
СИСТЕМИ ТЕХНІЧНОГО ЗОРУ І ШТУЧНОГО ІНТЕЛЕКТУ З ОБРОБКОЮ ТА РОЗПІЗНАВАННЯМ ЗОБРАЖЕНЬ

УДК 004.415.53 : 004.8

О.С. МОРОЗОВ, А.А. ЯРОВИЙ

ІНТЕЛЕКТУАЛЬНА МОДЕЛЬ ГЕНЕРАЦІЇ АДАПТИВНИХ ВЕБ-СЕЛЕКТОРІВ НА ОСНОВІ GNN

Вінницький національний технічний університет,

вул. Хмельницьке шосе, 95, м. Вінниця, Україна, 21021, e-mail: morozovsasha97@gmail.com

Анотація. У статті розглянуто проблему нестабільності DOM-селекторів, яка є однією з ключових причин появи флаку-тестів у сучасних системах автоматизованого тестування веб-ресурсів. Показано, що традиційні підходи до формування селекторів – як ручні, так і автоматизовані – не враховують глобальний контекст DOM-структури та є малоефективними у випадку динамічних змін інтерфейсу. Обґрунтовано доцільність використання графових нейронних мереж (GNN) як інструменту моделювання DOM-дерева у вигляді орієнтованого графа з урахуванням семантичних, атрибутивних та структурних ознак вузлів. Запропоновано теоретичну модель побудови адаптивних DOM-селекторів, що поєднує GNN з агентом прийняття рішень. Розроблено математичне подання DOM-графа, систему ознак вузлів (one-hot кодування тегів, бінарні атрибути id/class/data-testid, глибина, кількість нащадків, позиція серед сусідів), а також формалізовано метрики оцінювання селекторів: унікальність, точність, повнота, F1-міра та довжина. Сформовано функцію винагороди агента, яка оптимізує баланс між точністю, стабільністю та компактністю селектора. Показано, що поєднання графових репрезентацій DOM (Document object model) та механізму оптимізації рішень дає змогу створювати селектори, стійкі до структурних змін веб-сторінок, а також скорочує потребу у ручній підтримці тестів. Запропонована модель формує теоретичне підґрунтя для створення інтелектуальних систем автоматизованого тестування, здатних адаптувати локатори під час зміни DOM без необхідності повного аналізу інтерфейсу тестувальником.

Ключові слова. програмне забезпечення, інформаційні технології, інтелектуальні системи, автоматизоване тестування, графові нейронні мережі (GNN).

Abstract. The article discusses the problem of DOM selector instability, which is one of the key reasons for flaky tests in modern automated web resource testing systems. It is shown that traditional approaches to selector formation, both manual and automated, do not take into account the global context of the DOM structure and are ineffective in the case of dynamic interface changes. The feasibility of using graph neural networks (GNN) as a tool for modelling the DOM tree in the form of a directed graph, taking into account the semantic, attributive, and structural features of nodes, is justified.

A theoretical model for building adaptive DOM selectors, combining GNN with a decision-making agent, is proposed. A mathematical representation of the DOM graph, a system of node features (one-hot tag encoding, binary attributes id/class/data-testid, depth, number of descendants, position among neighbours) has been developed, and selector evaluation metrics have been formalised: uniqueness, accuracy, completeness, F1-measure and length. An agent reward function has been formed that optimises the balance between accuracy, stability, and compactness of the selector.

It has been shown that the combination of DOM (Document object model) graph representations and a decision optimisation mechanism allows the creation of selectors that are resistant to structural changes in web pages and reduces the need for manual test maintenance. The proposed model forms the theoretical basis for the creation of intelligent automated testing systems capable of adapting locators during DOM changes without the need for a complete analysis of the interface by the tester.

Keywords. software, information technology, intelligent systems, automated testing, graph neural networks (GNN).

DOI: 10.31649/1681-7893-2025-50-2-142-149

ВСТУП

У нинішніх умовах стрімкого розвитку веб-простору методи автоматичного вилучення та аналізу даних із веб-сторінок дедалі частіше стикаються зі складною, змінною та багато в чому непередбачуваною структурою розмітки.

СИСТЕМИ ТЕХНІЧНОГО ЗОРУ І ШТУЧНОГО ІНТЕЛЕКТУ З ОБРОБКОЮ ТА РОЗПІЗНАВАННЯМ ЗОБРАЖЕНЬ

Більшість традиційних інструментів, орієнтованих на використання статичних локаторів (наприклад, фіксованих CSS/XPath-селекторів), виявляються малоефективними у випадку навіть незначних змін у DOM-дереві, що є типовою ситуацією на сучасних сайтах. У результаті це призводить до значних витрат часу й ресурсів на ручне коригування скриптів.

Останні роки відзначається зростаючий інтерес до застосування машинного навчання, зокрема графових нейронних мереж, для обробки веб-сторінок. Ідея полягає в тому, що DOM-структура цілком природно може бути описана як дерево чи загальніше - як граф, де кожен елемент (вузол) пов'язаний із сусідами відношеннями «parent-child» або «sibling».

Мета роботи – розроблення теоретичної моделі побудови адаптивних DOM-селекторів із використанням графових нейронних мереж та підсилювального навчання для підвищення стабільності й ефективності автоматизованого тестування веб-ресурсів.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Значний прогрес досягнуто у методах пошуку веб-елементів за останнє десятиріччя, що пов'язано з активним впровадженням глибоких нейронних мереж і появою спеціалізованого інструментарію для роботи зі складними структурами. Водночас сучасні сайти постійно змінюють DOM, що ускладнює завдання підтримки фіксованих селекторів і потребує більш гнучких підходів.

На початкових етапах широко використовувались статичні алгоритми та евристики, засновані на регулярних виразах або жорстко зафіксованих CSS/XPath-селекторах [1]. Хоча вони можуть давати прийнятні результати в сталих умовах, будь-яка зміна назви атрибута чи структури HTML зазвичай ламає такі скрипти, що потребує ручного втручання. Подальша поява фреймворків на кшталт Selenium IDE або Puppeteer частково спростила завдання формування локаторів, однак не розв'язала проблему їх швидкого знецінення під час змін у розмітці. Для підвищення надійності виявлення цільових елементів було запропоновано моделі, що використовували машинне навчання (ML) як альтернативу статичним правилам [2]. Зокрема, SVM чи випадкові ліси (Random Forest) дозволяли аналізувати ознаки елементів DOM (наприклад, клас, текст, атрибути) та ефективніше знаходити потрібні вузли. Втім, тоді ще не було повноцінного врахування ієрархічних зв'язків, що є важливим у складних веб-інтерфейсах.

Впровадження графових нейронних мереж (Graph Neural Networks) стало ключовим кроком вперед, адже домоглося врахування структури DOM як графа. У роботах [3-4] продемонстровано, що GCN (Graph Convolutional Networks) і GAT (Graph Attention Networks) значно збільшують точність класифікації веб-елементів завдяки «придільненню уваги» різним типам відношень.

У подальших дослідженнях [4-5] почали застосовувати гетерогенні графові нейронні мережі (Heterogeneous GNN), що дають змогу розглядати DOM як різномірний граф із різними типами вузлів та ребер. Такий підхід виявляється особливо корисним у ситуаціях, де мають значення семантичні властивості тексту або позиційна інформація (розташування елемента в шаблоні верстки).

Окремий напрямок робіт, присвячений автоматизації селекторів, часто зосереджується саме на формуванні локатора, який не втрачати коректності при зміні структури HTML. У роботі [6] було запропоновано підхід на основі Reinforcement Learning (RL), де агент поетапно додає атрибути чи змінює шлях у DOM, прагнучи винагороди за унікальність і точність селектора на тестових сторінках. Проте в більшості таких напрацювань [6-7] бракує попереднього глибокого аналізу сторінки за допомогою GNN, що могло б прискорити навчання агента та зменшити кількість помилок.

Інший спосіб полягає в комбінаторній оптимізації (наприклад, Integer Linear Programming, ILP), де завдання полягає в мінімізації довжини локатора або, залежно від формулювання, у максимізації його стійкості [8]. Ці методи забезпечують формальні гарантії пошуку оптимального розв'язку, однак, на практиці можуть виявитися обчислювально складними, коли DOM має дуже велику кількість елементів.

