Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 3s (2025)

AI-Powered Multimodal Disaster Response Enhancement using Social Media Streams

Rohini Temkar[1], Gresha Bhatia[2], Sai Thikekar[3], Arya Banavali[4], Yash Chhaproo[5], Aradhya Ingle[6]

[1,2,3,4,5,6]Department of Computer Engineering, Vivekanand Education Society's Institute of Technology, Mumbai, India

Article History:

Received: *26-09-2024*

Revised: *04-11-2024*

Accepted: *19-11-2024*

Abstract:
Social media serves as a valuable source of real-time disaster data, providing timely updates that are critical for effective disaster monitoring and response. It plays an essential role in raising awareness, enhancing preparedness, and facilitating prompt action during emergencies. While much of the current research focuses on disaster prediction, real-time updates, which are key to saving lives and coordinating response efforts, are often overlooked. Our project addresses this gap by focusing on the collection and analysis of real-time data from various social media platforms, including disaster-related images and textual data like tweets.We use multiple APIs to gather real-time data from platforms such as Instagram, Facebook, and Twitter. To analyze visual data, we employ state-of-the-art multimodal models like ResNet50 and EfficientNet for image classification. For the textual data, advanced natural language processing models like BERT and XLNet are used for sentiment analysis, event detection, and severity assessment. By integrating and fusing the image and text data, our system is able to identify the type of disaster, estimate its severity, and provide actionable insights.

Keywords: Real-time disaster data, APIs, multimodal, ResNet50, EfficientNet, BERT, XLNet , disaster severity assessment.

I. INTRODUCTION

Natural disasters, including tsunamis, floods, and earthquakes, are destructive and often beyond human control. However, disaster management offers a structured approach to mitigate the devastating impacts of such events through preparedness, response, and recovery strategies. India, in particular, ranks third globally in human mortality from extreme weather events between 2000 and 2019. Over the period from 1983 to 2011, the country faced 190 floods and 54 cyclones, leading to severe human and economic losses, with approximately 60,919 deaths and 1.22 billion individuals affected by floods, and 20,360 deaths and 56 million people impacted by cyclones. These statistics emphasize the critical vulnerability of India to natural disasters and the urgent need for more effective disaster management and resilient infrastructure systems to reduce future risks [1].

In recent years, social media platforms, such as Twitter, Instagram, and Facebook, have played an increasingly important role in disaster management by providing real-time data that enhances situational awareness and facilitates rapid emergency response. The integration of social media analytics into disaster management frameworks has revolutionized how information is collected, processed, and acted upon during crises. Social media's ability to provide up-to-date, location-specific data enables authorities and responders to monitor dynamic situations and coordinate more efficient responses [2].

Further, multimodal data—text, images, and videos—has proven essential for enhancing the precision of urban event monitoring. For instance, [2] and related studies demonstrate that combining various data types significantly improves the spatial accuracy of event detection, such as urban flooding, achieving up to 100% precision in specific instances ("A Spacel Information Extraction Method based on Multimodal Social Media Data: A Case Study on Urban Inundation," 2023). Moreover, Chatterjee [3] propose an innovative approach for monitoring mental health through social media posts, using sentiment analysis and multimodal features to construct a mental health index, achieving 89% accuracy with Support Vector Machines (SVM). This real-time analysis has profound implications for disaster response, as psychological well-being is a critical aspect of managing the human impact of disasters.

Disaster response can also be enhanced by employing advanced natural language processing models like Late Dirichlet Allocation and BERT. [2] show that real-time classification of disaster-related topics using such models improves accuracy by 12% compared to earlier techniques. Despite these advancements, challenges remain, especially in managing the noisy and unstructured nature of social media data, which can affect the reliability of insights. Ensuring a balance between high accuracy and the unpredictability of social media remains a vital area for ongoing research.

II. RELATED WORK

Efficient data collection and aggregation from diverse sources, such as social media and news outlets, play a crucial role in real-time disaster management systems. In recent work, DeepScraper by Jaebeom [4] demonstrated a novel approach to tweet scraping using authenticated multiprocessing, significantly improving the collection speed and volume of social media data. This method successfully scraped over 5 million tweets, outperforming the standard Twitter API by 23.7 times, making it highly applicable for the social media data collection component of disaster response systems.

To complement social media data collection, the aggregation of disaster-related news is essential for a comprehensive crisis management system. Domala [5] proposed a system that automates the collection of disaster news using web scraping techniques and classifies the gathered data through machine learning models such as Support Vector Machines (SVM) and Logistic Regression. This system also incorporates Natural Language Processing (NLP) to preprocess the news data and geoparses relevant disaster locations. With an evaluation based on precision, recall, F1 score, and a confusion matrix, the approach ensures that only relevant news is included in crisis management workflows.

Accurate data validation is critical in disaster management to prevent the spread of misinformation, especially on social media platforms. Mishra [6] addressed this challenge through the Disaster Information Verification and Validation Application (DIVVA), which uses machine learning to verify disaster-related information by cross-referencing it with official government sources. The system utilizes a Bidirectional LSTM model to classify tweets as real or fake based on textual input analysis, achieving an accuracy of 84%. DIVVA's approach to ensuring the reliability of disaster information is highly relevant to our project, which focuses on aggregating and validating data from social media and news, ensuring only verified and actionable information is included in disaster response efforts.

Beyond data collection and validation, effective categorization of disaster-related information is essential for timely response. The work by Mishra [6] introduced a deep multimodal learning approach that integrates visual and textual features to classify disaster types from social media posts. The system employs ResNet50 for visual feature extraction and a BiLSTM model with an attention mechanism for textual analysis, showing a significant performance boost over unimodal models and even improving by 7% compared to other multimodal approaches. This method addresses the challenge of processing vast amounts of social media data, ensuring more accurate disaster categorization. Incorporating such a multimodal approach into our project can enhance the classification of disaster data, making the system more robust in identifying and organizing real-time information.

Effective classification of disaster-related tweets is crucial for timely response and rescue efforts. Ningsih [7] explored the use of the Bidirectional Encoder Representations from Transformers (BERT) model to enhance the identification of disaster-related tweets, particularly focusing on sentiment analysis. This approach helps differentiate critical tweets, such as rescue requests, from general disaster discussions, addressing the challenge of understanding complex language structures during emergencies. BERT's ability to manage context and sentiment uncertainty proves beneficial in improving tweet classification, thereby aiding disaster response teams in prioritizing rescue operations. Integrating BERT-like models into our project can significantly enhance the accuracy and relevance of disaster data classification.

III. PROPOSED METHODOLOGY

In natural disasters, platforms like Twitter become critical for real-time updates. The proposed solution addresses the challenge of collecting, processing, and classifying disaster-related tweets, overcoming Twitter's API restrictions and large data volumes. The system integrates web scraping, real-time streaming, and machine learning to efficiently filter and classify tweets, thereby improving disaster response.

3.1. Data Collection via Web Scraping

Due to Twitter’s recent API restrictions, web scraping has become an essential alternative for collecting tweets. This solution leverages Selenium and BeautifulSoup to scrape tweet content, metadata, and user information directly from Twitter's web interface. To handle Twitter’s rate limits and prevent detection, multiple techniques are employed. One such technique is proxy rotation, which uses a rotating pool of proxies to prevent IP bans and ensure consistent scraping throughout the process [8]. Additionally, automated scrolling is simulated using Selenium to load and extract older tweets that are beyond the initial page view [9].

Furthermore, throttling and user-agent spoofing are implemented to mimic human behavior and reduce the risk of being detected as a bot. By introducing randomized delays between requests and changing user-agents at regular intervals, the system closely emulates typical browsing patterns, allowing it to bypass Twitter’s anti-bot mechanisms [10]. These combined methods ensure scalable and efficient tweet collection despite the recent restrictions imposed by Twitter’s API, enabling a continuous stream of relevant data for disaster-related analysis.

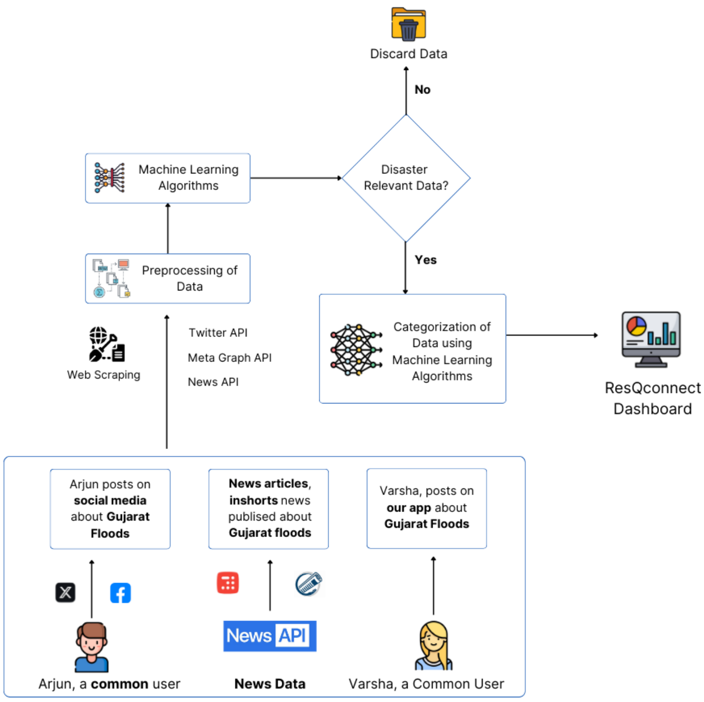

Fig 1. Methodolgy

Figure 1 illustrates the process flow of collecting data via web scraping and real-time streaming, followed by categorizing tweets using machine learning models. This flow starts from the data scraping process, streaming to Kafka topics, followed by NLP processing, classification, and storage.

3.2. Real-Time Data Ingestion and Processing Pipeline

To handle the influx of disaster-related tweets, a robust real-time data ingestion and processing pipeline is established using Apache Kafka and Apache Spark. This pipeline is critical for ensuring that the system can manage the large volume of tweets in real-time, particularly during high-traffic disaster periods.

3.3 Kafka-Based Data Ingestion

Tweets are streamed into Kafka topics, where they are tagged with disaster-related metadata such as "earthquake" or "flood." Kafka’s partitioning capability plays a significant role in this process, allowing the system to divide and handle tweets in parallel based on metadata, such as disaster type or geographic location. This enables the system to efficiently manage multiple streams of disaster-related data without bottlenecks [11]. In practice, the partitioning ensures that different disaster events are processed simultaneously, offering real-time insights on various ongoing disasters. Each Kafka topic serves as a queue for a particular category, which guarantees that no tweet is lost or delayed, even when traffic spikes. The combination of Kafka's durability and parallelism ensures a continuous flow of data into the processing system, ready for real-time analytics.

To avoid reliance on Twitter's paid API, we leveraged twscrape, a web-scraping tool that efficiently extracts tweets based on specific disaster-related queries. The extracted tweets are then streamed into Apache Kafka in real time, where each tweet is stored in a structured JSON format. The key code snippets for our pipeline are as follows:

Producer Code:

```python
from kafka import KafkaProducer
import twscrape
import asyncio

# Async function to scrape tweets and send to Kafka
async def fetch_and_send_tweets(query, kafka_topic):
    scraper = twscrape.TweetScraper(query)
    async for tweet in scraper.get_items():
        tweet_data = {'id': tweet.id, 'text': tweet.content,
                      'timestamp': tweet.date}
        producer.send(kafka_topic, value=tweet_data)
```

Consumer Code:

Here, tweets are scraped asynchronously and sent to the disaster_tweets topic in Kafka for further processing.

```python

from kafka import KafkaConsumer
import json

# Function to process tweets from Kafka
def process_tweets():
    for message in consumer:
        tweet = message.value
        print(f"Processing Tweet: {tweet['text']}
            from user: {tweet['username']}")

```

This consumer continuously listens for new tweets, processing them in real time, which is essential for tasks like disaster response monitoring.

