iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.68655

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 IV Apr 2025- Available at www.ijraset.com

Real Time Disaster Information Aggregation Model

Rita Kadam[1], Sourabh Tagad[2], Shashin Dherange[3], Siddhi Kanade[4], Akash Mali[5]
[1]*Artificial Intelligence and Data Science, School of Engineering and Technology, D Y Patil University, Ambi, Pune*
[2, 3, 4]*Computer Engineering Department, School of Engineering and Technology, D Y Patil University, Ambi, Pune*

Abstract: ***Disaster management and response rely heavily on real-time and accurate information. Previous research has focused on collecting disaster data from official government reports and structured news sources. While these methods provide reliable information, they often suffer from delays and lack immediate public sentiment analysis. In our project, we enhance disaster response by extracting data from Reddit and Google News, allowing access to both real-time information and public sentiment. Using web scraping and Natural Language Processing (NLP) techniques, we filter relevant disaster-related posts and news articles. Sentiment analysis is performed to assess the emotional tone of public reactions. Additionally, by applying classification models, we categorize the severity of the events, providing authorities with crucial insights. This integrated system offers a more immediate, diverse, and sentiment-aware disaster information pipeline compared to traditional methods, aiming to improve the speed and efficiency of disaster management efforts.***

Keywords: Disaster Management, Machine Learning, Real-Time Data Analysis, Reddit Data, Google News, Natural Language Processing (NLP), Sentiment Analysis.

I. INTRODUCTION

In today's world, the timely dissemination of disaster- related information is crucial to saving lives, minimizing damage, and coordinating effective responses.

Natural and man-made disasters, such as earthquakes, floods, and industrial accidents, often unfold rapidly, leaving little time for traditional reporting methods to reach the affected population.

In such situations, real-time aggregation of information from multiple sources can play a critical role.

This project presents the development of a Real-Time Disaster Information Aggregation Software, designed to gather, filter, and display relevant disaster data efficiently. Using news websites, Reddit discussions, and other publicly available sources, the system provides users with up-to-date, verified, and visually organized information. The backend, developed using Python and Django, integrates powerful libraries like Beautiful Soup and PRAW for data extraction, while the front end employs HTML, CSS, JavaScript, and Bootstrap to create a clean and responsive user experience.

In addition, the platform supports user authentication, personalized notifications, and intuitive dashboards for easy navigation. The ultimate goal of this system is to offer a centralized, reliable, and user-friendly portal for disaster information, ensuring that individuals, emergency responders, and organizations are better prepared to act quickly during critical events.

A. Problem Statement

The problem identified from the research is the existing gap in real-time disaster response due to the fragmentation and delayed delivery of crucial disaster-related information. Traditional systems rely on centralized reporting mechanisms, which are often slow to update and may not cover all critical aspects. Our approach addresses this gap by utilizing decentralized information sources such as Google News and Reddit, ensuring faster retrieval of disaster-related data. By tapping into these dynamic and diverse platforms, we aim to provide quicker, broader, and more accurate situational awareness during disaster events

B. Background and Motivation

The rapid occurrence of natural and man-made disasters, such as earthquakes, floods, wildfires, and industrial accidents, demands the swift distribution of critical information. Traditional news channels and official updates often face delays in reaching the public, causing a gap in disaster awareness and timely response. In recent years, the widespread use of the internet and social media platforms like Reddit has opened new avenues for gathering real-time data. Motivated by the need for quicker, centralized, and verified disaster information, this project focuses on developing a solution that aggregates data from multiple sources for immediate accessibility.

