Performance Analysis for Web Scraping Tools: Case Studies on Beautifulsoup, Scrapy, Htmlunit and Jsoup

Chapter · April 2024

DOI: 10.1007/978-3-031-56728-5_39

11 authors, including:

Yılmaz Dikilitaş
Kocaeli University
3 PUBLICATIONS **14** CITATIONS

SEE PROFILE

Ahmet Sayar
Kocaeli University
176 PUBLICATIONS **1,432** CITATIONS

SEE PROFILE

Performance Analysis for Web Scraping Tools : Case Studies on Beautifulsoup, Scrapy, Htmlunit and Jsoup

Yılmaz Dikilitaş[1][0000−0002−8892−574X], Çoşkun Çakal[2][0009−0008−0151−5520], Ahmet Can Okumuş[1][0009−0006−0429−6226], Halime Nur Yalçın[1][0009−0007−9044−4586], Emine Yıldırım[1][0009−0000−8751−8803], Ömer Faruk Ulusoy[1][0009−0008−0151−5520], Bilal Macit[2][0009−0001−0106−9793], Aslı Ece Kırkaya[2][0009−0003−5800−1364], Özkan Yalçın[2][0009−0004−7904−3237], Ekin Erdoğmuş[2][0009−0001−5443−9284], and Ahmet Sayar[2][0000−0002−6335−459X]

[1] Kocaeli University, Kocaeli 41001, Turkey
[2] Haratres Teknoloji, Kocaeli, Turkey

Abstract. Web scraping has become an indispensable technique for extracting valuable data from websites. With the growing demand for efficient and reliable web scraping tools, it is crucial to assess their performance to guide developers and researchers in selecting the most suitable tool for their needs. In this paper, we present a comprehensive performance analysis of four popular web scraping tools: BeautifulSoup, Scrapy, HtmlUnit, and Jsoup. Our study focuses on evaluating these tools based on metrics such as execution time, memory usage, and scalability. We conducted experiments using various websites and datasets to provide a comprehensive evaluation of the tools' performance. The results highlight the strengths and limitations of each tool, allowing users to make informed decisions when choosing a web scraping tool based on performance requirements. Additionally, we discuss real-world use cases and the impact of website structures on tool performance. This paper aims to assist developers and researchers in selecting the most appropriate web scraping tool for their specific needs, and it also identifies avenues for future research to further enhance the performance of these tools.

Keywords: Web Scraping · Scrapy · Beautifulsoup · Jsoup · HtmlUnit.

1 Introduction

Web scraping, the process of extracting data from websites, has gained significant importance in various domains such as data analytics, research, and business intelligence. It enables the automated collection of valuable information from a wide range of online sources. As web scraping continues to evolve, numerous tools have emerged to facilitate this task, each offering different functionalities, features, and performance characteristics.

The performance of web scraping tools plays a critical role in determining their effectiveness and efficiency in data extraction. The selection of an appropriate tool depends on factors such as execution time, memory usage, scalability, and adaptability to different website structures. However, the lack of comprehensive performance analysis often leaves developers and researchers uncertain about which tool to choose for their specific requirements.

In this paper, our aim is to bridge this gap by conducting a thorough performance analysis of four popular web scraping tools. BeautifulSoup, Scrapy, HtmlUnit, and Jsoup. These tools were chosen based on their popularity, widespread usage, and diverse functionalities. Our study focuses on evaluating its performance on various metrics to provide a comprehensive assessment of its strengths and limitations.

Web scraping tools such as Scrapy, BeautifulSoup, HtmlUnit, and Jsoup have been widely used in various research studies and applications. These tools have been used in different domains, including monitoring cyber trafficking, sentiment analysis, job market analysis, online reviews analysis, and disease detection. The following studies highlight the use and performance analysis of these web scraping tools in various contexts.

In a study by (Acerado, 2023), a web application for cyber monitoring and trafficking integrated with a web scraper was developed using Beautiful Soup. The sensitivity analysis conducted in the study determined that Beautiful Soup was the most suitable tool for developing web scraping algorithms based on performance, portability, and accuracy rates compared to Scrapy and Selenium[1].

Moro et al. (2019) conducted experiments using web scraped data from online sources to analyze hotel online reviews. They highlighted the advantages of web scraping, such as the retrieval of freely written opinions and the collection of a large volume of information at high speed[2].

Han and Anderson (2020) discussed popular Python libraries for web scraping, including BeautifulSoup and Scrapy, in the context of hospitality research. They provided instructions and insights on the use of these libraries for online data collection[3].

Wooldridge and King (2018) mentioned the use of web scraping tools and APIs to track the attention of online media in the context of alternative metrics (altmetrics) for the evaluation of research impact[4].

Zucco et al. (2019) conducted a review of sentiment analysis methods and tools, including web scraping tools, to mine text and social networks data. They compared and analyzed 24 tools based on criteria such as usability, flexibility, and type of analysis performed[5].

Pellert et al. (2020) built a self-updating monitor of emotion dynamics during the COVID-19 pandemic using web scraping and API access. They used Web scraping to retrieve data from news platforms, Twitter, and chat platforms for sentiment analysis[6].

Alrusaini (2023) used web scraping with BeautifulSoup, SERP API, and request libraries to obtain skin images for the detection of Monkeypox disease.

Deep learning models were trained on the scraped data for accurate disease detection[7].

Two points are very important in terms of performance in web scraping. The first is the time it takes to transfer the data and the second is the time it takes to parse the data. There have been several studies in the literature on data transfer acceleration. Some of them are related to changing the format of the data ([8],[9],[10] and [11]), some are related to data dilution ([12] and [13]) and some are related to parallel transfer of data ([14]and [15]). Since we will pull the data directly with the HTML protocol over port 8080, such approaches do not provide us with a solution. Analyzes in parsing the data rather than pulling the data will be done. The work presented in this paper will evaluate the performance of web scraping tools in parsing HTML pages and other identified features.

These studies demonstrate the diverse applications of web scraping tools in different research domains. Performance analysis of these tools highlights their suitability for specific tasks based on factors such as performance, accuracy, and ease of use.

The objectives of this research are two-fold. First, we aim to assess the execution time of each tool, considering factors such as parsing speed and network latency. Second, we analyze memory usage to understand the impact of each tool on system resources.

By conducting rigorous experiments and benchmarking against real-world websites and datasets, we provide an in-depth analysis of the performance characteristics of these web scraping tools. The insights gained from this study will help developers and researchers make informed decisions when selecting a tool based on their performance requirements.

The remainder of this paper is organized as follows. Section 2 provides a state-of-the-art on web scraping and performance analysis. Section 3 describes the methodology employed for our performance analysis, including the experimental setup and the metrics used for evaluation, presents the results of our performance analysis, highlighting the strengths and limitations of each tool. Finally, Section 4 concludes the paper by summarizing the key findings and providing recommendations for selecting a web scraping tool based on performance requirements.

2 State of Art

Web scraping is a technique that is used to extract data from websites automatically. It involves targeting specific webpages, extracting the underlying HTML code, parsing the relevant data, and saving it to a file system or database for further analysis (Darmawan et al., 2022). Web scraping has become an essential tool in various domains, including healthcare, psychology, website monitoring, entrepreneurship research, film analysis, and credibility analysis on social media platforms[16].

In the healthcare field, web scraping has been used to collect digital images of skin lesions for diseases such as Monkeypox, Chickenpox, Smallpox, Cowpox, and Measles (Islam et al., 2022). However, the lack of publicly available and reliable digital image databases for certain diseases, such as monkeypox, has led researchers to utilize web scraping to collect the necessary data (Islam et al., 2022)[17].

Psychological research has also benefited from web scraping, particularly in the extraction of big data from the Internet for use in studies (Landers et al., 2016). Researchers are encouraged to determine their research questions and hypotheses before using web scraping to address those questions, to avoid the pitfalls associated with hypothesizing after the results are known (Landers et al., 2016)[18].

Web scraping has proven to be valuable in website monitoring systems, allowing automatic checks of website availability (Arhandi et al., 2021). By using web scraping techniques and tools such as Raspberry Pi, researchers have achieved high accuracy in monitoring website availability (Arhandi et al., 2021)[19].

Entrepreneurship research has also used web scraping to collect large-scale data on entrepreneurial activities (Quinn et al., 2022). These data can be used to develop and test theories in entrepreneurial research (Quinn et al., 2022)[20].

In film analysis, web scraping has been used to study cultural phenomena such as hipster culture. By analyzing data collected through web scraping, researchers can gain insight into the prevalence and impact of hipster culture in various contexts.

Web scraping has also been used for credibility analysis on social media platforms such as Twitter. It has been compared to other extraction methods, such as API methods, to analyze credibility on social media platforms.

Web scraping techniques have been evaluated and compared in terms of their performance and effectiveness. For example, a study evaluated the performance of web scraping techniques using XPath, CSS Selector, Regular Expression, and HTML DOM with multiprocessing technical applications (Darmawan et al., 2022). The study found that web scraping is an effective and efficient technique for extracting and storing data from websites (Darmawan et al., 2022)[16].

In general, web scraping has become an important tool in various fields for data collection, analysis, and research purposes. It offers advantages such as efficiency, scalability, and the ability to collect data from diverse sources on the Internet. However, it is important for researchers to carefully consider their research questions and hypotheses before using web scraping to ensure its suitability for their specific research needs. Furthermore, researchers should be aware of the challenges associated with web scraping, such as changes in website structure and availability of data.

3 Methodology

In the performance analysis of web scraping tools, the selection of appropriate websites or data sets for testing and benchmarking plays a crucial role in

obtaining reliable and meaningful results. Researchers must carefully consider several factors to ensure a comprehensive evaluation of the performance of the tools. First, the choice of website should encompass a diverse range of structure and complexity. Moreover, websites featuring dynamic content generated through JavaScript should be included to assess the tools' handling of such scenarios. Real-world websites are essential for ensuring the evaluation's relevance to practical use cases. These websites often exhibit variations in coding practices, responsiveness, and content presentation, providing a more accurate representation of the challenges faced during actual web scraping tasks.

As a method, information of 24 products was taken on the homepage of an e-Commerce site. This process was repeated 100 times for all tools. The table 2 was created by taking the average of these epochs.Specifications of the computer used in this experiment; AMD Ryzen 5600X, 16 GB 3600 DDR5 RAM, 6600XT 8GB, 1TB SSD (1GB Read / 1GB Write).

The size and scale of the website used for testing are critical factors in covering a broad spectrum of scenarios. This includes large-scale website with extensive content to evaluate the tools' performance under varying data volumes. Additionally, the data set should contain diverse data types and formats, such as structured data such as tables, unstructured text, images, and multimedia content, to test the tools' parsing and extraction capabilities across different data representations. We choose very large e-commerce website for these reasons.

Ethical considerations are paramount that must obtain permission from website owners before conducting scraping activities to respect the websites' terms of service and legal constraints. Adherence to the website's robots.txt file is also essential to avoid overloading the servers and maintaining ethical scraping practices.

Table 1. Comparison of Web Scraping Libraries

Library	**Language**	**DOM Parsing**	**JavaScript Execution**
Scrapy	Python	No	Yes
BeautifulSoup	Python	Yes	No
Jsoup	Java	Yes	No
HtmlUnit	Java	Yes	Yes

Table 2. Performance Comparison of Web Scraping Libraries

Library	**Memory Usage**	**CPU Usage**	**Working Time**
Scrapy	2400 MB	2.8	8.1 sec
BeautifulSoup	8500 MB	3.7	8.6 sec
Jsoup	2150 MB	1.5	7.3 sec
HtmlUnit	2600 MB	4.1	9.7 sec

Memory Usage: This metric refers to the amount of memory consumed by each web scraping library during the scraping process. It provides an indication of the library's efficiency in managing memory resources. Libraries with low memory usage are more efficient in terms of memory consumption, which can be advantageous when dealing with large-scale scraping tasks or limited system resources. On the other hand, libraries with moderate or high memory usage may be more suitable for projects that require more advanced features or handling of complex data structures.

CPU Usage: This metric measures the CPU utilization of each web scraping library. Reflects the amount of computational resources the library requires to perform scraping tasks. Libraries with low CPU usage are more efficient in terms of utilizing system resources, which can result in faster scraping times and lower overall CPU load. However, libraries with moderate or high CPU usage may be necessary for projects that involve complex data processing, JavaScript execution, or other computationally intensive operations.

Working time: This metric represents the time it takes each web scraping library to complete a scraping task. Indicates the efficiency and speed of the library in retrieving and processing web data. Libraries with fast working times are advantageous when quick results are desired or when dealing with time-sensitive data. Moderate working times may be acceptable for most scraping tasks, while slower working times might be justified for projects that require extensive data processing or interaction with dynamic web elements.

By evaluating these metrics, developers can assess the performance characteristics of each library and choose the one that best suits their specific scraping requirements. It is important to consider the trade-offs between memory usage, CPU utilization, and working time based on the project's constraints and priorities.

The table 2 provides a comprehensive comparison of four popular web scraping libraries: Scrapy, BeautifulSoup, Jsoup, and HtmlUnit. The evaluation is based on three key variables: memory usage, CPU usage, and working time. Scrapy, a powerful Python framework, offers moderate memory usage and CPU utilization, resulting in a balanced working time. It provides a high-level interface for handling complex scraping tasks and is particularly useful for projects that require advanced features or support for JavaScript execution.

BeautifulSoup, a Python library focused on HTML parsing, stands out with its medium memory and low CPU usage. This makes it an efficient choice for scraping tasks that prioritize speed and resource efficiency. Excels at extracting data from HTML documents and offers a simple and intuitive API, making it suitable for smaller projects or cases where a lightweight solution is desired.

Jsoup, a Java library designed for HTML parsing and manipulation, exhibits characteristics similar to those of BeautifulSoup. It also boasts low memory and CPU usage, enabling fast scraping operations. The strength of Jsoup lies in its ability to navigate and manipulate the HTML structure, making it ideal for extracting specific data elements from HTML documents.

HtmlUnit, a Java-based library, takes a different approach by simulating a Web browser environment. While it requires a moderate amount of memory, it utilizes higher CPU resources due to its JavaScript execution capabilities. HtmlUnit is suitable for scraping tasks that involve dynamic web content or require interaction with JavaScript-based elements.

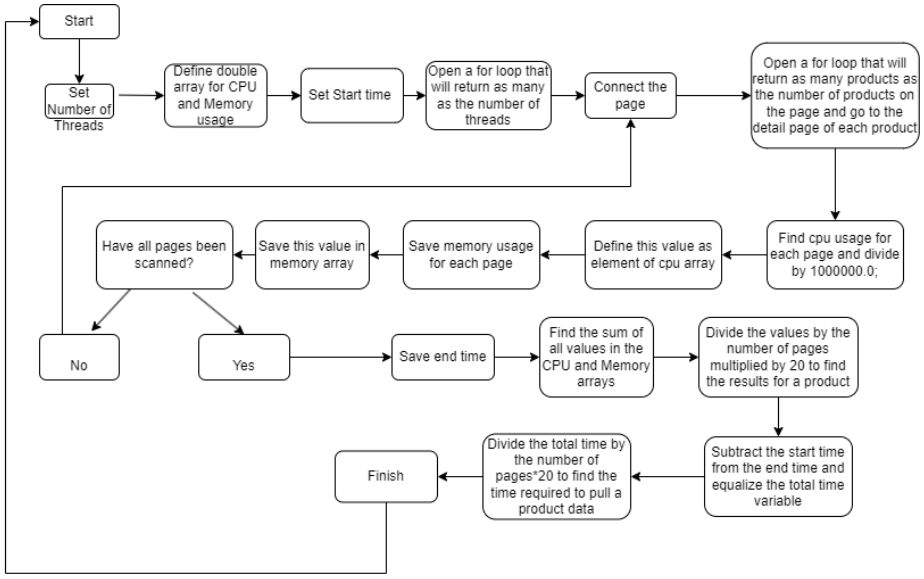

Fig. 1. Process of Used System

Performance analysis of web scraping libraries reveals several interesting findings. Figure 1 shows in detail how the process works and how the runtime is measured.

Firstly, Scrapy, a Python-based framework, demonstrates a balance between memory usage, CPU usage, and work time. This makes it a versatile choice for projects that require advanced features, support for JavaScript execution, and moderate resource consumption. However, for projects where speed and resource efficiency are prioritized over advanced functionality, BeautifulSoup and Jsoup emerge as strong contenders. Both libraries exhibit low memory and CPU usage, resulting in faster scraping operations and efficient resource utilization.

Another noteworthy finding is the impact of JavaScript execution on resource requirements. HtmlUnit, the Java-based library with JavaScript execution capabilities, demonstrates moderate memory usage and higher CPU utilization. This indicates that projects requiring dynamic web content or JavaScript interaction may benefit from HtmlUnit's capabilities, albeit with a trade-off in terms of resource consumption and working time.

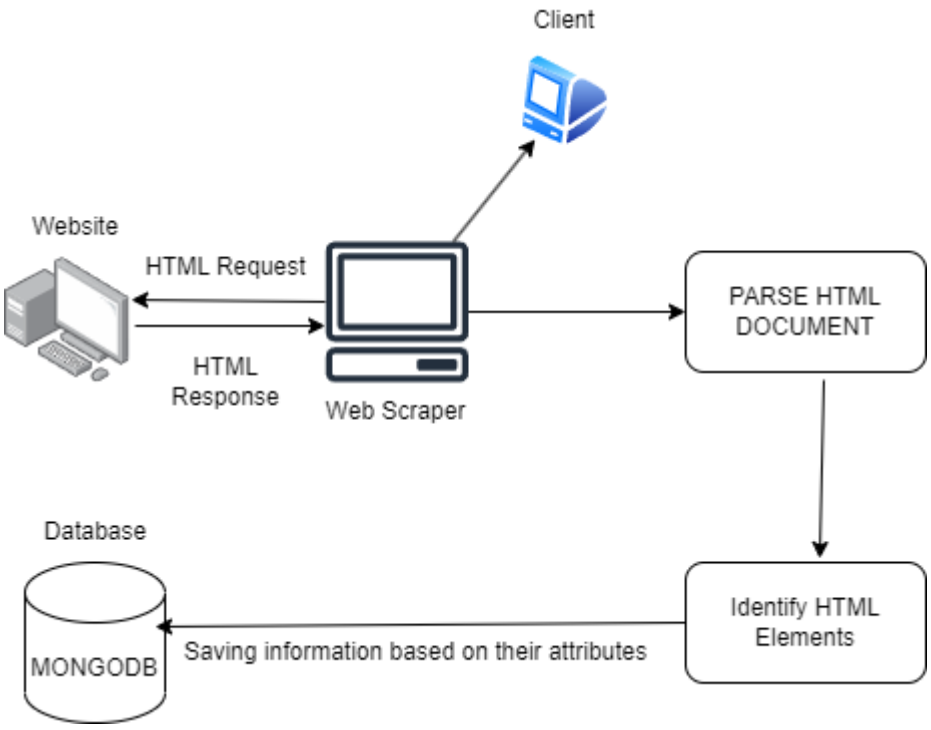

Fig. 2. Architecture of System

In general, the choice of a web scraping library depends on the specific requirements and constraints of the project. Figure 2 shows how the system works and is set up. For simpler scraping tasks that prioritize speed and resource efficiency, BeautifulSoup and Jsoup provide lightweight options. Scrapy, on the other hand, offers a comprehensive solution for more complex scraping scenarios, especially when advanced features and JavaScript execution are needed. HtmlUnit serves as a specialized option for projects with specific requirements for JavaScript handling.

It is crucial for developers to carefully evaluate the trade-offs between memory usage, CPU utilization, and working time based on the specific needs of their project. By considering these factors, developers can select the web scraping library that best aligns with their desired performance characteristics and resource constraints.

4 Conclusion

In this analysis, we compared the performance of four popular web scraping libraries: Scrapy, BeautifulSoup, Jsoup, and HtmlUnit. By evaluating metrics

such as memory usage, CPU utilization, and working time, we gained insight into the strengths and trade-offs of each library.

Scrapy stood out as a powerful framework that offers advanced features and JavaScript execution capabilities. Its moderate memory usage and CPU utilization make it suitable for complex scraping tasks that require extensive data processing. BeautifulSoup and Jsoup, on the other hand, excelled in efficiency with low memory and CPU usage, enabling fast scraping operations for simpler tasks.

The presence of JavaScript execution played a significant role in resource requirements. HtmlUnit, which supports JavaScript interaction, exhibited moderate memory usage and higher CPU utilization. This makes it a valuable choice for projects involving dynamic web content, but may come with a trade-off in terms of resource consumption and working time.

The selection of a web scraping library should be based on the specific requirements of the project. Factors such as the complexity of scraping tasks, the availability of system resources, the desired scraping speed, and the need for JavaScript execution should be considered. By carefully evaluating these factors, developers can make informed decisions to optimize memory usage, CPU utilization, and working time for efficient and effective Web data extraction.

The study's contributions lie in providing a comprehensive understanding of the performance characteristics of these web scraping tools, enabling developers and researchers to make informed decisions when selecting the most suitable tool for their specific scraping needs. By evaluating a diverse set of tools and metrics, the research broadened the scope of the existing literature on web scraping performance analysis and identified strengths and weaknesses that can aid users in optimizing their scraping processes. Additionally, the paper highlights the importance of considering website complexity, scalability, and JavaScript handling when choosing an appropriate web scraping tool for different use cases.

In conclusion, our analysis provides valuable insight into the performance characteristics of different web scraping libraries. It highlights the importance of considering trade-offs between memory usage, CPU utilization, and working time when selecting a library. Ultimately, the choice should align with the project's requirements, balancing efficiency, speed, and resource constraints to achieve successful web scraping outcomes.

References

[1] R. Acerado. "Cmata: Cyber Trafficking Monitoring and Tracking Prototype". In: *IJFCC* (2023), pp. 19–22. DOI: 10.18178/ijfcc.2023.12.1.598.

[2] S. Moro and S. Esmerado J.and Jalali. "Can We Trace Back Hotel Online Reviews' Characteristics Using Gamification Features?" In: *International Journal of Information Management* 44 (2019), pp. 88–95. DOI: 10.1016/j.ijinfomgt.2018.09.015.

[3] S. Han and C. Anderson. "Web Scraping For Hospitality Research: Overview, Opportunities, and Implications". In: *Cornell Hospitality Quarterly* 62 (1 2020), pp. 89–104. DOI: 10.1177/1938965520973587.

[4] J. Wooldridge and M. King. "Altmetric Scores: An Early Indicator Of Research Impact". In: *Journal of the Association for Information Science and Technolo* 70 (3 2018), pp. 271–282. DOI: 10.1002/asi.24122.

[5] C. Zucco et al. "Sentiment Analysis For Mining Texts and Social Networks Data: Methods And Tools". In: *WIREs Data Mining Knowl Discov* 10 (1 2019). DOI: 10.1002/widm.1333.

[6] M. Pellert et al. "Dashboard Of Sentiment In Austrian Social Media During Covid-19". In: *Front. Big Data* 3 (2020). DOI: 10.3389/fdata.2020.00032.

[7] O. Alrusaini. "Deep Learning Models For the Detection Of Monkeypox Skin Lesion On Digital Skin Images". In: *IJACSA* 14 (1 2023). DOI: 10.14569/ijacsa.2023.0140170.

[8] Suleyman Eken and Ahmet Sayar. "Performance Evaluations of Vector-Raster Satellite Image Transfers through Web Services". In: *2012 IEEE 36th Annual Computer Software and Applications Conference*. IEEE. 2012, pp. 346–347.

[9] Suleyman Eken and Ahmet Sayar. "Vectorization and spatial query architecture on island satellite images". In: *Procedia Comput. Sci. J* 2 (2012), pp. 37–43.

[10] Suleyman Eken, Eray Aydin, and Ahmet Sayar. "Vectorization of Large Amounts of Raster Satellite Images in a Distributed Architecture Using HIPI". In: *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*. IEEE. 2017, pp. 1–4.

[11] Suleyman EKEN and Ahmet SAYAR. "Uydu Görüntülerinin Yüksek Performansta İşlenmesi Üzerine Bir İnceme: Vektör Tabanlı Mozaik Örme Durum Çalışması". In: ().

[12] Ahmet Sayar. "Adaptive proxy map server for efficient vector spatial data rendering". In: *Journal of Applied Remote Sensing* 7.1 (2013), pp. 073498–073498.

[13] S Eken and A Sayar. "Vector modelling of island satallite images for spatial databases". In: *Proc. Of International Science and Technology Conference (ISTEC 11)*. 2011, pp. 25–30.

[14] Abdurrahim Ozel et al. "Web Servisler ile Paralel Görüntü İşleme Mimarisi: Raster İmgelerde Kenar Belirleme Uygulanması". In: (2012).

[15] Geoffrey C Fox et al. "Grids for real time data applications". In: *Parallel Processing and Applied Mathematics: 6th International Conference, PPAM 2005, Poznań, Poland, September 11-14, 2005, Revised Selected Papers 6*. Springer. 2006, pp. 320–332.

[16] I. Darmawan et al. "Evaluating Web Scraping Performance Using Xpath, Css Selector, Regular Expression, and Html Dom With Multiprocessing Technical Applications". In: *JOIV : Int. J. Inform. Visualization* 6 (4 2022), p. 904. DOI: 10.30630/joiv.6.4.1525.

[17] T. Islam et al. "Can Artificial Intelligence Detect Monkeypox From Digital Skin Images?" In: (2022). DOI: 10.1101/2022.08.08.503193.

[18] R. Landers et al. "A Primer On Theory-driven Web Scraping: Automatic Extraction Of Big Data From the Internet For Use In Psychological Research." In: *Psychological Methods* 21 (4 2016), pp. 475–492. DOI: 10.1037/met0000081.

[19] P. Arhandi, I. Mashudi, and F. Nugroho. "Automated Website Monitoring System Using Web Scraping and Raspberry Pi". In: *Telematika* 18 (2 2021), p. 222. DOI: 10.31315/telematika.v18i2.5506.

[20] L. Quinn et al. "Explaining Offenders' Longitudinal Product-specific Target Selection Through Changes In Disposability, Availability, and Value: An Open-source Intelligence Web-scraping Approach". In: *Crime Sci* 11 (1 2022). DOI: 10.1186/s40163-022-00164-1.