

O. C. Морозов, А. А. Яровий

Вінницький національний технічний університет, м. Вінниця, Україна

ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ПІДХОДІВ ДО ГЕНЕРАЦІЇ ТЕСТОВИХ СЕЛЕКТОРІВ ІЗ ЗАСТОСУВАННЯМ GNN У ПРОЦЕСІ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ВЕБДОДАТКІВ

У статті розглянуто проблему нестабільності тестових селекторів під час автоматизованого тестування вебдодатків. порушено питання адаптивності селекторів до змін у DOM-структурі, що є критично важливим для розроблення сучасних динамічних вебінтерфейсів. Виконано порівняльний аналіз трьох підходів до генерації селекторів: ручного (через Chrome DevTools), напівавтоматизованого (з використанням DevTools) та автоматизованого з використанням графових нейронних мереж (GNN).

Мета дослідження – визначити, який із підходів забезпечує найкраще співвідношення між точністю, повнотою, стабільністю та витратами часу на формування селекторів у реальних умовах тестування. У роботі застосовано експериментальний підхід, що передбачав тестування на 25 вебдодатках з різними типами DOM-структур: статичними, динамічними та SPA. Використано стандартні метрики якості: точність, повнота, F1-score, а також оцінено кількість помилок після змін у DOM та середній час генерації селектора.

Отримані результати свідчать про перевагу методу на основі GNN, зокрема у стабільності селекторів після оновлень інтерфейсу. Ручний підхід, хоча й демонструє високу точність, значно поступається за швидкістю та повнотою, а Chrome DevTools – хоч і швидкий, проте найменш надійний у динамічних середовищах. Отже, застосування графових нейронних мереж дає змогу створювати адаптивніші та надійніші рішення для автоматизованого тестування вебдодатків.

Крім того, дослідження демонструє доцільність подальшого розвитку підходів на основі GNN через інтеграцію механізмів self-supervised learning та Graph Attention Networks (GAT), що дасть змогу досягти ще більшої точності, масштабованості та продуктивності автоматизованих тестів у складних програмних системах. Запропонований підхід має потенціал для впровадження у фреймворки нового покоління для UI-тестування, що сприятиме поліпшенню якості цифрових продуктів на ринку.

Експериментально доведено доцільність застосування GNN для автоматизованої генерації селекторів, адаптованих до змін у DOM, а також виявлено, що такий підхід дає змогу зменшити кількість хибних спрацювань без втрати швидкодії. Отримані результати мають практичну цінність для підвищення надійності та ефективності систем автоматизованого тестування в умовах швидкозмінюваного інтерфейсу.

Ключові слова: програмне забезпечення, інформаційні технології, інтелектуальні системи, автоматизоване тестування, оптимізація, графові нейронні мережі, DOM-структура.

Вступ / Introduction

Автоматизоване тестування вже давно стало важливою частиною сучасного розроблення вебдодатків, особливо у великих і складних проєктах. Однією із основних проблем, з якою стикаються тестувальники, є нестабільність тестових селекторів. Це стосується тих випадків, коли через зміну DOM-структури, наприклад, після оновлення інтерфейсу, старі селектори більше не можуть коректно ідентифікувати елементи. Через це тести часто “падають” навіть без реальних змін у логіці роботи вебсайту.

Об’єкт дослідження – процес автоматизованого тестування вебдодатків.

Предмет дослідження – підходи до автоматизованої генерації тестових селекторів із застосуванням графових нейронних мереж.

Мета роботи – здійснити порівняльний аналіз підходів до генерації тестових селекторів на основі експериментального визначення вибраних показників ефективності у задачах автоматизованого тестування вебдодатків. Результати ґрунтуються на порівнянні за такими метриками: точність, повнота, F1-score, час генерації селектора та кількість помилок після змін інтерфейсу. Це дає змогу краще оцінити практичну користь кожного підходу, зокрема в умовах, коли інтерфейс часто оновлюється або змінюється динамічно.

Для досягнення цієї мети визначено такі основні завдання дослідження:

- проаналізувати підходи до автоматизованої генерації селекторів, зокрема із застосуванням графових нейронних мереж;
- здійснити експериментальні дослідження на прикладі реальних вебдодатків різних типів (статичних, динамічних, SPA);
- оцінити ефективність підходів та методів за метриками: точність, повнота, F1-score, час генерації та кількість помилок після змін у DOM (Document object model).

Матеріали та методи дослідження. У дослідженні використано вибірку із 25 вебдодатків різних типів (інтернет-магазини, новинні ресурси, медіасайти, стрімінгові сервіси та статичні вебсторінки), що охоплювали статичні, динамічні та SPA DOM-структури. Для порівняння застосовано три підходи до генерації селекторів: ручний (через Chrome DevTools), напівавтоматизований (Selenium IDE) та автоматизований із використанням графових нейронних мереж (GNN). HTML-код кожної сторінки перетворювали на графове подання, де вузли відповідали DOM-елементам, а ребра відображали їхні структурні та логічні зв'язки. Модель GNN навчалася на розмічених даних з урахуванням контексту елементів та їх атрибутів. Ефективність методів оцінювали за стандартними метриками (Precision, Recall, F1-score), а також за середнім часом генерації селекторів і кількістю помилок після змін у DOM. Для достовірності результатів використано усереднення показників всіх вибраних сайтів із подальшим аналізом варіації значень.

Аналіз останніх досліджень і публікацій. Графові нейронні мережі (GNN) демонструють високу ефективність у задачах номінації вебелементів. У дослідженні “The Klarna Product Page Dataset: Web Element Nomination with Graph Neural Networks and Large Language Models” представлено новий датасет з 51 701 вручну анотованою сторінкою продуктів з 8 175 e-commerce сайтів. Використання простого Convolutional GNN (GCN) перевершило складніші методи номінації елементів, а поєднання GCN з великими мовними моделями істотно підвищило точність номінації без необхідності додаткового навчання [1].

Інше дослідження, “Learning To Rank Resources with GNN”, запропонувало підхід до ранжування ресурсів у системах розподіленого інформаційного пошуку за допомогою GNN. Модель враховує як зв'язки між запитами та ресурсами, так і між самими ресурсами, що дає змогу досягти покращення точності на 6,4 % до 42 % порівняно з попередніми методами [2].

Дослідження “Feature Selection and Extraction for Graph Neural Networks” зосереджено на виборі та екстракції ознак для GNN. Автори запропонували алгоритм вибору ознак, який дає змогу зменшити розмірність вхідних даних без втрати точності, що важливо для ефективної роботи моделей GNN [3].

Ручний підхід до визначення селекторів вебелементів залишається поширеним серед тестувальників та розробників. Дослідження показали, що більшість

користувачів Selenium використовують локатори за ID, ім'ям, XPath та CSS. Однак лише 12,5 % респондентів застосовують патерн Page Object Model (POM), який сприяє зменшенню витрат на обслуговування тестів та підвищенню їхньої надійності [4, 5].

Selenium IDE – популярний інструмент для автоматизації вебтестування, особливо серед новачків. Він дає змогу записувати та відтворювати тести, використовуючи різні стратегії локаторів, такі як ID, ім'я, посилання, XPath та CSS. Однак автоматично згенеровані локатори можуть бути нестабільними на складних вебсайтах, тому потрібне ручне коригування або використання візуальних локаторів [6].

ChromeDevTools також використовують для визначення вебелементів. Він надає можливість переглядати та редагувати DOM, а також тестувати різні селектори в реальному часі. Цей інструмент корисний для швидкого аналізу та налагодження вебсторінок, але не забезпечує автоматизацію тестування на рівні Selenium IDE.

Однак існують інші джерела, які детально розглядають проблеми, пов'язані із використанням XPath та CSS-селекторів у вебавтоматизації. Наприклад, у статті “Test automation refactoring is a mess? Try CSS selector” автор описує, як зміни в DOM-структурі призвели до масових збоїв у тестах, що використовували XPath, і як перехід на CSS-селектори допоміг зменшити ці проблеми [7].

Також у статті “How to Choose Selectors for Automation to Make Your Life a Whole Lot Easier” розглянуто переваги та недоліки різних типів селекторів, зокрема рекомендації щодо вибору між XPath та CSS залежно від конкретних умов проєкту [8].

Результати дослідження та їх обговорення / Research results and their discussion

Для дослідження процесу застосування графових нейронних мереж у генерації тестових селекторів використано DOM-моделі з різних сайтів, які містили різноманітні вебсторінки із різними структурами (25 реальних сайтів) – інтернет-магазини, статичні сайти, новинні ресурси, медіаресурси, стрімінгові сервіси. Всі їх можна поділити на такі категорії:

- вебсторінки із незмінною структурою DOM, які використовують для перевірки точності та стабільності селекторів за відсутності динамічних змін;
- інтерфейси, що змінюються в реальному часі, ураховуючи елементи, які можуть бути додані або видалені без перезавантаження сторінки;
- вебресурси, де DOM-структура змінюється без повного перезавантаження сторінки, що є складною ситуацією для традиційних методів вибору селекторів, оскільки динамічні зміни інтерфейсу ускладнюють досягнення стабільності селекторів.

Вказаний підхід дав змогу перевірити ефективність методів у різних умовах і забезпечити широкую тестову базу для порівняння результатів.

Під час експерименту були використані такі метрики для вимірювання ефективності селекторів, генерованих різними методами:

- Точність – визначає, яку частину всіх знайдених селекторів вибрано правильно, тобто вони справді відповідають правильним елементам на сторінці після змін. У контексті GNN точність була значно вищою порівняно із ручним та напівавтоматизованим підходами, оскільки нейронні мережі здатні аналізувати складніші взаємозв'язки між елементами DOM, враховуючи їх контекст і зв'язки:

$$Precision(s) = \frac{TP}{TP + FP},$$

де TP – селектори, які правильно ідентифікували потрібні елементи, FP – селектори, які вказали на неправильні елементи.

- Повнота – визначає здатність знаходити всі необхідні селектори на сторінці, зокрема рідко використовувані чи складноструктуровані елементи:

$$Recall(s) = \frac{TP}{TP + FN},$$

де TP – селектори, які правильно ідентифікували потрібні елементи; FN – селектори, яких не було згенеровано, але які мали бути.

- F1-score є комбінованим показником точності та повноти, що забезпечує збалансованішу оцінку ефективності, особливо коли важливе не тільки досягнення високої точності, але й здатність знаходити всі необхідні селектори:

$$F1(s) = 2 \cdot \frac{Precision(s) \cdot Recall(s)}{Precision(s) + Recall(s)}.$$

- Час генерації – вимірюється час, необхідний для побудови селектора. Хоча GNN може бути дещо повільнішою за напівавтоматизовані методи, вона демонструє високу точність і здатність адаптуватися до змін у DOM.
- Помилки після змін DOM – кількість випадків, коли після оновлення вебсторінки селектор не спрацював.

Після отримання результатів обчислено усереднені значення кожної метрики за виразом:

$$X_{\text{сеп}} = \frac{1}{N} \sum_{i=1}^N X_i,$$

де $X_{\text{сеп}}$ – середнє значення метрики для усіх вебсайтів; N – кількість вебсайтів; X_i – значення метрики для конкретного сайту. Результати експерименту демонструють, що методи на основі GNN значно перевершують традиційні підходи:

Табл. 1. Порівняння точності, повноти, F1-score, часу генерації та кількості помилок у трьох методах формування селекторів / Comparison of accuracy, completeness, F1-score, generation time, and number of errors in three methods of selector formation

Метод	Точність, %	Повнота, %	F1-score, %	Час генерації, мс	Помилки після змін, шт.
Ручний	76,5	56,9	65	1200	5
Напівавтоматизований	78,5	63	69,8	45	22
GNN	88	84,7	86,3	60	5

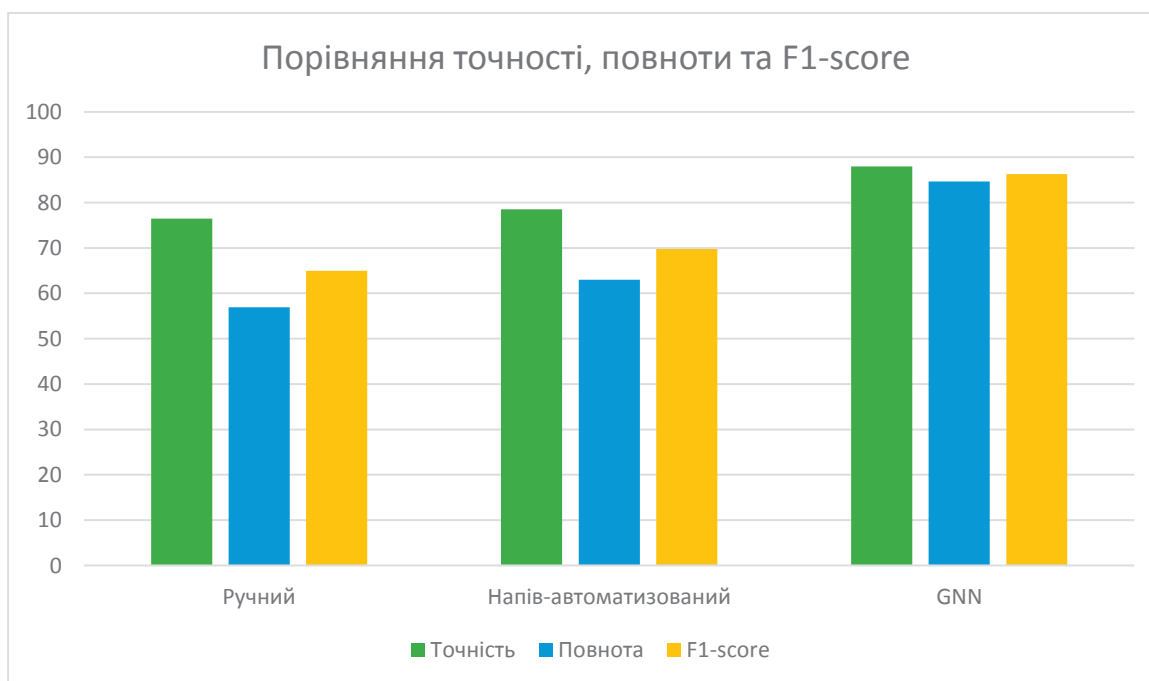


Рис. 1. Графічне порівняння показників точності (Precision), повноти (Recall) та F1-score для трьох підходів генерації селекторів / Graphical comparison of precision, recall, and F1-score metrics for three approaches to selector generation

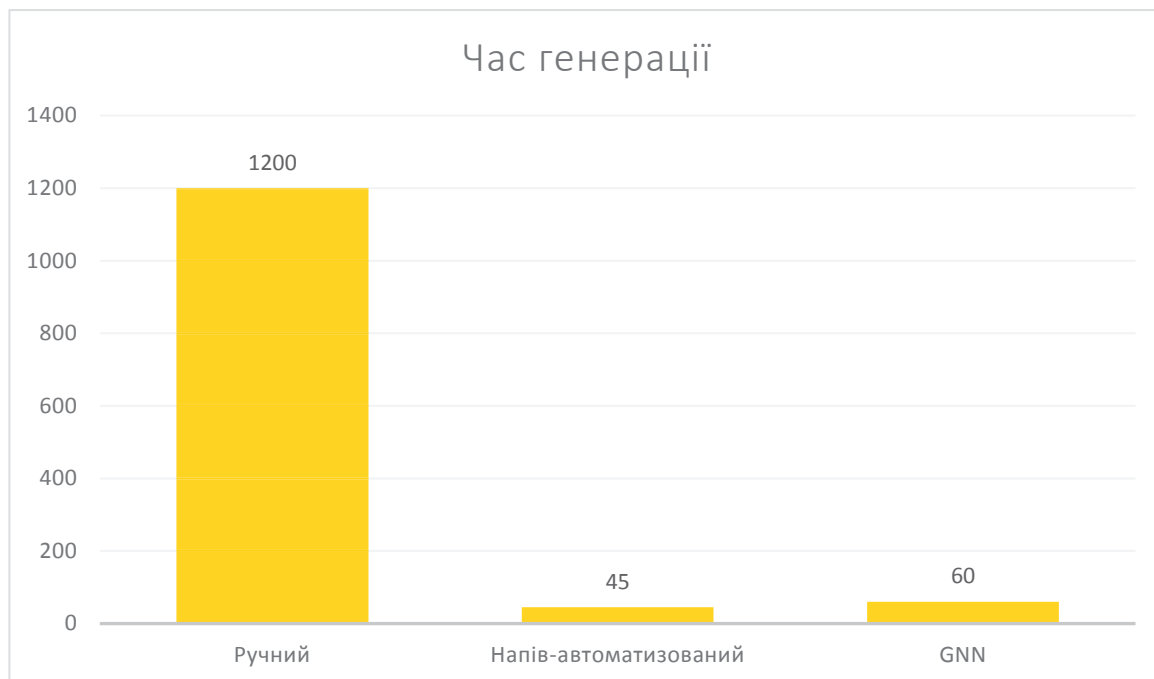


Рис. 2. Порівняння часу генерації тестових селекторів / Comparison of test selector generation times

Процес автоматизованої генерації тестових селекторів за допомогою графових нейронних мереж передбачає кілька основних етапів, кожен із яких відіграє важливу роль у досягненні стабільності й точності вибору елементів у DOM-структурі.

1. Збирання та попереднє оброблення даних. На першому етапі з HTML-коду вебсторінки формується графова структура, де вузлами є окремі DOM-елементи, а ребра відображають їхні структурні або логічні зв'язки. Враховується як вкладеність елементів, так і атрибутивна інформація (класи, ID, data-testid, теги), що дає змогу моделі ефективніше розпізнавати контекст елементів.

2. Навчання GNN-моделі. Після побудови графової структури модель графової нейронної мережі навчається на основі маркованих даних, де кожен вузол має відповідну мітку, що визначає його як цільовий елемент чи ні. Навчання може передбачати як традиційне надzorване навчання, так і методи з підкріпленням для підвищення адаптивності до змін у структурі DOM.

3. Генерація селекторів. На основі навченої моделі генеруються селектори у вигляді CSS або XPath, які повинні максимально точно ідентифікувати цільові елементи. Завдання полягає у мінімізації ризику хибних спрацювань – як позитивних, так і негативних, що досягається урахуванням структурного контексту DOM [9].

4. Адаптація до змін. Однією із головних переваг GNN є здатність моделі адаптуватися до змін у DOM без повного перенавчання. Цього досягають за допомогою оновлення ваг у графі, що враховують нові зв'язки між елементами вебсторінки. Отже, модель зберігає актуальність навіть після оновлення контенту на вебсайті.

Ці етапи створюють задачу оптимізації, в якій шукають такий набір селекторів S , що максимізує

стабільність на тестових даних. Формалізовано це можна подати як:

$$\text{Optimize}\{S\} = \arg \max \text{Stability}(S),$$

де $\text{Stability}(S)$ розраховують як функцію від метрик Precision та Recall з урахуванням змін у DOM після оновлень інтерфейсу; S – набір тестових селекторів, а стабільність визначається через точність і повноту тестів.

Це дає змогу досягти високої стабільності тестових селекторів у реальних умовах веброзроблення, де зміни в DOM є неминучими. Технологія GNN відкриває нові можливості для автоматизованого тестування вебдодатків, забезпечуючи надійніші та точніші результати тестування.

Для оцінювання ефективності GNN у реальних умовах було вибрано 25 вебсайтів з різними структурами, як із динамічними, так і зі статичними елементами. Основні труднощі під час тестування полягали в частих оновленнях інтерфейсу, змінах ієрархії елементів та появи нових блоків без зміни основного дизайну.

Під час експериментального порівняння методів зафіксовано значне зниження середнього часу, витраченого на оновлення тестових сценаріїв без втрати якості селекторів порівняно з напівавтоматичним режимом, який, хоч на 25 % швидший, але повністю не підходить для задач з динамічними елементами. Застосування GNN дало змогу залишити кількість помилок такою самою, як у разі ручного тестування, але істотно зменшило загальні витрати часу. Це свідчить про високу адаптивність і практичну ефективність графових моделей у контексті сучасного вебтестування.

Перспективи подальших досліджень можна виокремити на таких аспектах:

- Оптимізація архітектури GNN – використання Graph Attention Networks (GAT) для підвищення здатності моделі розрізняти важливі елементи DOM [3].
- Інтеграція підсилювачів навчання та мовних моделей (наприклад, Bard) для пошуку селекторів за текстом.
- Розширення тестової вибірки – введення більшої кількості вебдодатків, зокрема з мобільних платформ, щоб оцінити ефективність GNN у мобільному тестуванні.
- Оптимізація продуктивності – дослідження можливості зменшити тривалість генерації селекторів без втрати точності.

Упровадження цих покращень дасть змогу зробити автоматизоване тестування ще ефективнішим і стабільнішим, знижуючи витрати на підтримання тестів у великих вебпроектах.

Обговорення результатів дослідження. Результати експериментальних досліджень показали, що різні підходи до генерації тестових селекторів істотно відрізняються за точністю, повнотою та стабільністю після змін DOM-структури. Ручний метод забезпечує достатньо високу точність, проте потребує значних часових витрат та характеризується низькою повнотою, що ускладнює його застосування у масштабних проєктах. Напівавтоматизований підхід продемонстрував найвищу швидкодію, але виявився найнестабільнішим у динамічних середовищах, що призводить до великої кількості помилок після оновлення інтерфейсу.

Метод на основі GNN продемонстрував найкращий баланс між усіма ключовими метриками. Отримані значення точності, повноти та F1-score свідчать про здатність графових моделей ефективно враховувати як локальні атрибути елементів, так і їх структурні зв'язки, що забезпечує високу адаптивність до змін DOM. Час генерації селектора залишається прийнятним, а кількість помилок після змін інтерфейсу істотно зменшується.

Водночас експеримент має певні обмеження, зокрема обмежений обсяг вибірки (25 вебдодатків) та використання лише однієї архітектури графової моделі. Це відкриває простір для подальших досліджень, пов'язаних з тестуванням інших архітектур, таких як Graph Attention Networks, а також розширенням експериментальної бази за рахунок мобільних платформ та бізнес-критичних систем.

Отже, за результатами дослідження можна визначити наукову новизну та практичну цінність роботи.

Наукова новизна отриманих результатів дослідження – запропоновано підхід на основі використання графових нейронних мереж (GNN) для генерації тестових селекторів у контексті сучасного динамічного вебсередовища, що, на відміну від відомих підходів, враховує зміни DOM-структури разом із контекстними атрибутами та структурними зв'язками, що дало змогу істотно підвищити точність і повноту автоматичної генерації селекторів.

Практична значущість результатів дослідження – результати експериментальних досліджень підтверджують ефективність впровадження GNN у задачах автоматизованого тестування вебдодатків, особливо в умовах динамічного інтерфейсу. Запропонований підхід дає змогу:

- зменшити тривалість оновлення тестів після змін у DOM-структурі;
- зменшити кількість помилок, пов'язаних з нестабільністю селекторів;
- знизити загальні витрати на супровід тестових сценаріїв;
- підвищити надійність тестів без необхідності глибокого ручного втручання.

Отже, отримані результати можна безпосередньо застосовувати у практиці розроблення тестів у комерційних проєктах, а також покласти в основу розроблення нових інструментів і фреймворків для автоматизованого тестування.

Висновок / Conclusions

Дослідження показало, що застосування графових нейронних мереж (GNN) під час автоматизованого тестування вебдодатків істотно підвищує стабільність тестових селекторів. Результати експериментів свідчать про значне зниження кількості хибнопозитивних та хибнонегативних спрацювань порівняно з традиційними підходами, такими як ручне та навіавтоматизоване тестування.

У ході досліджень відзначено:

1. Підвищену точність та стійкість тестових селекторів. GNN дає змогу визначати надійніші селектори, на які менше впливають зміни у DOM-структурі.
2. Автоматичну адаптацію до змін. Використання графових моделей дає змогу нейромережі враховувати взаємозв'язки між елементами вебсторінки та адаптувати селектори відповідно до змін інтерфейсу.
3. Зменшення потреби у ручному налаштуванні тестів. Автоматична генерація стабільних селекторів істотно зменшує витрати на підтримку тестових сценаріїв.
4. Підвищення ефективності тестування. Дослідження реальних проєктів показало, що впровадження GNN дає змогу знизити кількість помилкових спрацювань приблизно на 27 % та покращити пропорцію хибних помилок після змін DOM-моделі до часу.

Як перспективи подальшого розвитку досліджень запропоновано інтеграцію Graph Attention Networks (GAT) для підвищення точності моделі, додаткового використання підсилювача та мовних моделей для збільшення адаптивності, а також дослідження питання оптимізації швидкодії роботи алгоритму.

Отже, впровадження GNN у процес автоматизації тестування є перспективним напрямом, що дає змогу істотно підвищити ефективність тестування вебдодатків, особливо у випадках динамічних інтерфейсів.

Подальші дослідження можуть бути спрямовані на оптимізацію архітектури моделі та розширення тестової вибірки для оцінювання ефективності GNN у різних середовищах веброзроблення.

References

1. Hotti, A., Risuleo, R. S., Magureanu, S., Moradi, A., & Lagergren, J. (2024). The Klarna Product Page Dataset: Web Element Nomination with Graph Neural Networks and Large Language Models. *arXiv*: 2111.02168 [cs.LG]. <https://doi.org/10.48550/arXiv.2111.02168>
2. Ergashev, U., Dragut, E., & Meng, W. (2023). Learning to rank resources with GNN. *Proceedings of the ACM Web Conference 2023*, 3247-3256. <https://doi.org/10.1145/3543507.3583360>
3. Acharya, D. B., & Zhang, H. (2020). Feature selection and extraction for graph neural networks. *ACM Southeast Conference*. <https://doi.org/10.48550/arXiv.1910.10682>
4. García, B., Gallego, M., Gortázar, F., & Munoz-Organero, M. (2020). A survey of the selenium ecosystem. *Electronics*, 9(7), 1067. <https://doi.org/10.3390/electronics9071067>
5. Uppal, N., & Chopra, V. (2012). Design and implementation in selenium IDE with web driver. *International Journal of Computer Application*, 46(12), 8-11.
6. Ui.Vision. (2023). Selenium IDE: Everything you need to know. Retrieved from <https://ui.vision/rpa/docs/selenium-ide/locators>
7. Yau, W. (2022). Test automation refactoring is a mess? Try CSS selector. *Medium*. Retrieved from <https://medium.com/ichef/test-automation-refactoring-is-a-mess-try-css-selector-c0084e4cd94>
8. Exadel. (2022). How to choose selectors for automation to make your life a whole lot easier. Retrieved from <https://exadel.com/news/how-to-choose-selectors-for-automation-to-make-your-life-a-whole-lot-easier/>
9. Morozov O. S., Yarovy A. A., Petryshyn S. I. (2025). Hybrid methods for testing selectors in dynamic web applications. *Youth in Science: Research, Problems, Prospects 2025*.

O. S. Morozov, A. A. Yarovy

Vinnitsia National Technical University, Vinnitsia, Ukraine

EXPERIMENTAL RESEARCH ON APPROACHES TO GENERATING TEST SELECTORS USING GNN IN THE PROCESS OF AUTOMATED TESTING OF WEB APPLICATIONS

The article discusses the problem of instability of test selectors in the process of automated testing of web applications. It raises the issue of selectors' adaptability to changes in the DOM structure, which is critically important in the development of modern dynamic web interfaces. A comparative analysis of three approaches to selector generation is conducted: manual (via Chrome DevTools), semi-automated (using DevTools), and automated using graph neural networks (GNN).

The aim of the study was to determine which approach provides the best balance between accuracy, completeness, stability, and time spent on selector formation in real testing conditions. The work used an experimental approach that included testing on 25 web applications with different types of DOM structures: static, dynamic, and SPA. Standard quality metrics were used: accuracy, completeness, F1-score, and the number of errors after changes in the DOM and the average selector generation time were also evaluated.

The results show the advantage of the GNN-based method, particularly in the stability of selectors after interface updates. Although the manual approach demonstrates high accuracy, it is significantly inferior in terms of speed and completeness, and Chrome DevTools, although fast, is the least reliable in dynamic environments. Thus, the use of graph neural networks makes it possible to create more adaptive and reliable solutions for automated testing of web applications.

In addition, the study demonstrates the feasibility of further developing GNN-based approaches through the integration of self-supervised learning and Graph Attention Networks (GAT) mechanisms, which will enable even greater accuracy, scalability, and performance of automated tests in complex software systems. The proposed approach has the potential to be implemented in next-generation frameworks for UI testing, which will contribute to improving the quality of digital products on the market.

Also, the feasibility of using GNN for automated generation of selectors adapted to changes in the DOM has been experimentally proven, and it has been found that this approach reduces the number of false positives without compromising performance. The results obtained are of practical value for improving the reliability and efficiency of automated testing systems in rapidly changing interface environments.

Keywords: software, information technology, intelligent systems, automated testing, optimisation, graph neural networks, DOM structure.

Інформація про авторів:

Морозов Олександр Сергійович, аспірант, кафедра комп'ютерних наук.

Email: Morozovsasha97@gmail.com; <https://orcid.org/0009-0006-3083-5449>

Яровий Андрій Анатолійович, д-р техн. наук, професор, завідувач кафедри комп'ютерних наук.

Email: a.yarovyy@vntu.edu.ua; <https://orcid.org/0000-0002-6668-2425>

Цитування за ДСТУ: Морозов О. С., Яровий А. А. Експериментальні дослідження підходів до генерації тестових селекторів із застосуванням GNN у процесі автоматизованого тестування вебдодатків. *Український журнал інформаційних технологій*. 2025, т. 7, № 2. С. 44-49.

Citation APA: Morozov, O. S., & Yarovy, A. A. (2025). Experimental research on approaches to generating test selectors using GNN in the process of automated testing of web applications. *Ukrainian Journal of Information Technology*, 7(2), 44-49. <https://doi.org/10.23939/ujit2025.02.44>