

УДК 681.518

А.М. Ковтко; І.Р. Козбур; В.Б. Савків, к.т.н., доц.; Г.В. Козбур, к.т.н., доц.
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

АРХІТЕКТУРА АВТОМАТИЗОВАНОЇ СИСТЕМИ ДЛЯ ГЕНЕРАЦІЇ МОДУЛЬНИХ ТЕСТІВ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ ТА МУТАЦІЙНОГО ТЕСТУВАННЯ

A.M. Kovtko; I.R. Kozbur; V.B. Savkiv, Ph.D., Assoc. Prof.; H.V. Kozbur, Ph.D., Assoc. Prof.

ARCHITECTURE OF AN AUTOMATED SYSTEM FOR GENERATION OF MODULE TESTS BASED ON ARTIFICIAL INTELLIGENCE AND MUTATIONAL TESTING

За основу автоматизованої системи генерації модульних тестів [1, 2] використано мутаційне тестування, котре застосовується у якості механізму оцінювання ефективності наявних тестів. Важливим етапом цього процесу є формування детальних звітів про виконання, дані яких у подальшому використовують для моделей LLM з метою оптимізації тестових наборів.

У сучасних умовах стрімкого розвитку інформаційних технологій критично необхідно створювати гнучкі програмні системи, здатні адаптуватися до появи нових інструментів та мов програмування. Це зумовлено високою динамікою ринку, швидкою еволюцією мов програмування і фреймворків. Гнучкість забезпечує масштабованість, зменшує витрати на підтримку та модернізацію і сприяє довгостроковій стійкості програмного продукту [3].

Фреймворки мутаційного тестування реалізовані для більшості популярних мов програмування, однак вони суттєво відрізняються за механізмами генерації та фільтрації мутантів, так і структурою вихідних звітів. Наприклад, Stryker (JavaScript/TypeScript) формує детальні JSON- та HTML-звіти зі статусами мутантів і посимвольним підсвічуванням змінених фрагментів коду. Фреймворк PIT для Java застосовує іншу модель представлення результатів, агрегуючи дані за класами, методами та типами мутаторів, формуючи XML- та HTML-звіти з показниками на кшталт Mutation Score, Line Coverage і Test Strength. Таким чином, хоча семантичний зміст звітів переважно збігається, їхня структура суттєво різниться, що ускладнює подальшу обробку.

Через зазначену відмінність звітів аналіз результатів мутаційного тестування в уніфікованому вигляді неможливий без попередньої стандартизації даних. Перед інтеграцією LLM до процесу генерації або вдосконалення тестів необхідно привести різні формати звітів до спільного проміжного представлення. Уніфікований формат має містити щонайменше: ідентифікатор мутанта, вихідний код і його модифікацію, статус виконання, інформацію про відповідні тести або покриття, а також контекст класу чи модуля. Стандартизація забезпечує узгоджене трактування метрик, мінімізує втрату інформації під час парсингу та спрощує формування запитів. У результаті LLM зможе надійно аналізувати якість тестового покриття, визначати слабкі місця та генерувати нові або покращені тести на основі структурованих і порівнянних даних, незалежно від джерела їх походження.

З урахуванням важливості забезпечення масштабованості та гнучкості запропоновано наступний набір модулів для автоматизованої системи генерації тестів (Рис. 1):

1. Адаптери під різні типи мутаційного тестування. Саме ці модулі забезпечуватимуть стандартизацію отриманих звітів про виконання мутаційного тестування, забезпечуючи можливість швидко додати підтримку нового мутаційного фреймворку за потреби.
2. На основі отриманих даних від адаптерів та аналізу контексту кожного конкретного програмного модулю створюються запити до LLM.
3. LLM модель, що виконує запит, створений на попередньому етапі. Передбачається підтримка декількох моделей із урахуванням завдання системи.
4. Отримання нового / редагованого набору тестів, які передаються на крок 1 для повторного оцінювання за допомогою мутаційного тестування.

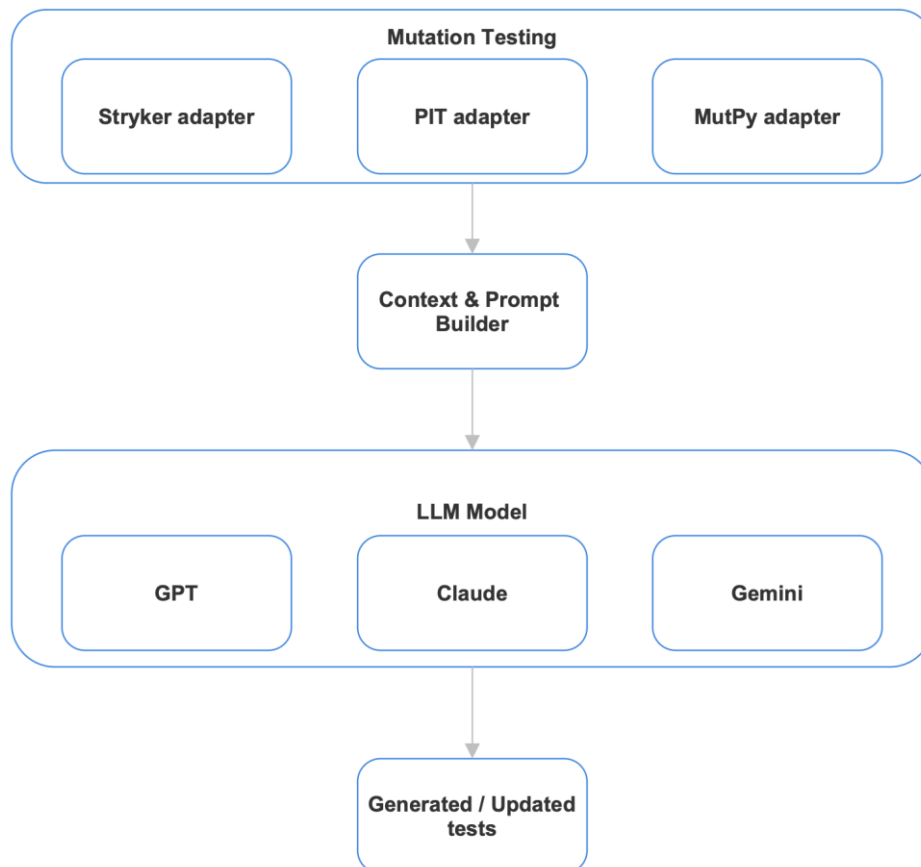


Рис. 1 Автоматизована система генерації модульних тестів.

Такий підхід забезпечує можливість використовувати різні фреймворки мутаційного тестування та LLM моделі з мінімальними змінами до решти складових системи.

Література

1. Kovtko, A., Savkiv, V., Kozbur, H., Kozbur, I., Trembach, R. Integration of mutation testing into unit test generation using large language models. *Proceedings of the CITI 2025: 4th International Conference on Computational Intelligence and Information Technology*, 2025.
2. Ковтко А.М. Автоматизована системи генерації модульних тестів на основі штучного інтелекту та мутаційного тестування / А.М. Ковтко, В.Б. Савків, І.Р. Козбур // Тези XIII МНПК „Актуальні задачі сучасних технологій“, 11-12 грудня 2024 року. 2024. С. 417–418. – (Комп’ютерно-інформаційні технології та системи зв’язку).

3. Tempero E., Blincoe K., Lottridge D. An Experiment on the Effects of Modularity on Code Modification and Understanding. *Proceedings of the 25th Australasian Computing Education Conference (ACE '23)*. 2023. P. 1–10. – ACM.

УДК 004.9

К. А. Кокурін

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ З ВИКОРИСТАННЯМ СЕНСОРА SDS011 ТА TELEGRAM-БОТА ДЛЯ СПОВІЩЕНЬ

К. А. Kokurin;

DEVELOPMENT OF AN AIR QUALITY MONITORING SYSTEM USING THE SDS011 SENSOR AND A TELEGRAM BOT FOR NOTIFICATIONS

У сучасних умовах зростання рівня урбанізації, інтенсивного розвитку промисловості та транспорту проблема контролю якості атмосферного повітря набуває особливої актуальності. Підвищені концентрації твердих частинок (PM2.5, PM10) негативно впливають на здоров'я населення, сприяють загостренню хронічних захворювань дихальної та серцево-судинної систем і погіршують загальний екологічний стан довкілля. Традиційні стаціонарні пости спостереження є дорогими в інсталяції та обслуговуванні, що обмежує їхню кількість та просторове покриття, особливо у невеликих населених пунктах та локальних зонах забруднення. Це зумовлює необхідність створення доступних, гнучких і масштабованих систем моніторингу на основі недорогих сенсорів.

Метою роботи є розробка мікросервісної системи моніторингу якості повітря з використанням сенсора SDS011 та інтегрованого Telegram-бота для оперативного інформування користувачів про рівень забруднення атмосферного повітря.

Запропоновано програмно-апаратну архітектуру розподіленої системи моніторингу, що базується на застосуванні сенсора твердих частинок SDS011 у поєднанні з обчислювальним модулем (Raspberry Pi), серверною частиною та сервісами доставки повідомлень. Збір даних здійснюється на низовому рівні через послідовний інтерфейс взаємодії із сенсором, після чого виміряні значення концентрацій PM2.5 та PM10 передаються до серверних мікросервісів для зберігання, попередньої фільтрації та аналітичного опрацювання.

Серверна частина реалізована у вигляді набору мікросервісів, які відповідають за приймання телеметрії, збереження даних у базі даних, формування агрегованих показників і виявлення фактів перевищення порогових значень. Використання мікросервісної архітектури забезпечує можливість масштабування системи, спрощує модифікацію окремих компонентів та інтеграцію нових джерел даних або аналітичних модулів без істотної зміни загальної структури.

Особливу увагу приділено розробці Telegram-бота, який забезпечує користувачам зручний доступ до актуальної інформації про якість повітря. Бот реалізує функції отримання поточних значень показників, перегляду останніх вимірювань, а також надсилання автоматичних сповіщень у випадку перевищення заданих порогів. Це підвищує оперативність прийняття рішень користувачами, сприяє формуванню екологічної обізнаності та залученню громадськості до контролю стану довкілля.