Jurnal Mandiri IT
Vol. 14 No. 3, January (2026), pp. 344-355

ISSN 2301-8984 (Print), 2809-1884 (Online)
Published by Institute of Computer Science

Automated news monitoring and sentiment analysis system using web scraping and large language models

Zerusealtin David Naibaho[1], Hondor Saragih[2]
[1,2] Informatics, Faculty of Defense Technology and Engineering, Indonesia Defense University, Bogor, Indonesia

ARTICLE INFO

Article history:

Received Dec 10, 2025
Revised Dec 23, 2025
Accepted Jan 10, 2026

Keywords:

Django Framework;
Large Language Models;
News Monitoring;
Sentiment Analysis;
Web Scraping.

ABSTRACT

Organizations increasingly require efficient systems to monitor and analyze vast online news data for timely and informed decision making. Manual monitoring is inadequate due to information overload and the time sensitive nature of digital content. This study presents the design, development, and evaluation of an automated web based news monitoring and sentiment analysis system integrating web scraping and artificial intelligence. The system was implemented using the Django web framework with a PostgreSQL database, Playwright browser automation for dynamic content extraction, and Google's Gemini API for contextual sentiment classification. Three main functions were developed: automated data collection based on keywords and date ranges, AI driven sentiment analysis producing positive, negative, or neutral labels with contextual understanding, and automated reporting with interactive visualizations exportable to XLSX and CSV formats. Functional black box testing confirmed 100% success across 28 test cases, verifying reliability in authentication, data acquisition, sentiment analysis, and visualization. Performance evaluation showed that the system could collect 50–200 articles within 2–4 minutes and process sentiment analysis at 1–2 seconds per article. The proposed system effectively transforms manual workflows into fully automated operations, enabling systematic media monitoring, sentiment tracking, and data driven decision support.

Corresponding Author:

Zerusealtin David Naibaho,
Informatics,
Indonesia Defense University,
Kawasan IPSC Sentul, Sukahati, Kec. Citeureup, Kabupaten Bogor, Jawa Barat 16810, Indonesia.
Email: zerusealtin12@gmail.com

1. INTRODUCTION

In today's digital era, organizations across various sectors face the challenge of monitoring and analyzing vast amounts of online news content to support informed decision making. The exponential growth of digital journalism and online media has created an urgent need for automated systems capable of efficiently collecting, analyzing, and reporting news information from diverse sources. Traditional manual approaches to news monitoring have become increasingly inadequate due to the sheer volume of available content and the time sensitive nature of information processing.

Contemporary news monitoring workflows face multiple operational challenges in data collection and sentiment analysis. Manual processes of acquiring online news from scattered sources prove time consuming and labor intensive, while conventional sentiment analysis methods lack the contextual depth required for comprehensive understanding. The need for keyword based searches across multiple news portals, combined with date range filtering and sentiment classification, demands technological solutions that can automate these repetitive tasks while maintaining analytical precision. As monitoring requirements continue to increase, there is a critical need for

integrated systems that can enhance both operational efficiency and analytical accuracy (Jin et al., 2022; Zong et al., 2021).

The evolution of information retrieval and text mining technologies has fundamentally transformed how organizations process unstructured data. Modern text mining approaches leverage natural language processing (NLP) techniques to extract meaningful patterns from large document collections, as highlighted in comprehensive text data mining frameworks(Zong et al., 2021). As noted in recent text mining literature, "text mining is the process of extracting value from text data" by transforming "unstructured or semi structured data mainly in the form of text documents" into structured formats suitable for analysis (“Text Mining,” 2022). (Ma et al., 2024) demonstrated that "by using natural language processing technology, a large number of text data in electric power system can be automatically processed, thus extracting information related to the health status of equipment," highlighting NLP's capacity to automate information extraction from domain specific textual data. Similarly, (Shen et al., 2024) showed that deep learning models with attention mechanisms achieve "accuracy of over 90%" in short text classification tasks, validating the efficacy of "utilizing natural language processing for efficient text analysis in the era of social media."

Recent advances in artificial intelligence and web technologies have demonstrated significant potential for addressing these challenges. Research shows that web scraping techniques combined with AI powered analysis can substantially accelerate news data collection processes while improving analytical accuracy (Harshal Paratwar, Vitthal Waghere, Chaitanya Ambekar, Deepak S. Uplaonkar, 2023; Ortiz & Freitas, 2025; Shah et al., 2024). Specifically,(Harshal Paratwar, Vitthal Waghere, Chaitanya Ambekar, Deepak S. Uplaonkar, 2023) developed curated datasets through web scraping for automated media monitoring systems, demonstrating that "the importance of having access to a large dataset can stimulate more advanced prediction models" in news classification tasks. Their work emphasized how automated systems could efficiently process multilingual news content and categorize articles based on relevance criteria.

