Knowledge Retrieval and Semantic Systems
https://jks.atu.ac.ir/

Allameh Tabataba'i University PRESS

13(46), 2026, 1-24
eISSN: 2783-1795 ISSN: 2980-8243

Article Type: Research Article

Received November 23, 2025
Accepted March 08, 2026

Received in revised form February 19, 2026
Published Online March 25, 2026

Using Dynamic Selectors and Artificial Intelligence to Harden Web Data Retrieval Codes

Farnaz Taghizadeh Kourayem [1], **Mohammadreza Kabaranzad Ghadim** [2]✉, and **Seyed Abdollah Amin Mousavi** [3]

1. Department of Information Technology Management, CT.C., Islamic Azad University, Tehran, Iran. E-mail: farnaz.taghizadehkourayem@iau.ac.ir
2. *Corresponding Author,* Department of Industrial Management, CT.C., Islamic Azad University, Tehran, Iran. E-mail: moh.kabaranzad@iauctb.ac.ir
3. Department of Information Technology Management, CT.C., Islamic Azad University, Tehran, Iran. E-mail: Saa.mousavi@iau.ac.ir

ABSTRACT

Today, the sustainability and resilience of web data extraction codes are one of the main challenges in software engineering and data mining, particularly when target websites use dynamic structures. Selenium, as one of the most widely used libraries for web scraping, is utilized in many industrial and research projects; however, its high sensitivity to changes in web page elements often results in errors, system interruptions, and the need for frequent code modifications. This research presents a novel method that combines "dynamic selectors" with an AI-based decision-making layer to provide a functional, reliable, and flexible approach for strengthening Selenium-based code. In this study, data were collected from the Digikala website over a two-year period with weekly intervals. The proposed method analyzes the behavior of page elements, automatically replaces the appropriate selectors, and applies multiple intelligent fallback strategies in case of failure. This approach increases the stability and reliability of selenium execution against website changes and can serve as an efficient model for web data collection systems operating in dynamic environments. Experimental results showed that when the proposed method was used for automatic and intelligent selection of HTML elements, all data extraction operations were completed without errors and without requiring manual code modifications. However, when the method was not applied, 67 extraction attempts required direct selector corrections by the developer. This performance difference demonstrates that the presented model significantly reduces maintenance costs, development time, and the risk of extraction process failure.

Keywords: Artificial intelligence, HTML code hardening, HTML structure of web pages, Selenium library, web data retrieval

Cite this Article: Taghizadeh Kourayem, F., Kabaranzad Ghamid, M. R, Mousavi, S. A. A. (2026). Using Dynamic Selectors and Artificial Intelligence to Harden Web Data Retrieval Codes. *Knowledge Retrieval and Semantic Systems*, *13*(46), 1-24. https://doi.org/10.22054/jks.2026.89870.1753

Publisher: Allameh Tabataba'i University Press
DOI: https://doi.org/10.22054/jks.2026.89870.1753

EXTENDED ABSTRACT

Introduction

The robustness of web scraping systems against structural changes in web pages remains a major challenge. Selenium-based solutions are highly vulnerable because they depend on fixed selectors. Existing robustness approaches often suffer from high computational costs or low reliability. To address this gap, this paper proposes an AI-based self-healing approach that dynamically analyzes element behavior and automatically selects the most suitable alternative selector, enabling continuous and reliable data extraction without manual intervention.

Literature Review

The studies by Kirinuki et al. (2019), Nass et al. (2023), and Coppola et al. (2025) propose alternative selectors to improve the reliability of web data extraction code using various cues. However, these methods exhibit relatively high error rates and are not suitable for pages with significant structural changes or elements with similar attributes. Although the use of large language models was proposed by Nass et al. (2024) to improve these methods and sometimes reduce errors, it raises privacy concerns and increases computational costs. In contrast, the present paper proposes a method that not only reduces errors but is also effective for pages with major structural changes while avoiding privacy and computational cost issues.

The works of Degaki et al. (2022), Khaliq et al. (2023), and Daneshvar and Wang (2024) propose using computer vision to predict elements on web pages. However, computer vision techniques show low accuracy in detection and prediction when the target information is limited, and they also involve high computational and time costs. In contrast, the present paper introduces a method that retrieves precise information while minimizing computational and time costs.

Taghizadeh et al. (2024) present a mechanism for time management and improving the accuracy of information retrieval from web platforms using Selenium. However, their method only identifies the location of errors and returns erroneous values as null, without providing a solution for code robustness. In contrast, the present paper builds on the method proposed by Taghizadeh et al. (2024) as a baseline for web data extraction and offers a solution for code robustness and automatic extraction of target values with the highest possible accuracy.

Saarathy et al. (2024) conducted a systematic review of existing studies on Selenium code robustness but did not propose a robustness model for web data extraction. In contrast, the present paper reviews existing studies and models and proposes a dedicated robustness model for Selenium code.

Healenium (2025) stores successful Selenium element information for future use but still requires manual intervention when errors occur. In contrast, the present paper automatically identifies alternative scripts using AI algorithms, reducing recovery time and human involvement.

Kluge and Stocco (2025) concluded that robust web scraping requires hybrid methods and automated validation, but they did not propose a new model. In contrast, the present paper introduces a model designed to overcome the limitations of existing approaches.

Methodology

This research presents an automated and robust framework for sustainable web data extraction (web scraping) that aims to overcome the inherent fragility of tools such as Selenium when facing structural changes in web pages. The proposed method is a hybrid system that integrates a relational database (SQL), Python and Selenium scripts, and artificial intelligence algorithms (particularly time-series models) to automatically identify and replace faulty selectors (XPath/CSS).

1. Data Infrastructure Design and Collection

Database: A SQL database was designed for structured storage, consisting of three main components:

- **Products table:** stores extracted product information (name, price, discount, rating, image, label, and date).
- **HTML Tags table:** records the complete tree structure (parent–child relationships) of HTML tags from the target website's product pages (e.g., Digikala). This table includes the XPath (XML Path Language) of each tag, the extraction date, and a Boolean flag called "used", which indicates whether a given XPath has previously been used successfully to extract a target element (e.g., a product name).
- **Tag Count View (virtual table):** calculates the frequency of each unique XPath in the HTML tags table. This view enables the algorithm to prioritize XPaths that appear an expected number of times (e.g., 20 times on a 20-product page) as candidates for repeated elements (such as product names).

2. Extraction and Retrieval Engine

Primary Extraction Script (Selenium): A Selenium-based script was implemented to extract product information from target web pages using initially defined fixed selectors.

Structure Retrieval and Analysis Module: If the primary script encounters an error or returns empty results, this indicates a possible structural change in the page. In such cases, a backend script is triggered that:

- Reads the full HTML structure of the page and processes it after removing static sections (e.g., headers and footers).
- Uses regular-expression (Regex) algorithms to extract the full XPath of all tags and stores them in the HTML tags table.

3. Intelligent Decision-Making and Automatic Repair Layer

AI Models (Time Series): The core of the repair system is a predictive model that applies time-series algorithms (such as LSTM) to analyze the historical behavior of the "used" flag for each XPath. The model was trained on six months of historical data initially collected and labeled through manual supervision.

Iterative Repair Process: When an error occurs:

- The model predicts and proposes replacement XPaths with a high probability of being marked as "used" in the future.
- Candidate XPaths are inserted into the Selenium script in priority order, and the script is re-executed.

Homepage://jks.atu.ac.ir/

- If execution succeeds, the corresponding flag is updated in the database. If it fails, unsuccessful candidates are discarded and the process continues with the next options.
- If no solution is found, the system falls back to a full rule-based traversal of the tags table.

4. Validation and Output Quality Control

To ensure the accuracy of data extracted using alternative selectors, the system incorporates validation rules:

- **Quantitative validation:** The number of extracted items for repeated elements (e.g., product names) must match the expected count (e.g., 20).
- **Qualitative validation (data patterns):** Extracted data must conform to predefined patterns (e.g., product names as strings within a maximum length, discount percentages as numbers followed by a % sign, and labels from predefined categories).

Experimental Evaluation

The system was evaluated on the Digikala website over a two-year period with weekly intervals. Evaluation metrics included extraction success rate, requirement for manual intervention, execution time, and prediction model accuracy. Results indicate that the proposed system achieved near-complete stability, eliminated the need for manual correction, and reduced the worst-case average execution time to approximately 9 minutes for 60 products, demonstrating significantly higher efficiency compared to alternative methods (e.g., image-based approaches). The LSTM model achieved very high accuracy in predicting correct selectors in many cases.

Conclusion

This study presents an automated, AI-based approach for robust web scraping. By analyzing the HTML structure, systematically extracting XPaths, and employing intelligent decision-making algorithms (such as LSTM) to dynamically select alternative selectors, the inherent fragility of tools such as Selenium is mitigated. The results demonstrate that this method can significantly reduce the data extraction failure rate and eliminate the need for manual intervention. The proposed system, by learning from historical data, evolves into a semi-autonomous web crawler that ensures the sustainability of knowledge and data retrieval systems in organizational and research applications.

Conflict of Interest

"The authors declare no conflict of interest."

بازیابی دانش و نظام‌های معنایی
https://jks.atu.ac.ir/

انتشارات دانشگاه علامه طباطبائی

دوره ۱۳، شماره ۴۶، ۱۴۰۵، ۱-۲۴
eISSN: 2783-1795 ISSN: 2980-8243

نوع مقاله: مقاله پژوهشی

تاریخ دریافت: ۱۴۰۴/۰۹/۰۲؛ تاریخ بازنگری: ۱۴۰۴/۱۱/۳۰
تاریخ پذیرش: ۱۴۰۴/۱۲/۱۷؛ تاریخ انتشار: ۱۴۰۵/۰۱/۰۵

استفاده از انتخابگرهای پویا و هوش مصنوعی برای مقاوم‌سازی کدهای بازیابی اطلاعات وب

فرناز تقی‌زاده کورایم [۱]، محمدرضا کاباران زاد قدیم [۲]✉، و سید عبداله امین موسوی [۳]

۱. گروه مدیریت فناوری اطلاعات، واحد تهران مرکزی، دانشگاه آزاد اسلامی، تهران، ایران. رایانامه: farnaz.taghizadehkourayem@iau.ac.ir
۲. *نویسنده مسئول*، گروه مدیریت صنعتی، واحد تهران مرکزی، دانشگاه آزاد اسلامی، تهران، ایران. رایانامه: moh.kabaranzad@iauctb.ac.ir
۳. گروه مدیریت فناوری اطلاعات، واحد تهران مرکزی، دانشگاه آزاد اسلامی، تهران، ایران. رایانامه: Saa.mousavi@iau.ac.ir

چکیده

امروزه پایداری و تاب‌آوری کدهای برداشت اطلاعات از وب یکی از چالش‌های اصلی در حوزه مهندسی نرم‌افزار و داده‌کاوی است، به‌ویژه زمانی که وب‌سایت‌های هدف از ساختارهای پویا استفاده می‌کنند. کتابخانه سلنیوم به‌عنوان یکی از پرکاربردترین ابزارهای برداشت اطلاعات وب، در بسیاری از پروژه‌های صنعتی و پژوهشی مورداستفاده قرار می‌گیرد، اما حساسیت بالای آن نسبت به تغییر عناصر صفحه اغلب منجر به بروز خطا، توقف سیستم و نیاز به اصلاحات مکرر کد می‌شود. این پژوهش با ارائه روش نوین مبتنی بر استفاده هم‌زمان از «انتخابگرهای پویا» و یک لایه تصمیم‌گیری مبتنی بر هوش مصنوعی، رویکردی کاربردی، قابل‌اعتماد و منعطف برای مقاوم‌سازی کدهای سلنیوم ارائه می‌دهد. در این مطالعه، برداشت اطلاعات از وب‌سایت دیجی‌کالا در بازه زمانی دوساله، با تناوب هفتگی انجام شد. روش پیشنهادی با تحلیل رفتار المان‌های صفحه، به طور خودکار انتخابگرهای مناسب را جایگزین و در صورت خطا از چندین استراتژی هوشمند استفاده می‌کند. این رویکرد پایداری و اطمینان اجرای سلنیوم را در برابر تغییرات سایت افزایش داده و می‌تواند الگویی کارآمد برای سامانه‌های جمع‌آوری داده از وب در محیط‌های پویا باشد. نتایج تجربی نشان داد هنگامی‌که از روش پیشنهادی جهت انتخاب خودکار و هوشمند المان‌های HTML استفاده شد، تمامی برداشت‌ها بدون خطا و بدون نیاز به تغییر دستی کد انجام شدند؛ اما هنگام عدم استفاده از این روش، ۶۷ برداشت نیازمند اصلاح مستقیم انتخابگرهای کد توسط توسعه‌دهنده بودند. این اختلاف عملکرد نشان می‌دهد که استفاده از مدل ارائه‌شده باعث کاهش چشمگیر هزینه نگه‌داری، زمان توسعه و ریسک شکست اجرای عملیات استخراج داده می‌شود.

کلیدواژه‌ها: بازیابی اطلاعات از بستر وب، ساختار HTML صفحات وب، کتابخانه سلنیوم، مقاوم‌سازی کدهای HTML، هوش مصنوعی

استناد به این مقاله: تقی‌زاده کورایم، فرناز، کاباران زاد قدیم، محمدرضا، و موسوی، سیدعبداله امین. (۱۴۰۵). استفاده از انتخابگرهای پویا و هوش مصنوعی برای مقاوم سازی کدهای بازیابی اطلاعات وب. *بازیابی دانش و نظام‌های معنایی*، ۱۳(۴۶)، ۱-۲۴.
https://doi.org/10.22054/jks.2026.89870.1753

ناشر: دانشگاه علامه طباطبائی

مقدمه

رشد روزافزون داده‌های موجود در وب و نیاز سازمان‌ها به بازیابی آنها، تحلیل مستمر و بلادرنگ اطلاعات موجود در این داده‌ها و درنهایت بازیابی دانش و تصمیم‌سازی معنایی بر اساس آن‌ها، اهمیت فرایند «برداشت داده از وب» یا وب‌اسکرپینگ[1] را بیش‌ازپیش آشکار کرده است. در این فرایند، خزنده‌های وب به‌عنوان برنامه‌های خودکار، صفحات اینترنتی را پیمایش کرده، داده‌های موردنیاز را شناسایی و استخراج می‌کنند. این سامانه‌ها امروز بخش جدایی‌ناپذیر از ربات‌های هوشمند بازیابی و ارائه دانش نظیر چت جی پی تی[2] و سیستم‌های مختلفی ازجمله سیستم‌های تست نرم‌افزاری تحت وب و گزارش خطا، تحلیل کسب‌وکار، پایش قیمت، رصد رفتار مشتریان، جمع‌آوری اطلاعات رقبا، پایش بازارهای مالی و بسیاری از خدمات مبتنی بر داده محسوب می‌شوند (Taghizadeh et al., 2024).

در صنایع کاربردی و محیط‌های پژوهشی، موفقیت سامانه‌های برداشت اطلاعات از وب زمانی تضمین می‌شود که این سیستم‌ها بتوانند در طول زمان عملکرد پایدار داشته و در برابر تغییرات ساختار صفحات وب مقاوم باشند. این مسئله به دلیل آن اهمیت دارد که امروزه اغلب وب‌سایت‌ها از ساختارهای پویای زبان نشانه‌گذاری ابرمتنی[3] و تغییرات مداوم در تگ‌ها و کلاس‌ها استفاده می‌کنند. چنین پویایی موجب می‌شود روش‌های کلاسیک استخراج داده با کوچک‌ترین تغییر در ساختار صفحه دچار خطا شده و نیازمند اصلاح و نگه‌داری مداوم باشند؛ مشکلی که هزینه‌های عملیاتی را افزایش داده و اعتبار سیستم‌های جمع‌آوری داده را تهدید می‌کند (Degaki et al., 2022).

در میان ابزارهای موجود، کتابخانه سلنیوم[4] یکی از پرکاربردترین فناوری‌ها برای برداشت اطلاعات از وب و شبیه‌سازی رفتار انسان در مرورگر است. قابلیت تعامل با عناصر پویا، پشتیبانی از مرورگرهای واقعی و کاربرد گسترده در تست نرم‌افزار باعث شده است سلنیوم انتخاب اصلی بسیاری از پژوهشگران و شرکت‌ها باشد. بااین‌حال نقطه‌ضعف بنیادی سلنیوم وابستگی مطلق آن به انتخابگرهای ثابت است. تغییر در ساختار صفحه – حتی در سطح یک کلاس یا تگ – می‌تواند موجب شکست کامل اجرای برنامه و توقف فرایند استخراج داده شود. این مسئله به‌ویژه در سناریوهایی که برداشت داده باید مستمر، بلندمدت و بدون مداخله دستی باشد به چالش مهمی تبدیل شده است (Nass et al., 2023).

در سال‌های اخیر روش‌های متنوعی برای افزایش پایداری استخراج داده از وب ارائه شده است. برخی از این روش‌ها بر انتخابگرهای سنتی و سلسله‌مراتبی تکیه دارند که در تغییرات جزئی عملکرد مناسبی دارند؛ اما در تغییرات اساسی نیازمند اصلاح دستی هستند. گروهی دیگر از پردازش تصویر برای تشخیص الگوهای بصری استفاده کرده‌اند که انعطاف‌پذیرند اما زمان‌بر بوده و برای المان‌های ریز ناکارآمد هستند. دسته سوم با استفاده از ویژگی‌های داخلی تگ‌ها مانند نام کلاس تلاش به افزایش پایداری کرده‌اند، اما این ویژگی‌ها به‌شدت ناپایدار بوده و تغییرات طراحی رابط کاربری موجب توقف استخراج داده توسط آنها می‌شود (Saarathy et al., 2024).

1. Web Scraping
2. ChatGPT
3. HTML
4. Selenium Library

این وضعیت نشان می‌دهد که شکافی مهم در دانش موجود وجود دارد: نبود رویکردهای مقاوم و خودترمیم‌شونده که در زمان تغییر ساختار صفحات وب بتوانند بدون افزایش قابل‌توجه هزینه پردازش، به‌صورت خودکار، دقیق‌ترین انتخابگرهای جایگزین معتبر را شناسایی و استفاده کنند؛ به‌عبارت‌دیگر، سیستمی که به‌صورت خودکار و بدون افزایش قابل‌توجه در هزینه‌های محاسباتی و زمانی، بتواند داده‌ها و دانش موجود بین آنها را از بستر وب بازیابی کند و یک نظام معنایی جامع و خودترمیم‌شونده را ایجاد کند، وجود ندارد. باتوجه‌به نیاز صنعت، کسب‌وکارها و پژوهشگران به سیستم‌های پایدار، رویکردهایی مبتنی بر هوش مصنوعی و انتخابگرهای پویا می‌توانند تحول قابل‌توجهی در این حوزه ایجاد کنند.

تحقیق حاضر با هدف پر کردن این شکاف، یک رویکرد خودکار مقاوم‌سازی کدهای سلنیوم مبتنی بر تحلیل رفتار عناصر، گزینش پویا و هوشمند انتخابگرهای جایگزین و تصمیم‌گیری مبتنی بر الگوریتم‌های هوش مصنوعی به‌منظور بازیابی خودکار دانش و ایجاد یک نظام معنایی جامع و خودترمیم‌شونده از داده‌های موجود در بستر وب ارائه می‌دهد. این روش به سیستم اجازه می‌دهد بدون نیاز به تغییر دستی کد در کوتاه‌ترین زمان ممکن، در هنگام ایجاد خطا، مجموعه‌ای از راهکارهای جایگزین را آزموده و انتخاب معتبرترین گزینه را انجام دهد. چنین قابلیتی زمینه ایجاد نسل جدیدی از خزنده‌های وب پایدار، هوشمند و کم‌هزینه را فراهم کرده و می‌تواند نقش مهمی در توسعه سامانه‌های جمع‌آوری داده سازمانی و پژوهشی ایفا کند.

پیشینه پژوهش

برای توسعه سامانه‌ای مقاوم و هوشمند جهت برداشت داده از وب با استفاده از سلنیوم، لازم است وضعیت موجود دانش و پژوهش‌های مرتبط به‌صورت نظام‌مند بررسی شود. ازاین‌رو در این بخش، پیشینه پژوهش در دو حوزه اصلی شامل «مقاوم‌سازی وب اسکریپت‌ها و انتخابگرهای پویا» و «کاربرد هوش مصنوعی در افزایش پایداری فرایندهای استخراج داده از وب» مرور شده است.

پژوهش‌های دو دهه اخیر نشان می‌دهند که یکی از چالش‌های اساسی در وب‌اسکرپینگ، وابستگی شدید انتخابگر تگ‌ها به ساختار صفحات وب است که با کوچک‌ترین تغییر در ویژگی تگ‌ها یا کلاس‌ها، باعث شکست کامل یا جزئی اجرای اسکریپت‌ها می‌شود. این مسئله در کاربردهای واقعی نظیر پایش قیمت، تحلیل رقبا، خزنده‌های وب و سیستم‌های هوشمند جمع‌آوری داده، به اتلاف زمان، افزایش هزینه نگه‌داری و نیاز به اصلاح مستمر کد منجر می‌شود. به همین دلیل، محققان رویکردهایی مانند یادگیری ویژگی‌های پایدار صفحات، تولید خودکار انتخابگرها، تحلیل رفتاری المان‌ها و الگوریتم‌های خودترمیمی را پیشنهاد کرده‌اند (Kluge & Stocco, 2025).

برای دستیابی به تصویری جامع و دقیق، ۱۱ مقاله معتبر و به‌روز از سال‌های اخیر بررسی شده‌اند. این مطالعات با معیارهایی نظیر هدف پژوهش، روش‌شناسی، متریک‌های ارزیابی، نوع داده‌ها و درنهایت شکاف یا محدودیت اعلام‌شده توسط نویسندگان تحلیل شده‌اند. مرور این مطالعات نشان می‌دهد که اگرچه پژوهش‌های خوبی در حوزه بهبود انتخابگرها و مدیریت خطا ارائه شده است، اما بیشتر آن‌ها در محیط‌های آزمایشگاهی یا داده‌های مصنوعی انجام شده‌اند و تعداد اندکی از آن‌ها به اجرای طولانی‌مدت در شرایط واقعی وب، آن هم در مقیاس زمانی چندساله، توجه کرده‌اند.