II. REVIEW OF LITERATURE

The existing literature on intelligent disaster management systems presents a variety of promising methodologies aimed at enhancing real-time disaster response and recovery efforts. Bouzidi et al. [1] developed an intelligent and real-time alert model that retrieves information from multiple online sources such as Facebook and Twitter, allowing rapid information dissemination during crises. However, the model's scalability is limited when facing massive volumes of data during widespread disasters, and it lacks robust data verification mechanisms, which could compromise the authenticity of critical information. Karimiziarani [2] conducted a comprehensive review of social media analytics techniques used in disaster response, highlighting the effectiveness of data mining, sentiment analysis, and event detection. Despite its technical depth, the review gives limited attention to practical deployment issues, such as internet outages during emergencies, and only briefly touches upon ethical and privacy concerns without offering concrete solutions. Sumon et al. [3] explored the application of machine learning algorithms for real-time disaster response in the U.S., emphasizing predictive modeling and automated decision-making. Nonetheless, their approach largely relies on historical datasets and simulated environments, raising concerns about the models' adaptability to dynamic, unpredictable real-world disaster scenarios, especially considering challenges like data imbalance across different disaster types. Kotagiri [4] proposed an AI-based decision support framework that integrates satellite imagery, predictive analytics, and social media data to assist sustainable humanitarian efforts. While technologically sophisticated, this approach assumes consistent access to high-end infrastructure, which might not be available in underdeveloped or disaster- stricken regions, thus limiting its practicality. Finally, Shuaibu and Tiwari [5] focused on real-time data analysis using ma- chine learning, emphasizing the need for fast and accurate decision-making in disaster management. However, their study does not address critical issues such as ensuring the quality and reliability of data inputs or mitigating biases in automated systems, which are crucial for fair and inclusive disaster response. Overall, although the reviewed studies contribute significant theoretical advancements, they collectively reveal the need for more scalable, reliable, and context-aware dis- aster management solutions that are robust against real-world operational challenges.

III. METHODOLOGY

In this study, we developed a real-time disaster management system by leveraging machine learning and social media analytics. Data was collected from multiple sources, including Google News and Reddit, to capture both official news re- ports and public discussions regarding ongoing disasters. The collected data was preprocessed to remove noise, irrelevant content, and duplicate entries. Natural Language Processing (NLP) techniques were applied to extract meaningful features such as event location, disaster type, severity, and public sentiment. Machine learning classifiers were then trained to categorize the information, assess the severity of the events, and generate timely alerts. The processed information was finally visualized on an interactive dashboard, offering an intuitive interface for decision-makers to monitor and respond to disasters in real time.

IV. MATHEMATICAL MODEL

A. *Definition of States (S)*

Let,

S = s — Set of all states of the system Where,
s0 = Start state (User sends request through browser) s1 = Flask API receives request
s2 = Flask connects to PostgreSQL database s3 = Database query is executed
s4 = Results are sent back to Flask
s5 = Flask renders HTML page with results
s6 = End state (User sees the Disaster Information on dashboard)

B. *Input, Process, Output*

Define:

I = User request parameters like category, page number
P = Establish DB connection, Execute Query, Fetch Results, Render HTML
O = Web page displaying filtered and paginated disaster reports

C. *System Function*

Thus, System Function F: $F : I \rightarrow O$

D. Constraints

Constraints:

- Database must be available. - Correct credentials are required for DB connection. - If no results are found, a message should be shown instead of empty table.

V. IMPLEMENTATION DETAILS

A. Technologies Used

1) Frontend: HTML, CSS (for dashboard visualization)
2) Backend: Python (Flask Framework)
3) Database: PostgreSQL
4) Other Tools:
- draw.io (for architecture diagram)
- pgAdmin (for managing PostgreSQL database)

B. Working Description

1) User sends a request from the web browser (like filtering by disaster category).
2) Flask app receives the request and processes it.
3) Flask sends a query to the PostgreSQL database to fetch the disaster records.
4) Database returns the requested data.
5) Flask takes the data and renders it into an HTML dashboard.
6) The user sees a clean, filtered, and paginated dashboard on the browser.

C. Key Features

1) Filtering by Disaster Type: (Earthquake, Flood, Wild- fire, etc.)
2) Pagination: To handle large datasets easily.
3) Responsive Dashboard: (Looks clean on different de- vices)
4) Error Handling:
- If no data found → proper message shown.
- If DB not available → system gracefully handles.

VI. RESULTS

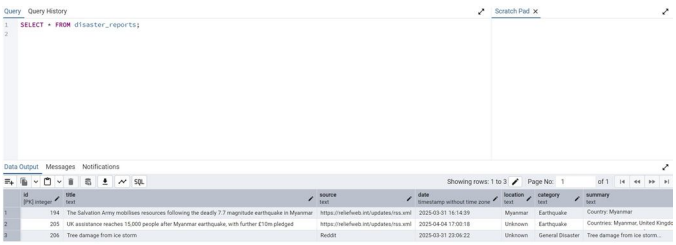


Fig. 1. PostgreSQL Database

A. Postgre SQL Database

The Disaster Management System project, titled ”Real- Time Disaster Information Aggregation Software,” is de- signed to gather, filter, and display disaster-related information in real time from multiple open sources like news websites and Reddit. Built using Python for the backend (with Django) and PostgreSQL for the database, it fetches and cleans data using libraries like BeautifulSoup, PRAW, Pandas, and NumPy. The frontend is developed with HTML, CSS, JavaScript, and Bootstrap to create a user-friendly dashboard for visualizing disaster updates. The system includes modules for News Aggregation, Reddit Parsing, Data Cleaning, Visual Display, Alerts, and User Authentication (using Django’s built-in auth system). It helps disaster response teams by providing real- time, categorized information for faster and more effective action. The project is deployed using GitHub and Heroku, with documentation tools like Overleaf and Google Drive aiding in collaboration. Twitter data was excluded due to tweet access limitations. Overall, the system aims to enhance disaster management efficiency by centralizing vital data in a simple and accessible format.

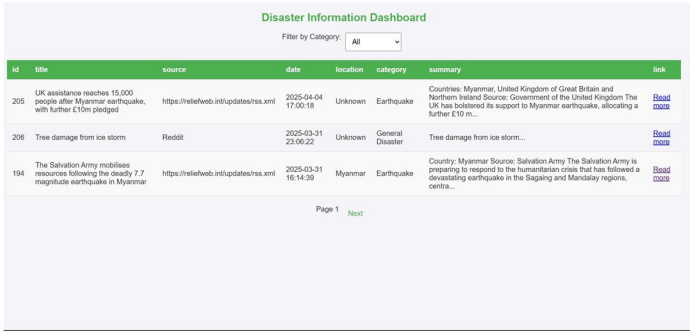


Fig. 2. Dashboard

B. Dashboard

The Real-Time Disaster Information Aggregation Software features a comprehensive dashboard designed to enhance sit- uational awareness for emergency responders and decision- makers. This user-friendly interface consolidates and visual- izes real-time data from diverse sources, including news outlets and Reddit, to provide an up-to-date overview of disaster events. By integrating interactive maps that display geolocated disaster incidents, live feeds streaming real-time updates, ana- lytics panels offering insights into disaster trends and severity assessments, and alert notifications for emerging disasters, the dashboard serves as a centralized hub for monitoring and managing disaster response efforts. This integration facilitates informed decision-making and efficient resource deployment, bridging the gap between official reporting and ground realities shared by the public.

VII. CONCLUSION

The Disaster Information Dashboard successfully provides a user-friendly platform to view and filter disaster-related data efficiently. Using technologies like Flask, PostgreSQL, and HTML/CSS, the system demonstrates how data-driven web applications can assist in organizing and presenting critical information. Although the current system handles static data well, future enhancements such as real-time data integration and predictive analysis can make it even more impactful. Overall, this project offers a strong foundation for further development in the field of disaster management systems.

