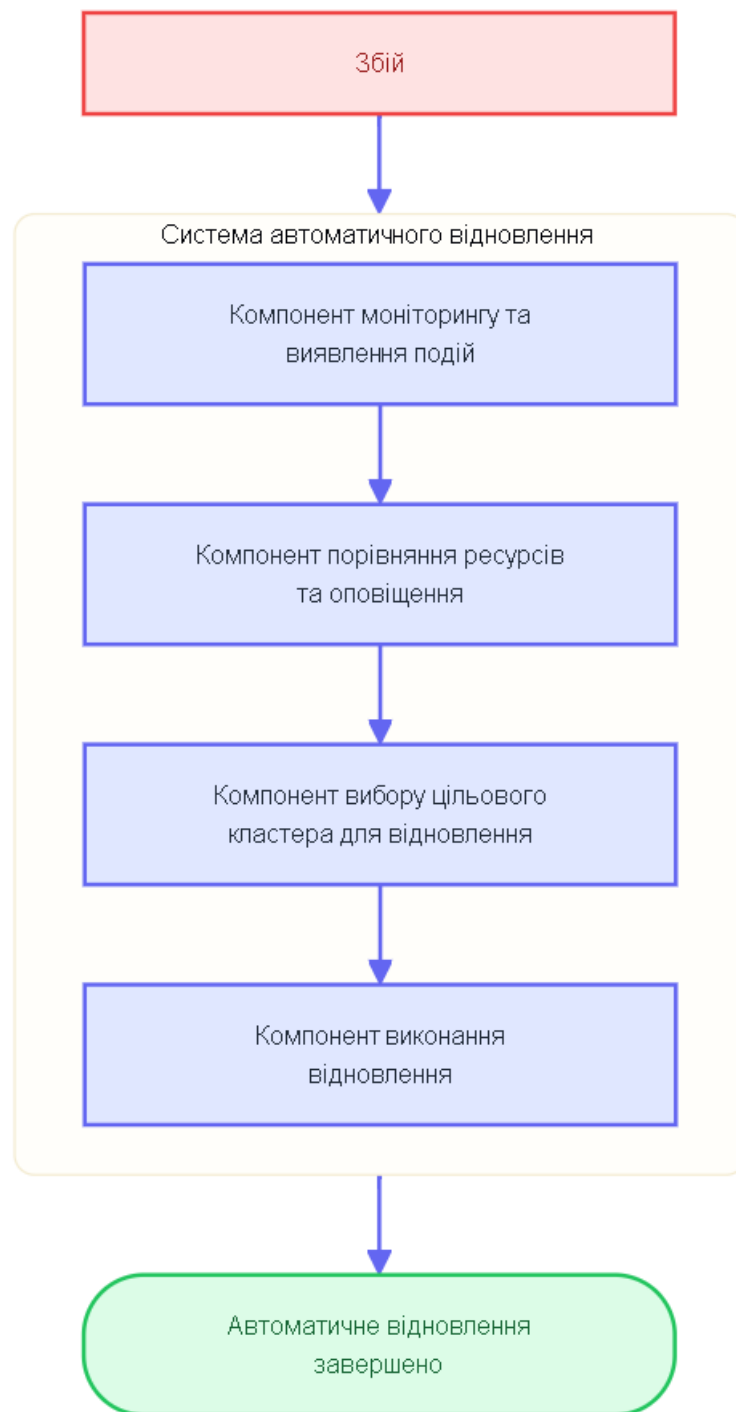


Рисунок 3.2 – Система автоматичного відновлення

Другою важливою частиною автоматизованого відновлення є автоматизоване керування виправленнями. Системи можуть автоматично застосовувати виправлення та негайно вирішувати відомі вразливості та помилки без втручання користувача. Перевага систем контролю версій полягає в

тому, що вони допомагають вам зберігати кілька версій програмного забезпечення та завжди повертатися до стабільного випуску, коли нещодавно розгорнуте оновлення не працює належним чином.

3.3 Адаптивні стратегії тестування

Адаптивні стратегії тестування є важливими для того, щоб процес тестування був ефективним та релевантним поточному операційному контролю. Дані в режимі реального часу та поведінка системи цих стратегій використовуються для динамічного коригування зусиль тестування, що покращує загальну надійність системи. Аналіз даних у режимі реального часу є центральним для адаптивного тестування. Системи можуть виявляти свою поведінку за різних умов, збираючи дані з кількох датчиків та журналів.

Алгоритми штучного інтелекту прогнозують потенційні сценарії тестування на основі цих даних, і можна досягти вищої якості тестування. Ще одним важливим компонентом є динамічна генерація тестових випадків. Це означає створення тестових випадків, таких як побудова системних умов, у міру їх використання. Тестування на основі ризиків додає подальшого вдосконалення до цього процесу, виконуючи тести на основі ймовірності та ризику несправностей, спрямовуючи обмежений ресурс у найбільш необхідні області. Чим більше процес тестування включає петлі зворотного зв'язку, тим ефективнішим буде процес тестування. Конвеєри безперервної інтеграції та розгортання (CI/CD) забезпечують інтеграцію зворотного зв'язку через конвеєр для команд, щоб вони могли налаштовувати тести відповідно до реальних результатів під час виконання. Організації можуть постійно оптимізувати свою стратегію тестування, тестуючи різні конфігурації тестування за допомогою альфа/бета-тестування або оптимізації.

Для подальшого підвищення ефективності можна додати самоналаштовувані параметри тестування. Навчаючи моделі машинного навчання на історичних результатах тестування та поточній продуктивності

системи, можна навчити їх адаптуватися до параметрів тестування. Штучний інтелект надає автоматизованій оркестрації тестів можливість координувати та визначати пріоритети тестів, упорядковуючи їх порядок для ефективного використання ресурсів та якомога швидше виконання критичних тестів.

3.4 Приклади з тематичних досліджень

Ефективність та використання впроваджень самовідновлення пояснюються на реальних прикладах. Крім того, ці тематичні дослідження ілюструють, як організації впровадили механізми самовідновлення для підвищення надійності та продуктивності системи.

Відомим прикладом самовідновлювальних систем Netflix є Chaos Monkey. Щоб перевірити стійкість інфраструктури Netflix, ми розробили інструмент, який випадковим чином завершує роботу екземплярів. Цей підхід непередбачуваний, тому система повинна адаптуватися та відновлюватися автономно, щоб гарантувати високу доступність та надійність. Netflix може імітувати збої контрольованим чином, виявляючи слабкі місця та зміцнюючи стійкість своєї системи.

Прикладом самовідновлення у великих масштабах є система Borg від Google. Ця система керування кластерами може автоматично перепланувати завдання для справних вузлів у разі виникнення збоїв самовідновлюваним способом. Високий час безвідмовної роботи Google зумовлений здатністю Borg виявляти збої та відновлюватися після них, одночасно заощаджуючи ресурси в масштабах своєї масивної інфраструктури.

Прикладом адаптивного управління ресурсами в хмарі є автоматичне масштабування Amazon Web Services (AWS). Ця функція автоматично масштабуватиме екземпляр на кількість екземплярів Amazon EC2 на основі запиту та перевірок справності. Завдяки автоматичному масштабуванню AWS додатки залишатимуться адаптивними, але водночас економічно ефективними,

забезпечуючи динамічне відновлення після збоїв та масштабування ресурсів відповідно до змінних потреб.

Service Fabric від Microsoft Azure надає надійну платформу для створення та розповсюдження мікросервісів з інтегрованою функцією самовідновлення. Завдяки перемиканню на резервний ПК, він забезпечує виняткову стійкість програм, автоматично виявляє збої та перерозподіляє робочі навантаження, щоб запобігти збоям під час виконання. Завдяки впровадженню цього проактивного методу робочі навантаження продовжують виконуватися, навіть коли виникають основні проблеми, що є важливим фактором для безперервності обслуговування. Це майбутнє механізмів самовідновлення, які змінять ландшафт управління системами та операційної досконалості на свою користь.

3.5 Системи самовідновлювального автоматизованого тестування на основі штучного інтелекту – проблеми та обмеження

Самовідновлювальні системи автоматизованого тестування на базі штучного інтелекту відкривають потенціал трансформаційних інновацій для підвищення надійності та ефективності програмного забезпечення. Тим не менш, існують труднощі та обмеження щодо їх впровадження, але вони виявляються неефективним та менш поширеним інструментом. Це має допомогти організаціям зрозуміти ці перешкоди та успішно використовувати ШІ для автоматизованого тестування. У цьому обговоренні ми розглянемо технічні труднощі, бар'єри впровадження та ширший вплив інтеграції ШІ в системи самовідновлення.

3.5.1 Технічні виклики

Технічна проблема номер один, з якою стикаються організації: інтеграція ШІ в існуючі системи. У випадку інтеграції ШІ зі застарілими системами виникає складність, пов'язана з інтеграцією ШІ з існуючими системами. Через зростаючу залежність від традиційних, застарілих технологій багато організацій стикаються

з проблемами сумісності в контексті вдосконалення алгоритмів ШІ. А ізолювані бази даних створюють ще одну перешкоду: системам ШІ потрібно бачити багато даних, щоб навчатися та приймати обґрунтовані рішення, але значна частина цих даних ізолювана між відділами або системами. Однак ця фрагментація може ускладнити збір та аналіз даних, і це погано впливає на ефективність наших моделей ШІ.

Ще однією важливою проблемою, з якою стикаються організації, є сумісність. Однак для самовідновлюваної системи інструменти штучного інтелекту повинні мати можливість безперешкодно взаємодіяти з іншими програмними компонентами. Однак досягти такого рівня сумісності складно, оскільки формати даних, протоколи та стандарти відрізняються залежно від системи. Завдяки гарній комунікації своїх можливостей, автоматизовані механізми самовідновлення за участю штучного інтелекту можуть повністю реалізувати свій потенціал, що призведе до неефективності та втрачених можливостей автоматизації.

Ще однією важливою проблемою під час впровадження фреймворків на основі штучного інтелекту є масштабованість. Багато алгоритмів ШІ, особливо екземпляри глибокого навчання, потребують ресурсомістких ресурсів та величезної обчислювальної потужності для виконання. Це особливо вузьке місце для організацій з обмеженою інфраструктурою, оскільки вони повинні задовольняти цю потребу в ресурсах. Механізми самовідновлення залежать від здатності отримувати дані в режимі реального часу. Однак масштабування рішень ШІ для обробки даних у режимі реального часу у великих та динамічних середовищах ефективно вимагає часу та зусиль. ШІ може вимагати більше існуючої інфраструктури, ніж він може підтримувати, і, ймовірно, потребуватиме дорогих оновлень або абсолютно нової системи для задоволення цих потреб.

Точність тестування є ключовою в автоматизації на основі штучного інтелекту, і багато речей можуть її порушити. Якість даних є ключем до якості даних, що подаються в моделі ШІ, але неправильні дані можуть призвести до

неточних прогнозів та ненадійних дій самовідновлення. Крім того, великою проблемою є упередженість моделі: як і люди, ШІ може відтворювати упередженості своїх навчальних даних, показуючи низькі результати перед обличчям прогнозування та виправлення. У динамічному середовищі умови швидко змінюються, і моделі ШІ повинні швидко навчатися, щоб підтримувати порядок. Ця потреба в швидкій адаптації може негативно вплинути на точність і надійність моделей, навчених на історичних даних, але є ненадійною в новій погодній ситуації.

3.5.2 Перешкоди для впровадження

Проблеми, що виходять за рамки технічних, перешкоджатимуть впровадженню самовідновлювальних систем автоматизованого тестування на основі штучного інтелекту. Цей процес включає багато економічних факторів. Впровадження технологій штучного інтелекту тягне за собою початкові витрати, які необхідно зменшити. Організації потребують величезних бюджетних інвестицій у нову інфраструктуру, навчання персоналу та придбання інструментів штучного інтелекту. Крім того, може бути складно чітко продемонструвати окупність інвестицій (ROI) для цих ініціатив, що ускладнює обґрунтування того, як покрити ці витрати для зацікавлених сторін. Ще більше ускладнюють фінансову оцінку та операційну стійкість постійні витрати на обслуговування, включаючи моніторинг та оновлення.

Такі умови не обмежуються лише фахівцями з обробки даних; організаційні фактори також впливають на перешкоди для впровадження фреймворків на основі штучного інтелекту. Ще однією сильною перешкодою є опір змінам співробітників та керівництва. Страх втратити роботу та подальше порушення узгоджених процесів. Процеси іноді змушують людей скептично ставитися до нових технологій. Стратегії управління змінами є ключем до подолання цього опору, і для подолання цього опору необхідна ефективна комунікація переваг штучного інтелекту. Крім того, дефіцит кваліфікованих

кадрів у компанії може значною мірою перешкодити запуску систем штучного інтелекту. Нестача компетентного персоналу для розробки, впровадження та підтримки фреймворків на базі штучного інтелекту може створити вузькі місця та обмежити адекватність цих видів діяльності.

Якщо організаційна культура не заохочує і не перешкоджає новим ідеям, впровадження структур, пов'язаних зі штучним інтелектом, може зіткнутися з певним опором. Ще однією проблемою ініціатив у сфері штучного інтелекту є забезпечення їх відповідності ширшим цілям та стратегіям організації. Проекти у сфері штучного інтелекту без чіткої стратегічної узгодженості, швидше за все, потребуватимуть ресурсів та підтримки, яких немає.

3.5.3 Майбутнє систем самовідновлювального автоматизованого тестування

Світ автоматизованого тестування за допомогою самовідновлення на основі штучного інтелекту стрімко розвивається, і в майбутньому відкриваються захопливі нові можливості. Функціональність та ефективність систем самовідновлення можна підвищити за допомогою найновіших інструментів та методологій у міру розвитку технологій штучного інтелекту. У цьому розділі обговорюються ці нові технології та виявляються важливі прогалини в дослідженнях, які можуть визначити напрямок розвитку систем самовідновлюваного автоматизованого тестування.

Пояснювальний ШІ (Explanatory Artificial Intelligence – XAI). Однією з найперспективніших технологій ШІ є пояснювальний ШІ, яка прагне зробити процес прийняття рішень за допомогою ШІ зрозумілим. XAI підвищує довіру, роблячи системи ШІ більш інтерпретованими, дозволяючи користувачам та зацікавленим сторонам розуміти процес прийняття рішень. Знання цього є важливим для налагодження, оскільки це дозволить нам виявляти та виправляти помилки моделей ШІ. Оскільки самовідновлювальні фреймворки ставатимуть

дедалі більш поширеними, прозорість автоматизованих систем стане необхідною для прийняття та довіри до самовідновлювальних систем.

Федеративне навчання – це ще один важливий розвиток моделей штучного інтелекту, що дозволяє навчатися на розподілених даних у кількох місцях без шкоди для конфіденційності. Однак проблема конфіденційності вирішується завдяки цьому децентралізованому підходу, оскільки дані залишаються локалізованими, що дуже важливо в середовищах тестування на основі даних. Здатність федеративного навчання стимулювати організації до створення надійних моделей штучного інтелекту, дотримуючись при цьому правил конфіденційності, робить його критично важливою технологією для появи самовідновлювальних фреймворків у майбутньому.

Додавання периферійних обчислень ще більше розширює можливості самовідновлювальних систем для обробки даних у режимі реального часу ближче до джерела. Це зменшує затримку та скорочує час виявлення та відновлення несправностей, що є критично важливим для забезпечення надійності системи. Периферійні обчислення також дозволяють розгортати самовідновлювальні системи в середовищах з обмеженими ресурсами, і таким чином збільшують потенціал застосування в різних галузях.

Завдяки новим можливостям, наприклад, квантовим комп'ютерам обчислювальна потужність розширюється для вирішення складних задач оптимізації, які традиційні комп'ютери не вирішують. Така здатність може суттєво вплинути на навчання та роботу моделей штучного інтелекту, а також підвищити точність та ефективність. Зрілість квантової технології можна було б використати для її інтеграції в самовідновлювальні фреймворки автоматизації та забезпечення нових рівнів продуктивності.

3.6 Технології самовідновлення

Майбутнє самовідновлювальних систем – це автономні обчислення. Також цей архетип включає можливості самокерування, які системи можуть

створювати, оптимізувати та захищати самі, потребуючи лише мінімального втручання людини. Автономні обчислення знімають операційне навантаження з організацій та підвищують стійкість системи, дозволяючи адаптивно реагувати на зміну умов та робочих навантажень.

Ще однією інноваційною технологією, яка може суттєво вплинути на самовідновлювальні системи, є цифрові двійники. Організації створюють віртуальні репліки фізичних систем для моделювання та прогнозування того, що станеться, коли – створення віртуальних реплік фізичних систем допомагає отримати уявлення про те, що станеться «коли». Моделювання цих рідкісних подій сприяє оптимізації, оскільки отримані знання можна застосовувати для підвищення стійкості та продуктивності системи. Цифрові двійники сприяють проактивному управлінню потенційними проблемами у фізичному світі, які насправді не виникають. Технологія блокчейн також надає особливу перевагу самовідновлювальним системам. Цілісність даних забезпечується шляхом надання незмінних записів про стан системи та транзакції для точної діагностики несправностей. Той факт, що блокчейн веде нас до розподіленого характеру співпраці між розподіленими системами, робить єдині точки відмови неіснуючими, а загальну надійність системи кращою.

3.7 Потреба в подальших дослідженнях

Хоча багатообіцяючих технологічних досягнень багато, для оптимальної реалізації самовідновлювальних систем необхідно заповнити кілька прогалів у дослідженнях. Оптимізація моделей є однією з найважливіших областей. Майбутня робота повинна включати розробку моделей штучного інтелекту, які можуть навчатися новим умовам і наборам даних і не потребують перенавчання з нуля. Як і у випадку з більшістю самовідновлювальних систем, необхідно забезпечити, щоб ці моделі залишалися на високому рівні продуктивності, надаючи зворотний зв'язок, щоб дозволити моделям навчатися, незважаючи на неочікувані зміни або неповні дані.

Також необхідним залишається подальше дослідження стратегій інтеграції. Дослідження повинні автоматично включати самовідновлювальні фреймворки без суттєвого порушення існуючої системи. Для забезпечення ширшої взаємодії та корисності цих фреймворків у різноманітних середовищах важливою буде сумісність на різних рівнях гетерогенності.

Ще однією важливою сферою досліджень є рішення масштабованості. Ключем до успіху самовідновлювальних фреймворків на основі штучного інтелекту буде дослідження методів оптимізації обчислювальних ресурсів та ресурсів зберігання даних у великомасштабних розгортаннях. Розробка розподілених архітектур дозволить розширити можливості переведення систем на розподілену обробку та відмовостійкість. Крім того, успішне використання систем штучного інтелекту вимагає орієнтованого на користувача дизайну, який призводить до ефективної співпраці між системами штучного інтелекту та їхніми аналогами «людина-оператор». Це дослідження має бути зосереджено на розробці та впровадженні інтуїтивно зрозумілих інтерфейсів та інструментів, які спрощують технології самовідновлення, дозволяючи користувачам використовувати та керувати цією системою.

З розвитком штучного інтелекту необхідно також враховувати кілька етичних та соціальних міркувань. Пом'якшення упередженості є одним з нагальних питань. Необхідно провести роботу з аналізу та викорінення упередженості в моделях штучного інтелекту, щоб результати були справедливими та рівноправними. Довіра до самовідновлювальних систем вимагає структур для підзвітності та прозорості в автономному прийнятті рішень.

Однак конфіденційність даних все ще залишається великою проблемою у системах, що використовуються штучним інтелектом. Визначення методів безпечної обробки та зберігання даних, а також захист конфіденційної інформації під час її збору та аналізу є вкрай важливим. Крім того, регулювання того, як системи штучного інтелекту використовують дані для дотримання

правових та етичних вимог щодо обміну даними, допоможе обмежити небезпеки, пов'язані з конфіденційністю даних.

Ще одним важливим міркуванням є вплив автоматизації на робочу силу. І як широке впровадження технологій самовідновлення змінить посадові ролі. Ті, хто шукає роботу, повинні будуть зіткнутися з переглядом поточного підходу власників систем до вирішення невизначеностей планування під час виконання проектів. Наша критична оцінка цих технологій необхідна для того, щоб забезпечити позитивні результати для всіх зацікавлених сторін, пов'язані з їх впровадженням у різні галузі.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Методи та засоби психофізіологічного розвантаження як допоміжний процес в розробці ПЗ

Для галузі ІТ інформаційна культура є необхідною умовою виживання, тому що зміна технологій в розробці програмного забезпечення відбувається кожні 6-8 місяців, а інвестиції на підготовку персоналу і освоєння нової технології величезні і великих компаніях варіюються від 1,5 до 2 млрд. доларів на рік [44].

Аналіз свідчить, що інформатизація та інтеграція комунікаційного простору України сприяє різкому підвищенню інформаційної та професійної компетентності, ділової активності, стимулюванню конкуренції, створенню інноваційних підприємств та організацій, нових робочих місць, зниженню витрат на утримання управлінського апарату [44]. Поряд із задачами і здобутками окреслилися негативи використання інформаційних технологій:

1) надмірне інформаційне навантаження, суть якого полягає у тому, що кількість корисної інформації, яка надходить до мережі, перевищує психофізіологічні можливості її сприйняття людиною;

2) велика кількість інформації, яка сприймається, але не є корисною для фахівців в даний момент;

3) інформаційний голод, причиною якого є саме надлишок інформації, викликаний інформаційним перенавантаженням;

4) «інформоманія» як хвороба людини, яка робить останню знеособленою, залежною від перебування в інформаційному просторі і роботи з комп'ютером і чому вона віддає перевагу, уникаючи «живого» спілкування з людьми;

5) поява «кіберспільнот», що за своїми соціокультурними характеристиками набагато ближчі до представників інших культур у

глобальному інформаційному просторі, ніж до своєї етнонаціональної спільноти чи решти населення, не охопленого Інтернетом;

б) індивідуалізм і дегуманізація способу життя «мешканців» Інтернету – відсутність готовності ділитися своїми знаннями.

Слід розуміти, що комп'ютерні технології істотно впливають на життєдіяльність людини, припускаючи глобалізацію і технократизацію суспільства. Але в ще більшій мірі цей вплив поширюється безпосередньо на центральну нервову систему, яка звикає працювати в дуже інтенсивному режимі багатозадачності, де вже переважають не тривалі логічні роздуми, а інтуїтивно-реактивні ланцюжки розумових формулювань у зв'язку з величезним обсягом оброблюваної щодня інформації, кількість якої зростає за експоненціальною швидкістю. Виникає припущення, що саме збільшення обсягу інформації та прискорення її обробки людиною може згубно вплинути на розвиток розумових здібностей людини.

У [44] наведено перелік протипоказань з боку органів зору та загальних (соматичних) протипоказань, які забороняють роботу на ЕОМ, а також комплекс вправ для поліпшення здоров'я і підвищення працездатності.

Таким чином, за умови високого рівня робіт з ЕОМ рекомендується психофізіологічне розвантаження у спеціально обладнаних приміщеннях (кімнати психофізіологічного розвантаження) під час регламентованих перерв або в кінці робочого дня.

При проведенні сеансів психофізіологічного розвантаження рекомендується використовувати деякі елементи методу аутогенного тренування, який ґрунтується на свідомому застосуванні комплексу взаємопов'язаних прийомів психічної саморегуляції й виконанні нескладних фізичних вправ зі словесним самонавіюванням.

У рекомендованому сеансі, який має проводитися у кімнаті психофізіологічного розвантаження з відповідним інтер'єром та кольоровим оформленням, виділяють такі три періоди, або фази:

Перший – абстрагування працівників від виробничої обстановки – відповідає фазі залишкового збудження. Звучить повільна мелодійна музика, пташиний спів. Обравши зручну позу, працівники адаптуються і психологічно готуються до наступних періодів.

Другий – заспокоєння – відповідає фазі відновлювального гальмування. Пропонується показ фото слайдів із зображеннями квітучого луку, березового гаю, гладенької поверхні ставка тощо. Через навушники транслюється спокійна музика.

Як функціональне освітлення застосовують зелене світло. Яскравість світла має поступово знижуватися впродовж періоду заспокоєння, а наприкінці його світло вимикається зовсім на 1–2 хв. Екран теж гасне.

Третій – активізація – відповідає фазі підвищеної збудженості.

На початку періоду світло вимкнене, через певний час на екрані з'являється червона пляма, розміри й яскравість якої поступово збільшуються.

Наприкінці періоду звучить бадьора музика. Вимовляються тричі мобілізуючі формули аутогенного тренування, яким мають передувати глибокий вдих та довгий глибокий видих.

Після сеансів психофізіологічного розвантаження у працівників зменшується відчуття втоми, з'являється бадьорість, хороший настрій. Загальний стан відчутно поліпшується.

4.2 Попередження аварій на виробництвах із застосуванням хлору

Хлор є частиною таблиці хімічних елементів і розташовується в ній під номером 17. У природі він зустрічається виключно у формі газу. Найчастіше він має специфічний зелений з жовтим переливом колір. Цей елемент важчий за повітря в 2,5 рази, тому накопичується в підвалах будинків, а на пересіченій місцевості в ярах і низинах. У воді ж хлор розчиняється без сліду і його наявність помітно тільки при великій концентрації (за рахунок специфічного запаху) [45].

В організмі людини в середньому міститься 95 г хлору. За добу людина споживає 5-10 г хлору (кухонна сіль). Він потрібен для вироблення в шлунку соляної кислоти, яка сприяє травленню і знищенню хвороботворних бактерій. Добова потреба хлору для людини становить 800 мг.

Хлор широко застосовується на виробництві, на його основі виготовляють отрутохімікати, розчинники, засоби для дезінфекції та миття, медикаменти. Хлор використовується в кольоровій металургії, у виготовленні пластмас тощо. Також хлор з успіхом застосовується і в побуті для очищення, відбілювання, прання. Завдяки незначним витратам і досить високій ефективності дезінфекції, хлор активно використовується для очищення і знезараження води в плавальних басейнах і питної водопровідної води.

Отруєння хлором можливе в разі:

- перевищення максимально допустимих концентрацій хлору для знезараження води в трубопроводі (сильний запах хлору);
- наявність хлору у великій кількості у воді басейну і часте купання в ньому;
- відбілювання і прання в закритому не провітрюваному приміщенні;
- аварії на підприємстві;
- використання хлору в якості зброї масового ураження.

В організм хлор потрапляє через слизові оболонки дихальної і травної систем, шкіру.

Ознаки отруєння хлором. До перших ознаках отруєння хлором відносяться:

- дискомфорт і подразнення слизової дихальних шляхів;
- підвищене слиновиділення і спазм голосових зв'язок;
- кашель і утруднене дихання;
- відчуття різі та печіння в очах, сльозотеча;
- нудота і гіркота у роті;
- головні болі і можливі судоми.

При попаданні на шкірний покрив або слизові спостерігається значний свербіж і гіперемія (почервоніння), вірогідні підшкірні крововиливи без пошкодження цілісності шкіри.

Тяжкість патологічного процесу та симптоми отруєння хлором знаходяться в прямій залежності від дози отруйної речовини (хлору) і тривалості його дії.

До прибуття медиків слід надати домедичну допомогу потерпілому:

- усунути джерело надходження отрути в організм – вивести або винести потерпілого поза зону дії отруйної речовини. При цьому необхідно пам'ятати про безпеку рятувальника – застосування марлевої маски або респіратора.

- забезпечити доступ чистого повітря;

- зняти забруднений одяг і теплою (не гарячою) водою промити контактуючі ділянки шкіри.

- у разі перорального надходження (проковтування) хлорвмісних рідин, потрібно промити шлунок. Промивати краще через зонд, або можна викликати блювання після рясного пиття.

- у разі пошкодження очей, промивання великою кількістю води або слабким розчином соди для зняття подразнення;

- полоскання ротової порожнини та носа содовими розчинами для мінімізації ушкодження слизових оболонок, застосування інгаляцій з додаванням соди для полегшення кашлю.

До профілактичних заходів отруєння хлором належать:

- забезпечення належних умов праці відповідно до санітарно-технічних вимог (вентиляція, провітрювання, справне обладнання);

- використання індивідуальних засобів захисту при роботі з хімікатами на виробництві;

- регулярні перевірки концентрацій хлору в повітрі робочої зони;

- проведення профілактичних медичних оглядів для виявлення схильності (доклінічних форм) і хронічних захворювань;

– дотримання вимог безпеки у використанні хлорвмісних рідин в побуті.

Суб'єкт господарської діяльності зобов'язаний забезпечити працівників хлорних об'єктів спеціальним одягом, спеціальним взуттям та іншими засобами індивідуального захисту відповідно до Положення про порядок забезпечення працівників спеціальним одягом, спеціальним взуттям та іншими засобами індивідуального захисту:

а) для захисту органів дихання – фільтруючими протигазами, ізолюючими дихальними апаратами та ізолюючими костюмами;

б) для захисту очей – захисними окулярами;

в) для захисту шкіри від їдких речовин – гумовими або прогумованими рукавицями, гумовими чоботами або шкіряними черевиками, сукняними костюмами.

Враховуючи значний ризик і широке застосування хлору, тяжкість ураження і високу можливість летального наслідку, у кожного повинен бути сформований алгоритм дій і чітка позиція – попередити отруєння легше і доцільніше, ніж лікувати і боротися з його наслідками.

ВИСНОВКИ

У цій кваліфікаційній роботі ми провели дослідження про удосконалення системи самовідновлюваного автоматизованого тестування на основі штучного інтелекту, яке є важливим кроком у тестуванні програмного забезпечення, оскільки воно пришвидшує процес і підвищує надійність.

У дослідженні виділено кілька важливих висновків. Інтеграція штучного інтелекту в автоматизоване тестування допомагає прогнозувати та відновлювати помилки в режимі реального часу. Це дозволяє моделям машинного навчання, таким як нейронні мережі, дерева рішень тощо, виявляти закономірності складних наборів даних. Наші механізми самовідновлення, у свою чергу, автоматизують виявлення, діагностику та виправлення помилок, зменшуючи залежність від людського втручання. Деякі технології, включаючи автономні обчислення та цифрових двійників, можна використовувати для створення стійких систем. Незважаючи на проблеми інтеграції та масштабованості, початкові витрати та проблеми опору організацій змінам залишаються. Новітні технології, такі як пояснимий штучний інтелект та блокчейн, можуть вирішити існуючі обмеження в сучасних системах соціальних мереж.

Існує кілька ключових рекомендацій, на яких повинні зосередитися зацікавлені сторони галузі. По-перше, ви повинні інвестувати в інфраструктуру, підвищити свою конкурентоспроможність та розглянути хмарні рішення. Не менш важливим є сприяння культурі інновацій шляхом сприяння постійному навчанню та забезпечення навчання для заповнення прогалин у навичках у сфері штучного інтелекту. Можна покращити конфіденційність та безпеку даних, отримуючи якісні дані за допомогою надійних практик управління та подальшого використання федеративного навчання. Крім того, впровадження системи самовідновлення буде ефективнішим, якщо буде сприяти співпраці між ІТ-підрозділами та бізнес-підрозділами, а також буде сприяння партнерству з постачальниками ІІІ та дослідницькими установами.

Це багатообіцяюче майбутнє штучного інтелекту для автоматизованого тестування з величезним впливом на галузі. У міру того, як фахівці зі штучного інтелекту стають зрілими, вони розроблятимуть передові рішення для прогнозування та відновлення після помилок, що дозволить організаціям підвищити якість програмного забезпечення та операційну ефективність. Щоб отримати ці переваги на максимальному рівні, необхідно вирішувати проблеми, а також використовувати можливості для інновацій. Галузь тестування програмного забезпечення ось-ось вступить у захопливу епоху еволюції, і організації можуть прокласти цей шлях, витрачаючи кошти на інфраструктуру, безперервне навчання та співпрацю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ihor, B., Oleksii, D., Aleksandr, K., Nataliia, K., Oleksandr, M., & Volodymyr, P. (2019, January). Multicriteria choice of software architecture using dynamic correction of quality attributes. In International Conference on Computer Science, Engineering and Education Applications (pp. 419-427). Cham: Springer International Publishing.
2. Bodnarchuk, I., Lisovyi, V., Kharchenko, O., & Galai, I. (2018, September). Adaptive method for assessment and selection of software architecture in flexible techniques of design. In 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT) (Vol. 1, pp. 292-297). IEEE.
3. Kharchenko, A., Raichev, I., Bodnarchuk, I., & Matsiuk, O. (2021, October). The Survey of Global Software Design Processes. In 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T) (pp. 291-294). IEEE.
4. Волович, В., Береженко, Б. М., & Боднарчук, І. О. (2022). Задача проєктування програмної архітектури в процесах забезпечення якості. Матеріали Х науково-технічної конференції „Інформаційні моделі, системи та технології “Тернопільського національного технічного університету імені Івана Пулюя, 104-106.
5. Боднарчук, І., Харченко, О., Хоміцький, Б., & Шимчук, Г. (2019). Проєктування архітектури програмних систем в проєктах з гнучкими методами управління. Матеріали XXI наукової конференції Тернопільського національного технічного університету імені Івана Пулюя, 46-48.
6. A. K. Sahoo, A. K. Tripathy, "Fault Detection and Recovery in Self-Healing Networks using Artificial Intelligence Techniques", International Journal of Advanced Computer Science and Applications, vol. 11, no. 2, 2020.

7. Z. Wang, L. Zhu, Y. Xiao, "Fault Detection and Recovery for Self-Healing Networks Based on Artificial Intelligence," 2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, 2018, pp. 690-694.
8. J. Tang, J. Wang, W. Huang, "Artificial intelligence-based fault detection and recovery strategy for self-healing network," 2018 37th Chinese Control Conference (CCC), Wuhan, China, 2018, pp. 6512-6516.
9. L. Xiao, W. Zhou, Z. Wang, "Research on Fault Detection and Recovery of Self-healing Network Based on Artificial Intelligence," 2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 2018, pp. 174-177.
10. K. Liu, C. Gao, Z. Fan, "Fault Detection and Recovery in Self-Healing Networks Using Reinforcement Learning," 2020 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chengdu, China, 2020, pp. 293-298.
11. H. Zhang, Y. Wang, H. Yang, "Fault Detection and Recovery for Self-Healing Network Based on Machine Learning," 2021 International Conference on Mechatronics, Control and Robotics (ICMCR), Wuhan, China, 2021, pp. 114-118.
12. Y. Chen, J. Zhang, Y. Liu, "Deep Learning Based Fault Detection and Recovery for SelfHealing Networks," 2020 IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 2020, pp. 2427-2431.
13. Q. Li, W. Liu, S. Li, "Artificial Intelligence Based Fault Detection and Recovery Strategy for Self-Healing Network," 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), Rotorua, New Zealand, 2019, pp. 136-143.
14. H. Li, Y. Wang, Y. Sun, "Fault Detection and Recovery in Self-Healing Networks Based on Artificial Neural Network," 2021 5th International Conference on Electronic Information Technology and Computer Engineering (EITCE), Chengdu, China, 2021, pp. 96-100.

15. M. Liu, C. Hu, L. Zhai, "A Novel Fault Detection and Recovery Method for Self-healing Network Based on Artificial Intelligence," 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), Chengdu, China, 2020, pp. 1-5.
16. Y. Zhang, X. Hu, H. Sun, "Fault Detection and Recovery in Self-healing Networks Based on Artificial Intelligence," 2021 3rd International Conference on Computer Communication and the Internet (ICCCI), Harbin, China, 2021, pp. 101-106.
17. J. Wang, S. Zhou, Y. Gu, "Fault Detection and Recovery in Self-healing Networks Based on Reinforcement Learning," 2020 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chengdu, China, 2020, pp. 524-529.
18. Battina, D. S. (2019). Artificial intelligence in software test automation: A systematic literature review. *International Journal of Emerging Technologies and Innovative Research* (www.jetir.org| UGC and issn Approved), ISSN, 2349-5162.
19. Khankhoje, R. (2023). An In-Depth Review of Test Automation Frameworks: Types and Trade-offs. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, 3(1), 55-64.
20. Pelluru, K. (2024). AI-Driven DevOps Orchestration in Cloud Environments: Enhancing Efficiency and Automation. *Integrated Journal of Science and Technology*, 1(6), 1-15.
21. Jiménez-Ramírez, A., Chacón-Montero, J., Wojdysky, T., & González Enríquez, J. (2023). Automated testing in robotic process automation projects. *Journal of Software: Evolution and Process*, 35(3), e2259.
22. Liu, Z., Chen, C., Wang, J., Chen, M., Wu, B., Che, X., ... & Wang, Q. (2023). Chatting with gpt-3 for zero-shot human-like mobile automated gui testing. *arXiv preprint arXiv:2305.09434*.
23. Schäfer, M., Nadi, S., Eghbali, A., & Tip, F. (2023). An empirical evaluation of using large language models for automated unit test generation. *IEEE Transactions on Software Engineering*, 50(1), 85-105.

24. Feldt, R., Kang, S., Yoon, J., & Yoo, S. (2023, September). Towards autonomous testing agents via conversational large language models. In 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE) (pp. 1688-1693). IEEE.
25. Kumar, S. (2023). Reviewing software testing models and optimization techniques: an analysis of efficiency and advancement needs. *Journal of Computers, Mechanical and Management*, 2(1), 43-55.
26. Pargaonkar, S. (2023). A Study on the Benefits and Limitations of Software Testing Principles and Techniques: Software Quality Engineering.
27. Elemam, S. M. (2018). Pragmatic Competence and the Challenge of Speech Expression and Precision (Master's thesis, University of Dayton).
28. Kothandapani, H. P. (2020). Application of machine learning for predicting us bank deposit growth: A univariate and multivariate analysis of temporal dependencies and macroeconomic interrelationships. *Journal of Empirical Social Science Studies*, 4(1), 1-20.
29. Kothandapani, H. P. (2019). Drivers and barriers of adopting interactive dashboard reporting in the finance sector: an empirical investigation. *Reviews of Contemporary Business Analytics*, 2(1), 45-70.
30. Kothandapani, H. P. (2021). A benchmarking and comparative analysis of python libraries for data cleaning: Evaluating accuracy, processing efficiency, and usability across diverse datasets. *Eigenpub Review of Science and Technology*, 5(1), 16-33.
31. Rahman, M.A., Butcher, C. & Chen, Z. Void evolution and coalescence in porous ductile materials in simple shear. *Int J Fracture*, 177, 129–139 (2012). <https://doi.org/10.1007/s10704012-9759-2>.
32. Rahman, M. A. (2012). Influence of simple shear and void clustering on void coalescence.
33. Alam, H., & De, A., & Mishra, L. N. (2015). *Spring, Hibernate, Data Modeling, REST and TDD: Agile Java design and development* (Vol. 1).

34. Ahuja, A., & Gupta, M. (2024). Optimizing Predictive Maintenance with Machine Learning And Iot: A Business Strategy For Reducing Downtime And Operational Costs.
35. Al Bashar, M., Taher, A., & Johura, F. T. (2019). Quality Control and Process Improvement in Modern Paint Industry.
36. Al Bashar, M., Taher, M. A., Islam, M. K., & Ahmed, H. (2024). The Impact of Advanced Robotics and Automation on Supply Chain Efficiency in Industrial Manufacturing: A Comparative Analysis Between the US and Bangladesh. *Global Mainstream Journal of Business, Economics, Development & Project Management*, 3(03), 28-41.
37. Ahmed, H., Al Bashar, M., Taher, M. A., & Rahman, M. A. (2024). Innovative Approaches To Sustainable Supply Chain Management in the Manufacturing Industry: A Systematic Literature Review. *Global Mainstream Journal of Innovation, Engineering & Emerging Technology*, 3(02), 01-13.
38. Vaithianathan, M. (2024). Real-Time Object Detection and Recognition in FPGA-Based Autonomous Driving Systems. *International Journal of Computer Trends and Technology*, 72(4), 145-152.
39. Vaithianathan, M., Patil, M., Ng, S. F., & Udkar, S. (2023). Comparative Study of FPGA and GPU for High-Performance Computing and AI. *ESP International Journal of Advancements in Computational Technology (ESP-IJACT)*, 1(1), 37-46.
40. Vaithianathan, M., Patil, M., Ng, S. F., & Udkar, S. (2024). Integrating AI and Machine Learning with UVM in Semiconductor Design. *ESP International Journal of Advancements in Computational Technology (ESP-IJACT) Volume, 2*, 37-51.
41. Zhu, Y. (2023). Beyond Labels: A Comprehensive Review of Self-Supervised Learning and Intrinsic Data Properties. *Journal of Science & Technology*, 4(4), 65-84.
42. Pei, Y., Liu, Y., & Ling, N. (2023, December). MobileViT-GAN: a generative model for low bitrate image coding. In *2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)* (pp. 1-5). IEEE.

43. Pei, Y., Liu, Y., Ling, N., Ren, Y., & Liu, L. (2023, May). An end-to-end deep generative network for low bitrate image coding. In 2023 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-5).

44. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин МОЗ України від 10.12.1998 № 7. // Офіційний сайт Верховної Ради України. – [Електронний ресурс]. – Режим доступу <https://zakon.rada.gov.ua/rada/show/v0007282-98>

45. Бідяк О. Профілактика отруєння хлором. // Офіційний сайт управління держпраці в Тернопільській області. – [Електронний ресурс]. – Режим доступу <https://te.dsp.gov.ua/profilaktyka-otruyennya-hlorom/>

ДОДАТКИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя (Україна)
Університет імені П'єра і Марії Кюрі (Франція)
Маріборський університет (Словенія)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедимінаса (Литва)
Наукове товариство ім. Т.Шевченка

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник
тез доповідей

**XIII Міжнародної науково-практичної
конференції молодих учених та студентів**
11-12 грудня 2024 року



УКРАЇНА
ТЕРНОПІЛЬ – 2024

14. **Наталія Гавришко** **381**
 ЗАСОБИ НАЛАГОДЖЕННЯ СОЦІАЛЬНО-ПСИХОЛОГІЧНОГО КЛІМАТУ В
 ТРУДОВОМУ КОЛЕКТИВІ
15. **Наталія Цюзь** **383**
 ЕМОЦІЙНЕ ВИГОРАННЯ ПЕДАГОГІВ
16. **О.П. Буцик** **385**
 АНІМАЛОТЕРАПІЯ ЯК ЗАСІБ ПОДОЛАННЯ ТРИВОЖНОСТІ В УМОВАХ ВІЙНИ
17. **Олена Кухар** **387**
 ВПЛИВ ДИТЯЧИХ ЕМОЦІЙНИХ ТРАВМ НА ПСИХОСОЦІАЛЬНИЙ РОЗВИТОК
 ОСОБИСТОСТІ
18. **О. О. Гарматюк, Ю. П. Дишкант** **389**
 ОСОБЛИВОСТІ РОЗВИТКУ РЕГІОНАЛЬНОЇ СИСТЕМИ ПІДПРИЄМНИЦТВА НА
 ОСНОВІ ВНУТРІШНІХ РЕСУРСІВ СФЕРИ ПРИВАТНОГО БІЗНЕСУ У КОНТЕКСТІ
 ЗАСТОСУВАННЯ АНТИКРИЗОВОГО МЕНЕДЖМЕНТУ

СЕКЦІЯ: КОМП'ЮТЕРНО-ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ ЗВ'ЯЗКУ

1. **В.І. Нетреб'як; О.В. Тотосько** **391**
 ДОСЛІДЖЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ОСОБИ ЗА
 ДОПОМОГОЮ ПЛІК
2. **Ю.Д. Сидорко; Я.Ф. Балан; Д.П. Стухляк** **394**
 РОЗРОБКА ТА ДОСЛІДЖЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ
 ТЕХНОЛОГІЧНИМ ПРОЦЕСОМ ВИРОБНИЦТВА ЦЕГЛИ НА БАЗІ ПЛАТФОРМИ
 FIT
3. **Н.А. Шевченко, Г.В. Шимчук, У.А. Гарматюк** **396**
 ІНТЕГРАЦІЯ ТЕХНОЛОГІЇ МЕРЕЖЕВОЇ ВІРТУАЛІЗАЦІЇ VRF У
 БАГАТОКОЛІЙНУ МАРШРУТИЗАЦІЮ
4. **Н.А. Шевченко, Г.В. Шимчук, У.А. Гарматюк** **398**
 ОПТИМІЗАЦІЯ МЕТРИК МАРШРУТИЗАЦІЇ ДЛЯ ЗАБЕЗПЕЧЕННЯ СТІЙКОСТІ ТА
 НАДІЙНОСТІ IP-МЕРЕЖ
5. **Т. Ланевич** **400**
 АРАСНЕ КАФКА ТА RABBITMQ: ПОРІВНЯННЯ АРХІТЕКТУР ТА
 МОЖЛИВОСТЕЙ
6. **Т. Ланевич** **402**
 ЗАСТОСУВАННЯ ДОМЕННО-ОРІЄНТОВАНОГО ПРОЄКТУВАННЯ У ГНУЧКІЙ
 АРХІТЕКТУРІ
7. **Назар Осідак, Олександр Цвігун** **404**
 ЗАСТОСУВАННЯ ШІ ДЛЯ ЗАДАЧ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ
8. **Руслан Матяшук, Тарас Постолок** **405**
 ЗНАЧЕННЯ ВПРОВАДЖЕННЯ SPI ДЛЯ ПРОГРАМНИХ ПРОЄКТІВ
9. **Святослав Мельничук** **406**
 ПІДХОДИ ДО ПОРІВНЯННЯ ПРОДУКТИВНОСТІ РЕЛЯЦІЙНИХ ТА
 НЕРЕЛЯЦІЙНИХ БАЗ ДАНИХ
10. **Тетяна Щеснюк** **407**
 ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ГНУЧКИХ ТЕХНОЛОГІЙ РОЗРОБКИ
 ДЛЯ IoT-СИСТЕМ

УДК 004.05; 004.42; 004.9

Назар Осідак, Олександр Цвігун

Тернопільський національний технічний університет імені Івана Пулюя

ЗАСТОСУВАННЯ ШІ ДЛЯ ЗАДАЧ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ

Nazar Osidak, Oleksandr Tsvihun

APPLICATION OF AI FOR AUTOMATED TESTING TASKS

Штучний інтелект (ШІ) має великий потенціал у автоматизації тестування програмного забезпечення, але його застосування залежить від конкретних завдань. Існують типові ситуації, коли варто використовувати ШІ для автоматизації тестування, а є також випадки, коли традиційні методи можуть бути більш ефективними [1].

Штучний інтелект найкраще підходить для автоматизації тестів, де є потреба в адаптації, обробці великих обсягів даних, або в ситуаціях, коли специфікація тестів неповна або має суперечності. Традиційні методи автоматизації залишаються ефективними для стабільних, стандартизованих тестів.

Опишемо коротко завдання, які можна вирішити за допомогою ШІ. В першу чергу це тестування на основі поведінки користувача (User Behavior Testing): ШІ може аналізувати дані поведінки користувачів та автоматично генерувати тести, що імітують реальні сценарії взаємодії з продуктом. Наприклад, використання алгоритмів машинного навчання для створення сценаріїв тестування, що адаптуються під реальні патерни поведінки.

ШІ може автоматично визначати, які тестові сценарії виконувати в першу чергу, щоб мінімізувати час тестування при максимальному покритті. Таким чином вирішується задача оптимізації сценаріїв тестування. Алгоритми, зокрема на основі навчання з підкріпленням, можуть оптимізувати вибір тестів, залежно від результатів попередніх запусків і змін у коді. У випадках, коли програма працює з великими обсягами даних або складними системами, ШІ може застосовувати методи аналізу великих даних для пошуку потенційних дефектів у системах, що важко протестувати вручну. ШІ може допомогти в тестуванні програм, де специфікація неповна або містить суперечності. Застосування технологій, таких як нейронні мережі, може допомогти адаптувати тестування до змін у поведінці програми.

Існує також не менший список задач, які краще вирішувати без ШІ. Регресійне тестування для випадків, коли програма має добре визначену структуру та стабільний функціонал. Тоді традиційні методи автоматизації тестів (наприклад, за допомогою Selenium або JUnit) можуть бути більш ефективними. Це дозволяє виконувати стандартні перевірки без додаткових затрат на навчання моделей ШІ.

Статичні тести на навантаження або стрес-тести, які вимірюють стабільність і продуктивність програми, можуть бути виконані без ШІ. Вони потребують чітких та вимірних результатів, які краще піддаються автоматизації без залучення складних технологій. У випадку функціонального тестування прості функціональні тести, коли потрібно перевірити, чи працюють конкретні функції програми відповідно до вимог, можуть бути виконані без ШІ за допомогою традиційних інструментів автоматизації. У таких випадках ШІ не дасть значного покращення в продуктивності.

Література

1. Цісарук Д. В. Методи і засоби тестування програмного забезпечення комп'ютерних систем з використанням алгоритмів машинного навчання : кваліфікаційна робота магістра за спеціальністю „123 — комп'ютерна інженерія“ / Д. В. Цісарук. – Тернопіль, 2021. – 90 с.