iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** X **Month of publication:** October 2025

DOI: https://doi.org/10.22214/ijraset.2025.74721

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue X Oct 2025- Available at www.ijraset.com

Forecasting and Visualization of RSS Feeds Trends and Historical News Data Using Clustering and Prophet

S. Venkata Laksmi[1], N. U. Ajia[2], T. S. Dharshini[3]
[1]*Associate Professor / CSE,* [2, 3]*UG Scholars, Department of Computer Science and Engineering,*
K.L.N. College of Engineering, Pottapalayam, Sivagangai, India

Abstract: ***The exponential growth of digital news media has led to an overwhelming influx of unstructured information, posing challenges in extracting actionable insights and identifying emerging trends. This paper presents an Advanced News Analytics System designed to automate the analysis, clustering, and forecasting of news data aggregated from real-time RSS feeds and historical archives. Leveraging machine learning and time-series forecasting techniques, the system employs Feedparser for data acquisition, Scikit-learn for clustering via K-Means and DBSCAN, and Facebook Prophet for trend forecasting. The platform, implemented using Streamlit, provides an intuitive dashboard that visualizes historical topic dynamics, real-time distributions, and future trend predictions. Unlike conventional aggregators, this system transcends basic content display by enabling automated topic modeling, trend evolution tracking, and predictive analytics, thereby empowering users with data-driven insights into media trends. The study underscores the system's potential to enhance strategic decision-making through scalable, unbiased, and interpretable news trend forecasting.***

Keywords: News analytics, RSS feed processing, Machine learning, Clustering, K-Means, DBSCAN, Prophet forecasting, Time-series analysis, Streamlit dashboard, Trend prediction, Data visualization, Predictive media intelligence.

I. INTRODUCTION

The rapid expansion of digital news media has led to an overwhelming influx of unstructured information, making it increasingly challenging to identify emerging patterns and forecast future trends. Traditional news aggregation systems focus primarily on content collection rather than analytical interpretation, offering limited insight into topic evolution [2,3]. Leveraging advancements in machine learning and time-series forecasting, the proposed Advanced News Analytics System utilizes Scikit-learn for clustering, BERTopic for neural topic modeling, and Facebook Prophet for temporal trend prediction [1,2,4,13]. Built with FastAPI and Streamlit, the platform aggregates real-time RSS feeds and historical archives to generate interactive visualizations through Chart.js [5–10,17–20]. This paper explores the system's framework and implementation, emphasizing its potential to enhance media trend analysis, predictive insight generation, and data-driven decision-making in the digital information ecosystem.

II. METHODOLOGY

The proposed system, Forecasting and Visualization of RSS Feed Trends and Historical News Data using Clustering and Prophet, is built upon a robust and modular architecture designed to ensure efficient data collection, intelligent analysis, and interactive visualization of news dynamics. The platform integrates multiple technologies and frameworks to deliver a scalable, high-performance, and user-friendly analytical solution. The data acquisition process begins with the Feedparser library in Python, which continuously extracts real-time articles from diverse RSS feeds while simultaneously incorporating historical news datasets for temporal analysis. These heterogeneous data sources are merged, cleaned, and structured through Pandas and NumPy, ensuring consistency, integrity, and readiness for analytical processing.

The data preprocessing and clustering phase employs advanced machine learning algorithms, including K-Means and DBSCAN, implemented via Scikit-learn. This enables the automatic categorization of news content into thematic clusters, effectively identifying relationships and emerging topics from large-scale unstructured data. The clustered data is then stored in an Aggregated News Database, ensuring efficient retrieval and long-term scalability.For trend prediction and temporal analysis, the system leverages Facebook Prophet, a powerful time-series forecasting model capable of handling seasonality, trend shifts, and irregularities in news frequency.

This forecasting module predicts the trajectory of emerging topics, allowing users to anticipate future trends based on historical and current data patterns.

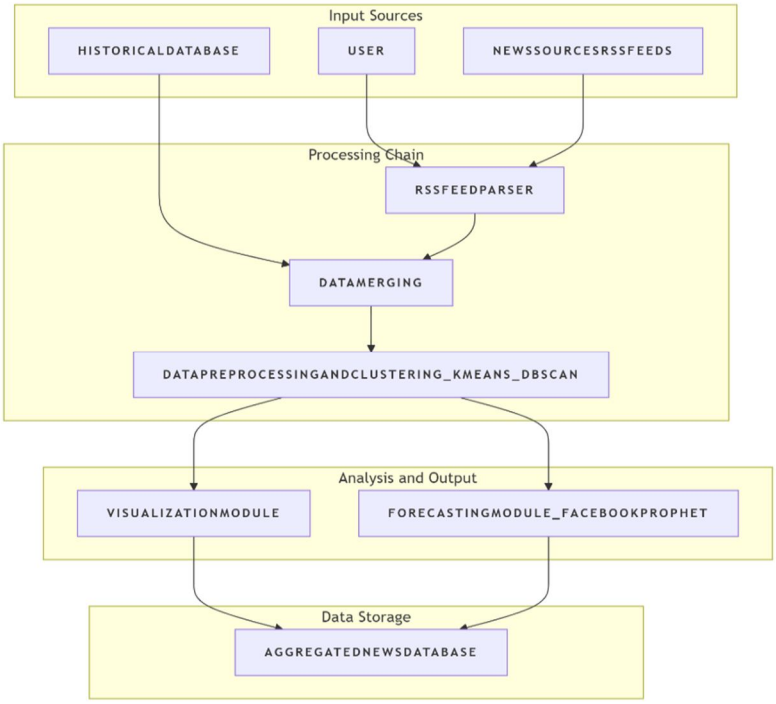


Fig: 1 Flow diagram

The analytical results are presented through an interactive and visually enriched Streamlit-based dashboard, which serves as the system's front-end interface. This dashboard provides dynamic visualizations of topic distributions, historical patterns, and forecasted trends, ensuring intuitive exploration and insight-driven interpretation. The system architecture is designed for modular extensibility, allowing future integration of additional data sources or advanced NLP techniques for sentiment and relevance analysis. By combining automated news aggregation, unsupervised learning, and predictive forecasting, the proposed system transcends the capabilities of conventional news aggregators. It provides a comprehensive analytical environment for monitoring information flows, understanding thematic evolution, and supporting data-driven decision-making in media analytics.

III. RSS AND HISTORICAL DATA COLLECTION

The RSS and Historical Data Collection module forms the foundational layer of the system, responsible for aggregating and unifying diverse information sources. It leverages RSS feed parsing techniques to extract real-time articles, metadata, and publication timestamps from multiple news outlets, ensuring continuous data acquisition. Simultaneously, it integrates historical news archives to enrich the dataset with temporal depth, enabling retrospective trend analysis and longitudinal forecasting. By combining live and past data, this module ensures a holistic representation of topic evolution over time. It utilizes automated schedulers and data validation mechanisms to maintain the integrity and timeliness of incoming feeds, eliminating redundancy and ensuring data reliability. This seamless integration of historical and real-time information serves as the cornerstone for subsequent clustering, analysis, and predictive modeling, supporting accurate insight extraction from the rapidly evolving digital news ecosystem.

IV. NEW DATA PROCESSING AND CLUSTERING

The Data Processing and Clustering module transforms raw textual data into structured, meaningful representations through systematic preprocessing and unsupervised learning techniques. It employs natural language processing (NLP) operations such as tokenization, stop-word elimination, lemmatization, and vectorization to clean and standardize the incoming news corpus. Following this, machine learning algorithms like K-Means and DBSCAN are applied to automatically categorize news articles into coherent clusters based on thematic similarity and semantic relationships.

This enables the system to identify emerging topics, hidden patterns, and related discussions across multiple sources. The module is designed for adaptability and scalability, ensuring efficient performance even with large and heterogeneous datasets. By converting unstructured text into analytically rich clusters, this component facilitates advanced exploration, comparative analysis, and trend tracking essential for predictive forecasting and visualization processes in later stages

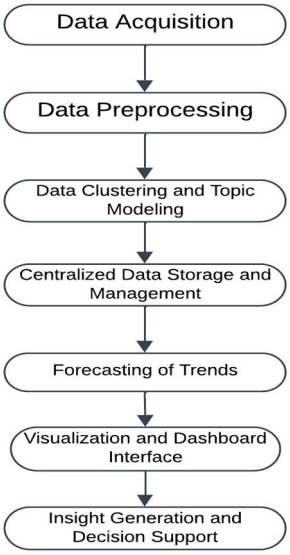


Fig: 2 Process flow

V. CETRALIZED DATA STOREAGE AND MANAGEMENT.

The Centralized Data Storage and Management module ensures seamless organization, retrieval, and maintenance of aggregated news data from both real-time and historical sources. Implemented through a robust database architecture, it enables structured storage of clustered datasets, metadata, and forecasting results for consistent access across system components. The module supports optimized indexing and querying mechanisms, facilitating efficient data retrieval for analysis and visualization tasks. It incorporates data validation, version control, and redundancy elimination to uphold data consistency and accuracy. Furthermore, it ensures security and scalability through access control, backup protocols, and dynamic allocation of storage resources as data volume grows.

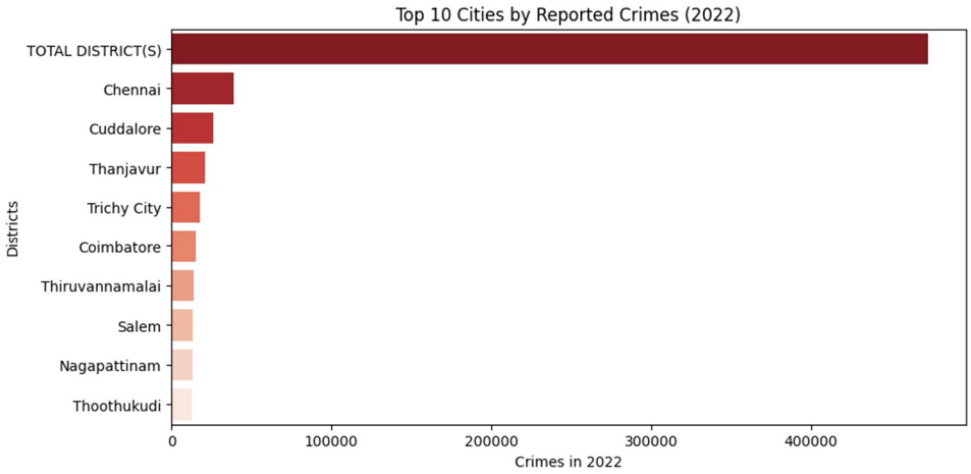


Figure: 3 Visualized Chart

This centralized repository not only supports efficient processing workflows but also guarantees traceability and reproducibility of results, forming the backbone of the analytical pipeline by providing a reliable and high-integrity foundation for all downstream operation

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue X Oct 2025- Available at www.ijraset.com

VI. VISUALIZATION AND TREND FORECASTING

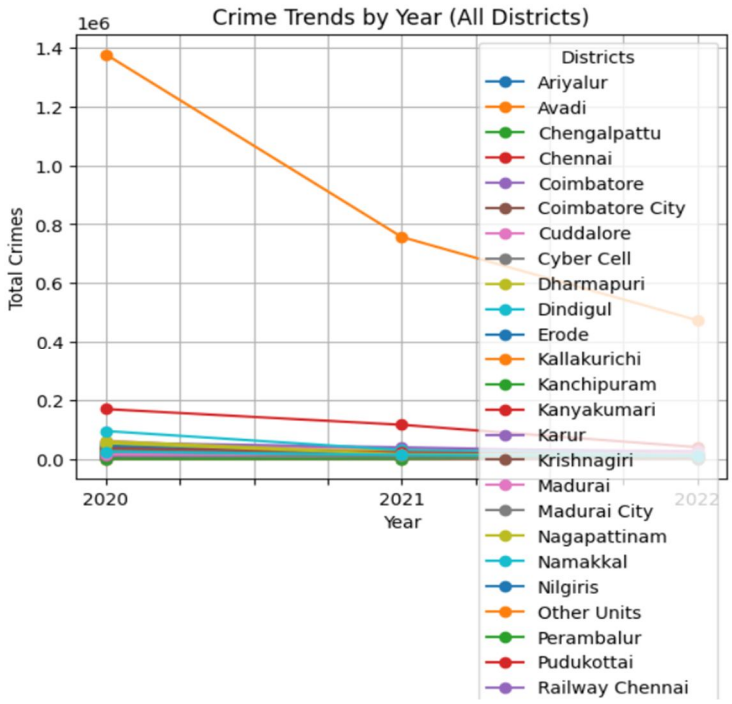


Figure: 3 Visualized Graph

The Visualization and Trend Forecasting module serves as the analytical interface of the system, transforming processed data into comprehensible insights through interactive visual representations and predictive analytics. Leveraging libraries like *Matplotlib*, *Plotly*, and *Prophet*, it enables users to explore topic distributions, temporal patterns, and projected trends with precision and clarity. Forecasting is achieved through *Facebook Prophet*, which models seasonality, trend shifts, and historical fluctuations to predict future topic prominence.

The visualization component employs intuitive dashboards, charts, and time-series plots developed using *Streamlit*, ensuring accessibility and real-time interaction for end-users. This dual capability of visualization and forecasting empowers users to monitor topic evolution dynamically, identify emerging narratives, and make data-driven strategic decisions. By presenting complex analytical outcomes in a clear and actionable format, the module bridges the gap between data science and user understanding, driving informed media intelligence and foresight.

VII. CONCLUSION

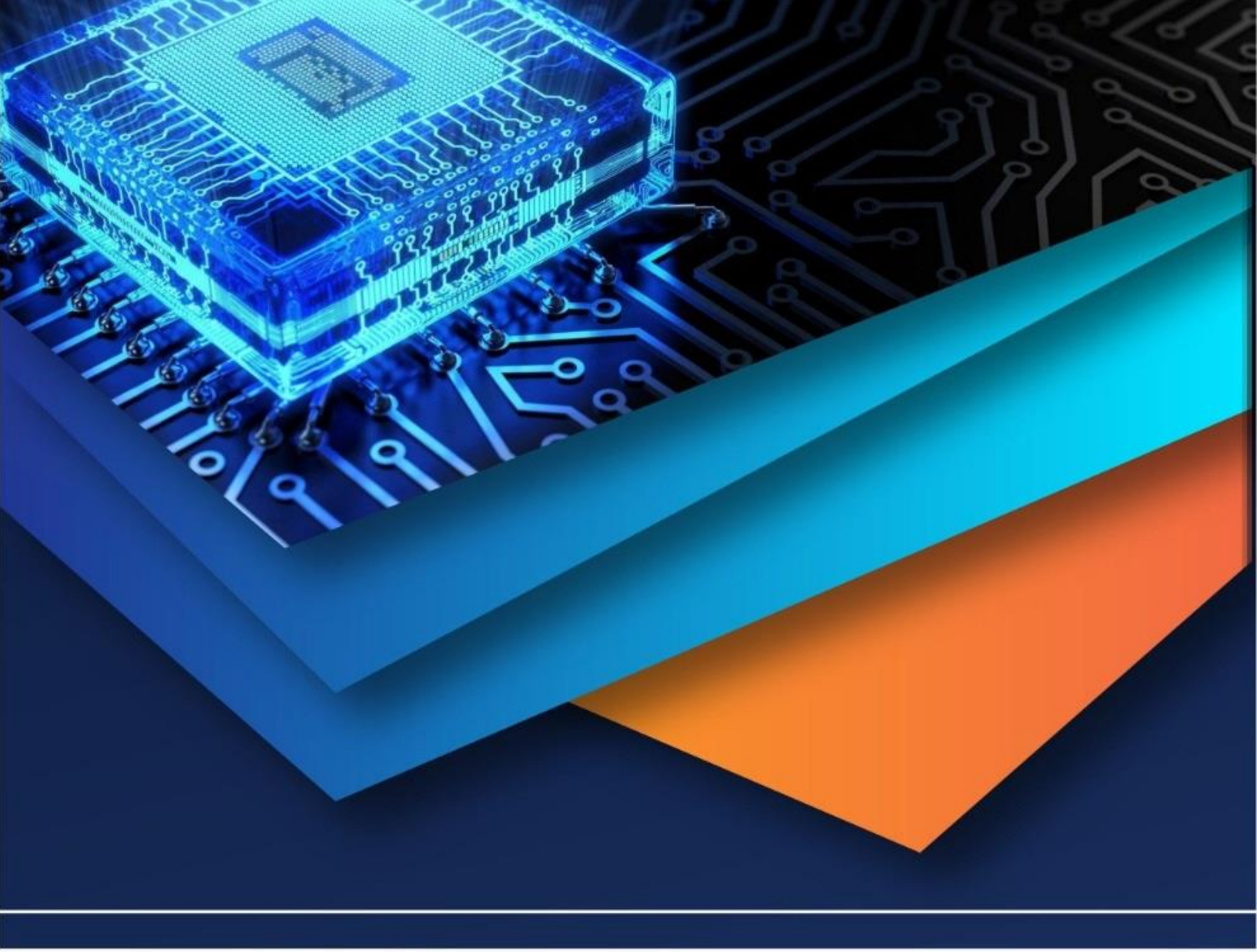
The News Analytics and Forecasting System presents a comprehensive framework for automating the collection, processing, and analysis of news data from multiple sources. By integrating RSS feeds and historical datasets, the system ensures continuous and reliable access to real-time and archival information. This combination strengthens the analytical foundation by enabling both immediate insights and long-term trend evaluation, offering users a holistic perspective on news patterns over time.

The inclusion of data processing, clustering, and centralized storage allows for efficient handling of vast textual information. Through the application of intelligent clustering algorithms, the system identifies topic-based groupings that enhance content organization and retrieval. Furthermore, the structured database design ensures scalability, security, and seamless integration with analytical tools for advanced operations like anomaly detection and temporal trend monitoring.

Finally, the visualization and forecasting modules bridge the gap between raw data and actionable insights. Using advanced graphical dashboards and predictive modeling techniques such as Prophet or ARIMA, users can visualize historical trends and forecast potential future developments. This end-to-end workflow transforms unstructured news data into an intelligent, data-driven analytical system capable of supporting informed decision-making, proactive media monitoring, and enhanced understanding of evolving news dynamics.

REFERENCES

[1] Facebook Research. (2023). Prophet: Forecasting at Scale. Retrieved from: https://facebook.github.io/prophet/
[2] Grootendorst, M. (2023). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv preprint arXiv:2203.05794. Retrieved from: https://arxiv.org/abs/2203.05794.
[3] Muennighoff, N., et al. (2023). MTEB: Massive Text Embedding Benchmark. arXiv preprint arXiv:2305.13823. Retrieved from: https://arxiv.org/abs/2305.13823
[4] Pedregosa, F., et al. (2023). *Scikit-learn: Machine Learning in Python (Version 1.3).* Journal of Machine Learning Research, 24(1), 1–8. Retrieved from: https://jmlr.org/papers/v24/23-0348.html.
[5] PostgreSQL Global Development Group. (2023). PostgreSQL 15.2 Documentation. Retrieved from: https://www.postgresql.org/docs/15/index.html
[6] Redis Ltd. (2023). Redis Documentation (Version 7.0). Retrieved from: https://redis.io/documentation/.
[7] Celery Project. (2023). Celery: Distributed Task Queue. Documentation. Retrieved from: https://docs.celeryq.dev/en/stable/
[8] FastAPI. (2023). FastAPI Documentation (Version 0.100.0). Retrieved from: https://fastapi.tiangolo.com/
[9] Meta Open Source. (2023). React Documentation. Retrieved from: https://react.dev/
[10] Chart.js Community. (2023). Chart.js Documentation (Version 4.0). Retrieved from: https://www.chartjs.org/docs/latest/
[11] Facebook Research. (2023). Prophet: Forecasting at Scale. Retrieved from: https://facebook.github.io/prophet/
[12] Grootendorst, M. (2023). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv preprint arXiv:2203.05794. Retrieved from: https://arxiv.org/abs/2203.05794
[13] Muennighoff, N., et al. (2023). MTEB: Massive Text Embedding Benchmark. arXiv preprint arXiv:2305.13823. Retrieved from: https://arxiv.org/abs/2305.13823
[14] Pedregosa, F., et al. (2023). *Scikit-learn: Machine Learning in Python (Version 1.3).* Journal of Machine Learning Research, 24(1), 1–8. Retrieved from: https://jmlr.org/papers/v24/23-0348.html
[15] PostgreSQL Global Development Group. (2023). PostgreSQL 15.2 Documentation. Retrieved from: https://www.postgresql.org/docs/15/index.html
[16] Redis Ltd. (2023). Redis Documentation (Version 7.0). Retrieved from: https://redis.io/documentation/
[17] Celery Project. (2023). Celery: Distributed Task Queue. Documentation. Retrieved from: https://docs.celeryq.dev/en/stable/
[18] FastAPI. (2023). FastAPI Documentation (Version 0.100.0). Retrieved from: https://fastapi.tiangolo.com/
[19] Meta Open Source. (2023). React Documentation. Retrieved from: https://react.dev/
[20] Chart.js Community. (2023). Chart.js Documentation (Version 4.0). Retrieved from: https://www.chartjs.org/docs/latest/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY