https://doi.org/10.31891/2307-5732-2025-359-128
УДК 004.91

ІВАНОВ ДЕНИС
Національний технічний університет «Дніпровська політехніка»
https://orcid.org/0000-0001-8660-0928
e-mail: ivanov.d.v@nmu.one
ГАРКУША ІГОР
Національний технічний університет «Дніпровська політехніка»
https://orcid.org/0000-0003-1190-1501
e-mail: garkusha.i.m@nmu.one

МЕТОДОЛОГІЯ ПАРСИНГУ СТОРІНОК ВЕБ-САЙТІВ ДЛЯ АВТОМАТИЗАЦІЇ ЗБОРУ РІЗНОСТРУКТУРОВАНИХ ДАНИХ

У статті розглянуто методологію парсингу як комплексну систему методів і засобів, що забезпечує ефективне вилучення як структурованих (таблиці, списки, JSON, XML), так і неструктурованих (текстові блоки, статті, новини) даних із веб-сайтів. Запропоновано узагальнену архітектуру процесу парсингу, яка охоплює етапи аналізу джерела даних, вибору інструментів (наприклад, BeautifulSoup, Scrapy, Selenium, Puppeteer), побудови правил парсингу, обробки виняткових ситуацій (капча, JavaScript-рендеринг, динамічне завантаження контенту), а також збереження результатів у форматах, придатних для подальшого аналізу. Основна увага зосереджена на порівняльному аналізі методів обробки структурованих та неструктурованих даних, включно з підходами до семантичного аналізу, регулярними виразами, застосуванням NLP-технологій та алгоритмів машинного навчання. Визначено основні проблеми, що виникають при автоматичному зборі інформації, зокрема обмеження політики доступу (robots.txt), правові аспекти, складність обробки мультимодального контенту та зміна структури веб-сторінок. Наведено практичні поради для підвищення стійкості парсерів до змін веб-ресурсів і дотримання етичних норм під час обробки відкритих даних. Ефективність описаної методології підтверджено на прикладі створення системи автоматичного моніторингу новинних сайтів, яка здійснює регулярне вилучення текстів публікацій, їх попередню обробку та збереження у базі даних для подальшої тематичної класифікації. Результати дослідження можуть бути застосовані в практичних інформаційних системах, що потребують регулярного збору даних, а також у наукових дослідженнях, де важливе значення має об'єктивне й масштабоване вилучення інформації з веб-середовища.

Ключові слова: *парсинг, структуровані дані, неструктуровані дані, веб-скрейпінг, автоматизація збору інформації, веб-сайти, аналіз контенту, BeautifulSoup, Scrapy, семантичний аналіз, обробка тексту, інформаційні системи.*

IVANOV DENYS
GARKUSHA IGOR
Dnipro University of Technology

METHODOLOGY OF WEB PAGE PARSING FOR AUTOMATED COLLECTION OF HETEROGENEOUS DATA

The article examines the methodology of web page parsing as a comprehensive and systematic framework of methods and tools that enables efficient extraction of both structured data (tables, lists, JSON, XML) and unstructured data (text blocks, articles, news reports) from websites. A generalized architecture of the parsing process is proposed, encompassing the stages of data source analysis, tool selection (e.g., BeautifulSoup, Scrapy, Selenium, Puppeteer), construction of parsing rules, handling exceptional cases (captchas, JavaScript rendering, dynamically loaded content), as well as storing results in formats suitable for subsequent large-scale analysis and the integration into information systems involves a thorough comparative evaluation of methods designed for the processing of structured and unstructured data. The analysis focuses on approaches such as semantic interpretation, the application of regular expressions, advanced natural language processing (NLP) technologies, and algorithmic solutions based on machine learning. The study identifies key challenges in automated information retrieval, such as access policy restrictions, legal and ethical considerations, the complexity of multimodal content processing, and frequent changes in the underlying structure of web pages. Practical recommendations are offered to improve the robustness and adaptability of parsers against modifications of online resources while maintaining compliance with data ethics standards. The effectiveness of the proposed methodology is validated by the development of an automated news monitoring system that performs regular extraction of publication texts, their preprocessing, and structured storage in a database for subsequent thematic classification and knowledge discovery. The results of the study may be utilized in practice not only in the design of practical information systems that require continuous and scalable data collection, but also in scientific research domains where objective, reproducible, and high-volume information extraction from the web environment serves as a key factor.

Keywords: *parsing, structured data, unstructured data, web scraping, information collection automation, websites, content analysis, BeautifulSoup, Scrapy, semantic analysis, text processing, information systems.*

Стаття надійшла до редакції / Received 18.09.2025
Прийнята до друку / Accepted 05.11.2025

Постановка проблеми

В умовах сучасного цифрового простору кількість даних, доступних у відкритому доступі через веб-сайти, зростає експоненційно. Ці дані становлять важливе джерело інформації для наукових досліджень та бізнес-аналітики, моніторингу суспільних процесів, даних для журналістики та державного управління. Водночас, дані на веб-ресурсах представлені у різних форматах – як структурованих (таблиці, XML, JSON, RDF), так і неструктурованих (текстові повідомлення, коментарі, статті, огляди) [1].

Для ефективного отримання таких даних все частіше застосовуються методи автоматизованого збору – веб-скрейпінг (web scraping) та парсинг. Проте традиційні інструменти парсингу (наприклад, BeautifulSoup, Scrapy, Selenium) все частіше стикаються з технічними бар'єрами. Сучасні веб-сайти використовують складні технології клієнтського рендерингу на основі JavaScript, динамічне завантаження контенту, механізми захисту, такі як CAPTCHA та Cloudflare, що ускладнюють доступ до інформації [2].

Ще одним викликом є обробка неструктурованих даних. Вони потребують застосування методів

обробки природної мови (NLP), машинного навчання, а також новітніх підходів, таких як великі мовні моделі (LLM), які можуть здійснювати семантичний аналіз текстів та автоматичне вилучення сутностей [3].

Окрім технічних проблем, існують також правові та етичні обмеження, пов'язані з використанням веб-даних. Зокрема, у відповідь на активне використання штучного інтелекту великі онлайн-платформи оновлюють політики доступу до своїх API та блокують ботів (наприклад, оновлення протоколу robots.txt, як це зробив Reddit) [4].

Паралельно з цим, бізнес-середовище дедалі більше потребує надійних, масштабованих рішень для автоматизованого збору та аналізу даних із мережі. За даними звітів аналітичних компаній, очікується суттєве зростання попиту на технології веб-скрейпінгу в найближчі роки, що підкреслює актуальність створення ефективних, адаптивних систем збору інформації [5].

Таким чином, існує потреба в комплексній методології, яка враховує особливості різних типів даних, адаптивність до змін веб-структур, дотримання етичних норм, а також застосування сучасних технологій інтелектуальної обробки. Така методологія має забезпечити не лише ефективне отримання даних, а й їхнє коректне зберігання, фільтрацію, трансформацію та використання в інформаційних системах для прийняття рішень [6].

Аналіз останніх публікацій

Протягом останніх років можна відзначити активний розвиток методів парсингу сторінок веб-сайтів, зокрема у контексті обробки різноструктурованої інформації. Це зумовлено зростаючою потребою в автоматизованому зборі даних для аналітики, досліджень та навчання моделей штучного інтелекту.

У роботі Vijayaragavan Pichiyan та ін. [7] розглянуто інтеграцію методів веб-скрейпінгу з NLP - технологіями для ефективного вилучення та аналізу неструктурованих текстових даних з веб-джерел. Дослідники акцентують увагу на поєднанні традиційних методів парсингу, таких як XPath та бібліотеки BeautifulSoup і Scrapy, з сучасними NLP-підходами, включаючи токенізацію, стемінг, видалення стоп-слів та розпізнавання іменованих сутностей (NER), що дозволяє перетворювати неструктуровані тексти у структуровану інформацію для подальшого аналізу.

У дослідженні Paliwal U. [8] проведено порівняльний аналіз методів веб-скрейпінгу для вилучення структурованих даних. Автори публікації оцінюють точність та ефективність різних підходів до парсингу, зокрема, використання DOM-аналізу та CSS-селекторів, що дозволяє визначити оптимальні стратегії для автоматизованого збору даних з веб-сайтів.

Автори публікації Jain S. та ін. [9] у своєму огляді методів аналізу та парсингу неструктурованих даних підкреслюють важливість автоматизації процесів обробки логів та інших системних файлів. У роботі запропоновано загальну структуру для автоматичного парсингу, яка дозволяє ефективно обробляти великі обсяги неструктурованої інформації для системного моніторингу та виявлення аномалій.

У роботі Chandrasekharuni Y. [10] запропоновано новий підхід до представлення веб-додатків у вигляді графів знань. Цей підхід дозволяє моделювати стан веб-додатків як структуровані представлення, що сприяє кращому розумінню їхньої функціональності та поведінки, а також полегшує процес автоматизованого тестування та аналізу поведінки користувачів.

У дослідженні Lotfi C. та ін. [11] проведено огляд технік та застосувань веб-скрейпінгу в контексті аналізу великих даних. Автори публікації акцентують увагу на важливості застосування методів веб-скрейпінгу для вилучення корисної інформації з різноманітних веб-джерел, включаючи соціальні медіа, веб-сайти та онлайн-платформи, що є критично важливим для прийняття зважених бізнес-рішень.

Таким чином, здійснений аналіз демонструє, що сучасні дослідження зосереджені на вдосконаленні технічних методів парсингу, а також на розгляді правових та етичних аспектів автоматизованого збору даних з веб-сайтів. Це підкреслює необхідність розробки комплексної методології, яка враховує як технічні, так і нормативні вимоги.

Формулювання цілей статті

Метою роботи є обґрунтування та розробка методології парсингу різноструктурованих даних зі сторінок веб-сайтів для забезпечення ефективного, масштабованого та автоматизованого збору інформації, що може бути застосована у практичних інформаційних системах та наукових дослідженнях, пов'язаних з обробкою веб-контенту.

Виклад основного матеріалу

У сучасних інформаційних системах автоматизований збір даних з веб-ресурсів є важливим елементом у створенні баз знань, аналітичних звітів та підтримці процесу прийняття рішень. Залежно від структури джерела, розрізняють два основних підходи до парсингу: робота зі структурованими даними (таблиці, JSON, XML тощо) та неструктурованими даними (HTML-сторінки з довільною верстkою, тексти без чітких форматів). Методологія, що пропонується в цій роботі, передбачає поетапний підхід до обробки обох типів даних із врахуванням специфіки веб-середовища.

На початковому етапі здійснюється ідентифікація типу даних, доступних на сайті. Це може бути:

- структурований формат: дані, що доступні через API у форматах JSON, XML або CSV;
- напівструктурований або неструктурований формат: дані, розміщені в HTML-розмітці сторінки, без чіткої структури.

Аналіз включає вивчення DOM-структури HTML-документа, ідентифікацію шаблонів повторюваних елементів (наприклад, картки товарів, пости в блозі) та визначення ключових вузлів, що містять цільову

інформацію.

Запропонована методологія парсингу передбачає поетапну архітектуру процесу, що складається з таких основних етапів (рис. 1).

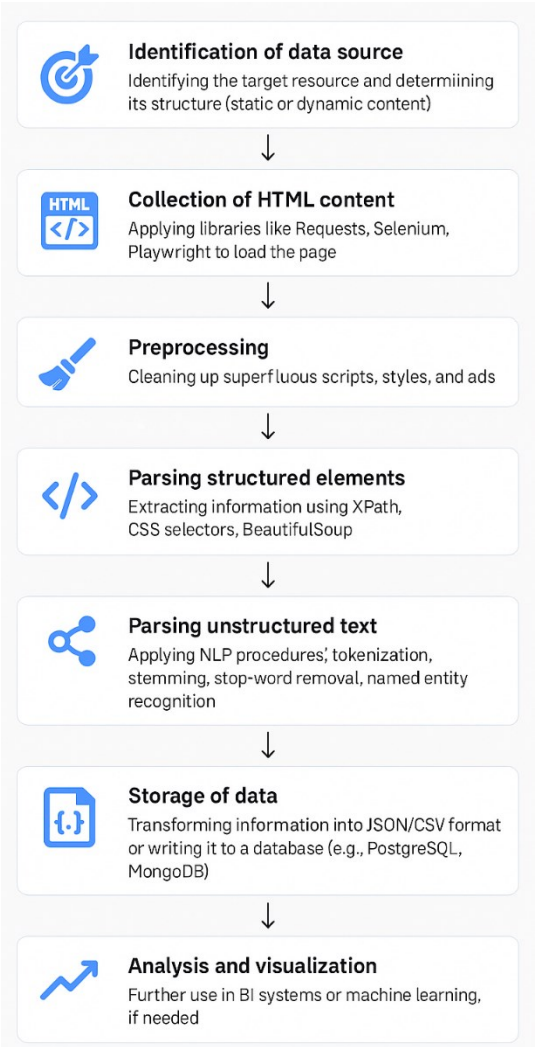

Рис. 1. Архітектура методології парсингу структурованих та неструктурованих даних

1. Ідентифікація джерела даних. На цьому етапі здійснюється аналіз веб-ресурсу, з якого планується витяг інформації. Визначається структура сторінки (наприклад, чи є контент статичним або динамічно згенерованим через JavaScript), що критично для вибору інструментів доступу. Також аналізуються частота оновлення даних, наявність API, політика доступу (включаючи файли robots.txt та CAPTCHA-захист), а також правові аспекти використання даних.

2. Збір HTML-контенту. В залежності від типу сайту, застосовуються різні бібліотеки: для статичних сторінок ефективним є використання requests або httpx, тоді як для роботи з динамічними сайтами необхідне використання headless-браузерів через Selenium, Playwright або Puppeteer. На даному етапі необхідно гарантувати стабільність завантаження сторінок, враховуючи можливі обмеження з боку серверу, таймаути, редиректи та асинхронне завантаження елементів DOM. Фрагмент коду парсеру для задання параметрів та отримання посилань на архів погоди представлено на рисунку 2.

3. Попередня обробка. Отриманий HTML-документ піддається фільтрації з метою видалення нерелевантного контенту – скриптів, стилів, рекламних блоків, аналітичних трекерів тощо. Метою є спрощення подальшого аналізу, зменшення обсягу даних та уникнення «шуму», що заважає точному вилученню інформації.

4. Парсинг структурованих елементів. Структуровані елементи веб-сторінок (таблиці, списки, блоки з визначеними класами або ID) обробляються за допомогою методів XPath, CSS-селекторів, або бібліотек таких як BeautifulSoup, lxml тощо. Особлива увага приділяється стабільності парсингу у випадку змін DOM-структури, тому доцільним є використання стратегій з «fallback»-селекторами або регулярним оновленням шаблонів.

5. Парсинг неструктурованого тексту. Для обробки вільного тексту, що не має чіткої HTML-структури, застосовуються NLP-метод. Ключові етапи процесу охоплюють токенізацію, нормалізацію (стемінг або лематизація), видалення стоп-слів, частиномовне тегування (POS-tagging), NER та ін., залежно від цілей аналізу. Застосовуються бібліотеки, такі як spaCy, NLTK, Stanza або інші мовно-орієнтовані фреймворки.

6. Збереження даних. Після парсингу дані структуруються та серіалізуються у зручному для подальшого аналізу форматі – найчастіше це JSON, CSV, або безпосередньо імпортуються у системи керування базами даних, зокрема реляційні (PostgreSQL, MySQL) чи документоорієнтовані (MongoDB, Elasticsearch). Враховується питання узгодженості схем, обробки помилок та дублювання інформації. Фрагмент колу парсеру для зберігання та конвертації в JSON представлено на рисунку 3.

7. Аналіз та візуалізація. Зібрані та збережені дані можуть бути використані для побудови аналітичних звітів, візуалізації трендів (через бібліотеки Plotly, Matplotlib, Seaborn, Tableau тощо), або слугувати вхідними даними для моделей машинного навчання. На цьому етапі також може здійснюватись інтеграція з BI-системами для формування висновків на підставі зібраної інформації.

Залежно від типу даних, використовуються різні програмні засоби:

1. Для структурованих даних: Scrapy, Puppeteer, BeautifulSoup, lxml.
2. Для неструктурованих: spaCy, NLTK, Transformers (Hugging Face), TextBlob.
3. Для збереження – бази даних SQLite, PostgreSQL, MongoDB.

Автоматизація збору інформації несе ризики порушення прав власності на дані. Методологія враховує:

- дотримання вимог robots.txt;
- обмеження на кількість запитів (throttling, backoff-алгоритми);
- анонімізацію доступу (наприклад, через проксі).

```python
class Weather:

    options = webdriver.ChromeOptions()
    prefs = {"download.default_directory": DOWNLOADS_PATH}
    options.add_argument("--start-maximized")
    options.add_experimental_option("prefs", prefs)
    # options.add_argument("--headless=new")

    driver = webdriver.Chrome(options=options)
    wait = WebDriverWait(driver, 5)

    def get_url(self, url):

        self.driver.get(url)

        accept = self.driver.find_element('xpath', '//button[@aria-label="Погоджуюсь"]')
        accept.click()

        history = self.driver.find_element('xpath', '//a[@class="icon-history-climate nav-icon"]')
        history.click()

        archive_url = self.driver.find_element('xpath', '//a[span[@itemprop="name"][text()="Архів погоди"]]').get_attribute("href")
        self.get_archive(archive_url)


    def get_archive(self, url):

        self.driver.get(url)

        select_period = int(input("Select period:\n1. Month\n2. Year\nInput number: "))

        select_year = int(input("Input year from 2023 to 2024: "))
        year = self.driver.find_element('xpath', '//label[@for="year_{}"]'.format(select_year))
        year.click()

        if select_period == 1:
```

Рис. 2. Фрагмент коду парсеру для задання параметрів та отримання посилань на архів погоди

```python
def csv_to_json(csv_file_path, json_file_path):

    data = []

    with open(csv_file_path, 'r') as csv_file:
        timestamps_flag = False

        csv_reader = csv.reader(csv_file)
        for row in csv_reader:
            row_dict = dict()
            row_dict[row[0]] = row[-1]

            if row[0] == "timestamp":
                timestamps_flag = True
                timestamps = dict()
                timestamps['timestamps'] = []
                data.append(timestamps)
                name1 = row[0]
                name2 = row[-1]
                continue

            if timestamps_flag == True:
                timestamp = dict()
                timestamp[name1] = row[0]
                timestamp[name2] = row[-1]
                data[-1]['timestamps'].append(timestamp)
            else:
                data.append(row_dict)


    with open(json_file_path, 'w') as json_file:
        json.dump(data, json_file, indent=4)
```

Рис. 3. Фрагмент коду парсеру для конвертації з csv до json

Таким чином було реалізовано експериментальну систему збору інформації про погодні умови з веб-ресурсу, що містить актуальні новини та звіти. Система здійснює парсинг структурованих та неструктурованих даних, зокрема обробляє заголовки новин, основний текст повідомлень, дату їх публікації, а також категорію матеріалу. Для забезпечення роботи з динамічними компонентами веб-сторінок, які завантажуються асинхронно (наприклад, через JavaScript), було застосовано фреймворк Selenium, що дозволяє імітувати взаємодію користувача з браузером і отримувати повний HTML-код сторінки після рендерингу.

Окрім основних функцій, система була налаштована для регулярного збору даних із заданою періодичністю та обробки помилок, що можуть виникати під час взаємодії з веб-ресурсом (наприклад, при зміні структури HTML-коду або тимчасовій недоступності сайту). Такий підхід дозволяє забезпечити стабільність збору інформації та високу якість отриманих даних для подальшого використання у дослідницьких або практичних цілях.

Для оцінки ефективності запропонованої методології парсингу архівних даних про погоду використовується стандартна метрика точності (accuracy), що визначає співвідношення правильно витягнутих погодних параметрів до загальної кількості очікуваних параметрів, що обчислюється за формулою:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}, \quad (1)$$

де

TP (True Positives) – кількість істинно позитивних випадків, що належать до позитивного класу (наприклад, температура, вологість, атмосферний тиск);

TN (True Negatives) – кількість істинно негативних випадків, що належать до негативного класу;

FP (False Positives) – кількість хибно позитивних випадків;

FN (False Negatives) – кількість хибно негативних випадків, тобто кількість даних, що мали бути витягнуті, але були пропущені парсером.

Крім того, для комплексної оцінки застосовуються:

- Precision (прецизійність):

$$Precision = \frac{|TP|}{|TP|+|FP|}, \quad (2)$$

- F1-міра:

$$F1 = 2 \cdot \frac{Precision \cdot \text{Accuracy}}{Precision + \text{Accuracy}}, \quad (3)$$

Під час тестування парсера на вибірці з 100 веб-сторінок архівних погодних даних було визначено наступні результати:

- загальна кількість очікуваних записів (температура, вологість, тиск) = 1000;
- кількість правильно витягнутих записів (TP) = 950;
- кількість пропущених записів (FN) = 50;
- кількість хибно витягнутих записів (FP) = 20.

Процес оцінювання базується на основних параметрах, таких як температура повітря, відносна вологість, атмосферний тиск та інші показники для кожної дати та географічного пункту. Для неструктурованих форматів (наприклад, новинні замітки з описом погодних умов) застосовується попереднє виділення сутностей (Named Entity Recognition) та трансформація у структурований вигляд перед оцінкою.

Для забезпечення об'єктивності тестування використовується незалежний еталонний набір даних, а також методологія крос-валідації. Також для перевірки коректності роботи парсеру були використані невалідні теги. Відповідні веб-сторінки містили заголовки, що тематично належали до потрібного розділу, однак їхній зміст включав некоректні або спотворені дані.

Аналіз отриманих показників демонструє високу ефективність розробленого парсера архівних погодних даних. Точність на рівні 95% свідчить про здатність системи стабільно вилучати більшість необхідних параметрів з веб-джерел. Прецизійність у 97.9% підтверджує низький рівень хибних спрацювань, що є особливо важливим для задач, де критичним є уникнення додавання невірних даних. F1-міра на рівні 96.4% підкреслює загальний баланс між прецизійністю та повнотою, що дозволяє зробити висновок про ефективність методології як для структурованих, так і для неструктурованих форматів архівних погодних записів.

Отримані результати свідчать про доцільність впровадження даної методології у системи автоматизованого збору погодних даних, зокрема для створення історичних кліматичних баз даних, що потребують високої точності та повноти.

Висновки

У статті запропоновано та обґрунтовано методологію парсингу різноструктурованих даних для автоматизації збору архівних погодних даних зі сторінок веб-сайтів. Розглянуто специфіку обробки різних форматів даних, зокрема HTML-таблиць, JSON-структур та неструктурованих текстів, що дозволяє підвищити універсальність парсера.

Розроблена система продемонструвала високу ефективність при тестуванні на вибірці реальних даних: точність склала 95%, прецизійність – 97.9%, F1-міра – 96.4%. Це свідчить про надійність запропонованого підходу та його здатність забезпечувати якісний збір інформації навіть у випадках з неоднорідною структурою джерел.

Методологія може бути масштабована для інших доменів, що потребують автоматизації збору даних, зокрема в екологічних, соціологічних та економічних дослідженнях. Подальші дослідження планується спрямувати на вдосконалення алгоритмів обробки неструктурованих текстів з використанням передових

методів обробки природної мови, а також інтеграцію механізмів автоматичної валідації зібраних даних для підвищення їхньої достовірності.

Розроблена методологія має практичне застосування в рамках розробленого веб-сервісу, де для побудови зон затоплення територій використовується збір даних про опади за допомогою парсингу сторінок веб-сайту [12].

Таким чином, запропонована методологія є перспективним інструментом для побудови гнучких систем збору та обробки різноструктурованих даних зі сторінок веб-сайтів, що дозволяє автоматизувати науково-практичні задачі в сфері збору та обробки даних..

Література

1. Ahluwalia A., Wani S. Leveraging Large Language Models for Web Scraping [Електронний ресурс] // arXiv. – 2024. – № arXiv:2406.08246. – Режим доступу: https://arxiv.org/abs/2406.08246

2. Chandrasekharuni Y. Representing Web Applications As Knowledge Graphs [Електронний ресурс] // arXiv. – 2024. – № arXiv:2410.17258. – Режим доступу: https://arxiv.org/abs/2410.17258

3. Brown M. A. та ін. Web Scraping for Research: Legal, Ethical, Institutional, and Scientific Considerations [Електронний ресурс] // arXiv. – 2024. – № arXiv:2410.23432. – Режим доступу: https://arxiv.org/abs/2410.23432

4. Reddit to update web standard to block automated website scraping [Електронний ресурс] // Reuters. – 2024. – Режим доступу: https://www.reuters.com/technology/reddit-update-web-standard-block-automated-website-scraping-2024-06-25/

5. ScrapeOps. Web Scraping Market Report 2025 [Електронний ресурс]. – 2025. – Режим доступу: https://scrapeops.io/web-scraping-playbook/web-scraping-market-report-2025/

6. Kalvo J. Web Scraping: Unlocking Business Insights In A Data-Driven World [Електронний ресурс] // Forbes Technology Council. – 2025. – Режим доступу: https://www.forbes.com/councils/forbestechcouncil/2025/01/27/web-scraping-unlocking-business-insights-in-a-data-driven-world/

7. Pichiyan V., Muthulingam S., Sathar G., Nalajala S., Akhil C., Das M. N. Web Scraping using Natural Language Processing: Exploiting Unstructured Text for Data Extraction and Analysis // Procedia Computer Science. – 2023. – Vol. 230. – P. 193–202. – DOI: 10.1016/j.procs.2023.12.074.

8. Paliwal U. Web scraping for extraction of structured data: A comparative study. – Erasmus University Rotterdam, 2022. – URL: http://hdl.handle.net/2105/63571.

9. Jain S., de Buitléir A., Fallon E. A review of unstructured data analysis and parsing methods. – 2020. – URL: http://research.thea.ie/handle/20.500.12065/3475.

10. Chandrasekharuni Y. Representing Web Applications As Knowledge Graphs. – 2024. – arXiv:2410.17258. – URL: https://arxiv.org/abs/2410.17258.

11. Lotfi C., Srinivasan S., Ertz M., Latrous I. Web Scraping Techniques and Applications: A Literature Review // SCRS Conference Proceedings on Intelligent Systems. – 2022. – DOI: 10.52458/978-93-91842-08-6-38.

12. Іванов Д.В., Каштан В.Ю., Гнатушенко В.В. Комп'ютерна програма «Веб-сервіс інтерактивна карта комп'ютерного моделювання затоплення території при надзвичайних ситуаціях на Дніпровській ГЕС» // Свідоцтво про реєстрацію авторського права на твір № 136440; Заявл. 21 квітня 2025 р. № с202503615; Опубл. 21 травня 2025 р. Авторське право і суміжні права.

References

1. Ahluwalia A., Wani S. Leveraging Large Language Models for Web Scraping [Elektronnyi resurs] // arXiv. – 2024. – № arXiv:2406.08246. – Rezhym dostupu: https://arxiv.org/abs/2406.08246

2. Chandrasekharuni Y. Representing Web Applications As Knowledge Graphs [Elektronnyi resurs] // arXiv. – 2024. – № arXiv:2410.17258. – Rezhym dostupu: https://arxiv.org/abs/2410.17258

3. Brown M. A. та ін. Web Scraping for Research: Legal, Ethical, Institutional, and Scientific Considerations [Електронний ресурс] // arXiv. – 2024. – № arXiv:2410.23432. – Rezhym dostupu: https://arxiv.org/abs/2410.23432

4. Reddit to update web standard to block automated website scraping [Elektronnyi resurs] // Reuters. – 2024. – Rezhym dostupu: https://www.reuters.com/technology/reddit-update-web-standard-block-automated-website-scraping-2024-06-25/

5. ScrapeOps. Web Scraping Market Report 2025 [Elektronnyi resurs]. – 2025. – Rezhym dostupu: https://scrapeops.io/web-scraping-playbook/web-scraping-market-report-2025/

6. Kalvo J. Web Scraping: Unlocking Business Insights In A Data-Driven World [Elektronnyi resurs] // Forbes Technology Council. – 2025. – Rezhym dostupu: https://www.forbes.com/councils/forbestechcouncil/2025/01/27/web-scraping-unlocking-business-insights-in-a-data-driven-world/

7. Pichiyan V., Muthulingam S., Sathar G., Nalajala S., Akhil C., Das M. N. Web Scraping using Natural Language Processing: Exploiting Unstructured Text for Data Extraction and Analysis // Procedia Computer Science. – 2023. – Vol. 230. – P. 193–202. – DOI: 10.1016/j.procs.2023.12.074.

8. Paliwal U. Web scraping for extraction of structured data: A comparative study. – Erasmus University Rotterdam, 2022. – URL: http://hdl.handle.net/2105/63571.

9. Jain S., de Buitléir A., Fallon E. A review of unstructured data analysis and parsing methods. – 2020. – URL: http://research.thea.ie/handle/20.500.12065/3475.

10. Chandrasekharuni Y. Representing Web Applications As Knowledge Graphs. – 2024. – arXiv:2410.17258. – URL: https://arxiv.org/abs/2410.17258.

11. Lotfi C., Srinivasan S., Ertz M., Latrous I. Web Scraping Techniques and Applications: A Literature Review // SCRS Conference Proceedings on Intelligent Systems. – 2022. – DOI: 10.52458/978-93-91842-08-6-38.

12. Ivanov D.V., Kashtan V.Yu., Hnatushenko V.V. Komp'iuterna prohrama «Veb-servis interaktyvna karta komp'iuternoho modeliuvannia zatoplennia terytorii pry nadzvychainykh sytuatsiiakh na Dniprovskii HES» // Svidotstvo pro reiestratsiiu avtors'koho prava na tvir № 136440; Zaiavl. 21 kvitnia 2025 r. № c202503615; Opubl. 21 travnia 2025 r. Avtors'ke pravo i sumizhni prava.