ازسوی‌دیگر، اکثر پژوهش‌ها تنها به اصلاح انتخابگر پس از وقوع خطا پرداخته‌اند و کمتر به مسئله «یادگیری تدریجی و کاربرد تجارب اجرای قبلی» برای افزایش تاب‌آوری بلندمدت سیستم اشاره کرده‌اند. این شکاف پژوهشی، ضرورت توسعه روشی را مطرح می‌کند که ضمن اجرای واقعی در محیط وب پویا، قادر باشد انتخابگرهای پایدارتر تولید کرده، تجربه اجرای پیشین را در تصمیم‌گیری لحاظ کند و بدون نیاز به مداخله انسانی پایداری اسکریپت را تضمین نماید.

بررسی شکاف‌های شناسایی‌شده در پژوهش‌های گذشته، ضرورت انجام مطالعه حاضر را تقویت می‌کند؛ زیرا پژوهش جاری با استفاده از هوش مصنوعی و انتخابگرهای پویا، در یک بازه زمانی طولانی دوساله و کاملاً واقعی، به ارزیابی عملی پایداری و قابلیت خودترمیمی سیستم پرداخته است. در ادامه ابتدا تعریف برخی از مفاهیم استفاده شده در این پژوهش و سپس هدف، روش، متریک، نتایج، داده‌ها، و چالش و شکاف‌های تحقیقاتی موجود در پژوهش‌های پیشین آمده است (جدول ۱).

۱) المان هدف در صفحه سایت: المان هدف، به یک جزء مشخص از ساختار صفحه وب اطلاق می‌شود که ازنظر پژوهشگر یا الگوریتم، مورد تحلیل، استخراج، شناسایی یا پردازش قرار می‌گیرد. این المان می‌تواند شامل نام یک محصول، قیمت آن، یک برچسب، یک بخش متنی، تصویر، لینک یا هر عنصر دیگری باشد که در لایه ارائه صفحه وب حضور داشته و توسط مرورگر قابل‌نمایش به کاربر نهایی است.

۲) تگ: تگ یا برچسب، کوچک‌ترین واحد نشانه‌گذاری در یک سند «زبان نشانه‌گذاری ابرمتنی»[1] است که برای تعریف ماهیت، نقش و رفتار یک جزء در صفحه وب مورداستفاده قرار می‌گیرد. هر تگ معمولاً از یک تگ باز و یک تگ بسته تشکیل شده و می‌تواند متن، المان‌های فرزند یا هر داده دیگری را در برگیرد. تگ‌ها ساختار سلسله‌مراتبی سند زبان نشانه‌گذاری ابرمتنی را شکل داده و اساس مدل شیءگرای سند را تشکیل می‌دهند. مثلاً تگ p یک پاراگراف در صفحه نشان می‌دهد و به‌صورت سلام نوشته می‌شود و می‌تواند تگ‌های زیادی را به‌عنوان فرزند در خود نگه دارد یا خود فرزند تگی دیگر باشد. به‌عنوان‌مثال در hi تگ p فرزند تگ div است.

۳) ساختار «زبان نشانه‌گذاری ابرمتنی» صفحه وب: ساختار «زبان نشانه‌گذاری ابرمتنی»، بیانگر سازمان‌دهی سلسله‌مراتبی عناصر در یک سند وب است که در قالب مجموعه‌ای از تگ‌ها و المان‌های تودرتو تعریف می‌شود. این ساختار تعیین می کند که چگونه داده‌ها در صفحه وب توصیف، گروه‌بندی و نمایش داده شوند و مبنای ایجاد درخت‌واره نحوه نمایش در مرورگر است. ساختار «زبان نشانه‌گذاری ابرمتنی» نه‌تنها بیانگر توالی و ارتباط والد-فرزندی میان عناصر است، بلکه بیان‌کننده نقش معنایی هر بخش از محتوا نیز است.

٤) ویژگی‌های تگ: ویژگی‌های یک تگ، مجموعه‌ای از داده‌های کمکی هستند که در درون تگ بازتعریف شده و به آن معنا، رفتار، شناسه یا ویژگی‌های کنترلی اضافه می‌کنند. ویژگی‌هایی مانند نام کلاس، لینک، اندازه و موارد مشابه، از مهم‌ترین انواع ویژگی‌ها در «زبان نشانه‌گذاری ابرمتنی» به شمار می‌روند. این ویژگی‌ها می‌توانند توسط مرورگر، موتور جاوا اسکریپت و سایر ابزارها برای کنترل نمایش، تعامل، شناسایی و پردازش محتوا مورداستفاده قرار گیرند.

۵) انتخابگر: انتخابگر، الگوی نحوی مورداستفاده برای شناسایی و ارجاع به المان‌های خاص در یک سند «زبان نشانه‌گذاری ابرمتنی» است. انتخابگرها در زبان‌های نشانه‌گذاری و طراحی و همچنین در ابزارهای پیمایش یک سند «زبان نشانه‌گذاری

1. HTML

ابرمتنی» مانند مسیر اختصاصی[1] مورداستفاده قرار می‌گیرند. یک انتخابگر می‌تواند بر اساس نام تگ، شناسه، کلاس، ویژگی ها، روابط سلسله‌مراتبی والد-فرزند یا ترکیبی از این موارد، یک یا چند المان را در ساختار سند انتخاب و هدف‌گذاری کند و یا آن را بازگرداند.

٦) وب اسکرپینگ: یک نرم‌افزار خودکار است که با پیمایش سیستماتیک صفحات وب، داده‌ها و محتوای موجود در اینترنت را بازیابی و ذخیره‌سازی می‌کند. خزنده‌ها معمولاً از یک یا چند نقطه شروع آغاز کرده و با تحلیل ساختارهای موجود در صفحات، به‌صورت بازگشتی اطلاعات صفحات را کشف و برداشت می‌کنند. این ابزارها نقش بنیادی در موتورهای جستجو، پایگاه‌های داده محتوای وب، سیستم‌های پایش تغییرات وب و سامانه‌های استخراج داده ایفا می‌کنند. استفاده از خزنده‌های وب امکان جمع‌آوری سازمان‌یافته داده از منابع گسترده اینترنت را فراهم کرده و مبنای بسیاری از فرآیندهای تحلیل داده و بازیابی اطلاعات در پژوهش‌های علمی و کاربردهای صنعتی محسوب می‌شود.

جدول ۱. هدف، روش، متریک، نتایج، داده‌ها، و چالش و شکاف‌های تحقیقاتی موجود در پژوهش‌های پیشین

ردیف	نویسندگان و سال	هدف	روش	متریک‌ها و نتایج	داده‌ها	چالش/شکاف
۱	(Kirinuki et al., 2019)	توصیه انتخابگر جایگزین با استفاده از سرنخ‌های متنوع (ویژگی‌های تگ، متن، تصویر، موقعیت) برای تعمیر اسکریپت‌های شکست‌خورده	ترکیب سرنخ‌های ساختاری و تصویری و وزن‌دهی آن‌ها برای رتبه‌بندی کاندیدها و توصیه انتخابگرهای محتمل	در برنامه‌های کاربردی تحت وب مختلف دقت بین ۷۷ تا ۹۳ درصد گزارش شده است	چند پروژه متن-باز و مقایسه بین نسخه‌ها	خوب برای تغییرات ساختاری نه‌چندان شدید. در تغییرات ریشه‌ای ظاهر/سیاست‌گذاری‌های جدید یا در صفحات کاملاً بازطراحی‌شده، محدودیت دارد
۲	(Degaki et al., 2022)	بررسی اینکه ترکیب سرنخ‌های بصری (هوش مصنوعی و پردازش تصویر) و تگ‌های صفحه تا چه حد می‌تواند پایداری را بالا ببرد.	ابتدا با بینایی ماشین المان‌های بصری را تشخیص می‌دهند و سپس با تگ‌های صفحه اعتبارسنجی می‌کنند	precision/recall visual detection و robustness در تشخیص تصویر؛ گزارش بهبود نسبت به هر روش تکی (فقط بینایی ماشین و فقط تگ در کتابخانه سلنیوم)	نمونه‌های صنعتی و مطالعات موردی از وب‌سایت‌ها	هماهنگ‌سازی بین فضای تصویر (پیکسل) و فضای تگ‌ها (عنصر ساختاری) پیچیده است؛ همچنین هزینه‌های محاسباتی و تأخیر برای کاربردهای بلادرنگ یک مشکل عملی است.
۳	(Khaliq et al., 2023)	کاربرد یادگیری عمیق برای کمک به شناسایی المان‌ها	چارچوب مبتنی بر شبکه‌های کانولوشنال/encoder	دقت شناسایی المان، repair-rate، coverage تست؛	مجموعه صفحات و تصاویر	نیاز به دیتاست‌های برچسب‌خورده زیاد برای آموزش؛ مشکلات در

1. XPath

ردیف	نویسندگان و سال	هدف	روش	متریک‌ها و نتایج	داده‌ها	چالش/شکاف
		از تصویر/ویژگی های تگ	برای شناسایی المان از اسکرین‌شات و تولید/تکمیل اسکریپت‌های سلنیوم	گزارش بهبود نسبت به روش‌های سنتی	محصول/سبد خرید از اپلیکیشن‌های تجارت الکترونیک	صفحات با چینش‌های متنوع و واکنش‌گرا؛ ادغام تصویر و ویژگی‌های تگ برای تعادل، دقت و پایداری هنوز حل نشده است
٤	(Nass et al., 2023)	بازیابی المان هدف پس از تغییرات صفحه با محاسبه امتیازِ تشابه بین چند ویژگی تگ. نام مدل طراحی شده Similo است	ویژگی‌های یک تگ مثل نام، id، کلاس، متن، موقعیت، اندازه، و ... را با پارامترهای همان عنصر در نسخه قدیمی مقایسه می‌کند و به آن امتیاز می‌دهد و بهترین کاندید را انتخاب می‌کند	۷۲ شکست از ۵۹۸ مورد بازیابی اطلاعات	داده‌های قابل بازیابی ٤٠ وب‌سایتِ پرکاربرد و مقایسه بین نسخه‌های بازیابی مختلف در زمان‌های متفاوت	امکان دارد چند تگ نام، کلاس یا ویژگی‌های کاملاً یکسانی داشته باشند یا در طول زمان ویژگی های تگ به‌صورت کامل در طول زمان تغییر کند.
٥	(Taghizadeh et al., 2024)	شناسایی موقعیت‌هایی که نه خطا بازمی‌گردد، نه کد شکست می خورد و نه کد مربوط به تگ اجرا می‌شود و همچنین مدیریت زمان در اسکریپت‌های سلنیوم	استفاده از اسکریت های نوشته‌شده به زبان پایتون	مقایسه تعداد خطاها هنگام استفاده از اسکریپت و عدم استفاده از اسکریپت. هنگام استفاده از اسکریپت هیچ خطایی اعلام نشده و اطلاعات به‌صورت دقیق در جای خود نشسته‌اند	مطالعه سایت دیجی‌کالا	فقط مقادیر دارای خطا و محل خطا را اعلام می کند. همچنان نیاز به دست‌کاری دستی کدها وجود دارد
٦	(Daneshvar & Wang, 2024)	ارزیابی مدل‌های مدرن object-detection در تشخیص عناصر با استفاده از اسکرین‌شات	استفاده از کتابخانه تشخیص تصاویر یولو روی تصاویر سایت	precision / map / recall	دیتاستی از هزاران اسکرین‌شات برچسب‌خورده از برنامه‌های کاربردی تحت وب	عملکرد خوب در تشخیص اما مشکلِ همپوشانی، آیتم‌های ریز و چگالی بالای عناصر؛ ترکیب با تگ‌های صفحه برای کاهش false positives توصیه شده است. سرعت بسیار پایین

ردیف	نویسندگان و سال	هدف	روش	متریک‌ها و نتایج	داده‌ها	چالش/شکاف
						دارد؛ چراکه برای یک تگ کوچک و مشخص و ثابت هم از هوش مصنوعی استفاده می‌کند که احتمال خطا وجود دارد.
۷	(Nass et al., 2024)	بازیابی المان هدف پس از تغییرات صفحه با محاسبه امتیازِ تشابه بین چند ویژگی تگ و استفاده از یک مدل زبانی بزرگ GPT-4 برای بازیابی دقیق‌تر	ویژگی‌های یک تگ مثل نام، id، کلاس، متن، موقعیت، اندازه، و ... را با پارامترهای همان عنصر در نسخه قدیمی مقایسه می‌کند و به آن امتیاز می‌دهد و بهترین کاندیدها را انتخاب می‌کند. سپس یک مدل زبان بزرگ استفاده می‌شود تا از میان این کاندیدها، بهترین گزینه را انتخاب کند	٤٠ شکست از ٨٠٤ مورد بازیابی اطلاعات	داده‌های قابل بازیابی ٤٨ وب‌سایتِ پرکاربرد و مقایسه بین نسخه‌های بازیابی مختلف در زمان‌های متفاوت	استفاده از یک مدل زبانی بزرگ کمک‌کننده است؛ ولی هزینه‌ها بالا و پاسخ‌ها گاهی غیردقیق و نیاز به کنترل اعتبارسنجی انسانی دارد؛ همچنین مسائل حریم خصوصی هم مطرح‌اند
۸	(Saarathy et al., 2024)	جمع‌بندی وضعیت مقالات و مطالعات موجود در زمینه پایداری کدهای کتابخانه سلنیوم	مرور نظام‌مند مقالات و پروژه‌های صنعتی، طبقه‌بندی روش‌ها و تحلیل مزایا/معایب	جمع‌بندی گزارش‌های نرخ موفقیت، میزان کاهش اصلاح کدها به‌صورت دستی، هزینه محاسباتی، تعداد شکست‌های برطرف شده.	جمیع نتایج مطالعات تجربی، مخازن کدهای متن-باز	توافق در متریک‌ها و مقادیر اندازه‌گیری کم است؛ فقدان استاندارد آزمایشی و قابل‌تکرار برای ارزیابی جامع دیده می‌شود.
۹	(Coppola et al., 2025)	بررسی الگوریتم‌های امتیازدهی برای انتخاب مناسب‌ترین کاندید	استخراج ویژگی‌های تگ و مقادیر تگ و آموزش مدل‌های رتبه‌بندی برای بهینه‌کردن k انتخاب برتر.	دقت پیشنهاد	برنامه‌های توسعه یافت برای similo	یادگیری بر اساس رتبه نیاز به داده‌های برچسب‌خورده زیاد دارد و مناسب‌سازی کد برای همه سایت‌ها هنوز چالش است؛ همچنین تأخیر در

ردیف	نویسندگان و سال	هدف	روش	متریک‌ها و نتایج	داده‌ها	چالش/شکاف
		از لیست کاندیدها (مثل Similo)				محیط عملیاتی مسئله‌ساز است
۱۰	(Healenium, 2025)	ایجاد لایه میانی که شکست اسکریپ های سلنیوم را کشف و به طور خودکار تلاش به ترمیم/جایگزینی آنها کند	اسکریپت‌های سلنیوم اجرا می‌شود و اگر خطایی وجود نداشت، کدهای مربوط به المان هدف در دیتابیس ذخیره می شوند تا اگر بازیابی های بعدی با خطا مواجه شد از آنها استفاده گردد	نرخ شکست از ۳۰ درصد به ۵ درصد کاهش یافته	نمونه‌های صنعتی، دموها و مخازن نمونه شرکت؛ پیاده‌سازی عملی در پروژه‌های شرکت	عملکرد عملی دارد اما نیاز به تنظیم دستی، مانیتورینگ، دارد و اغلب روی سناریوهای خاص تنظیم می‌شود؛ نیاز به ارزیابی گسترده‌تر دارد
۱۱	(Kluge & Stocco, 2025)	بازتولید و مقایسه الگوریتم‌های موجود (ازجمله Similo و...) در سایت‌های بزرگ‌تر و تکرار الگوریتم‌ها	پیاده‌سازی و ارزیابی الگوریتم‌های موجود	...	سایت‌های بزرگ با هزاران المان هدف	نشان می‌دهد هیچ راه‌حل واحدی برای همه موارد وجود ندارد و نیاز به روش‌های ترکیبی و معیارهای اعتبارسنجی خودکار است

در ادامه به ارتباط بین هر یک از پژوهش‌های موجود در جدول ۱ و مسئله موردبررسی در این مقاله و شکافی که مقاله حاضر در تلاش برای برطرف نمودن آن است، پرداخته شده. مقاله‌های کیرینوکی و همکاران[1] (۲۰۱۹)، نس و همکاران[2] (۲۰۲۳) و کوپلا و همکاران[3] (۲۰۲۵) با استفاده از سرنخ‌های مختلف، انتخابگرهای جایگزین را برای مقاوم‌سازی کدهای برداشت اطلاعات از بستر وب پیشنهاد می‌دهند. بااین‌حال، در این روش‌ها خطای پیشنهادهای ارائه شده، نسبتاً بالا است و برای صفحاتی با تغییرات ساختاری شدید و دارای تگ‌های با ویژگی‌های یکسان، مناسب نیست. همچنین باوجود اینکه استفاده از یک مدل زبانی بزرگ توسط نس و همکاران (۲۰۲۴) به‌منظور بهبود این روش‌ها ارائه شده که گاهی خطا را کاهش می‌دهد، اما هم مسائل حریم خصوصی ایجاد می‌کند و هم هزینه محاسباتی را افزایش می‌دهد. در مقابل، مقاله حاضر در تلاش است تا روشی ارائه دهد که نه‌تنها خطا را کاهش می‌دهد، بلکه برای صفحاتی با تغییرات ساختاری شدید نیز مناسب باشد و مسائل حریم خصوصی و هزینه محاسباتی نیز ایجاد نکند. مقاله‌های دگکی و همکاران[4] (۲۰۲۲)، خلیق و همکاران[5] (۲۰۲۳) و دانشور و ونگ[6] (۲۰۲۴) استفاده از بینایی ماشین را برای پیش‌بینی المان‌های موجود در صفحات

1. Kirinuki et al.
2. Nass et al.
3. Coppola et al.
4. Degaki et al.
5. Khaliq et al.
6. Daneshvar & Wang

وب پیشنهاد داده‌اند. بااین‌حال، پردازش تصویر و بینایی ماشین، هنگامی‌که اطلاعات موجود در صفحه بسیار ریز باشند، دقت بسیار پایینی در تشخیص و پیش‌بینی دارند و همچنین هزینه محاسباتی و زمانی این روش‌ها بسیار بالا است. در مقابل مقاله حاضر، روشی ارائه داده که هم بتواند مقدار دقیق اطلاعات را بدون هیچ خطایی بازگرداند و هم هزینه محاسباتی و زمانی آن حداقل باشد. مقاله تقی‌زاده‌کورایم و همکاران (۱۴۰۳) به ارائه سازوکاری برای مدیریت زمان و افزایش دقت اطلاعات برداشت‌شده از بستر وب با کمک کتابخانه سلنیوم پرداخته است. بااین‌حال، در این مقاله فقط محل خطا مشخص می‌گردد و مقادیر دارای خطا با مقدار تهی بازمی‌گردند، همچنین در این مقاله، راه‌حلی برای مقاوم‌سازی کدها ارائه نشده است. در مقابل، مقاله حاضر ضمن استفاده از روش پیشنهادشده توسط تقی‌زاده‌کورایم و همکاران (۱۴۰۳) به‌عنوان کدهای پایه برای برداشت اطلاعات از بستر وب، راه‌حلی برای مقاوم‌سازی کدها و برداشت خودکار مقادیر موردنظر از بستر وب با بالاترین دقت ممکن ارائه می‌دهد. ساراسی و همکاران[1] (۲۰۲۴) به‌مرور نظام‌مند مقالات موجود در زمینه پایداری کدهای کتابخانه سلنیوم پرداخته‌اند، اما مدلی برای مقاوم‌سازی کدهای برداشت اطلاعات از بستر وب ارائه ننموده‌اند. در مقابل، مقاله حاضر ضمن بررسی مقالات و مدل‌های موجود، مدلی را برای مقاوم‌سازی کدهای کتابخانه سلنیوم ارائه کرده است. هیلنیوم[2] (۲۰۲۵)، مدلی ارائه داده که در آن ابتدا اسکریپت‌های سلنیوم اجرا می‌شوند و اگر خطایی وجود نداشت، کدهای مربوط به المان هدف در دیتابیس ذخیره می‌شوند تا اگر بازیابی‌های بعدی با خطا مواجه شد از آنها استفاده گردد. بااین‌حال، اگر اسکریپت‌ها با خطا مواجه شوند، نیاز به تنظیمات دستی است. در مقابل مقاله حاضر در تلاش است تا مدلی ارائه دهد که در صورت بروز خطا، اسکریپت جایگزین با کمک الگوریتم‌های هوش مصنوعی در کمترین زمان، شناسایی شوند. کلوگه و استوکو[3] (۲۰۲۵) به مقایسه و بازتولید الگوریتم‌های موجود پرداخته و عنوان نموده‌اند که برای مقاوم‌سازی کدهای برداشت اطلاعات از بستر وب، هیچ راه‌حل واحدی وجود ندارد و نیاز به روش‌های ترکیبی و معیارهای اعتبارسنجی خودکار است. بااین‌حال، این مقاله هیچ مدلی ارائه نداده و تنها به ترکیب، مقایسه و بررسی مدل‌های موجود پرداخته است. در مقابل مقاله حاضر، مدلی ارائه نموده که بتواند کاستی‌های مدل‌های پیشین را کاهش دهد.

روش

این پژوهش ازنظر هدف، کاربردی و ازنظر ماهیت و روش اجرا، ترکیبی از روش‌های تحلیل داده‌های وب، توسعه الگوریتم‌های مقاوم‌سازی، و تحلیل کمی-تجربی است. باتوجه‌به ماهیت داده‌ها، مطالعه حاضر مبتنی بر جمع‌آوری داده‌های واقعی از سایت دیجی‌کالا و استخراج اطلاعات محصولات آن از طریق وب‌اسکرپینگ انجام شده است. سایت دیجی‌کالا به این دلیل انتخاب گردید که شامل داده‌ها و مشتریان واقعی است و باتوجه‌به مشتریان و محصولات زیاد، ترافیک و سرعت آن برای یک مطالعه پژوهشی واقعی است. همچنین سایت دیجی‌کالا، یکی از سایت‌هایی است که برنامه آن همواره در حال به‌روزرسانی و بهبود است، درنتیجه ساختار صفحات وب آن در طول زمان به دلایل مختلف تغییر می‌کند، درنتیجه می‌تواند برای مقاله حاضر، نمونه مطالعاتی مناسبی باشد. همچنین رویکرد پژوهش، استقرایی–تحلیلی بوده و تلاش دارد از داده‌های مشاهده‌شده، الگوها و شاخص‌های عملکردی در برداشت داده‌ها و پایداری اسکریپت‌ها استخراج و تبیین کند.

1. Saarathy et al.
2. Healenium
3. Kluge & Stocco

جامعه آماری شامل تمامی محصولات موجود در وب‌سایت دیجی‌کالا (دسته‌بندی داروهای مکمل) است که اطلاعات آن‌ها در بازه زمانی مهر ۱۴۰۲ تا مهر ۱۴۰۴ در دسترس بوده است. داده‌ها ازنظر حجم و تنوع، توانایی ارزیابی پایداری اسکریپت‌های سلنیوم و عملکرد الگوریتم‌های انتخابگر پویا را دارند. برداشت داده‌ها به‌صورت هفته‌ای یک بار و درمجموع ۱۰۰ برداشت کامل انجام شده است.

روایی محتوایی: شاخص‌های اولیه شامل اطلاعات ساختاری و محتوایی صفحات محصول (عنوان، قیمت، ویژگی‌ها، کد کالا و سایر داده‌های متنی) استخراج شدند و با تحلیل مفهومی و دسته‌بندی منطقی بررسی گردیدند تا تنها متغیرهای مرتبط با پایداری اسکریپت‌ها و موفقیت برداشت داده در مدل باقی بمانند.

روایی سازه: ساختار نهایی شاخص‌ها با استفاده از الگوریتم‌های هوش مصنوعی و یادگیری ماشین شامل سری‌های و تحلیل ویژگی‌ها، اعتبارسنجی شد و با چارچوب نظری سیستم‌های خودترمیمی و مقاوم‌سازی اسکریپت‌های سلنیوم هم‌راستا است. ازجمله الگوریتم‌های سری زمانی استفاده شده شامل الگوریتم‌های LSTM، ANN، SVR و XGBoost بودند که برای ارزیابی دقت آنها از متریک ضریب تعیین R^2 استفاده شد. ورودی الگوریتم از جدول تگ‌های HTML با ستون‌های تاریخ برداشت اطلاعات، مسیر اختصاصی المان و ستون «استفاده شد» است و خروجی الگوریتم پیش‌بینی مقدار ستون «استفاده شد» برای سطرهایی است که مقادیر ستون تاریخ برداشت اطلاعات آنها با تاریخ روز یکسان باشد. در پایان هر برداشت نیز مقدار ستون «استفاده شد» در جدول تگ‌های HTML به‌روز می‌گردد و هر بار تقریباً ۲۰ هزار سطر به جدول اضافه می‌شود تا برای برداشت‌های بعدی و افزایش دقت الگوریتم‌های سری زمانی، مورداستفاده قرار گیرد.

پایایی داده‌ها: فرایند استخراج داده‌ها توسط اسکریپت‌های پایتون و کتابخانه سلنیوم چندین بار اجرا شد تا از ثبات و دقت نتایج اطمینان حاصل گردد و خطاهای احتمالی وب‌اسکرپینگ به حداقل برسد.

پایایی تحلیل‌ها: مدل‌ها و الگوریتم‌های هوش مصنوعی برای تولید و انتخاب انتخابگرهای پویا با استفاده از معیارهای کمی شامل درصد موفقیت، نرخ بازیابی خودکار خطا، و تعداد برداشت موفق ارزیابی شدند تا نتایج در اجرای مجدد از ثبات کافی برخوردار باشند.

شبه‌کد روش پیشنهادی: در ادامه مراحل مختلف روش پیشنهادی به‌منظور ارائه نمای کلی از روش پیشنهادی ارائه شده است:

۱. ایجاد جداول موردنیاز:
   i. جدول اطلاعات محصول
   ii. جدول تگ‌های HTML
   iii. جدول شمارش تگ‌ها، جهت تشخیص انتخابگرهای محتمل
۲. مراحل اجرای اسکریپت‌های پایتون
   i. اسکریپت برداشت اطلاعات از سایت دیجی‌کالا
   ii. در صورت عدم بروز خطا و برداشت کامل اطلاعات، شامل اطلاعات محصولات و تگ‌ها آنها در جداول مربوطه ذخیره می‌شوند و در صورت بروز خطا، اسکریپت‌های مراحل بعد اجرا می‌گردند.
   iii. استخراج تمام تگ‌ها و ساختارهای HTML محصولات و پاک‌سازی آن‌ها

iv. استخراج Xpathها

v. اجرای اسکریپت‌های مربوط به هوش مصنوعی به‌منظور پیش‌بینی و تعیین Xpathهای صحیح و پس از آن اجرای مجدد اسکریپت برداشت اطلاعات از بستر وب (به‌عبارت‌دیگر، رفتن به مرحله i از اجرای اسکریپت‌های پایتون)

جمع‌بندی: روش‌شناسی حاضر به‌گونه‌ای طراحی شده است که پایداری و قابلیت اعتماد اسکریپت‌های سلنیوم در برداشت داده‌های واقعی از وب بررسی و تقویت شود و امکان تحلیل دقیق اثر الگوریتم‌های انتخابگر پویا و هوش مصنوعی بر موفقیت استخراج داده‌ها فراهم گردد.

این پژوهش با هدف افزایش پایداری و دقت کدهای برداشت اطلاعات از وب‌سایت دیجی‌کالا، چارچوبی ترکیبی شامل پایگاه داده SQL، اسکریپت‌های سلنیوم و پایتون و الگوریتم‌های هوش مصنوعی طراحی کرد که هرکدام از این‌ها به‌تفصیل در ادامه آمده‌اند. این رویکرد یک سامانه خودکار و مقاوم برای برداشت داده از وب ارائه می‌دهد که تغییرات ساختار صفحات را مدیریت کرده و نیاز به اصلاح دستی را کاهش می‌دهد.

۱. طراحی پایگاه داده

در گام نخست به‌منظور ذخیره‌سازی اطلاعات محصولات سایت دیجی‌کالا و همچنین داده‌های مربوط به تگ‌های HTML و انتخابگرها، چند جدول در پایگاه داده sql ایجاد شد. این جداول شامل:

۱-۱. جدول اطلاعات محصولات: این جدول مشخصاتی از قبیل «نام محصول، قیمت اصلی، درصد تخفیف، قیمت پس از تخفیف، امتیاز کاربران، لینک تصویر، برچسب محصول و تاریخ روز» را برای هر محصول در بازه‌های زمانی هفتگی ذخیره می‌کند.

۱-۲. جدول تگ‌های HTML دیجی کالا: این جدول تمام تگ‌های موجود در صفحه HTML ۲۰ محصول اول دیجی کالا را به‌صورت ساختارمند و والد–فرزندی ذخیره می‌کند. هر سطر شامل نام تگ، xpath یا مسیر اختصاصی تگ (به‌صورت شماره‌گذاری‌شده در صورت تکرار چندباره مثل [2] a/ [3] div]، تاریخ روز یا همان تاریخ برداشت اطلاعات و ستون «استفاده شد» که با کلمه‌های «بله» و «خیر» پر می‌شود، است. نحوه برداشت کل تگ‌ها و محاسبه مسیر تگ‌ها یا همان xpath آن‌ها در بخش اسکریپت‌های سلنیوم و پایتون به‌تفصیل و با ارائه کد نوشته شده است. درصورتی‌که اسکریپت سلنیوم برداشت اطلاعات از وب بدون خطا و با اطلاعات کامل، اجرا شود؛ پس از اتمام اجرا فقط یک xpath برای هر یک از مقادیر نام محصول، قیمت اصلی، درصد تخفیف، قیمت پس از تخفیف، امتیاز کاربران، لینک تصویر، برچسب محصول به جدول افزوده‌شده و ستون «استفاده شد» مقدار بله را می‌گیرد؛ اما درصورتی‌که اسکریپت سلنیوم با خطا و مشکل روبرو شوند، تمام تگ‌های html دیجی کالا به جدول اضافه‌شده و ستون «استفاده شد» با مقدار پیش‌فرض تهی پر می‌شود و پس اجرای تمام اسکریپت‌های بعدی و مشخص شدن مقدار درست xpath برای هرکدام از المان‌های صفحه نظیر نام محصول و ...، این ستون با مقدار بله که به معنی xpath موجود و درست برای المان‌های موردنظر نظیر نام محصول و ... پر می‌شود و بقیه سطرها مقدار خیر را می‌گیرند.

۱-۳. جدول شمارش تگ‌ها جهت تشخیص انتخابگرهای محتمل: بعد از استخراج کامل HTML و تکمیل جدول تگ‌های HTML دیجی کالا، این جدول ساخته می‌شود. این جدول شامل ستون‌های «مسیر اختصاصی تگ بدون تکرار، فراوانی

ستون مسیر اختصاصی تگ از جدول تگ‌های HTML دیجی کالا برای هر مسیر اختصاصی و ستون تاریخ ثبت مسیر اختصاصی» است. برای تکمیل این جدول، تگ‌های مشابه یا همان مسیرهای اختصاصی مشابه، شمارش شده است. به این دلیل این جدول تهیه می‌شود که اکثر سایت‌های فروشگاهی تعداد محدودی محصول را در یک صفحه نشان می‌دهند که این عدد ثابت است. به‌عنوان‌مثال، سایت دیجی کالا در هر صفحه از صفحات محصول خود تنها ۲۰ محصول را نشان می‌دهد، مثلاً در یک صفحه ما ۲۰ عنوان کالا، ۲۰ قیمت، ۲۰ تصویر و ... داریم. باتوجه‌به ثابت بودن این عدد، وجود آن در این جدول به الگوریتم‌های هوش مصنوعی و اسکریپت‌های برداشت اطلاعات می‌تواند کمک قابل‌توجهی کند. درنتیجه انتخابگرهایی تعداد ۲۰ یا کمتر دارند، احتمال بیشتری دارد که بتوانند اطلاعات موردنظر ما را برداشت کنند. یک نکته مهم اینکه، این جدول درواقع یک view ساخته‌شده در دیتابیس sql است که از روی جدول تگ‌های HTML دیجی کالا ساخته شده است و به دلیل راحتی در جمله‌بندی و بیان صریح‌تر از کلمه جدول استفاده شده است. ساخت view باعث افزایش سرعت بسیار خوبی در اجرای کد شده و پیچیدگی‌های ساختاری برنامه را به‌شدت کاهش می‌دهد و به‌واسطه آن نیازی به اسکریپت نویسی و اتصال فرانت‌اند و بک‌اند و اتصال اسکریپت به بک‌اند برای ثبت اطلاعات در دیتابیس وجود ندارد و جدول از طریق view با کسری از ثانیه تولید شده و به‌راحتی می‌توان از آن در اسکریپت‌های مختلف استفاده نمود.

۲. اسکریپت‌های پایتون و سلنیوم

۲-۱. اسکریپت برداشت اطلاعات محصول از سایت دیجی کالا: این بخش بر اساس کدهای موجود در پژوهش انجام شده توسط تقی زاده و همکاران (۲۰۲۴) نوشته شد تا با کمک آن بتوان مقادیر بدون خطا را دقیق در جای مناسب خود دریافت نمود و همچنین محل خطا را نیز شناسایی نمود. این کار باعث سرعت بخشیدن به برنامه می‌شود، چراکه مقادیر فقط دارای خطا از طریق اسکریپت‌های بعدی شناسایی و اصلاح می‌شوند. البته کدهای نوشته شده توسط تقی زاده و همکاران (۲۰۲۴) برای مدیریت زمان و نامتقارن بودن و تأخیر استفاده شده‌اند و در این پژوهش برای خطای ایجاد شده در تغییرات ساختار html صفحه استفاده نموده‌ایم. این تفاوت تنها در ارائه خروجی‌ها قابل‌نمایش است، چراکه در خروجی موضوع مطرح‌شده توسط تقی زاده و همکاران (۲۰۲۴) تعدادی از اطلاعات موجود در یک ستون مثلاً قیمت خالی نمایش داده می‌شود ولی در خروجی این پژوهش باتوجه‌به اینکه ساختار html صفحه تغییر یافته است، کل ستون مربوط به قیمت خالی نمایش داده می‌شود که این ستون توسط اسکریپت‌های بعدی اصلاح و پر خواهد شد.

۲-۲. استخراج تمام تگ‌ها و ساختار HTML صفحه اول محصولات دیجی کالا (یا به‌عبارت‌دیگر ۲۰ محصول اول): باتوجه‌به تحقیقات گذشته و مطالعات انجام شده و همچنین اصول طراحی وب‌سایت با استفاده از جاوا اسکریپت، ری اکت و سایر زبان‌های برنامه‌نویسی، ساختار تمام محصولات در یک سایت فروشگاهی مشابه هم هستند، اما ممکن است برخی محصولات تگ‌های بیشتر یا کمتری داشته باشند؛ به‌عبارت‌دیگر، ساختار والد فرزندی قیمت محصول در تمام محصولات یکسان است و تنها تفاوت در یکی از تگ‌هاست که آن هم مربوط به والد اصلی هر محصول است که هر محصول از طریق آن با محصول دیگر متفاوت می‌شود. به همین دلیل در این بخش، تنها تگ‌ها و ساختار HTML فقط صفحه اول محصولات برداشت می‌شود. در ادامه اسکریپت مربوط به این بخش آمده است (شکل ۱). این اسکریپت پس از جمع‌آوری تمام تگ‌ها و ساختارهای HTML صفحه اول آن را در متغیری به نام html_content می‌ریزد که با استفاده از این متغیر و دستورات regex که در بخش بعد آمده است، می‌توان XPATH یا همان مسیر اختصاصی المان‌های صفحه یا همان اشیاء صفحه را استخراج نمود (شکل ۱).

شکل ۱. اسکریپت مربوط به استخراج تمام تگ‌ها و ساختار HTML صفحه اول محصولات دیجی کالا (یا به‌عبارت‌دیگر ۲۰ محصول اول)

```
from selenium import webdriver
from webdriver_manager.firefox import GeckoDriverManager
from selenium.webdriver.firefox.service import Service
from selenium.webdriver.firefox.options import Options
from bs4 import BeautifulSoup
import time
URL = "https://www.digikala.com/search/category-nutritional-supplements/?page=1"
def fetch_full_html_with_divs(url):
    options = Options()
    options.add_argument("--headless")
    options.add_argument("--disable-gpu")
    service = Service(GeckoDriverManager().install())
    driver = webdriver.Firefox(service=service, options=options)
    try:
        driver.get(url)
        time.sleep(5)
        last_height = driver.execute_script("return document.body.scrollHeight")
        while True:
            driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
            time.sleep(5)
            new_height = driver.execute_script("return document.body.scrollHeight")
            if new_height == last_height:
                break
            last_height = new_height
        html = driver.page_source
        soup = BeautifulSoup(html, "html.parser")
        div_count = len(soup.find_all("div"))
        print(f"تعداد کل divها در صفحه: {div_count}")
        return html
    finally:
        driver.quit()
html_content = fetch_full_html_with_divs(URL)
```

۲-۳. اسکریپت استخراج Xpath تمامی تگ‌ها: در این مرحله تگ‌ها به ترتیب والد–فرزند با یک الگوریتم پیمایشی یا همان رجکس استخراج می‌شوند. در ادامه کد مربوط به این اطلاعات و آمده است. این کد لیستی از مسیرهای اختصاصی هر تگ به نام paths را بازمی‌گرداند. در این روش از متد رجکس که با re مشخص‌شده و همچنین حلقه‌های while و for تودرتو استفاده شده است و ساختار html را به مسیر اختصاصی هر تگ بر اساس والد و فرزندی آن تبدیل می‌کند. البته قبل از اجرای این اسکریپت، اسکریپت‌های مربوط به پاک‌سازی ساختار انجام شد و بخش‌هایی مانند هدر و فوتر و برخی بخش‌های ثابت از ساختار html حذف شدند تا متغیر html_content حجم سبک‌تری از داده را در خود نگهداری نماید (شکل ۲).

شکل ۲. اسکریپت استخراج Xpath تمامی تگ‌ها

```
import re
from collections import defaultdict
tag_pattern = re.compile(r'<(/?)(\w+)[^>]*>')

paths = []  # خروجی نهایی مسیرها
stack = []  # مسیر فعلی
child_counters_stack = []  # شمارنده فرزندان هر سطح

pos = 0
while pos < len(html_content):
    match = tag_pattern.search(html_content, pos)
    if not match:
        break
    is_closing, tag_name = match.groups()
    pos = match.end()

    if not is_closing:  # تگ باز
        stack.append(tag_name)
        child_counters_stack.append(defaultdict(int))
    else:  # تگ بسته
        if stack:
            # شمارنده فرزند مشابه در والد
            parent_counter = child_counters_stack[-1]
            parent_counter[tag_name] += 1
            index = '' if parent_counter[tag_name] == 1 else f'[{parent_counter[tag_name]}]'

            # ساخت مسیر کامل از والد تا فرزند
            full_path_parts = []
            for i, t in enumerate(stack):
                if i < len(stack)-1:
                    # بررسی تعداد فرزندان مشابه در سطح والد
                    cnt = child_counters_stack[i][t]
                    part = t if cnt <= 1 else f"{t}[{cnt}]"
                else:
                    # خود تگ بسته شده
                    part = t + index
                full_path_parts.append(part)
            full_path = '/'.join(full_path_parts)
            paths.append(full_path)
            stack.pop()
            child_counters_stack.pop()
```

۲-٤. ذخیره‌سازی و تکمیل اطلاعات: اسکریپت بخش‌های ۳-۲ و ۲-۲، تنها زمانی اجرا می‌شوند که اسکریپت بخش ۱-۲ مقادیری خالی از محصولات صفحه را بازگرداند یا دچار خطا شود. پس از اجرایی کامل اسکریپت بخش‌های ۳-۲ و ۲-۲، جدولِ تگ‌های HTML دیجی کالا بر اساس لیست paths پر می‌شود و سپس جدولِ شمارش تگ‌ها جهت تشخیص انتخابگرهای محتمل بر اساس جدولِ تگ‌های HTML دیجی کالا تکمیل می‌گردد.

۳. اسکریپت‌های مربوط به یادگیری ماشینی و هوش مصنوعی برای انتخاب انتخابگرها: برای انتخاب انتخابگر یا همان xpath مناسب، ابتدا مقادیر مربوط به ستون مسیر اختصاصی از جدولِ تگ‌های HTML دیجی کالا با فیلتر تاریخ روز از پایگاه داده برداشت‌شده و مقادیر مربوط به ستون «استفاده شد» آن بر اساس داده‌های تاریخی همان جدول با کمک الگوریتم‌های سری زمانی پیش‌بینی می‌گردد و xpath هایی که مقدار «استفاده شد» در آنها بله باشد، جایگزین xpath موجود در اسکریپت سلنیوم می‌شوند و درنهایت کد سلنیوم مجدد اجرا می‌گردد و درصورتی‌که کد بدون خطا اجرا گردید، مقادیر «استفاده شد» در جدولِ تگ‌های HTML دیجی کالا اصلاح می‌گردد و در صورت بروز خطا تمام مقادیری که ستون «استفاده شد» آن‌ها بله بود، حذف‌شده و مجدد الگوریتم‌های پیش‌بینی سری‌های زمانی فراخوانی می‌شوند. همچنین درصورتی‌که برخی المان‌ها

مثلاً فقط المان «درصد تخفیف» خطا داشت، xpath هایی که استفاده شده‌اند حذف می‌شوند و xpath هایی که خطا نداشتند، مقادیر «استفاده شد» در جدول تگ‌های HTML دیجی کالا آنها بله می‌شود و الگوریتم تا زمان یافتن xpath درست ادامه پیدا می‌کند. اگر برنامه نتواند با کمک سری‌های زمانی xpath درست را بیابد، آنگاه تگ‌های موجود در جدول تگ‌ها را پیمایش می‌کند. در ادامه جدول مربوط به الگوریتم‌های هوش مصنوعی استفاده شده و معیارهای سنجش آنها آورده شده است. شایان‌ذکر است که در شش ماه اول از این پژوهش، اصلاح اسکریپت سلنیوم به‌صورت دستی و با نظارت مستقیم انجام گردید تا سطرهای این جدول برای آموزش مدل‌های هوش مصنوعی مناسب شود.

جدول ۲. الگوریتم‌های سری زمانی استفاده شده برای تحلیل و ارزیابی درستی پیش‌بینی xpath ها

الگوریتم‌ها	۱۴۰۴-۰۶-۲۶	۱۴۰۴-۰۵-۲۹	۱۴۰۴-۰۴-۲۵	۱۴۰۴-۰۳-۲۸
	(R^2 ضریب تعیین)			
LSTM	۱	۱	۹۹۹/۰	۱
ANN	۱	۱	۹۹۸/۰	۹۹۹/۰
SVR	۹۸۷/۰	۹۸۹/۰	۹۸۲/۰	۹۸۷/۰
XGBoost	۹۶۴/۰	۹۸۹/۰	۹۷۷/۰	۹۸۲/۰

٤. بررسی آنچه انتخابگرها ارائه داده‌اند: در این مرحله، انتخابگرها تمام المان‌های اصلی صفحه که برنامه در تلاش برای جمع‌آوری آن بود را جمع‌آوری نموده‌اند، اما سؤال اینجاست که از کجا انتخابگرها همان المان موردنظر را بازمی‌گردانند و به‌عنوان‌مثال مقدار یک فیلتر در صفحه یا یک مقدار کاملاً غیر مرتبط را بازنمی‌گردانند؟. برای جلوگیری از چنین مشکلاتی چندین راهکار متفاوت در طول برنامه مورداستفاده قرار گرفته است. یکی اینکه تعداد xpath مشابه برای نام محصول، قیمت و تصویر باید حتماً ۲۰ عدد باشد و برای درصد تخفیف و برچسب محصول کمتر از ۲۰ یا ۲۰ باشد. دوم نام محصول یک مقدار رشته‌ای است و طول آن نهایتاً ۲۵۰ کاراکتر است، درصد تخفیف مقداری عددی است که در کنار آن علامت % قرار دارد، تصویر محصول فایلی از نوع تصویر است و برچسب محصول هم رشته‌ای است شامل مقادیر ثابت نظیر فروش شگفت‌انگیز، فروش ویژه و بلک فرایدی. با ترکیب این ۲ شرط در طول برنامه و در بخش‌های مختلف نظیر جداول و اسکریپت‌ها، کدهایی برای آنها نوشته شده است که در صورت وجود مشکل، برنامه از آن مطلع و انتخابگرها را اصلاح می‌نماید.

یافته‌ها

طبق بررسی‌های انجام شده، استفاده از روش پیشنهادی باعث ایجاد پایداری و مقاومت در کدهای سلنیوم برای برداشت اطلاعات از بستر وب گردید. طرح پیشنهادی از تاریخ ۵ مهر ۱۴۰۲ به‌صورت هفته‌ای یک‌بار شروع به جمع‌آوری تگ‌های موجود در ساختار html صفحه اول محصولات در دیجی کالا نمود و شش ماه اول هر هفته تقریباً ۲۰ هزار xpath استخراج‌شده از سایت دیجی کالا را به جدول تگ‌ها اضافه نمود و پس‌ازآن هفته‌هایی که هیچ خطایی وجود نداشت تنها ۷ xpath اصلی به جدول افزوده شدند. درنتیجه در تاریخ ۱۴۰۴-۰۶-۲۶ که آخرین برداشت صورت گرفت، جدول تگ‌ها تقریباً دارای یک‌میلیون سطر داده بود. همچنین دقت اجرای الگوریتم‌های سری زمانی از طریق ضریب تعیین در ۴ تاریخ

مختلف اندازه‌گیری شد که الگوریتم lstm در بسیاری از مواقع دقت ۱۰۰ درصدی داشته است. همچنین مدت‌زمان اجرای برنامه و وضعیت خطاهای آن در صورت استفاده و عدم استفاده از روش پیشنهادی در جدول ۳ آمده است که در بهترین حالت، یعنی زمانی که از اسکریپتی مشابه اسکریپت هفته قبل استفاده شد و خطا هم وجود نداشت، مدت‌زمان برداشت اطلاعات ۶۰ محصول از دیجی کالا ۱۸۰ ثانیه یا همان ۳ دقیقه طول کشیده است و در بدترین حالت که هم اسکریپت سلنیوم خطا داشته و هم الگوریتم هوش مصنوعی و برنامه مجبور به پیمایش کل جدول تگ‌ها شده، مدت‌زمان اجرای برنامه ۵۵۰ ثانیه یا همان ۹ دقیقه و میانگین مدت‌زمان اجرای کامل برنامه برای ۶۰ محصول در ۱۸ برداشت ۲۵۵/۲۲ ثانیه طول کشیده است که این مقدار به نسبت تکنیک‌های پردازش تصویر که ساعت‌ها اجرای آنها طول می‌کشد، بسیار بهتر است (جدول ۳). همچنین جدول ۴ گزارشی از تعداد اطلاعات برداشت‌شده به‌صورت خودکار از بستر وب و عدم اعمال تغییرات دستی در کد، هنگام استفاده و عدم استفاده از روش پیشنهادی و میانگین تعداد اطلاعاتی که با موفقیت برداشت شده‌اند را ارائه می‌دهد. بر اساس جدول ۴، در صورت عدم استفاده از روش پیشنهادی، تقریباً اطلاعات ۲۶ محصول از ۶۰ محصول به‌صورت کامل برداشت شده است؛ به‌عبارت‌دیگر، در صورت عدم استفاده از روش پیشنهادی در هنگام برداشت اطلاعات به‌صورت کاملاً خودکار، بیش از ۵۶ درصد از اطلاعات از بین می‌روند. درحالی‌که، هنگام استفاده از روش پیشنهادی ۱۰۰ درصد اطلاعات برداشت شده‌اند.

جدول ۳. مقایسه وجود خطا و عدم وجود خطا، قبل و بعد از اجرای روش پیشنهادی و مدت‌زمان اجرای کامل برنامه

ردیف	تاریخ	خطا در اسکریپت سلنیوم		مدت‌زمان اجرای کامل برنامه برای ۶۰ محصول (ثانیه)
		قبل از اجرای روش پیشنهادی	بعد از اجرای روش پیشنهادی	
۱	۳۱-۰۲-۱۴۰۴	خیر	خیر	۱۸۰
۲	۰۷-۰۳-۱۴۰۴	خیر	خیر	۱۸۰
۳	۱۴-۰۳-۱۴۰۴	بله	خیر	۲۸۰
۴	۲۱-۰۳-۱۴۰۴	بله	خیر	۳۱۰
۵	۲۸-۰۳-۱۴۰۴	بله	خیر	۲۱۰
۶	۰۴-۰۴-۱۴۰۴	بله	خیر	۲۱۰
۷	۱۱-۰۴-۱۴۰۴	خیر	خیر	۱۸۰
۸	۱۸-۰۴-۱۴۰۴	خیر	خیر	۱۸۰
۹	۲۵-۰۴-۱۴۰۴	بله	خیر	۳۴۴
۱۰	۰۱-۰۵-۱۴۰۴	خیر	خیر	۱۸۰
۱۱	۰۸-۰۵-۱۴۰۴	بله	خیر	۵۵۰
۱۲	۱۵-۰۵-۱۴۰۴	بله	خیر	۴۲۰
۱۳	۲۲-۰۵-۱۴۰۴	خیر	خیر	۱۸۰
۱۴	۲۹-۰۵-۱۴۰۴	بله	خیر	۲۰۰
۱۵	۰۵-۰۶-۱۴۰۴	خیر	خیر	۱۸۰
۱۶	۱۲-۰۶-۱۴۰۴	بله	خیر	۳۷۵
۱۷	۱۹-۰۶-۱۴۰۴	خیر	خیر	۱۸۰
۱۸	۲۶-۰۶-۱۴۰۴	بله	خیر	۲۵۵
	میانگین مدت‌زمان اجرای کامل برنامه برای ۶۰ محصول در ۱۸ برداشت:			۲۵۵/۲۲

جدول ٤. تعداد اطلاعات برداشت شده به‌صورت خودکار از بستر وب و عدم اعمال تغییرات دستی در کد، هنگام استفاده و عدم استفاده از روش پیشنهادی

ردیف	تاریخ	عدم استفاده از روش پیشنهادی				استفاده از روش پیشنهادی			
		نام محصول	قیمت	تخفیف	نوع تخفیف	نام محصول	قیمت	تخفیف	نوع تخفیف
۱	۳۱-۰۲-۱٤۰٤	٦۰	٦۰	۱۷	۱۷	٦۰	٦۰	۱۷	۱۷
۲	۰۷-۰۳-۱٤۰٤	٦۰	٦۰	۲۲	۲۲	٦۰	٦۰	۲۲	۲۲
۳	۱٤-۰۳-۱٤۰٤	۱۱	۱۱	٥	٤	٦۰	٦۰	۲۹	۲۹
٤	۲۱-۰۳-۱٤۰٤	۲	۲	۰	۰	٦۰	٦۰	۱۲	۱۲
٥	۲۸-۰۳-۱٤۰٤	٤	٤	۲	۲	٦۰	٦۰	۱٤	۱٤
٦	۰٤-۰٤-۱٤۰٤	۱۲	۱۱	٤	۰	٦۰	٦۰	۳۷	۳۷
۷	۱۱-۰٤-۱٤۰٤	٦۰	٦۰	۳۱	۳۱	٦۰	٦۰	۳۱	۳۱
۸	۱۸-۰٤-۱٤۰٤	٦۰	٦۰	۲۱	۲۱	٦۰	٦۰	۲۱	۲۱
۹	۲٥-۰٤-۱٤۰٤	۰	۰	۰	۰	٦۰	٦۰	۲٥	۲٥
۱۰	۰۱-۰٥-۱٤۰٤	٦۰	٦۰	۳۳	۳۳	٦۰	٦۰	۳۳	۳۳
۱۱	۰۸-۰٥-۱٤۰٤	۰	۰	۰	۰	٦۰	٦۰	۱۹	۱۹
۱۲	۱٥-۰٥-۱٤۰٤	٥	۰	۰	۰	٦۰	٦۰	۱۳	۱۳
۱۳	۲۲-۰٥-۱٤۰٤	۸	۸	٤	۲	٦۰	٦۰	۲۷	۲۷
۱٤	۲۹-۰٥-۱٤۰٤	٦	٦	٥	٥	٦۰	٦۰	۱٥	۱٥
۱٥	۰٥-۰٦-۱٤۰٤	٦۰	٦۰	٤۰	٤۰	٦۰	٦۰	٤۰	٤۰
۱٦	۱۲-۰٦-۱٤۰٤	۰	۰	۰	۰	٦۰	٦۰	۳۸	۳۸
۱۷	۱۹-۰٦-۱٤۰٤	٦۰	٦۰	۲۷	۲۷	٦۰	٦۰	۲۷	۲۷
۱۸	۲٦-۰٦-۱٤۰٤	۱	۰	۰	۰	٦۰	٦۰	۲۱	۲۱
میانگین تعداد اطلاعات برداشت شده		۲٦/۱۱	۲٥/۷۸	۱۲/٥٦	۱۲/۱۷	٦۰	٦۰	۲٥/۳۳	۲٥/۳۳

مزایای این پژوهش و استفاده از روش پیشنهادی باعث می‌شود که نیاز به تغییر دستی انتخابگرها بعد از هر تغییر ظاهری سایت از بین برود، انتخابگرهای پایدار به‌صورت خودکار استخراج شوند، با حذف تلاش انسانی، سرعت و پایداری سیستم جمع‌آوری داده افزایش یابد، از داده‌های تاریخی برای تصمیم‌گیری خودکار در استخراج اطلاعات استفاده شود.

این مطالعه به‌منظور طراحی، پیاده‌سازی و ارزیابی یک سامانه مقاوم برای برداشت خودکار اطلاعات از وب انجام شده است. در گام نخست، ساختار کلی سیستم پیشنهادی طراحی شد که شامل دو بخش اصلی است:

۱) ماژول تولید انتخابگرهای پویا،

۲) لایه تصمیم‌گیری مبتنی بر هوش مصنوعی برای انتخاب بهینه انتخابگر در زمان اجرا.

در مرحله اول، مجموعه‌ای از داده‌های نمونه از ساختار صفحات وب جمع‌آوری شد تا تغییرات محتمل در چینش و ویژگی‌های موجود مورد تحلیل قرار گیرد. برای این منظور، صفحات هدف در بازه‌های زمانی مختلف ذخیره و مقایسه

شدند تا الگوهای تغییر و رفتار عناصر قابل‌شناسایی باشند. خروجی این مرحله، مجموعه‌ای از ویژگی‌های قابل‌استناد برای طبقه‌بندی و انتخاب انتخابگرهای مناسب بود.

در مرحله دوم، الگوریتمی طراحی شد که برای هر عنصر هدف، چندین انتخابگر با اولویت‌های مختلف تولید می‌کند؛ این انتخابگرها در یک پایگاه داده ذخیره‌شده و برای تحلیل‌های بعدی در اختیار لایه تصمیم‌گیری قرار گرفتند.

در مرحله سوم، یک مدل یادگیری ماشین برای تصمیم‌گیری بین انتخابگرهای تولیدشده توسعه یافت. این مدل با استفاده از داده‌های ثبت‌شده از اجرای واقعی سیستم آموزش داده شد و توانست در صورت وقوع خطا، مناسب‌ترین انتخابگر جایگزین را بدون دخالت انسان انتخاب کند. در طراحی این مدل، معیارهایی همچون نرخ موفقیت انتخابگر، عمق جستجو، میزان وابستگی به ویژگی‌های تغییرپذیر و سرعت اجرا لحاظ شد.

در مرحله چهارم، سیستم پیشنهادی به‌صورت عملی بر روی وب‌سایت دیجی‌کالا پیاده‌سازی و به مدت دو سال، با تناوب هفتگی اجرا شد. در طول این مدت، داده‌های مربوط به تعداد شکست‌ها، نیاز به اصلاح دستی، زمان اجرای عملیات و میزان پایداری سیستم جمع‌آوری و ثبت گردید.

در مرحله نهایی، عملکرد روش پیشنهادی با روش استاندارد مبتنی بر انتخابگرهای ثابت مقایسه شد. معیارهای ارزیابی شامل نرخ شکست، نیاز به مداخله انسانی، هزینه نگه‌داری، پایداری در برابر تغییرات ساختار صفحه و زمان اجرای عملیات بود. نتایج نشان داد که روش پیشنهادی توانسته است تعداد شکست‌ها و نیاز به اصلاح دستی را به شکل معنی‌داری کاهش دهد و در محیطی پویا عملکرد قابل‌اتکاتری ارائه کند.

بحث و نتیجه‌گیری

هدف این پژوهش ارائه روشی بود که بدون نیاز به بازنویسی مداوم انتخابگرها پس از تغییر ساختار HTML سایت‌ها، بتوان فرایند بازیابی و استخراج داده و در راستای آن بازیابی دانش را پایدار و خودکار نمود. در حالت معمول، تغییر در ساختار صفحه (مثلاً تغییر کلاس CSS یا لایه‌بندی جدید) باعث شکست کامل اسکریپت استخراج داده می‌شود. رویکرد ارائه‌شده در این نوآوری، ۱. تحلیل کامل HTML صفحات هدف، ۲. استخراج همه تگ‌ها و مسیر کامل XPath با شماره‌گذاری ساختاری، ۳. استخراج الگوی تکرار و رفتار پایداری انتخابگرها و ۴. استفاده از الگوریتم تصمیم‌گیر هوش مصنوعی برای انتخاب انتخابگر مناسب توانست این مشکل را حل کند. از نتایج ثبت‌شده در جدول شمارش تگ‌ها مشخص شد که برخی انتخابگرها در طول زمان پایداری بیشتری دارند. این موضوع نشان داد که انتخابگرهای پایدار معمولاً وابسته به ساختار اصلی سایت و نه لایه‌بندی ظاهری هستند و می‌توانند مبنای تصمیم‌گیری الگوریتمی قرار گیرند.

نتایج حاصل از این پژوهش نشان می‌دهد که: انتخابگرهای پویا می‌توانند به‌صورت خودکار و بدون دخالت انسان انتخاب شوند، میزان شکست استخراج داده تا بیش از ۹۹٪ کاهش یافت، با استفاده از مدل‌های یادگیری، انتخابگرها بر اساس سری زمانی پایدارسازی شدند، همچنین سیستم طراحی شده قابلیت رشد دارد و با افزایش داده‌های تاریخی، دقت مدل هوش مصنوعی بهتر می‌شود. این روش می‌تواند به‌عنوان راهکاری عمومی برای وب‌اسکرپینگ مقاوم در برابر تغییرات سایت‌ها به کار رود. همچنین مدل داده‌محور استفاده‌شده، سیستم را به یک خزنده نیمه‌خودمختار تبدیل می‌کند که بدون بازنویسی کد قادر به ادامه فعالیت است.

درنتیجه، با استفاده از روش پیشنهادی می‌توان داده‌های موردنیاز را به‌صورت خودکار، در هر زمان و با دقت بالا، بازیابی نمود. در این روش، خزنده‌های وب به‌عنوان برنامه‌های خودکار، ضمن برطرف نمودن نیاز سازمان‌ها و پژوهشگران به بازیابی و تحلیل مستمر داده‌های موجود در بستر وب، می‌توانند به بازیابی دانش و اطلاعات موجود در این داده‌ها و ایجاد نظام‌های معنایی ارزشمند و واقعی و ساخت سامانه‌هایی نظیر بات‌های هوشمند بازیابی و ارائه دانش مانند چت جی پی تی و سیستم‌های مختلفی ازجمله سیستم‌های تست نرم‌افزاری تحت وب و گزارش خطا، تحلیل کسب‌وکار، پایش قیمت، رصد رفتار مشتریان، جمع‌آوری اطلاعات رقبا، پایش بازارهای مالی و بسیاری از خدمات مبتنی بر داده کمک شایانی کنند.

پیشنهادهای آتی

باتوجه‌به این‌که این برنامه برای یک وب‌سایت فروشگاهی نوشته شده، درنتیجه برای سایت‌های دارای اشیا با ویژگی‌های متفاوت (به‌عنوان‌مثال در اینجا شیء محصول و ویژگی‌های نام، قیمت، تصویر، تخفیف و برچسب در آن) مناسب است؛ به‌عبارت‌دیگر، این برنامه برای سایت‌هایی نظیر دکتردکتر، آمازون یا ترب که ویژگی‌های مشابهی دارند به‌خوبی کار می کند؛ ولی برای سایت‌هایی مانند python.org یا برنامه‌های کاربردی تحت وب که ساختار متفاوتی دارند، بررسی نشد است. برای پژوهش‌های آتی می‌توان سایت‌هایی که ساختارهای متفاوتی دارند را بر اساس راهکار پیشنهادی این پژوهش مورد تحلیل و بررسی قرار داد.

ملاحظات اخلاقی

پیروی از اصول اخلاق‌پژوهش

نویسندگان اصول اخلاقی را در انجام و انتشار این پژوهش علمی رعایت نموده‌اند و این موضوع موردتأیید همه آن‌هاست.

تعارض منافع

بنا بر اظهار نویسندگان این مقاله تعارض منافع ندارد.

حامی مالی

مقاله حاضر بدون حمایت مالی انجام شد.

سپاسگزاری

نویسندگان مراتب قدردانی خود را از گروه مدیریتی مجله بازیابی دانش و نظام‌های معنایی اعلام می‌دارند. از داوران محترم به خاطر ارائه نظرهای ساختاری و علمی سپاسگزاری می‌شود.

ORCID

Farnaz Taghizadeh kourayem http://orcid.org/0000-0003-2676-2495
Mohammadreza Kabaranzad Ghadim http://orcid.org/0000-0002-4372-5226
Seyed Abdollah Amin Mousavi http://orcid.org/0009-0005-3052-5910

منابع

تقی‌زاده‌کورایم، فرناز، کاباران‌زادقدیم، محمدرضا و موسوی، سید عبداله امین. (۱٤۰۳). سازوکاری برای مدیریت زمان و افزایش دقت اطلاعات هنگام استفاده از کتابخانه سلنیوم. *بازیابی دانش و نظام‌های معنایی*، *٤۱*(۱۱)، ۲۰٤–۲۲۵. https://doi.org/10.22054/jks.2024.80235.1660

References

Coppola, R., Feldt, R., Nass, M., & Alégroth, E. (2025). Ranking approaches for similarity-based web element location. *Journal of Systems and Software*, *222*, Article 112286. https://doi.org/10.1016/j.jss.2024.112286

Degaki, R. H., Colonna, J. G., Lopez, Y., Carvalho, J. R., & Silva, E. (2022, November). Real time detection of mobile graphical user interface elements using convolutional neural networks. In *Proceedings of the Brazilian Symposium on Multimedia and the Web* (pp. 159–167). ACM. https://doi.org/10.1145/3539637.3558044

Healenium. (2025). *Healenium* [Open-source project]. GitHub. https://github.com/healenium

Khaliq, Z., Khan, D. A., & Farooq, S. U. (2023). Using deep learning for selenium web UI functional tests: A case-study with e-commerce applications. *Engineering Applications of Artificial Intelligence*, *117*, Article 105446. https://doi.org/10.1016/j.engappai.2022.105446

Kirinuki, H., Tanno, H., & Natsukawa, K. (2019, February). COLOR: Correct locator recommender for broken test scripts using various clues in web application. In *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)* (pp. 310–320). IEEE. https://doi.org/10.1109/SANER.2019.8667976

Kluge, A., & Stocco, A. (2025). *Web element relocalization in evolving web applications: A comparative analysis and extension study*. arXiv. https://doi.org/10.48550/arXiv.2505.16424

Nass, M., Alégroth, E., & Feldt, R. (2024). Improving web element localization by using a large language model. *Software Testing, Verification and Reliability*, *34*(7), Article e1893. https://doi.org/10.1002/stvr.1893

Nass, M., Alégroth, E., Feldt, R., Leotta, M., & Ricca, F. (2023). Similarity-based web element localization for robust test automation. *ACM Transactions on Software Engineering and Methodology*, *32*(3), 1–30. https://doi.org/10.1145/3571855

Saarathy, S. C. P., Bathrachalam, S., & Rajendran, B. K. (2024). Self-healing test automation framework using AI and ML. *International Journal of Strategic Management*, *3*(3), 45–77. https://doi.org/10.47604/ijsm.2843

Shayan Daneshvar, S., & Wang, S. (2024). *GUI element detection using SOTA YOLO deep learning models*. arXiv. https://doi.org/10.48550/arXiv.2408.03507

Taghizadeh Kourayem, F., Kabaranzad Ghadim, M., & Mousavi, S. A. A. (2024). A mechanism to manage time and increase data accuracy when using the Selenium library. *Knowledge Retrieval and Semantic Systems*, *11*(41), 204–225. https://doi.org/10.22054/jks.2024.80235.1660 [In Persian]