VIII. FUTURE SCOPE

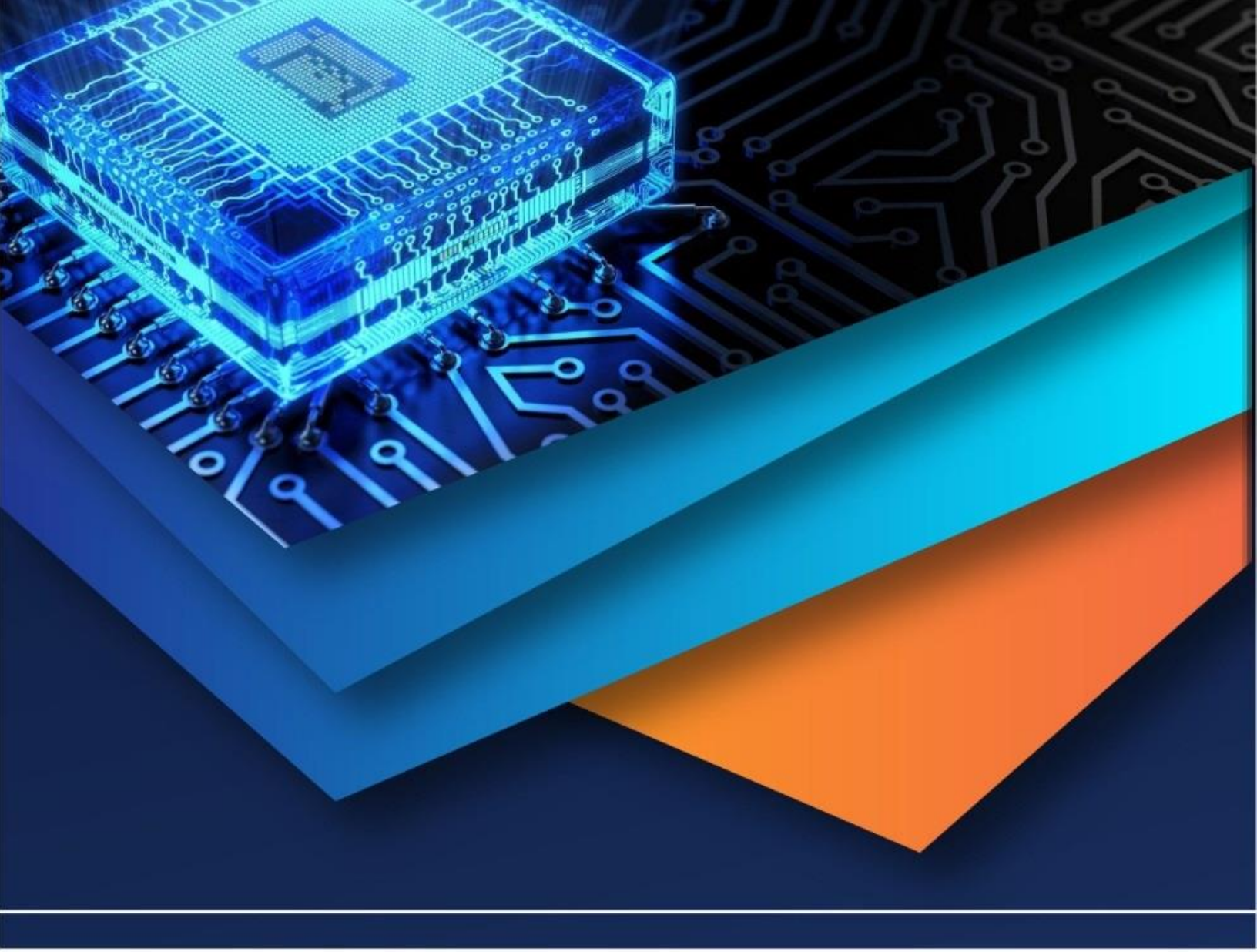
1) Integration of real-time disaster feeds using APIs to keep the database updated automatically.
2) Implementation of push notifications or alert systems for critical disasters.
3) Adding graphical analytics like charts and heatmaps for better data visualization.
4) Extending the system to handle larger and more complex datasets efficiently.
5) Making the dashboard mobile-friendly for better acces- sibility.
6) Incorporating machine learning models to predict possi- ble future disaster events.

IX. ACKNOWLEDGMENT

I would like to express my sincere gratitude to my project guide, Prof. Rita Kadam , for their invaluable guidance and constant support throughout this project. I would also like to thank Dy Patil, Ambi, School of Engineering for providing the necessary resources and infrastructure. Finally, I am grateful to my family and friends for their continuous encouragement.

REFERENCES

[1] Z. Bouzidi, M. Amad, and A. Boudries, “Intelligent and Real-Time Alert Model for Disaster Management Based on Information Retrieval from Multiple Sources,” International Journal of Advanced Media and Communication, ACM Digital Library, 2019.
[2] M. Karimiziarani, “Social Media Analytics in Disaster Response: A Comprehensive Review,” arXiv preprint, 2023.
[3] M. F. I. Sumon, M. A. Khan, and A. Rahman, “Machine Learning for Real-Time Disaster Response and Recovery in the U.S.,” International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence, 2023.
[4] O. Kotagiri, “Enhancing Disaster Response and Recovery Through AI: Real-Time Decision Support for Sustainable Humanitarian Efforts,” In- ternational Journal of AI-Assisted Medicine, 2022.
[5] A. A. Shuaibu and S. Tiwari, “Real-Time Data Analysis for Disaster Management: A Machine Learning Approach,” Journal of Intelligent Decision Technologies and Applications, 2024.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY