УДК 004.738.5

Синьков Егор Дмитриевич,
студент 4 курс
ФГБОУ ВО «Нижегородский государственный технический университет им. Р. Е. Алексеева», г. Нижний Новгород

РАЗРАБОТКА АДАПТИВНОГО ПАРСЕРА ДЛЯ E-COMMERCE ПЛАТФОРМ С ДИНАМИЧЕСКИМ ИЗМЕНЕНИЕМ СТРУКТУРЫ DOM

Аннотация. *Статья посвящена проблеме извлечения данных из современных e-commerce платформ, активно использующих технологии динамического рендеринга, обфускации и непрерывного изменения структуры DOM. В статье рассматриваются алгоритмы адаптивного парсинга, основанные на эвристическом анализе и машинном обучении, позволяющие осуществлять устойчивое распознавание веб-элементов без привязки к жестким CSS-селекторам и XPath. Проанализированы статистические данные внедрения SPA-фреймворков в электронной коммерции, доказывающие необходимость отказа от классических подходов к веб-скрапингу. Предложена архитектура гибридного парсера, способного автоматически подстраиваться под A/B-тестирование и изменения верстки целевых сайтов.*

Ключевые слова: *веб-скрапинг, парсинг данных, электронная коммерция, динамический DOM, машинное обучение, автоматизация, системный анализ.*

Annotation: *The article is devoted to the problem of data extraction from modern e-commerce platforms that actively use dynamic rendering technologies, obfuscation, and continuous changes in the DOM structure. The article considers adaptive parsing algorithms based on heuristic analysis and machine learning, which allow robust recognition of web elements without being tied to rigid CSS*

selectors and XPath. Statistical data on the implementation of SPA frameworks in e-commerce are analyzed, proving the need to abandon classic approaches to web scraping. An architecture of a hybrid parser capable of automatically adapting to A/B testing and layout changes of target sites is proposed.

Key words: *web scraping, data parsing, e-commerce, dynamic DOM, machine learning, automation, systems analysis.*

В современных условиях стремительного развития электронной коммерции (e-commerce) доступ к актуальным рыночным данным является критическим фактором конкурентоспособности. Мониторинг ценообразования, анализ ассортимента конкурентов и агрегация отзывов требуют автоматизированных систем сбора информации. Однако процесс веб-скрапинга сталкивается с серьезным технологическим барьером: эволюцией подходов к фронтенд-разработке. Если в начале 2010-х годов преобладали статические HTML-страницы, то сегодня доминируют технологии Single Page Application (SPA), такие как React, Vue и Angular. Согласно статистическим исследованиям рынка веб-технологий, на конец 2025 года более 82% ведущих мировых e-commerce платформ используют динамический рендеринг и технологии генерации CSS-классов на лету (например, CSS-in-JS, Tailwind), что делает структуру Document Object Model (DOM) крайне нестабильной [2, с. 114]. Это означает, что классические методы парсинга, опирающиеся на фиксированные XPath-пути или имена классов, теряют свою актуальность. При любом обновлении сайта или проведении A/B-тестирования такие парсеры неминуемо выходят из строя, требуя постоянного ручного вмешательства и переписывания кода разработчиком [1, с. 45].

Для решения данной проблемы в рамках системного анализа предлагается подход, основанный на разработке адаптивного парсера. Адаптивный парсер — это программный комплекс, способный самостоятельно идентифицировать нужные узлы (nodes) в дереве DOM, опираясь не на их синтаксические атрибуты (id, class), а на семантические,

топологические и визуальные признаки. В основе предлагаемой архитектуры лежит гибридный метод идентификации элементов, включающий три основных уровня анализа:

1) Текстово-семантический анализ. Использование методов обработки естественного языка (NLP) для поиска ключевых паттернов в текстовых узлах. Например, цена товара чаще всего представляет собой числовой паттерн, рядом с которым находится символ валюты.

2) Анализ иерархии и соседей (топологический). Если целевой элемент "Кнопка купить" изменил свой класс, его все равно можно найти по относительному расположению к изображению товара или блоку с ценой в графовом представлении DOM.

3) Визуально-геометрический анализ. При использовании headless-браузеров (Puppeteer, Selenium) парсер может анализировать Bounding Client Rect (координаты и размеры элементов на экране). «Визуальная» кнопка добавления в корзину имеет определенные геометрические стандарты и располагается в первой трети экрана в карточке товара.

Реализация адаптивного механизма требует построения пайплайна предварительной обработки DOM-дерева. Изначально HTML-код преобразуется в абстрактное синтаксическое дерево (AST), из которого удаляются все незначимые для анализа теги (script, style, meta). Затем к оставшимся узлам применяется алгоритм векторизации, где каждый узел получает вектор признаков: размер текста, наличие числовых значений, глубина вложенности, геометрические координаты. На основе собранных признаков обучается модель классификации (например, Random Forest или градиентный бустинг XGBoost), которая предсказывает вероятность принадлежности неизвестного DOM-узла к искомому классу данных (например, Product_Title, Product_Price, SKU). Практические эксперименты показывают, что такой подход снижает частоту сбоев парсера при изменении

верстки целевого сайта на 94% по сравнению с жесткой привязкой к селекторам [3, с. 72].

В таблице 1 представлен сравнительный анализ классического и предлагаемого адаптивного подходов к парсингу e-commerce платформ.

Таблица 1.

Сравнительный анализ методов парсинга

Критерий сравнения	**Классический парсер (XPath/CSS)**	**Адаптивный парсер (Эвристика + ML)**
Устойчивость к изменениям классов	Низкая (требует переписывания кода)	Высокая (опирается на семантику и геометрию)
Скорость разработки под новый сайт	Быстрая (ручной поиск селекторов)	Средняя (требует сбора датасета или настройки правил)
Потребление ресурсов (CPU, RAM)	Низкое (обычно работает через GET-запросы)	Высокое (требует запуска Headless-браузера)
Устойчивость к A/B тестированию	Неустойчив	Устойчив
Обработка динамической подгрузки (Ajax)	Требует реверс-инжиниринга API	Обрабатывает нативно (ждет рендеринга DOM)

Переход от директивного парсинга к адаптивному требует больших вычислительных затрат на этапе выполнения скрипта, так как системе необходимо не просто скачать HTML, но и выполнить полный рендеринг страницы в виртуальном браузере [5]. Однако в долгосрочной перспективе с точки зрения системной инженерии и управления проектами, затраты на вычислительные мощности окупаются многократно за счет радикального снижения расходов на поддержку и обновление кода.

Таким образом, разработка адаптивного парсера для платформ с динамическим изменением структуры DOM является актуальной и необходимой задачей. Использование методов системного анализа, машинного обучения и геометрического анализа веб-страниц позволяет создать отказоустойчивую систему сбора данных, способную стабильно

функционировать в условиях современной веб-разработки и активного противодействия скрапингу. Внедрение подобных систем открывает новые возможности для глубокой аналитики рынка электронной коммерции и автоматизации бизнес-процессов [4, с. 25].

Список литературы:

1. Митчелл, Р. Скрапинг веб-сайтов с помощью Python. Сбор данных из современного интернета / Р. Митчелл. — 2-е изд., перераб. и доп. — М.: ДМК Пресс, 2021. — 280 с.

2. Смирнов, А. В. Статистический анализ применения SPA-технологий и динамического рендеринга в сегменте электронной коммерции: тенденции и проблемы / А. В. Смирнов, П. С. Иванов // Вестник информационных технологий. — 2025. — № 4. — С. 112–118.

3. Козлов, Д. И. Применение алгоритмов машинного обучения для устойчивого извлечения данных из веб-страниц / Д. И. Козлов // Системный анализ и программная инженерия. — 2023. — № 2. — С. 68-75.

4. Чернов, А. А. Проблемы агрегации данных в условиях обфускации и непрерывной мутации DOM-дерева / А. А. Чернов, Е. М. Соколова // Инновационные технологии в экономике и управлении. — 2022. — № 8. — С. 22-29.

5. Puppeteer Documentation. Headless Chrome Node.js API [Электронный ресурс]. URL: https://pptr.dev/ (дата обращения: 15.07.2026).