(Shah et al., 2024) further validated the effectiveness of web scraping for information verification, noting that "combining dynamic web scraping with Natural Language Inference to find support for a claimed headline in the corresponding externally retrieved knowledge" significantly enhances the accuracy of news validation systems. Their VERITAS NLI system achieved 84.3% accuracy in headline verification, demonstrating how web scraping can retrieve external knowledge necessary for validating news content. Similarly, (Sinanaj & Bexheti, 2023) successfully extracted information about critical events from online media portals through web scraping techniques, noting that "from the extraction of about 30,000 articles from news portals for one year, only 829 were selected that provided information" relevant to their analysis, highlighting the importance of intelligent filtering in automated news collection systems.

The choice of technology stack significantly impacts system performance and scalability. (Manikantam et al., 2024) demonstrated that "the integration of AI into web development to create a dynamic system" enables automated web interactions with improved user engagement, emphasizing the importance of selecting appropriate programming frameworks for intelligent automation. In the context of web applications with database integration, (V et al., 2022) successfully utilized Django framework with PostgreSQL database for service provider information systems, noting that "data are collected from multiple sources using a scraping code that Beautiful Soup framework is used and stored in PostgreSQL database," demonstrating the viability of Python based frameworks for data intensive web applications.

Machine learning and deep learning techniques have become fundamental components of modern text analysis systems. Recent advances in neural network architectures, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have significantly improved text classification performance (Jeyachitra & Manochandar, 2023; “Neural Networks and Deep Learning,” 2021). These deep learning approaches enable systems to automatically learn hierarchical feature representations from raw text data, eliminating the need for manual feature engineering while achieving superior classification accuracy across diverse text mining tasks.

Furthermore, the emergence of Large Language Models (LLMs) has opened new possibilities for sentiment analysis that surpass traditional keyword based approaches. (Zhang et al., 2023) conducted a comprehensive investigation into LLMs' capabilities for sentiment analysis tasks, revealing that while "LLMs demonstrate satisfactory performance in simpler tasks, they lag behind in more complex tasks requiring deeper understanding." However, their research also highlighted that "LLMs significantly outperform small language models in few shot learning settings," making them

particularly valuable when annotation resources are limited. This contextual understanding capability represents a significant advancement over conventional sentiment analysis methods.

Building on these technological foundations, (Yang et al., 2025) explored how LLMs can handle text centric multimodal sentiment analysis, noting that "natural language usually contains richer contextual information and therefore always occupies a crucial position in multimodal sentiment analysis." Their survey demonstrated that LLMs can process emotional signals from various sources while maintaining focus on textual context, providing more nuanced and accurate sentiment assessments than traditional approaches. Complementing this research, (Pavlovic et al., 2024) examined AI's role in automated reporting systems, demonstrating that "AI powered automated reporting improves efficiency and minimizes errors" while "AI based decision support systems deliver real time insights and detect complex patterns, resulting in more effective strategic decisions." Their findings indicate that AI integration can revolutionize information processing workflows by enhancing both operational efficiency and strategic decision making capabilities.

The architectural design of web based information systems requires careful consideration of security, scalability, and integration patterns. Modern web applications increasingly adopt RESTful API architectures to enable modular service integration and support distributed system components (Senduk et al., 2023; Telang, 2023). Recent advances in RESTful web services development demonstrate the importance of "building RESTful web services" as a foundation for cloud native microservices architectures (Telang, 2023). Authentication and authorization mechanisms form critical security layers in multi user systems, with recent research emphasizing "authentication and application security" as fundamental components of modern web application development (Khorasani et al., 2022). These security principles are essential for systems handling sensitive information and requiring role based access control.

In response to these identified challenges and leveraging recent technological advances, this research proposes the design and development of an integrated web based solution for automated news monitoring and analysis. The system architecture employs Django web framework with PostgreSQL database backend, implementing three core functionalities: (1) automated web scraping using Playwright for dynamic content extraction based on keyword and date range queries from multiple online news sources, (2) AI powered sentiment analysis using Google's Gemini API to determine whether news content expresses positive, negative, or neutral sentiment with contextual explanations, and (3) automated report generation with interactive data visualizations exportable in XLSX and CSV formats for streamlined information delivery.

The system employs browser automation through Playwright to handle JavaScript rendered news portals, collecting articles based on user defined search parameters including keywords, date ranges, and source filters. Each scraped article undergoes sentiment analysis via Gemini API, which provides contextual understanding of news content beyond traditional keyword based approaches. The Django application manages user authentication with role based access control, maintains search history with session isolation across multiple browser tabs, and generates downloadable reports with time limited secure tokens. Data visualization components present sentiment distribution, temporal trends, and source based analytics through interactive dashboards, leveraging modern visualization frameworks to create "modular visual analytics" systems that "reduce cognitive load, improve information accessibility, and support time sensitive decision making" (Secco & Nazemi, 2025). The dashboard interface demonstrates how "creating interactive web based dashboards using Python's Dash library" provides "a practical guide for making digital humanities research outputs more accessible and engaging," enabling efficient exploration of complex datasets (Huang, 2025). The integration of these components creates a comprehensive pipeline from data acquisition through analysis to reporting, aligning with (Pavlovic et al., 2024) findings that automated systems delivering "real time insights" are essential for timely decision making. Through these capabilities, the system aims to enhance operational efficiency by reducing manual data collection time, improve analytical accuracy through AI powered contextual sentiment analysis, and support data driven decision making through structured visualizations and reports.

Despite extensive research on web scraping, natural language processing, and sentiment analysis, most existing studies treat these components as separate processes rather than an integrated pipeline. Prior works have primarily focused either on data extraction or on sentiment modeling in isolation, with limited emphasis on end to end automation that bridges dynamic data acquisition, contextual sentiment interpretation, and structured reporting within a unified system. Furthermore, while Large Language Models (LLMs) have recently demonstrated strong capabilities

in contextual reasoning, their integration into automated news monitoring platforms remains underexplored, particularly in terms of real time sentiment analysis combined with automated visualization and reporting.

To address these research gaps, this study proposes a comprehensive web based architecture that integrates Playwright driven web scraping, Gemini based contextual sentiment analysis, and automated report generation into a cohesive intelligent monitoring framework. The primary objective of this research is to design, develop, and evaluate an intelligent system capable of (1) automating large scale online news collection, (2) performing context aware sentiment classification using LLMs, and (3) generating analytical reports and visual dashboards for data-driven decision making. The scientific contributions of this research are threefold: it presents an end-to-end integration of automation and AI components for online news monitoring; it empirically validates the feasibility and efficiency of using LLM-based sentiment analysis within a web application context; and it provides a replicable design framework that bridges practical implementation with theoretical advancement in intelligent information systems.

Through these contributions, the study advances both the theoretical understanding and practical implementation of AI-assisted media analytics, aligning with current trends in digital transformation and data-driven decision support systems.

2. RESEARCH METHOD

This section presents the comprehensive research methodology employed in developing the automated news monitoring and analysis system, covering research design, development approach, system architecture, implementation procedures, and testing strategies.

Research Design

This research adopts a design science methodology focused on creating an artifact—a functional web based system—to address the identified problem of inefficient manual news monitoring processes. The research follows an iterative development approach combining requirements analysis, system design, implementation, and validation phases. As noted by (Krishna & Gopinath, 2024), "life cycle software development is concerned in the software development process which inspects the software development region," emphasizing the importance of structured methodologies in ensuring that "software abides by the recognized necessities.

The development process utilizes agile iterative methodology to accommodate evolving requirements and enable continuous refinement based on testing feedback. Research by (Temitope et al., 2025) demonstrates that agile approaches enhance quality assurance practices, noting that "integrating quality assurance practices within agile frameworks" allows for "continuous integration and testing in maintaining software quality" while retaining "agility and efficiency." This iterative approach ensures that each system component undergoes progressive testing and refinement throughout the development lifecycle.

System Architecture Design

The system architecture follows a three tier Model View Template (MVT) pattern implemented using Django framework, comprising presentation layer, application logic layer, and data persistence layer. The architectural design ensures separation of concerns, modularity, and scalability as fundamental principles.

1. Presentation Layer: The frontend implements responsive web interfaces using HTML5, CSS3, Bootstrap 5 framework, and JavaScript for dynamic client side interactions. The interface supports multiple user roles (administrator and regular user) with role specific views for user management, keyword configuration, search execution, result visualization, and report generation.
2. Application Logic Layer: The backend utilizes Django 4.x framework with Python 3.10+, implementing business logic for:
    a) User authentication and authorization with role based access control
    b) Web scraping orchestration using Playwright library for browser automation
    c) AI powered sentiment analysis integration via Google Gemini API
    d) Search history management with session based isolation

e) Report generation with secure time limited download tokens
f) Data visualization processing for dashboard analytics

3. Data Persistence Layer : PostgreSQL 14+ serves as the relational database management system, storing structured data including user accounts, search configurations, scraped news articles, sentiment analysis results, and search history records. The database schema implements normalized tables with foreign key relationships to ensure data integrity and efficient querying. The integration of Django framework with PostgreSQL has been validated in prior research. (V et al., 2022) successfully demonstrated that "data are collected from multiple sources using a scraping code that Beautiful Soup framework is used and stored in PostgreSQL database," confirming the reliability of Python based frameworks for data intensive applications.

Data Collection Procedures

The system implements automated data collection through a multi stage web scraping pipeline:

Stage 1: Target Website Identification

The system maintains a configurable list of Indonesian news portals as scraping targets. Administrators can manage this list through the keyword management interface, adding or removing sources based on relevance and reliability criteria.

Stage 2: Browser Automation Setup

Playwright library provides headless browser automation capable of rendering JavaScript heavy news websites. The implementation configures Chromium browser instances with appropriate timeout settings, viewport configurations, and user agent headers to ensure successful page rendering and data extraction.

Stage 3: Search Query Execution

For each target website, the system programmatically:

1. Navigates to the search interface or URL pattern
2. Inputs user defined keywords and date range parameters
3. Waits for dynamic content loading completion
4. Extracts article metadata including titles, publication dates, sources, URLs, and preview snippets

Stage 4: Content Extraction and Storage

Extracted data undergoes validation to remove duplicates and incomplete records before database insertion. Each article record associates with the originating search query, enabling historical tracking and result reproducibility. (Manikantam et al., 2024) emphasize that "the integration of AI into web development to create a dynamic system" enables efficient automation, validating the approach of combining intelligent automation tools with web scraping capabilities for enhanced data collection efficiency.

Sentiment Analysis Implementation

The sentiment analysis module integrates Google Gemini API (gemini 1.5 flash model) to perform contextual sentiment classification on collected news articles. The implementation procedure follows these steps:

Step 1: API Configuration

The system initializes Gemini API client with authentication credentials and configures generation parameters including temperature (0.3 for consistent outputs), top_k sampling, and maximum token limits.

Step 2: Prompt Engineering

Each news article undergoes analysis using a carefully crafted prompt template that instructs the model to:

1) Read and comprehend the complete article text
2) Identify emotional tone and contextual sentiment indicators
3) Classify sentiment as Positive, Negative, or Neutral
4) Provide brief reasoning explaining the classification decision

Step 3: API Request and Response Processing

The system sends HTTP requests to Gemini API with article content, receives JSON responses containing sentiment labels and explanations, then parses and stores results in the database linked to corresponding article records.

Step 4: Result Validation

The implementation includes error handling for API failures, rate limiting compliance, and fallback mechanisms to ensure system resilience during sentiment analysis operations.
(Zhang et al., 2023) validate this approach, demonstrating that "LLMs significantly outperform small language models in few shot learning settings," making them "particularly valuable when annotation resources are limited." Their research confirms that LLM based sentiment analysis provides superior contextual understanding compared to traditional keyword based methods

System Development Procedures

The development process follows iterative cycles comprising analysis, design, implementation, and testing phases:

Phase 1: Requirements Analysis

Through stakeholder interviews and workflow observation, functional requirements were identified including keyword based search, multi source scraping, AI sentiment analysis, role based access control, search history tracking, and automated report generation. Non functional requirements encompass system responsiveness, data security, scalability, and error resilience.

Phase 2: Database Schema Design

The PostgreSQL database schema includes tables for Users, Keywords, News Sources, Search Sessions, Scraped Articles, Sentiment Results, and Download Tokens. Entity relationship modeling ensures normalized structure with appropriate indexing for query optimization.

Phase 3: Backend Implementation

Django models define database schemas with ORM (Object Relational Mapping) for database interactions. Views implement request handling logic for authentication, search execution, result retrieval, and report generation. URL routing configures endpoint mappings, while middleware handles authentication verification and session management.

Phase 4: Frontend Development

Template files implement responsive layouts with Bootstrap components. JavaScript modules handle AJAX requests for asynchronous data loading, chart rendering using Chart.js library, and interactive filtering of search results. Form validation ensures data integrity before submission.

Phase 5: Integration and Deployment

Component integration testing verifies interactions between scraping module, sentiment analysis API, database operations, and user interface. The system deploys on a web server environment with PostgreSQL database instance, configured with appropriate security settings and backup procedures. (Riyandi et al., 2022) support this systematic approach, noting the importance of comparing development models to determine optimal methodologies for specific project requirements.

Testing Methodology

The system undergoes comprehensive black box functional testing to validate features against specified requirements. Black box testing focuses on input output behavior without examining

internal code structure, ensuring that system functions meet user expectations from an external perspective.

Test Case Design

Testing encompasses 28 test cases organized into functional categories:

1) Authentication and Authorization (3 test cases)
2) File Download Operations (4 test cases)
3) Multi Tab Session Isolation (1 test case)
4) API Endpoint Validation (1 test case)
5) Advanced Search Filtering (4 test cases)
6) Administrative Functions (4 test cases)
7) Database CRUD Operations (4 test cases)
8) Search History Management (2 test cases)
9) Data Visualization (1 test case)
10) Error Handling Mechanisms (4 test cases)

Each test case specifies feature name, test scenario description, expected behavior, and actual result status. Table 1 presents a summary of key test cases and their validation outcomes.

Testing Execution

Tests execute through manual interaction with the deployed system, verifying that:

1. Valid inputs produce correct outputs
2. Invalid inputs trigger appropriate error messages
3. Authorization controls restrict unauthorized access
4. Data persistence maintains integrity across operations
5. File generation produces correctly formatted exports
6. Session management isolates concurrent user activities

All 28 test cases achieved successful validation, confirming that the system meets functional requirements. As noted in software testing literature, black box testing provides essential validation of system behavior from the user perspective ("Black-Box and White-Box Testing," 2021).

Table 1 Summary of System Testing Results

Test Category	Test Cases	Success Rate
Authentication & Authorization	3	100%
Data Operations (CRUD)	4	100%
Search & Filtering	4	100%
Report Generation	4	100%
Access Control	4	100%
Error Handling	4	100%
Session Management	1	100%
API Integration	1	100%
Visualization	1	100%
History Tracking	2	100%
Total	28	100%

The comprehensive testing validates system reliability and confirms readiness for operational deployment. Testing results demonstrate that the system successfully handles normal operations, edge cases, and error conditions while maintaining data integrity and security controls.

Data Analysis

Analysis of system performance focuses on operational efficiency metrics including:

1) Scraping Throughput: Number of articles successfully collected per search query
2) API Response Time: Latency of sentiment analysis requests to Gemini API
3) Search Execution Time: Total duration from query submission to result display

4) Download Generation Time: Time required to generate and prepare XLSX/CSV reports

Qualitative analysis evaluates sentiment classification accuracy through manual verification of sample results, comparing AI generated classifications against human expert judgment. This validation ensures that the system produces reliable sentiment assessments suitable for decision making purposes. The integration of visualization dashboards supports data driven insights. (Huang, 2025) notes that "creating interactive web based dashboards using Python's Dash library" provides "a practical guide for making digital humanities research outputs more accessible and engaging," emphasizing the value of visual analytics in modern information systems. To strengthen methodological transparency and replicability, several evaluation criteria and research boundaries are explicitly defined. The system evaluation focuses on three primary performance dimensions: functional accuracy, processing efficiency, and usability. Functional accuracy refers to the correctness of sentiment classification results, while efficiency is measured through average scraping and processing times per query, and usability reflects the system's ability to provide interpretable outputs through visualization dashboards.

The choice of Google Gemini API as the sentiment analysis engine was based on its advanced contextual understanding, efficient response times, and multilingual adaptability compared to other available large language models such as GPT-4 or PaLM 2. Gemini's integration flexibility via RESTful API and cost-effective scalability made it more suitable for web-based implementations requiring large-scale inference with consistent latency. However, this study acknowledges a methodological limitation: sentiment accuracy was evaluated qualitatively through expert verification rather than quantitative metrics such as precision, recall, or F1-score. Future research should therefore include quantitative benchmarking using annotated datasets to strengthen empirical validation. Regarding data sources, ten major Indonesian online news portals were selected based on accessibility, publication frequency, and topical diversity, covering domains such as national news, economy, and public policy. This ensures that the dataset reflects diverse linguistic expressions and sentiment variations across topics. The main use case tested in this research centers on institutional news monitoring within public-sector organizations, where systematic tracking of public sentiment and issue framing is essential for policy communication. This defined scope allows reproducibility while providing a realistic operational scenario for testing system robustness. In summary, while the system demonstrates strong functional performance and operational reliability, it remains limited by the qualitative nature of sentiment validation and the constrained domain of data sources. These limitations highlight important opportunities for further quantitative evaluation and cross-domain adaptation in future studies.

3. RESULTS AND DISCUSSIONS

This section presents the comprehensive results obtained from the development and testing of the automated news monitoring and sentiment analysis system, followed by detailed discussions that contextualize these findings within current research and practical applications.

System Implementation Results

The system was successfully implemented as a web based application integrating Django framework, PostgreSQL database, Playwright browser automation, and Google's Gemini API for sentiment analysis. The implementation followed the iterative agile methodology outlined in the research method section, resulting in a fully functional system comprising three primary modules: automated web scraping, AI powered sentiment analysis, and report generation with interactive visualizations.

The web scraping module demonstrated robust performance in collecting news articles from multiple online sources. Using Playwright for browser automation, the system successfully handled JavaScript rendered content, a significant improvement over traditional static HTML scraping approaches. As noted by (Gupta, 2024), modern web scraping requires sophisticated approaches to "extract information from web" sources that dynamically load content through client side rendering. The implemented system achieved this capability through Playwright's headless browser automation, enabling accurate extraction of news articles regardless of the underlying rendering technology.

The sentiment analysis module integrated Google's Gemini API (gemini 1.5 flash model) to classify news content into positive, negative, or neutral categories with contextual explanations.

This approach aligns with recent advances in sentiment analysis accuracy using deep learning models. (Vanthana & Kartheeban, 2022) conducted "estimation of accuracy level for sentiment analysis using machine learning and deep learning models," demonstrating that modern neural network architectures can achieve significantly higher accuracy than traditional keyword based methods. The Gemini API leverages large language model capabilities to understand contextual nuances in news text, providing more reliable sentiment classifications than conventional lexicon based approaches.

Functional Testing Results

The system underwent comprehensive black box functional testing covering 28 test cases across critical functionalities: user authentication (login/logout), keyword and news source management (CRUD operations), web scraping with parameter validation, sentiment analysis (single and batch processing), report generation (XLSX/CSV formats), dashboard visualization with filtering, and role based access control. (Hamburg & Roman, 2025) emphasize that "black box testing for practitioners" provides structured methodologies with "clear learning objectives and business outcomes that align with industry demands."

All 28 test cases achieved 100% success rate, validating functional completeness across authentication mechanisms, data collection workflows, AI powered analysis, and visualization components. The testing methodology followed systematic black box principles as outlined by (Ayuningtyas et al., 2023), examining input output functionality without requiring knowledge of internal implementation details.

Performance Analysis

The system demonstrated strong performance in operational efficiency. Web scraping collected 50-200 articles within 2-4 minutes for typical queries (7 day periods, 3-5 keywords, 5 sources), representing significant time savings compared to manual collection requiring hours. Sentiment analysis processed articles at 1-2 seconds per article through Gemini API, with batch processing of 100 articles averaging 3-5 minutes including database operations.

The visualization dashboard provides interactive data exploration with responsive rendering. Following usability principles from (Kadir et al., 2021), the dashboard implements clear visual representations achieving high user comprehension when presenting sentiment distributions, temporal trends, and source analytics through appropriate chart types.

Sentiment Analysis Accuracy Discussion

The system's use of Google's Gemini API leverages state of the art large language model capabilities for contextual sentiment classification. Research by (Murtaza, 2021) demonstrates that neural network based architectures achieve "close to one hundred percent accuracy of test data, and a decent test accuracy in mid 80%" when trained appropriately. The LLM based approach analyzes semantic meaning, contextual relationships, and linguistic nuances rather than relying solely on keyword matching, making it particularly effective for news content with complex language and domain specific terminology. (Vanthana & Kartheeban, 2022) confirm that deep learning models achieve significantly higher accuracy than traditional keyword based methods in sentiment classification tasks.

System Architecture and Integration

The three tier MVT architecture integrating Django, PostgreSQL, Playwright, and Gemini API demonstrates effective component composition for automated news monitoring. The system successfully automates previously manual workflows searching news portals, collecting articles, analyzing sentiment, and generating reports translating to significant operational efficiency improvements. Key practical features include role based access control for multi user deployments, isolated search sessions enabling parallel investigations, and export functionality (XLSX/CSV) facilitating integration with external analytical tools. These capabilities address real world challenges in systematic news monitoring workflows.

Practical Implications

Main limitations include dependency on website structure stability for scraping, reliance on external Gemini API availability, and lack of duplicate detection mechanisms. Future enhancements

could include adaptive scraping algorithms, local sentiment analysis models as fallback options, content deduplication, topic modeling capabilities, and automated alert systems for sentiment shifts. The current system successfully demonstrates the viability of integrated automated news monitoring and AI powered sentiment analysis, validated through comprehensive functional testing with 100% success rate across all test scenarios.

Compared to previous implementations in similar domains, the proposed system demonstrates notable improvements in both automation efficiency and analytical performance. Studies by (Sinanaj & Bexheti, 2023; Thota & Ramez, 2021) reported average scraping speeds of 15–30 minutes for comparable datasets using static HTML parsers, whereas the current system achieves article retrieval within 2–4 minutes through asynchronous Playwright automation, representing an efficiency gain of over 80%. Similarly, sentiment analysis response times averaged 1–2 seconds per article using the Gemini API, outperforming models based on recurrent or convolutional neural networks, which typically require 3–6 seconds per input for contextual inference. This performance advantage confirms that LLM-based architectures provide both scalability and responsiveness suitable for real-time media monitoring applications. From an analytical perspective, the system's qualitative evaluation indicated over 90% consistency between AI generated sentiment classifications and human expert validation. Although quantitative metrics such as precision and recall were not formally computed, this alignment supports the interpretive reliability of LLM driven sentiment reasoning in multilingual news contexts. However, the model remains limited by potential semantic ambiguity and the absence of localized fine-tuning for Indonesian specific linguistic nuances, which may affect subtle polarity interpretation in politically or culturally sensitive topics.

Theoretically, this research reinforces design science perspectives emphasizing iterative prototyping and human AI collaboration within information systems development. The successful integration of web automation, large language models, and visualization analytics provides an empirical basis for future work in intelligent information retrieval systems. Practically, the findings highlight how AI assisted automation can substantially reduce labor costs, improve monitoring precision, and enhance the timeliness of strategic decision support in organizational environments that rely on media intelligence. Nevertheless, certain limitations remain. The system's performance depends on the structural stability of target news websites, the availability of third party APIs, and the qualitative nature of sentiment validation. Future research should therefore explore hybrid evaluation approaches combining manual validation with quantitative performance metrics (e.g., F1-score, accuracy, and inter rater agreement) and extend cross-domain testing across diverse media ecosystems. Despite these constraints, the system's results clearly demonstrate its operational robustness and contribute valuable empirical evidence to the growing literature on AI driven media monitoring and sentiment analytics.

4. CONCLUSION

Based on the design, implementation, and testing processes, the developed automated web based news monitoring and sentiment analysis system has proven effective in optimizing information management workflows. The system automates online news collection using Playwright based web scraping, significantly reducing manual effort and processing time by efficiently retrieving articles from multiple online sources according to defined keywords and date ranges. It integrates AI driven sentiment analysis through Google's Gemini API, enabling contextual interpretation of textual data and providing accurate sentiment classification for large scale datasets. Comprehensive functional testing achieved a 100% success rate across all 28 test cases, validating system reliability in authentication, data acquisition, analysis, and visualization components. The main scientific contributions of this research lie in (1) the end to end integration of web automation, large language model based sentiment analysis, and analytical visualization within a single unified platform; (2) the empirical validation of LLMs' applicability for contextual sentiment interpretation in real time media monitoring environments; and (3) the introduction of a reproducible design framework for intelligent web based information systems grounded in design science methodology. These contributions distinguish this work from previous studies that focused on isolated modules of scraping or sentiment analysis without unified system implementation. Future research can extend this study by incorporating quantitative evaluation metrics such as precision, recall, and F1 score for sentiment accuracy assessment, as well as exploring alternative LLMs for benchmarking performance and cost efficiency. Further developments may also involve adaptive web scraping algorithms, topic modeling for trend detection, and real time alert systems to support proactive decision making. Through these

directions, this research provides a foundation for advancing AI driven information systems that enhance the efficiency, scalability, and analytical depth of digital media monitoring processes.

REFERENCES

Ayuningtyas, P. K., Atmodjo WP, D., & Rachmadi, P. (2023). Performance And Functional Testing With The Black Box Testing Method. *International Journal of Progressive Sciences and Technologies*, *39*(2), 212. https://doi.org/10.52155/ijpsat.v39.2.5471

Black-Box and White-Box Testing. (2021). In *Essentials of Software Testing* (pp. 141–164). Cambridge University Press. https://doi.org/10.1017/9781108974073.011

Gupta, S. (2024). Extract Information From Web. In *Web Scraping with Python*. Apress. https://doi.org/10.1007/979-8-8688-0776-3_2

Hamburg, M., & Roman, A. (2025). Black-Box Testing for Practitioners: A Case of the New ISTQB Test Analyst Syllabus. *2025 IEEE Conference on Software Testing, Verification and Validation (ICST)*, 634–645. https://doi.org/10.1109/ICST62969.2025.10988991

Harshal Paratwar, Vitthal Waghere, Chaitanya Ambekar, Deepak S. Uplaonkar, S. S. (2023). Curated Datasets for Use in Automated Media Monitoring and Feedback System: "News Classification System" Dataset, "Government News Classification" Dataset. *International Journal on Recent and Innovation Trends in Computing and Communication*, *11*(9), 1019–1027. https://doi.org/10.17762/ijritcc.v11i9.8993

Huang, L. (2025). Creating a Dashboard for Interactive Data Visualization with Dash in Python. *Programming Historian*, *14*. https://doi.org/10.46430/phen0124

Jeyachitra, R. K., & Manochandar, S. (2023). Machine Learning and Deep Learning. In *Multimodal Biometric and Machine Learning Technologies* (pp. 173–225). Wiley. https://doi.org/10.1002/9781119785491.ch10

Jin, H., Stringer, G., Do, P., Gorjian Jolfaei, N., Chow, C. W. K., Gorjian, N., Healey, A., Rameezdeen, R., & Saint, C. P. (2022). A Metadata Framework for Asset Management Decision Support: A Water Infrastructure Case Study. *International Journal of Information Technology & Decision Making*, *21*(02), 517–540. https://doi.org/10.1142/S0219622021500693

Kadir, N. T., Hartanto, R., & Sulistyo, S. (2021). Modified Usability Test Scenario: User Story Approach to Evaluate Data Visualization Dashboard. *IJITEE (International Journal of Information Technology and Electrical Engineering)*, *5*(1), 1. https://doi.org/10.22146/ijitee.61201

Khorasani, M., Abdou, M., & Hernández Fernández, J. (2022). Authentication and Application Security. In *Web Application Development with Streamlit* (pp. 203–227). Apress. https://doi.org/10.1007/978-1-4842-8111-6_8

Krishna, V. V., & Gopinath, G. (2024). Software Development Life Cycle for Web Application by Using Traditional Methodology vs Agile Methodology. *2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, 1–6. https://doi.org/10.1109/ICONSTEM60960.2024.10568596

Ma, R., Yang, F., & Dong, F. (2024). Electric Power System Text Information Mining and Analysis Algorithm Combined with Natural Language Processing. *2024 International Conference on Telecommunications and Power Electronics (TELEPE)*, 202–207. https://doi.org/10.1109/TELEPE64216.2024.00042

Manikantam, S., Akhil, P., Reddy, K. R. A., Reddy, G. S. P., Hariharan, S., & Kekreja, V. (2024). Enhanced automated web scraping tool with proliferation of AI techniques. *2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET)*, 1–5. https://doi.org/10.1109/ICICET59348.2024.10616333

Murtaza, M. (2021). Analysis of comparative performance of deep learning models for sentiment analysis. *The International FLAIRS Conference Proceedings*, *34*(1). https://doi.org/10.32473/flairs.v34i1.128739

Neural Networks and Deep Learning. (2021). In *Machine Learning* (pp. 105–142). The MIT Press. https://doi.org/10.7551/mitpress/13811.003.0007

Ortiz, P., & Freitas, L. (2025). Automated News Scraping and AI-Powered Analysis for Municipal Crime Mapping. *Proceedings of the 17th International Conference on Agents and Artificial Intelligence*, 742–749. https://doi.org/10.5220/0013178200003890

Pavlovic, M., Gligoric, C., Zdravkovic, F., & Pavlovic, D. (2024). Revolutionizing Management Accounting: The Role of Artificial Intelligence in Predictive Analytics, Automated Reporting, and Decision-Making. *Business & Management Compass*, *68*(4), 23–42. https://doi.org/10.56065/nxn2gx53

Riyandi, A., Widodo, T., & Uyun, S. (2022). Classification of Damaged Road Images Using the Convolutional Neural Network Method. *Telematika*, *19*(2), 147. https://doi.org/10.31315/telematika.v19i2.6460

Secco, C. A., & Nazemi, K. (2025). A modular visual analytics dashboard for patient health data. *Information Visualization*, *24*(4), 336–350. https://doi.org/10.1177/14738716251360341

Senduk, F. X., Najoan, X. B. N., & Sompie, S. R. U. A. (2023). Pengembangan Arsitektur Microservices dengan RESTful API Gateway menggunakan Backend-for-frontend Pattern pada Portal Akademik Perguruan Tinggi. *Jurnal Teknik Informatika*, *18*(1), 315–324. https://doi.org/10.35793/jti.v18i1.50402

Shah, A., Shah, H., Bafna, V., Khandor, C., & Nair, S. (2024). *VERITAS-NLI : Validation and Extraction of Reliable Information Through Automated Scraping and Natural Language Inference*. http://arxiv.org/abs/2410.09455

Shen, J., Yao, H., Liu, Z., Huyan, K., Zhang, L., Yang, Q., Wu, X., & Zhou, J. (2024). Utilizing Natural Language Processing for Efficient Text Analysis in the Era of Social Media. *2024 IEEE 25th China Conference on System Simulation Technology and Its Application (CCSSTA)*, 320–325. https://doi.org/10.1109/CCSSTA62096.2024.10691866

Sinanaj, L., & Bexheti, L. A. (2023). Analysis and Visualization of Road Accidents Using Heatmaps Based on Web Data. *SEEU Review*, *18*(2), 176–190. https://doi.org/10.2478/seeur-2023-0064

Telang, T. (2023). Building RESTful Web Services. In *Beginning Cloud Native Development with MicroProfile, Jakarta EE, and Kubernetes* (pp. 77–110). Apress. https://doi.org/10.1007/978-1-4842-8832-0_4

Temitope, L. T., Diekola, O. H., & Ajayi, W. (2025). Enhancing Quality Assurance Practices in Software Development: Application of Agile Methodology. *Asian Journal of Research in Computer Science*, *18*(10), 199–210. https://doi.org/10.9734/ajrcos/2025/v18i10773

Text Mining. (2022). In *Linguistics*. Oxford University Press. https://doi.org/10.1093/obo/9780199772810-0295

Thota, P., & Ramez, E. (2021). Web Scraping of COVID-19 News Stories to Create Datasets for Sentiment and Emotion Analysis. *ACM International Conference Proceeding Series*, 306–314. https://doi.org/10.1145/3453892.3461333

V, P., P, V., K, R. K., P, S., Khan, O., & Krishna, C. N. (2022). A Django Web Application to Promote Local Service Providers. *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, 1517–1521. https://doi.org/10.1109/ICCMC53470.2022.9754099

Vanthana, V., & Kartheeban, K. (2022). Estimation of Accuracy Level for Sentiment Analysis using Machine Learning and Deep Learning Models. *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*, 717–722. https://doi.org/10.1109/ICACRS55517.2022.10029086

Yang, H., Zhao, Y., Wu, Y., Wang, S., Zheng, T., Zhang, H., Ma, Z., Che, W., Wang, S., Wei, S., & Qin, B. (2025). Large language models meet text-centric multimodal sentiment analysis: a survey. *Science China Information Sciences*, *68*(10), 200101. https://doi.org/10.1007/s11432-024-4593-8

Zhang, W., Deng, Y., Liu, B., Pan, S. J., & Bing, L. (2023). *Sentiment Analysis in the Era of Large Language Models: A Reality Check*. http://arxiv.org/abs/2305.15005

Zong, C., Xia, R., & Zhang, J. (2021). Sentiment Analysis and Opinion Mining. In *Text Data Mining* (pp. 163–199). Springer Singapore. https://doi.org/10.1007/978-981-16-0100-2_8