На підставі огляду робіт [1-8] можна виокремити дві головні проблеми. По-перше, надійні класифікатори для веб-елементів (GNN та інші глибинні моделі) зазвичай не пропонують формальних методів побудови селекторів. По-друге, методи оптимізації локаторів (RL, ILP) часто ігнорують контекст DOM, який можна було б врахувати із застосуванням GNN. В результаті відсутні комплексні системи, що забезпечували б і точність розпізнавання, і гнучкість у формуванні стійкого локатора.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

DOM – це інтерфейс прикладного програмування (API) для валідних HTML-документів та правильно сформованих XML-документів. Вона визначає логічну структуру документів та спосіб доступу до них і роботи з ними [9]. HTML-документ доцільно розглядати як граф, де кожен тег відповідає окремому вузлу. Зв'язки між вузлами формуються на підставі відношень батько-нащадок (parent-child), а також, за потреби, сусідніх елементів (sibling). Математична модель DOM-графа має вигляд:

$$G = (V, E) \quad (1)$$

де V - множина вузлів(елементів), E - множина ребер (зв'язки між елементами). Розглянемо властивості графів у контексті оптимізації пошуку оптимального селектору: глибину графа, ступінь вузла, зв'язність. Глибина вузла в графі визначається як довжина найкоротшого шляху від кореневого вузла (зазвичай це або `html`, або `body`) для поточного вузла. Максимальна глибина – найбільша з глибин усіх вузлів.

$$depth(v_i) = d(v_{root}, v_i)$$

де v_i – довільний вузол у DOM-графі, v_{root} – кореневий вузол (`html` або `body`), $d(v_{root}, v_i)$ – кількість ребер, що з'єднують корінь вузлом v_i по найкоротшому шляху.

Ступінь вузла - це кількість його суміжних вузлів. У напрямленому графі, як у DOM-структурі розрізняються: вхідний ступінь - кількість батьківських елементів, вихідний ступінь – кількість дочірніх елементів.

$$\begin{aligned} deg^-(v_i) &= |\{v_j \in V \mid (v_j, v_i) \in E\}| \\ deg^+(v_i) &= |\{v_k \in V \mid (v_i, v_k) \in E\}| \\ deg(v_i) &= deg^-(v_i) + deg^+(v_i) \end{aligned} \quad (2)$$

де $deg^-(v_i)$ – кількість батьківських елементів вузла v_i , $deg^+(v_i)$ – кількість дочірніх вузлів, $deg(v_i)$ – загальна кількість суміжних елементів. Ступінь є важливим для аналізу структурної ролі вузла DOM, вузли з високим ступенем можуть бути ключовими при генерації селектора.

Зв'язність графа характеризує наявність шляхів між вузлами. Для орієнтовних графів: сильна зв'язність – між будь-якими двома вузлами існують шляхи в обох напрямках, слабка зв'язність – існує шлях в прямому або зворотному напрямку. Для структури DOM як дерева є притаманна слабка зв'язність, що допомагає побудувати єдиний шлях від кореня до кожного елемента. Розглянемо частину структури DOM та візуалізуємо його як граф:

```
<div id='root'>
  <h1 class='title'></h1>
  <div class='content'>
    <p></p>
  </div>
</div>
```

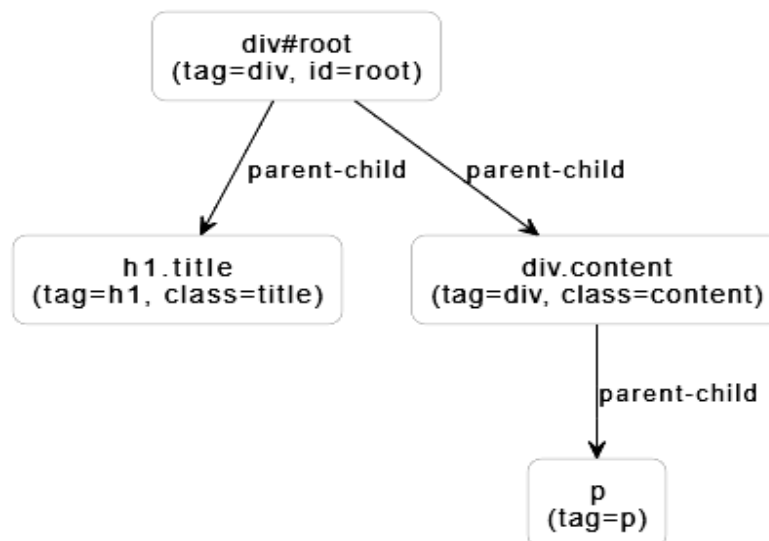


Рисунок 1 – Візуалізація DOM як граф

СИСТЕМИ ТЕХНІЧНОГО ЗОРУ І ШТУЧНОГО ІНТЕЛЕКТУ З ОБРОБКОЮ ТА РОЗПІЗНАВАННЯМ ЗОБРАЖЕНЬ

Опишемо, які ознаки обираються для вузлів DOM-графа: one-hot кодування, бінарні ознаки атрибутів, структурні ознаки. One-hot кодування – це метод перетворення категоріальних даних у бінарне векторне представлення. Кожна категорія представлена вектором, довжина якого дорівнює загальній кількості унікальних категорій [10]. В задачі побудови селектору за допомогою GNN метод використовується для кодування тегів. Це дозволяє розрізняти типи елементів, наприклад <div>, , <a> та інші. Розглянемо приклад:

Для кожного вузла DOM-графа визначається його тег (наприклад: <div>, <a>, , тощо). Нехай існує множина можливих тегів:

$$\mathcal{T} = \{t_1, t_2, t_3, \dots, t_T\} \quad (3)$$

де T – кількість різних типів тегів, що зустрічаються в корпусі документів.

Тоді кожен v_i , який відповідає тегу t_k , кодується у вигляді one-hot вектора довжини T :

$$onehot(v_i) = [0, 0, \dots, 1, \dots, 0] \in \mathbb{R}^T \quad (4)$$

де одиниця стоїть на k -ій позиції, яка відповідає тегу t_k .

Нехай маємо множину тегів:

$$\mathcal{T} = \{div, span, a, ul\}, T=4$$

Для кожного кожного вузла маємо кодування:

$$\begin{aligned} \langle div \rangle - x &= [1, 0, 0, 0] \\ \langle span \rangle - x &= [0, 1, 0, 0] \\ \langle a \rangle - x &= [0, 0, 1, 0] \\ \langle ul \rangle - x &= [0, 0, 0, 1] \end{aligned}$$

Таким чином, one-hot кодування забезпечує уніфіковане представлення типу тегу, яке може бути інтегроване у матрицю ознак $X \in \mathbb{R}^{N \times d}$ для подальшої обробки GNN. При цьому ознака тегу є категоріальною, і one-hot кодування дозволяє уникнути накладання «штучних» відстаней між різними тегами (на відміну від простого числового кодування).

Для атрибутів використовуються бінарні ознаки. Атрибутами виступають – id, class, data-testid та інші. Для кожного атрибуту додається ознака 1, якщо він присутній і 0 – якщо відсутній. Це критично важливо, оскільки в автоматизованих тестах найчастіше використовується пошук саме за допомогою атрибутів.

Також використовуються структурні ознаки, які дозволяють GNN краще враховувати структуру DOM-дерева. Серед них глибина вузла, кількість дочірніх вузлів (ступінь вузла), позиція серед усіх сусідів. Такі структурні глобальні та локальні ознаки допомагають моделі диференціювати вузли навіть за ідентичними тегами, що важливо для контекстуального розуміння структури [10, 11]. Всі ознаки вузлів DOM описані в таблиці 1.

Таблиця 1 – Ознаки вузлів DOM

Категорія	Ознака	Тип/Значення	Пояснення
Семантичні	One-hot тегу	Вектор довжини T	Кодування типу елемента: <div>, , <a> тощо.
Атрибути	Наявність id	0/1	1 - Елемент має id, 0 - відсутній
	Наявність class	0/1	1 - Елемент має class, 0 - відсутній
	Наявність data-testid	0/1	1 - Елемент має data-testid, 0 - відсутній
Структурні	Глибина вузла	Ціле число	Відстань від кореня DOM-дерева
	Кількість дочірніх елементів	Ціле число	Скільки дочірніх елементів має вузол
	Позиція серед дочірніх елементів	Ціле число	Порядковий номер вузла у списку сусідніх елементів

СИСТЕМИ ТЕХНІЧНОГО ЗОРУ І ШТУЧНОГО ІНТЕЛЕКТУ З ОБРОБКОЮ ТА РОЗПІЗНАВАННЯМ ЗОБРАЖЕНЬ

Таким чином, вектори ознак усіх вузлів об'єднуються у матрицю ознак:

$$X \in \mathbb{R}^{N \times d} \quad (5)$$

де N – кількість вузлів у DOM-графі, d – кількість ознак.

Ключовим елементом графового подання DOM-структури є визначення зв'язків між вузлами, які формалізуються у вигляді матриці суміжності або списку ребер. Нехай задано граф $G = (V, E)$, де $V = \{v_1, v_2, \dots, v_N\}$ – множина вузлів, а $E \subset V \times V$ – множина ребер, що відображають структурні відношення у DOM. Тоді матриця суміжності має вигляд:

$$A \in \{0, 1\}^{N \times N}, A_{ij} = \begin{cases} 1, & \text{якщо } (v_i, v_j) \in E \\ 0, & \text{інакше} \end{cases} \quad (6)$$

де ребра визначаються зв'язками parent-child, коли потрібно sibling.

Існують такі типи ребер у DOM-графі:

- Parent-child - основні зв'язки, що відображають ієрархію DOM-дерева (наприклад $\langle ul \rangle \rightarrow \langle li \rangle$)
- Sibling - додаткові зв'язки між сусідніми елементами, які знаходяться на одному рівні (наприклад $\langle li \rangle \rightarrow \langle li \rangle$)
- Self-loop - петля вузла на самого себе, яка дозволяє зберегти його власні ознаки у процесі агрегації.

Одним з ключових аспектів оптимізації процесу автоматизованого тестування є побудова селекторів, які одночасно є стабільними, унікальними та компактними. Для цього необхідна формалізація критеріїв якості селектора у вигляді математичних метрик, які дозволяють здійснювати кількісне порівняння різних підходів до їх генерації. Такий підхід забезпечує можливість об'єктивної оцінки та використання алгоритмів машинного навчання, зокрема графових нейронних мереж, для їх автоматичної побудови.

Унікальність є базовою вимогою до селектора, оскільки лише у випадку однозначної ідентифікації DOM-елемента можна гарантувати коректність тесту. Введемо формальне визначення:

$$u(s) = \begin{cases} 1, & \text{якщо } |Match(s)| = 1, \\ 0, & \text{інакше} \end{cases} \quad (7)$$

де $Match(s)$ – множина елементів, які знаходить селектор.

Унікальність особливо важлива у випадках використання селекторів у масштабних регресійних тестах, де відсутність однозначності призводить до флак-тестів.

При побудові складних селекторів, особливо коли вони базуються на класах чи вкладених елементах, часто виникають ситуації, коли:

- селектор повертає зайві елементи (FP, False Positive),
- не охоплює всі релевантні елементи (FN, False Negative).

Для оцінки таких випадків використовуються метрики інформаційного пошуку Precision (точність), Recall (повнота) [11]:

$$Precision(s) = \frac{TP}{TP+FP} \quad (8)$$

$$Recall(s) = \frac{TP}{TP+FN} \quad (9)$$

Для збалансованої оцінки застосовується гармонічне середнє:

$$F1(s) = 2 \cdot \frac{Precision(s) \cdot Recall(s)}{Precision(s) + Recall(s)} \quad (10)$$

Крім точності та унікальності, важливо враховувати довжину селектора, яка впливає на його зручність використання, стабільність та швидкість пошуку. Довжина селектора є важливим критерієм під час написання автоматизованих тестів, так як він виконує декілька функцій:

- пришвидшує роботу тестів;
- покращує стабільність тестів;
- сприяє покращенню читабельності тестового коду та полегшує його супровід.

