



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XII **Month of publication:** December 2025

DOI: <https://doi.org/10.22214/ijraset.2025.75971>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

GovPulse AI-powered News Intelligence and Sentiment Alert System

Arnav Singh, Ayush, Sarthak Jha, Devjeet Sahoo, Dr.MonikaArora

Department of Artificial Intelligence and Data Science Bhagwan Parshuram Institute of Technology Rohini, sector-17, New Delhi-110089

Abstract: Identifying biases, sentiments, and relevance for efficient governance has become difficult due to the rapid expansion of digital news across numerous platforms. Conventional approaches don't have automated systems to group news by government agencies or to quickly draw attention to important issues. Additionally, the variety of regional languages makes timely decision-making and extensive monitoring more difficult. An automated framework for digital news crawling, classification, and sentiment analysis with an integrated feedback system is presented in this work. The framework gathers videos and articles from various national and regional media sources, uses machine learning models to categorize them into their respective ministries based on the content, and uses natural language processing for sentiment analysis. Real-time notifications to the relevant departments are triggered by negative news items, allowing for prompt intervention. Direct links to original sources, department-wise filters, sentiment visualization, and multilingual support are all features of an intuitive interface. Future developments will involve implementing the system as a mobile application, adding more regional languages, and enhancing model accuracy with larger datasets. This strategy helps government agencies make timely and well-informed policy decisions while raising public awareness, which promotes better governance and social cohesion.

Keywords: NLP, sentiment analysis, classification, news crawling, and multilingual processing

I. INTRODUCTION

With the rapid growth of online news platforms, an enormous amount of information is continuously published across national, regional, and local media. While this stream of digital news offers valuable insights into public sentiment and emerging developments, it also creates significant challenges for real-time monitoring. Manually reviewing such large volumes of content is slow, inconsistent, and prone to missing early signals that may require timely attention [20]. As a result, important issues can go unnoticed, ultimately impacting the effectiveness of governance, policy responses, and public trust.

To address these challenges, *GovPulse* introduces an AI-driven system designed to automatically monitor and analyze governance-related news. The system uses the Scrapy framework to fetch news articles from selected online sources, classify them into their respective government ministries, and conduct sentiment analysis to determine if the coverage is positive, neutral, or negative. If a news item is flagged as negative or questionable, the system automatically alerts the relevant department for appropriate action at the right time. Additionally, *GovPulse* provides an interactive web interface where users can easily explore, filter, and interpret categorized news in a clear and intuitive manner [21].

1) Key Contributions of this Work:

- Extract news articles using the Scrapy framework, for efficient data gathering on a large scale [9].
- Classification of the governance-related articles to the correct government ministries by using appropriate supervised learning and clustering techniques [22]. Sentiment analysis is supported by transformer-based models like DistilBERT for accurate polarity detection [3].
- Real-time alert mechanism sends notifications via email to concerned ministries when negative news is detected.
- Web-based dashboard that provides sentiment filtering, visual analytics, and smooth integration of the front-end with the back-end.

2) Motivation and Significance:

- It is very inefficient to monitor such large volumes of digital news manually and ascertain the need for automated tools.
- Structured and sentiment-labeled news enables objective analysis and reduces dependence on subjective interpretation.
- Turning raw news into actionable insight strengthens transparency, responsiveness, and accountability of governance [23].

Overall, *GovPulse* aims to make public information monitoring more efficient by delivering timely, sentiment-aware, and ministry-specific insights. Its modular design also makes the system easy to extend, whether by adding support for more regional languages, integrating advanced multilingual models, or deploying it as a mobile application in the future.

II. LITERATURE REVIEW

Research in the fields of automated news monitoring, text categorization, and sentiment analysis has evolved significantly in recent years. A variety of machine learning and deep learning techniques have been explored to extract meaningful insights from digital media. However, gaps remain in the areas of multilingual processing, real-time alerts, and integration with governance systems. This section reviews key related works.

Patro et al. (2020) [1] The authors developed a real-time news classification framework using supervised machine learning algorithms such as Naïve Bayes, Random Forest, and Support Vector Machines. Their system was effective in classifying news into high-level categories like sports, politics, and business. However, the approach was limited by scalability challenges and was not designed to handle multilingual data streams.

Zhu (2021) [2] This work investigated the use of Convolutional Neural Networks (CNNs) for large-scale news text classification. By combining feature weighting and hashing techniques, the model achieved better accuracy than traditional ML methods. Nonetheless, CNN architectures faced limitations in processing long text dependencies and could not adequately address multilingual contexts.

Bu'yu'ko'zet al. (2020) [3] The study compared ELMo and DistilBERT for socio-political news classification. Results demonstrated that transformer-based models outperformed earlier neural networks in contextual understanding. While promising, the research was restricted to English-only datasets, leaving unanswered questions about performance in multilingual and low-resource language settings.

Valmiki and Ambili (2023) [4] The author highlighted the role of machine learning in analyzing and predicting sentiment from news data. Their experiments with modern algorithms showed improvements in accuracy but pointed out challenges in generalizing across diverse populations and activity domains. These findings underscore the importance of building robust, domain-adapted multilingual sentiment systems.

Yadav (2015) [5] This survey paper focused on sentiment analysis techniques for Hindi text, discussing both lexicon-based and supervised methods. It revealed a lack of annotated corpora and standardized benchmarks for Indian languages. This resource scarcity remains a critical barrier to developing reliable multilingual sentiment models.

Gupta et al. (2021) [6] The authors conducted a systematic review of news classification methods, comparing traditional ML, deep learning, and transformer-based approaches. They identified issues such as interpretability, dataset imbalance, and difficulty in real-world deployment. The study emphasized the need for scalable architectures that combine classification, sentiment detection, and practical deployment features.

Patel and Sharma (2019) [7] This paper introduced a hybrid technique for web page classification on news feeds, focusing on extraction and categorization. While efficient for structured content, it struggled with unstructured data and lacked multilingual capabilities. Integrating such techniques with advanced AI models could enhance robustness and applicability.

Overall, the reviewed literature demonstrates strong progress in text analytics but highlights persistent challenges in multilingual support, real-time alerting, and government-specific applications. *GovPulse* addresses these gaps by combining scalable news crawling, transformer-based classification, sentiment analysis, and instant feedback mechanisms into one integrated system.

III. PROPOSED WORK

The proposed work of the project, *GovPulse – AI-powered News Intelligence and Sentiment Alert System*, is to build an end-to-end framework capable of automatically collecting, classifying, and analyzing digital news. The system is designed to handle both static and dynamic web pages, automatically categorize news articles under the appropriate government ministries, determine their sentiment, and deliver actionable insights in real time [24].

News articles are collected from a variety of online platforms, including e-newspapers and web-based news portals. Large-scale extraction of static content is handled using the *Scrapy* framework [9], whereas websites that require dynamic rendering are processed through Selenium [10]. For video-based sources, the spoken content is converted into text using available captions or speech-to-text transcription methods.

Thereafter, the data is filtered through a preprocessing pipeline that cleans and normalizes the text of articles. The refined articles then pass through transformer-based models like DistilBERT for ministry classification.

Next, sentiment analysis is done on them, categorizing each article as positive, neutral, or negative. Articles classified as strongly negative automatically trigger an alert mechanism to send notifications to the relevant government departments.

The processed articles, with their sentiment scores and classification, are then visualized on a React.js and TailwindCSS-based dashboard [18], [19].

The developed dashboard supports filtering by ministry and sentiment, making it easy to view and draw conclusions from the news under analysis.

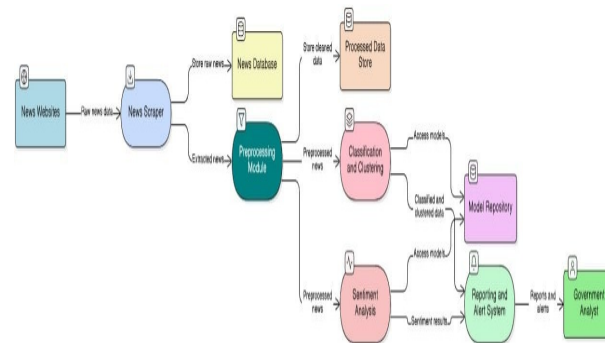


Fig.1: System Architecture of GovPulse

Figure 1 illustrates the proposed architecture, showing the flow from data collection to preprocessing, classification, sentiment analysis, and final visualization.

IV. METHODOLOGY

The methodology of GovPulse is structured into several sequential stages, ensuring the transition from raw news data to actionable insights.

- 1) **Data Acquisition:** News articles and videos are collected using automated crawlers. Scrapy [9] is the core framework for scalable crawling of static and dynamic pages. For video content, captions and audio transcripts are extracted using speech-to-text models. All collected data is stored in structured CSV/JSON formats.
- 2) **Preprocessing and Translation:** Preprocessing includes removal of duplicates, tokenization, stopword removal, and lemmatization [25]. Non-English content is translated into English using IndicTrans [29] or Google Translate. This ensures consistency across multilingual sources.
- 3) **Embeddings and Clustering:** Sentence embeddings are generated to capture semantic meaning of articles. Dimensionality reduction using UMAP [11] enhances efficiency before clustering with algorithms such as K-means, DBSCAN [12], or HDBSCAN [13]. Clusters are assigned to ministries based on frequent keywords, generating labeled datasets for supervised training.
- 4) **Supervised Classification:** Using the labeled dataset, transformer models (DistilBERT, XLM-R [30]) are fine-tuned to classify unseen articles into ministries. Classical models such as Random Forest and Linear SVM are used as baselines for comparison [14].
- 5) **Sentiment Analysis:** Pretrained Roberta [15] and DistilBERT [3] sentiment classifiers are applied to assign polarity scores. Each article is categorized into Positive, Neutral, or Negative with associated probabilities [26].
- 6) **Alert Mechanism:** Negative articles automatically trigger the alert module, which sends structured notifications through Gmail SMTP or Nodemailer to the relevant ministries.
- 7) **Visualization:** Results are displayed on a React.js + TailwindCSS dashboard that fetches predictions via Django REST APIs [17]. Features include filtering, sentiment analytics, and auto-refresh.

Figure 2 presents the complete methodology, covering scraping, preprocessing, classification, sentiment analysis, and dashboard delivery.

V. IMPLEMENTATION

Implementation of the GovPulse—AI-powered News Intelligence and Sentiment Alert System follows a modular pipeline with well-defined stages. In this work, there is seamless integration of automated web scraping, multilingual preprocessing, clustering, classification, sentiment analysis, and visualization of the project. Major components are in Python, where Scrapy is employed for data acquisition, HuggingFace

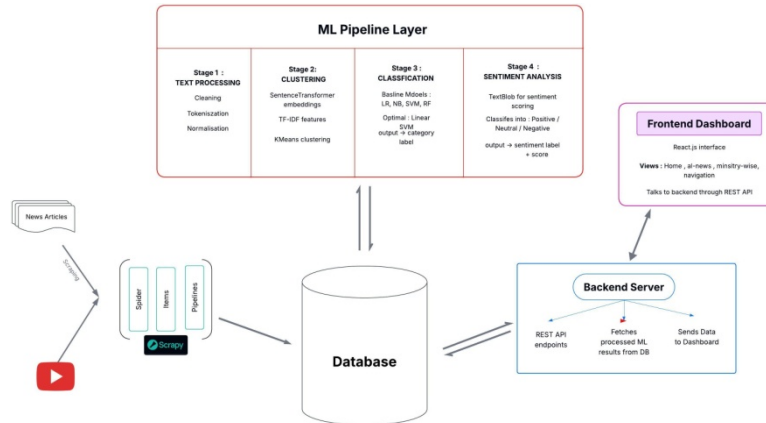


Fig.2: Workflow of the GovPulse Pipeline

Transformers[8] provided deep learning, and Django REST APIs enable frontend integrations.

A. Data Collection

GovPulse utilizes two complementary datasets for real-time monitoring and model development. The first is a **real-time scraped** dataset, collected using the Scrapy framework and supplemented with YouTube video transcripts and captions retrieved through YouTube APIs. This dataset contains detailed attributes such as Author, Category, Content, Article Date, Headline, Image URL (when available), Keywords, Source, Summary, URL, and WordCount. These records form the live input for ministry classification, sentiment analysis, and alert generation.

The second is a historical dataset composed of archived news articles with fields including Heading, Content, and URL, with 7645 rows. During the development workflow, this dataset is enhanced by assigning Category labels through clustering and adding Sentiment tags after training transformer-based models. The resulting classification and sentiment models are then applied to the real-time dataset to ensure accurate and consistent analysis of incoming news. This dual-dataset approach supports robust model training while enabling comprehensive and timely monitoring of both written and video-based news content.

B. Data Storage

GovPulse uses MongoDB as its primary data storage layer owing to its scalability, schema flexibility, and suitability for semi-structured news content[27]. Given that scraped articles, transcripts, and model outputs vary significantly in length and structure, a document-oriented NoSQL model offers a more flexible and efficient representation than rigid relational schemas.

The database consists of four core collections—raw_articles, clean_articles, media_assets, and model_outputs—supporting the complete ML pipeline. The raw_articles collection stores unprocessed data scraped from national and

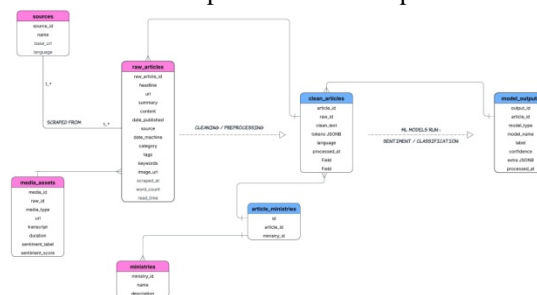


Fig.3: ER diagram of the GovPulse database schema

regional news portals, including headlines, article text, timestamps, metadata, and source information. After text preprocessing and normalization, refined content is stored in clean_articles, preserving a clear boundary between raw and processed data. Media assets such as YouTube transcripts and video-linked media are maintained in media_assets. Final model outputs—classification labels, sentiment predictions, confidence scores, and processing timestamps—are stored in model_outputs and linked to corresponding article identifiers.

The storage layer is optimized using indexes on frequently queried fields, including source, date_published, ministry, and sentiment_label, enabling low-latency dashboard queries and alerting. MongoDB's aggregation pipelines further support efficient computation of ministry-level article counts, sentiment distributions, and temporal trends. Collectively, this architecture forms a scalable and resilient backbone for GovPulse, enabling real-time ingestion alongside analytical workloads.

C. Web Scraping and Automation

The scraping process is automated using the Scrapy-based *Scraper* module, with multiple spiders crawling different news domains in parallel. Key features include:

- 1) Asynchronous large-scale crawling: The scraper uses Twisted's event-driven architecture to fetch high volumes of articles efficiently. Dedicated spiders handle national sources (Times of India, NDTV, Indian Express) and regional sources (The Telegraph), each tailored to their DOM structures.
- 2) YouTube transcript extraction: The `ytvideo_spider` spider uses the YouTube Data API to collect auto-generated or manual captions from video-based sources.
- 3) Standardized data schema: The schema defined in `items.py` ensures consistent output fields before data is exported to CSV or inserted into MongoDB.
- 4) Data cleaning pipelines: The pipeline module (`pipelines.py`) removes HTML tags, scripts, and advertisements while extracting structured fields such as headline, body, URL, date, and source.
- 5) Robust middleware: Custom middleware (`middlewares.py`) manages request retries, user-agent rotation, throttling via `DOWNLOAD_DELAY`, and optional proxy rotation for stable crawling [9].
- 6) Automated scheduling: Recurring scraping tasks run via `cronjobs` or Scrapy's `CrawlerProcess scheduler`, enabling periodic (hourly/daily) extraction without manual intervention.

Dynamic, JavaScript-rendered pages are handled using Selenium or Playwright. For video-based news, speech-to-text models are used when transcripts are unavailable, ensuring multimodal coverage.

D. Preprocessing

After scraping, the collected text undergoes a structured preprocessing pipeline to ensure it is suitable for downstream machine learning tasks. The objective of this stage is to clean, normalize, and standardize the raw data so that the models can effectively capture semantic and contextual patterns [25].

The preprocessing steps include:

- 1) Removal of duplicates, noise, and irrelevant metadata to eliminate repeated entries, HTML tags, boilerplate text, and other non-content elements.
- 2) Handling missing or incomplete data by either correcting incomplete fields or discarding unusable entries.
- 3) Normalization operations such as lowercasing, punctuation removal, and cleaning special characters to maintain consistent text formatting.
- 4) Tokenization using SpaCy to break the text into meaningful linguistic units [28].
- 5) Lemmatization using SpaCy to convert each token into its base form and reduce vocabulary complexity.
- 6) Stopword removal to eliminate non-informative words and improve signal-to-noise ratio.

The resulting preprocessed dataset provides a clean and consistent textual foundation for embedding generation, clustering, classification, and sentiment analysis. This step plays a critical role in enhancing model accuracy and ensuring reliable downstream performance.

E. Embeddings and Clustering

Each article was transformed into numerical embeddings to represent its semantic meaning in a high-dimensional vector space. These embeddings were generated using transformer-based sentence encoders, which capture contextual relationships between words more effectively than traditional bag-of-words or TF-IDF representations [16].

Once the embeddings were obtained, dimensionality reduction was performed using UMAP, a technique well-suited for preserving both local and global structure while significantly reducing computational complexity [11]. The reduced vectors were then clustered using the K-Means algorithm to identify groups of semantically similar articles. To determine the most appropriate number of clusters, multiple values of k were evaluated using the Silhouette Score, as shown in Figure 4, which measures the cohesion within clusters and the separation between them [31].

While $k=7$ yielded a marginally higher SilhouetteScore, the value $k=10$ was selected because it offered clearer thematic separation across article groups and improved interpretability for downstream tasks.

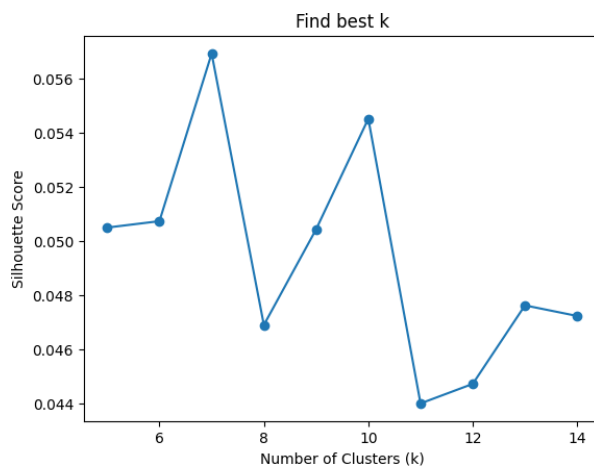


Fig. 4: Silhouette score analysis for determining the optimal number of clusters (k). The score for $k=10$ was selected to ensure thematic separation.

After clustering, each group was examined by analyzing the most frequent keywords and representative articles within it. This qualitative review enabled the assignment of meaningful category labels to the clusters, ultimately forming the basis for supervised classification and ministry mapping in later stages of the pipeline.

F. Classification

Supervised classification techniques were employed to build predictive models using the labeled dataset. Two major families of models were explored during experimentation:

Baseline Models: Random Forest, Naïve Bayes, SVM, and Logistic Regression were implemented to establish strong initial performance benchmarks [14].

Model performance was evaluated using standard classification metrics, including accuracy and F1-score. Table I summarizes the comparative results of the models. Among all models, Linear SVM achieved the highest accuracy (88.9%) and F1-score (0.88), demonstrating superior overall performance. Logistic Regression also performed competitively, while Naïve Bayes and Random Forest provided reasonable baseline results.

<u>Model</u>	<u>Accuracy(%)</u>	<u>F1-Score</u>
LogisticRegression	88.2	0.81
NaiveBayes	84.3	0.83
LinearSVM	88.9	0.88
RandomForest	85.2	0.85

TABLE I: Performance comparison of the implemented classification models.

G. Sentiment Analysis

Neutral Sentiment Thresholding: A news article’s tone can be identified as neutral, negative, or positive using GovPulse’s sentiment analysis. Only a polarity score of precisely 0.0 was categorized as Neutral by the original TextBlob-based method, which is too restrictive for language used in everyday situations. Mild expressions that produce polarity values near zero are frequently found in news articles. **This refined threshold helps the system better capture subtle or weakly expressed sentiments found in factual news reporting.** A neutral margin is added to prevent misclassification.

- Negative: $\text{polarity} < -0.05$
- Neutral: $-0.05 \leq \text{polarity} \leq 0.05$
- Positive: $\text{polarity} > 0.05$

This thresholding allows for a more realistic separation of content with strong sentiments from that with weak sentiments. Reliability is also increased by using transformer-based sentiment models, such as DistilBERT and RoBERTa [15], especially for ambiguous or multilingual text. These models capture contextual cues that are missed by Lexicon-based approaches. Their outputs are combined with polarity scores to generate the final sentiment label with higher confidence.

The resulting sentiment tags, which are stored with probabilities, enable more precise trend analysis on the dashboard and aid the alert system by highlighting articles that are highly negative.

H. Dashboard Visualization

The final stage is a user-friendly web dashboard developed using React.js and TailwindCSS, which connects with Django REST API to present processed news. Key features include:

- Newscards showing article title, summary, ministry label, and sentiment scores.
- Filtering by sentiment, ministry, and language.
- Auto-refresh every hour, with manual refresh capability.
- Analytics graphs showing sentiment trends across ministries.

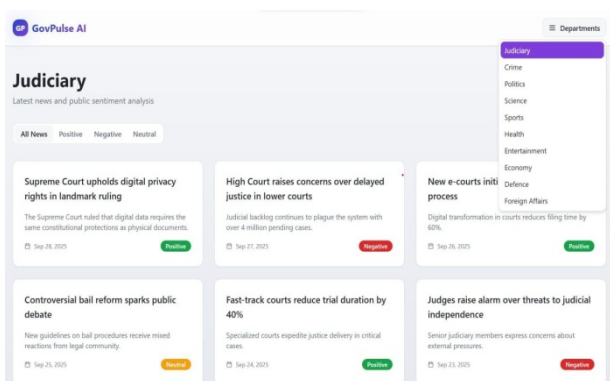


Fig. 5: GovPulse Dashboard Displaying Classified and Sentiment-tagged News

I. Summary

The system showcases the integration of web scraping automation, transformer-based classification, and sentiment-driven alerts to create a comprehensive news intelligence platform. Utilizing Scrapy for automated data gathering, sophisticated NLP models for classification, and a dashboard for interactive visualization, GovPulse offers a complete solution for multilingual news tracking and real-time support for governmental decision-making.

VI. RESULTS AND DISCUSSION

The GovPulse system was evaluated on multiple criteria, including classification of articles, sentiment analysis and the usability of the dashboard. The results demonstrate that transformer-based architecture significantly outperformed traditional machine learning baselines in terms of accuracy, robustness, and adaptability to multilingual content.

Figure 6 shows the comparative performance of the models, where Linear SVM achieved the highest accuracy of **88.9%** and an F1-score of **0.88**.

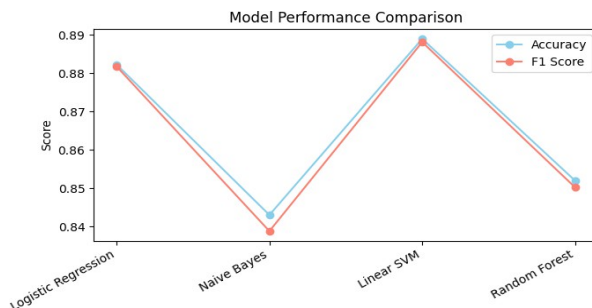


Fig. 6: Performance comparison of baseline and transformer models on the ministry classification task based on Accuracy, Precision, Recall, and F1-Score.

A deeper analysis of the best-performing model’s pre- dictions is provided by the confusion matrix in Figure 7, which highlights common misclassifications between related ministries.

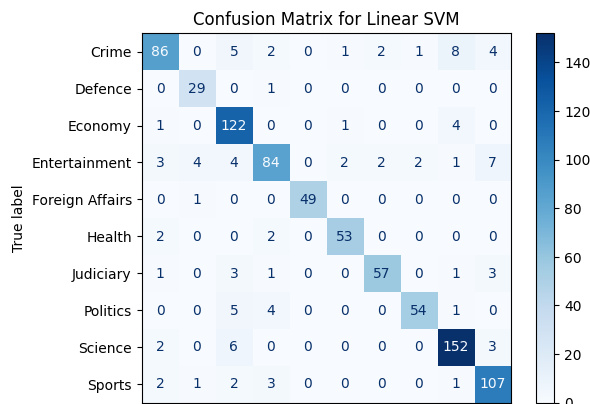


Fig. 7: Confusion matrix for the Linear SVM classification model. The x-axis shows predicted ministry labels, and the y-axis shows true labels.

The React.js dashboard was also tested with a small focus group of users. Feedback highlighted the effectiveness of clear categorization, color-coded sentiment tagging, and filtering by ministry or sentiment. Suggestions for improvement included bilingual visualization (showing both original and translated content) and the addition of temporal analytics to observe sentiment trends over time.

Figure 8 provides an example of the sentiment distribution graph displayed on the dashboard.

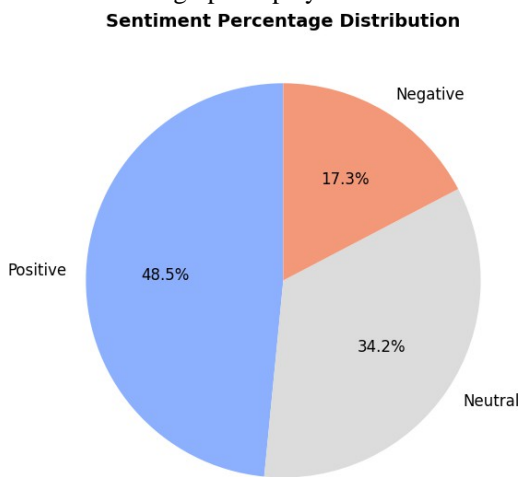


Fig. 8: Overall distribution of sentiment (Positive, Neutral, Negative) across all collected news articles in the dataset.

Overall, the results highlight several important findings: transformer-based models are essential for high accuracy in multilingual classification and sentiment analysis; negative sentiment detection paired with automated alerts provides immediate utility for governance applications; and the scraping framework, implemented using Scrapy with Selenium/Playwright for dynamic content, ensured timely and scalable news acquisition. However, certain limitations remain, including uneven performance across underrepresented regional languages and infrastructure challenges for scaling real-time deployment. Future research will focus on fine-tuning larger multilingual models, integrating multimodal features such as images and videos, and enhancing the dashboard with predictive analytics to forecast emerging issues.

VII. CONCLUSION AND FUTURE PLAN

This work presented GovPulse, an AI-driven system for large-scale monitoring, classification, and sentiment analysis of governance-related news. By combining automated web scraping, multilingual preprocessing, and transformer-based NLP models, the system effectively processes high-volume digital news and organizes it into ministry-specific categories.

Experimental results demonstrate that transformer models significantly outperform classical baselines, with the *LinearSVM* achieving an accuracy of 88.9% and an F1-score of 0.88. The integrated alert mechanism and the interactive dashboard further enhance the system's practical utility by enabling timely detection of negative news and providing transparent, actionable insights to stakeholders.

Although the system performs robustly across major sources, challenges remain in handling code-mixed and low-resource regional languages, processing noisy multimedia content, and scaling real-time pipelines at a national level. Addressing these limitations is essential for improving the model's coverage and deployment readiness. Future enhancements will focus on incorporating stronger multilingual transformer models such as mBERT and XLM-R, extending the pipeline to multimodal analysis using images and videos, and integrating predictive analytics to forecast emerging public issues. Scaling GovPulse through cloud-based distributed pipelines will further improve real-time performance and ensure long-term operational sustainability.

Overall, GovPulse establishes a strong foundation for AI-enabled news intelligence and offers meaningful potential for strengthening transparent governance, proactive policymaking, and informed citizen engagement.

REFERENCES

- [1] R. Patro, S. Sahu, S. Mohanty, Real-time News Classification using Machine Learning Algorithms, *International Journal of Computer Applications*, vol. 176, no. 37, pp. 1–7, 2020.
- [2] J. Zhu, Deep Learning for Large-Scale Text Classification: Convolutional Neural Networks with Feature Hashing, *Proc. 26th ACM Int. Conf. on Information and Knowledge Management (CIKM)*, pp. 55–64, 2017.
- [3] S. Bu'yu'ko'z, E. Ku'c, u'k, S. O'ztu'rk, Comparative Study of ELMO and DistilBERT for News Classification, *Natural Language Engineering*, vol. 26, no. 6, pp. 691–707, 2020.
- [4] N. Valmiki, A. P. S., Machine Learning Approaches for Sentiment Analysis and Prediction, *EPR International Journal of Multidisciplinary Research*, vol. 9, no. 2, pp. 45–51, 2023.
- [5] A. Yadav, Sentiment Analysis in Hindi: A Survey, *Proc. IEEE Int. Conf. on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2349–2353, 2015.
- [6] M. Gupta, K. Sharma, P. Yadav, A Review on News Classification: Traditional vs Deep Learning Approaches, *Journal of Information and Knowledge Management*, vol. 20, no. 4, pp. 1–15, 2021.
- [7] K. Patel, V. Sharma, Hybrid Approaches for Web Page and News Feed Classification, *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 7, pp. 112–118, 2019.
- [8] T. Wolf et al., Transformers: State-of-the-Art Natural Language Processing, *Proc. 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 38–45, 2020.
- [9] Scrapy Developers, Scrapy: An Open Source Framework for Scalable Web Crawling, 2023.
- [10] SeleniumHQ, Selenium WebDriver, 2023.
- [11] L. McInnes, J. Healy, J. Melville, UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, *arXiv preprint arXiv:1802.03426*, 2018.
- [12] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, pp. 226–231, 1996.
- [13] R. Campello, D. Moulavi, J. Sander, Density-Based Clustering Based on Hierarchical Density Estimates, *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pp. 160–172, 2013.
- [14] F. Pedregosa et al., Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [15] Y. Liu et al., RoBERTa: A Robustly Optimized BERT Pretraining Approach, *arXiv preprint arXiv:1907.11692*, 2019.
- [16] J. Devlin et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proc. NAACL*, pp. 4171–4186, 2019.
- [17] Django Software Foundation, Django: The Web Framework, 2023.
- [18] Meta, React – A JavaScript Library for Building User Interfaces, 2023.
- [19] Tailwind Labs, Tailwind CSS – Rapidly Build Modern Websites, 2023.
- [20] S. Ghosh, S. K. Ghosh, Bias detection in online news: A comprehensive survey, *Telematics and Informatics*, vol. 64, p. 101690, 2021.
- [21] A. Vaswani et al., Attention is All You Need, *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [22] T. K. Landauer, P. W. Foltz, D. Laham, An introduction to latent semantic analysis, *Discourse Processes*, vol. 25, no. 2-3, pp. 259–284, 1998.
- [23] H. Allcott, M. Gentzkow, Social media and fake news in the 2016 election, *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–236, 2017.
- [24] M. A. Rosli et al., A survey of web crawling algorithms, *Proc. Int. Conf. on Computing and Informatics*, pp. 1–6, 2019.
- [25] S. Bird, E. Klein, E. Loper, *Natural Language Processing with Python*, O'Reilly Media, Inc., 2009.
- [26] C. Hutto, E. Gilbert, VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text, *Proc. ICWSM*, 2014.
- [27] K. Chodorow, *MongoDB: The Definitive Guide*, O'Reilly Media, 2013.
- [28] M. Honnibal, I. Montani, spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing, 2017.
- [29] R. Gala et al., IndicTrans: A Transformer-based Model for Indic Language Translation, *arXiv preprint arXiv:2205.12218*, 2022.
- [30] A. Conneau et al., Unsupervised Cross-lingual Representation Learning at Scale, *Proc. ACL*, pp. 8440–8451, 2020.
- [31] P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)