Для визначення довжини селектору запропоновано функцію з використанням вагових коефіцієнтів, що покращує визначення довжини селектора:

СИСТЕМИ ТЕХНІЧНОГО ЗОРУ І ШТУЧНОГО ІНТЕЛЕКТУ З ОБРОБКОЮ ТА РОЗПІЗНАВАННЯМ ЗОБРАЖЕНЬ

$$L(S) = \sum_{i=1}^k w(c_i) \quad (11)$$

де k – кількість компонентів, c_i – компонента селектора (теги, класи, id, оператори вкладеності тощо), $w \in (0; 1]$ – ваговий коефіцієнт. Розподілення вагових коефіцієнтів можна реалізувати наступним чином:

Таблиця 2 – Вагові коефіцієнти для компонентів селектору

Тип елементу	Ваговий коефіцієнт
Ter	0.5
Id	0.2
Class	0.5
Data-testid	0.4
nth-child	0.8
Вкладеність (>)	0.5

Враховуючи зазначені обмеження пропонується використання графових нейронних мереж у поєднанні з агентом прийняття рішень. Такий підхід дозволяє врахувати як локальні ознаки вузлів (тег, атрибути, позицію серед сусідів), так і їхній контекст у DOM-дереві. На відміну від простих евристичних або текстових методів, GNN формує векторні репрезентації елементів, що містять інформацію не лише про сам вузол, а й про його зв'язки з батьківськими та дочірніми елементами. Це забезпечує можливість робити висновки про стабільність потенційного селектора навіть у випадку, коли атрибут, наприклад клас, виявляється повторюваним у багатьох елементах [12].

Агент, взаємодіючи з репрезентаціями, що генерує GNN, здійснює вибір дій, спрямованих на побудову оптимального селектора. Ці дії можуть включати використання або ігнорування певних атрибутів, визначення глибини вкладеності, а також вибір стратегії формування шляху у вигляді CSS чи XPath. Метою агента є максимізація функції винагороди, яка базується на метриках якості селектора, а саме: унікальності, точності, повноті, F1-score та довжині. Таким чином, агент навчається віддавати перевагу тим ознакам і стратегіям побудови, які забезпечують високу стабільність та компактність селекторів.

Формальна модель побудови адаптивних селекторів передбачає навчання агента таким чином, щоб він максимізував якість побудованих локаторів. Як уже встановлено, ключові метрики якості: унікальність $u(s)$, точність $Precision(s)$, повнота $Recall(s)$, F1-міра $F1(s)$ та довжина селектора $L(s)$. Ці показники формують основу для функції винагороди [13-14].

Математично модель винагороди може бути виражена як:

$$R(s) = \alpha \cdot u(s) + \beta \cdot F1(s) - \gamma \cdot L(s) \quad (12)$$

де $u(s)$ – бінарний показник унікальності, $F1(s)$ – збалансована міра точності і повноти, $L(s)$ – нормалізована довжина селектору, α, β, γ – вагові коефіцієнти, які коригують вплив кожного компонента. Це дозволяє агенту оптимізувати баланс між стабільністю, точністю та компактністю.

Функція винагороди $R(s)$ дозволяє навчити агента надавати перевагу селекторам, які є не лише коректними, тобто унікальними і точними, але й компактними й стабільними.

Особливо важливим є аспект адаптації. У разі зміни DOM-структури агент, спираючись на контекстні ознаки вузлів, здатний перебудувати селектор, замінивши нестабільний атрибут на інший або скориставшись альтернативним шляхом у графі. Це дозволяє суттєво знизити кількість нестабільних тестів і мінімізувати ручне втручання у підтримку автоматизованих сценаріїв.

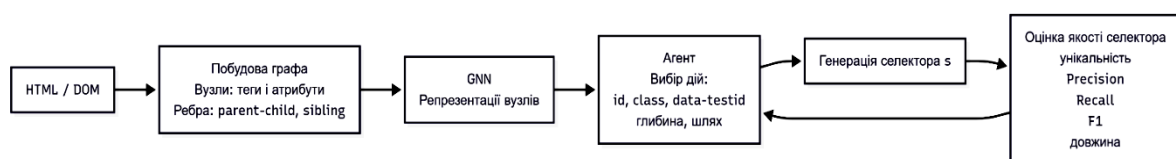


Рисунок 2 – Узагальнена модель функціонування системи

СИСТЕМИ ТЕХНІЧНОГО ЗОРУ І ШТУЧНОГО ІНТЕЛЕКТУ З ОБРОБКОЮ ТА РОЗПІЗНАВАННЯМ ЗОБРАЖЕНЬ

Теоретичну модель роботи запропонованої системи можна описати так: на першому етапі HTML-документ перетворюється у граф, де вузли відповідають тегам та атрибутам, а ребра - відношенням parent-child чи sibling. Побудований граф подається на вхід GNN, яка генерує векторні представлення вузлів. Далі агент, використовуючи ці репрезентації, послідовно формує селектор і перевіряє його якість за визначеними метриками. Отриманий зворотний зв'язок застосовується для оновлення стратегії агента, що підвищує ефективність подальшої побудови селекторів.

Таким чином, поєднання GNN та агента забезпечує теоретично обґрунтовану основу для створення адаптивних селекторів. На відміну від класичних методів, запропонований підхід дозволяє враховувати контекст, прогнозувати стабільність і реагувати на зміни DOM, що робить його перспективним інструментом у задачах оптимізації процесу автоматизованого тестування веб-ресурсів.

ОБГОВОРЕННЯ РЕЗУЛЬТАТІВ

Запропонована модель формування адаптивних селекторів базується на поєднанні графових нейронних мереж (GNN) із агентом прийняття рішень, що використовує механізм оцінювання дій через функцію винагороди. Використання GNN дозволило врахувати не лише локальні атрибути елементів, а й контекстні зв'язки у структурі DOM-графа, що підвищує точність і стійкість побудованих селекторів.

У порівнянні з традиційними евристичними методами, запропонований підхід забезпечує кращу адаптивність до змін у DOM та дозволяє скоротити кількість нестабільних локаторів. Крім того, запропонована система метрик – унікальність, точність, повнота, F1-міра та довжина селектора – створює можливість кількісного порівняння різних підходів і формує основу для подальших експериментів. Отримані результати свідчать про те, що графове подання DOM у поєднанні з агентом оптимізації рішень є ефективною теоретичною основою для створення інтелектуальних систем тест-автоматизації, здатних адаптуватися до змін у веб-структурах без ручного втручання.

Наукова новизна дослідження полягає у розробленні теоретичної моделі побудови адаптивних DOM-селекторів на основі графових нейронних мереж із використанням агента, що реалізує принципи оптимізації на основі функції винагороди. На відміну від існуючих підходів, у роботі формалізовано векторне подання DOM-графа з урахуванням семантичних, атрибутивних та структурних ознак вузлів, а також функцію оцінки селекторів, що поєднує унікальність, точність, повноту, F1-міру та довжину. Запропонований підхід дозволяє здійснювати оцінку та побудову селекторів із урахуванням контексту DOM-графа без необхідності повного циклу підсилювального навчання, що спрощує обчислення й робить модель більш придатною до практичного впровадження в інструменти тест-автоматизації.

Практична цінність отриманих результатів полягає у створенні методологічної основи для автоматичного формування та адаптації DOM-селекторів у процесі автоматизованого тестування. Використання графового подання структури веб-сторінки та системи метрик якості дозволяє зменшити кількість flaky-тестів, скоротити час на підтримку тестових сценаріїв і підвищити стабільність систем тест-автоматизації. Розроблена модель може бути інтегрована у сучасні фреймворки (Selenium Grid, Cypress, Puppeteer) або використана для створення інтелектуальних модулів, які автоматично аналізують і перебудовують селектори під час змін DOM, що безпосередньо підвищує ефективність процесу тестування веб-ресурсів.

ВИСНОВКИ

Здійснено формалізацію подання HTML-документа у вигляді графа, де вузли відображають DOM-елементи, а ребра - структурні зв'язки типу parent-child та sibling. Запропоновано систему ознак вузлів, які включають як атрибутивні характеристики (нааявність id, class, data-testid), так і структурні (глибина, кількість нащадків). Побудовано матрицю ознак вузлів, яка формується шляхом об'єднання векторів ознак для кожного вузла, з подальшим масштабуванням та подачею на вхід GNN.

Визначено систему метрик для оцінювання якості побудованого селектора: унікальність, точність (Precision), повнота (Recall), F1-міра та довжина. Запропоновано формулу оцінювання довжини з урахуванням вагових коефіцієнтів для різних типів компонентів селектора (id, class, nth-child тощо), що дозволяє об'єктивно вимірювати компактність локатора. Такий підхід забезпечує математичну основу для навчання агента на послідовності дій, спрямованих на побудову оптимального селектора. Описано теоретичну модель функціонування системи: побудова DOM-графа → генерація ознак → GNN-репрезентації вузлів → робота агента над побудовою селектора → обчислення метрик → оновлення стратегії на основі винагороди. Особливої уваги надано здатності системи до адаптації — при зміні

СИСТЕМИ ТЕХНІЧНОГО ЗОРУ І ШТУЧНОГО ІНТЕЛЕКТУ З ОБРОБКОЮ ТА РОЗПІЗНАВАННЯМ ЗОБРАЖЕНЬ

структури DOM агент може перебудувати селектор, використовуючи альтернативні шляхи у графі, що дозволяє мінімізувати флаку-тести й витрати на підтримку тестів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ / REFERENCES

1. Lawson, R. *Web Scrapping with Python*; Packt Publishing: Birmingham, UK, 2015
2. K. Parikh, D. Singh, D. Yadav, and M. Rathod, "Detection of web scraping using machine learning," Open access international journal of Science and Engineering, pp. 114–118, 2018.
3. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations (2017). DOI: <https://doi.org/10.48550/arXiv.1609.02907>.
4. Veličković, P. et al. Graph attention networks. *International Conference on Learning Representations* (ICLR, 2017). DOI: <https://doi.org/10.48550/arXiv.1710.10903>.
5. Jianan Zhao, Xiao Wang, Chuan Shi, Binbin Hu, Guojie Song, and Yanfang Ye. 2021. Heterogeneous graph structure learning for graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 4697–4705. DOI: <https://doi.org/10.1609/aaai.v35i5.16600>.
6. Zheng Y, Liu Y, Xie X, et al. Automatic web testing using curiosity-driven reinforcement learning. In: *Proceedings of the 43rd International Conference on Software Engineering*, 2021. 423–435. DOI: <https://doi.org/10.48550/arXiv.2103.06018>.
7. Vu Nguyen, Thanh To, and Gia-Han Diep. 2021. Generating and selecting resilient and maintainable locators for Web automated testing. *Software Testing, Verification and Reliability* 31, 3 (2021), e1760. DOI: <https://doi.org/10.1002/stvr.1760>.
8. Nguyen, L. M., Hui, S. C., & Fong, C. A. (2013). Large-scale multiobjective static test generation for web-based testing with integer programming. *IEEE Transactions on Learning Technologies*, (1), 46–59. DOI: <https://doi.org/10.1109/TLT.2012.22>.
9. M. T. Taky, Automated Testing With Cypress. PhD thesis, 2021.
10. Sanjay Dutta, What is One-hot Encoding in Machine Learning? A Comprehensive Guide with examples. Medium. Режим доступу https://medium.com/@sanjay_dutta/what-is-one-hot-encoding-in-machine-learning-a-comprehensive-guide-with-examples-090f037a6bbe
11. B. Yeoh and H. Wang, "Grown+ up: A "graph representation of a webpage " network utilizing pre-training," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 2372–2382, 2022. DOI: <https://doi.org/10.1145/3511808.3557340>.
12. Sheng Jia, Jamie Kiros, Jimmy Ba, "DOM-Q-NET: Grounded RL on Structured Language". International Conference on Learning Representations (2017). DOI: <https://doi.org/10.48550/arXiv.1902.07257>
13. Y. Lan, Y. Lu, Z. Li, M. Pan, W. Yang, T. Zhang, and X. Li, "Deeply reinforcing android gui testing with deep reinforcement learning," in *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*, 2024, pp. 1–13. DOI: <https://doi.org/10.1145/3597503.3623344>
14. Ali Hassaan Mughal. 2025. An Autonomous RL Agent Methodology for Dynamic Web UI Testing in a BDD Framework. DOI: <https://doi.org/10.48550/arXiv.2503.08464>

Надійшла до редакції 17.07.2025 р.

МОРОЗОВ ОЛЕКСАНДР СЕРГІЙОВИЧ – аспірант кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, **e-mail: morozovsasha97@gmail.com**

ЯРОВИЙ АНДРІЙ АНАТОЛІЙОВИЧ – доктор технічних наук, професор, завідувач кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, **e-mail: a.yarovyy@vntu.edu.ua**

O.S.MOROZOV, A.A. YAROVYI
INTELLECTUAL MODEL FOR GENERATING ADAPTIVE WEB SELECTORS BASED ON GNN
Vinnytsia National Technical University