3.4 Real-Time Processing with Spark Streaming

Once the tweets are ingested into Kafka, they are passed to Apache Spark Streaming, which processes the data in micro-batches for real-time analytics. Key tasks performed during this stage include sentiment analysis, where text mining techniques are applied to assess the sentiment of tweets. This provides immediate insight into public reactions and emotions concerning specific disasters [12]. Additionally, Spark's NLP tools are used to extract key disaster-related keywords and location data from the tweets, helping in the identification of affected regions and relevant discussions. The use of Spark ensures that the system can scale to handle massive amounts of tweet data efficiently, even during disaster events with high tweet volumes. By leveraging its distributed processing capabilities, the system guarantees that the data is processed quickly and effectively, making the insights available in real-time. This approach ensures fast, scalable, and reliable processing, which is essential during critical disaster scenarios.

3.5 Multimodal Model for Tweet Categorization

Disaster-related tweets often contain both textual and visual information, making a multimodal approach necessary for accurate categorization. To handle this, the system employs natural language processing (NLP) models for text and convolutional neural networks (CNNs) for images. The integration of these two modalities improves the classification of disaster-related tweets.

3.5.1 Text Processing using NLP Models

For text classification, BERT (Bidirectional Encoder Representations from Transformers) and XLNet models are fine-tuned on a dataset of disaster-related tweets. These models excel at contextual understanding, which is essential for accurately identifying disaster-related information within short, often informal tweets. BERT, in particular, considers the context of words bidirectionally, which allows for a more nuanced understanding of the text. This bidirectional context improves the model's ability to classify tweets with higher accuracy [13]. XLNet, on the other hand, is trained using a permutation-based language modeling approach, which allows it to handle longer sequences more effectively. This makes XLNet particularly suited for longer tweets or more complex sentence structures, where context and sequence are important for classification accuracy [14]. Both models, when fine-tuned, enhance the system's ability to differentiate disaster-related tweets from unrelated content based on their context.

3.5.2 Image Processing using CNNs

In addition to text, many disaster-related tweets include images that provide visual evidence of the event. For this, the system uses EfficientNetB3, DenseNet201, and ResNet50—all powerful CNN architectures trained on disaster imagery datasets. Among these, EfficientNetB3 demonstrated the best performance, striking a balance between accuracy and computational cost. This makes it particularly suitable for real-time image classification in disaster scenarios, where time is of the essence and resource efficiency is crucial [14]. DenseNet201 and ResNet50 also contribute to the model's ability to classify images by leveraging their depth and feature extraction capabilities. By using a combination of these CNNs, the system ensures that the visual data from tweets is processed effectively, providing additional context to the disaster categorization.

3.5.3 Multimodal Fusion Model Architecture

The integration of text and image data occurs in the multimodal fusion model, as depicted in Fig. 2. This internal flow diagram shows how features extracted from both modalities (text and image) are combined before the final classification. The fusion of both textual and visual features enables the model to more accurately detect disaster-related content. By leveraging both types of data, the system improves overall classification performance, ensuring that even tweets with limited text but strong visual content—or vice versa—are accurately categorized.

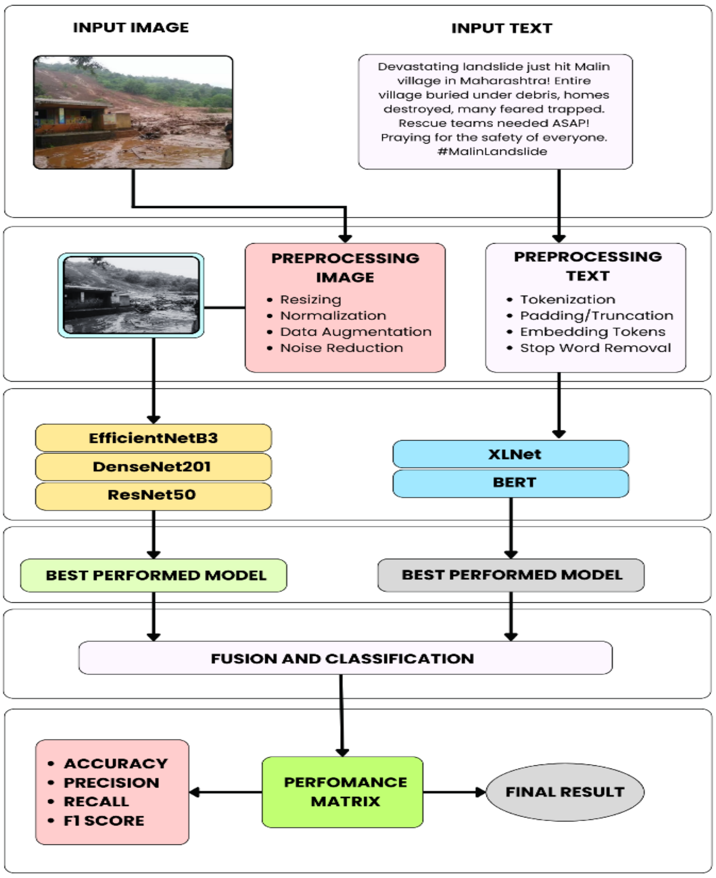

Fig 2. Multimodal Fusion Model Architecture

This multimodal approach provides a comprehensive solution to classifying disaster-related tweets, enhancing the reliability of detection by utilizing both language and imagery. 3. Fusion of Text and Image Data

The system enhances tweet classification by combining text and image features through a feature-level fusion approach. In this approach, the features extracted from both the textual data (using NLP models) and the visual data (using CNNs) are combined before being passed to the classifier. This fusion ensures that the model leverages the complementary strengths of both textual and visual data, leading to a more accurate classification of disaster-related tweets. Studies have shown that this multimodal fusion outperforms approaches that rely on a single modality, such as only text or only images, particularly in complex scenarios like disaster detection [9].

The fusion model is vital for handling tweets with mixed or incomplete data, where visual information or text alone may not be sufficient to classify the tweet accurately.

3.6. Filtering Irrelevant Tweets with Gemini LLM

To handle the vast influx of tweets, the system employs Gemini LLM, a large language model (LLM) designed specifically for filtering out irrelevant content. Tweets related to disasters can often be mixed with spam, advertisements, or other noise, which may not contribute meaningfully to disaster management efforts. Gemini LLM analyzes the context of tweets and effectively differentiates between disaster-relevant information and extraneous content. The model's ability to understand and assess tweet context in real-time significantly reduces the clutter in the dataset, ensuring that only meaningful tweets are processed and analyzed [15]. This filtering mechanism is crucial in maintaining the system's overall efficiency and focus during disaster events when accurate and relevant information is essential.

3.7. Data Storage and Search

After processing, the system stores tweets in MongoDB and utilizes Elasticsearch for indexing and searching. MongoDB is used for storing unstructured data, such as tweet content, images, and associated metadata, which makes it well-suited for handling diverse tweet formats. It ensures the scalability of the storage, as the volume of tweets can increase dramatically during disaster situations. In parallel, Elasticsearch provides the capability for fast, full-text search, enabling real-time querying and retrieval of tweets based on disaster-related keywords, geographic locations, or other metadata. This combination allows users to search the vast database quickly and efficiently. To visualize trends and provide insights into tweet patterns, the system also integrates Kibana, a data visualization tool that works seamlessly with Elasticsearch. Kibana allows real-time monitoring of tweet trends, helping disaster response teams make data-driven decisions based on the most up-to-date information [16].

3.8. Performance Evaluation

The system's performance is evaluated using several key metrics to ensure the effectiveness of tweet classification and processing. Accuracy is the primary metric used to assess the overall proportion of correctly classified tweets. However, given the imbalance in datasets—where some disaster types may be more frequent than others—additional metrics like precision, recall, and the F1-score provide a more granular view of performance. Precision measures the proportion of correctly identified disaster tweets out of all tweets classified as disaster-related, while recall evaluates the system's ability to identify all relevant disaster tweets. The F1-score, which is the harmonic mean of precision and recall, provides a balanced evaluation, particularly in cases of imbalanced datasets [17]. Finally, the system also monitors latency, measuring the time between tweet ingestion and classification. This ensures that the system can handle real-time tweet processing, an essential feature during fast-paced disaster events.

These combined efforts ensure that the system is both accurate and responsive, capable of processing large amounts of tweet data efficiently during critical times, and providing actionable insights to disaster response teams.

IV. IMPLEMENTATION

The proposed solution integrates various components for real-time disaster tweet collection, processing, and classification. The system is built using web scraping tools, Apache Kafka, Apache Spark, MongoDB, Elasticsearch, and a multimodal classification model that leverages BERT, EfficientNetB3, and other state-of-the-art machine learning tools.

4.1 Dataset

The dataset used in this research consists of social media posts with associated text and image files. Each post is classified into one of several categories, including damage-related events like floods or fires and non-damage events. The dataset structure looks like this:

Filename	Tweet	Label	Image
ad_2017-11-25_10-36-26.txt	⭐ We are really getting into the Christmas spirit...	non_damage	ad_2017-11-25_10-36-26.JPG
building_2017-10-30_17-26-34.txt	IJOY uv board has a competitive price and very...	non_damage	building_2017-10-30_17-26-34.JPG
floodwater_2017-09-04_04-46-10.txt	Arriving in Kalkundi island destroyed in #bangladesh...	flood	floodwater_2017-09-04_04-46-10.JPG
accrafloods_2015-06-06_16-59-56.txt	Hi my lovelies, check out my firsthand experience of...	flood	accrafloods_2015-06-06_16-59-56.JPG
buildingfire_2016-10-02_03-07-17.txt	The Hamilton fire service during an exercise at...	fires	buildingfire_2016-10-02_03-07-17.JPG

Table 1. Dataset overview

The dataset consists of columns for the filename, tweet (text data), label (classification of the event), and image (associated visual data). The label column indicates whether the post refers to flood, fire, or non-damage events. This multimodal dataset is essential for training models that can handle both textual and visual data to predict disaster-related events.

4.2. Data Collection using Web Scraping

Given the limitations of the Twitter API, Selenium and BeautifulSoup were employed to scrape tweets directly from Twitter's web interface. This method allows us to collect the necessary tweets, user metadata, and other related information for further processing.

Steps Involved:

1. Login Management: Selenium automates the login process and manages cookies to maintain a valid session.

2. Automated Scrolling: Selenium simulates user

3. scrolling to load older tweets dynamically.

4. Content Extraction: BeautifulSoup is used to extract tweet content, metadata (such as timestamp, user location), and associated images.

5. Proxy and Anti-Bot Mechanisms: Techniques like proxy rotation and user-agent spoofing were implemented to mimic human browsing behavior and bypass Twitter's anti-bot detection.

Once scraped, the data (tweets, metadata, and images) is streamed into Apache Kafka for further processing.

```python
from selenium import webdriver
from bs4 import BeautifulSoup

# Initialize Selenium WebDriver
driver = webdriver.Chrome(executable_path='/path/to/chromedriver')
driver.get('https://twitter.com/search?q=%23disaster')

# Scroll and load tweets
for i in range(10):
    driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
    soup = BeautifulSoup(driver.page_source, 'html.parser')
    tweets = soup.find_all('div', {'data-testid': 'tweet'})
    for tweet in tweets:
        # Extract tweet content, metadata, etc.
        print(tweet.text)

```

Code for Webscraping

Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 3s (2025)

4.3. Real-Time Data Ingestion with Apache Kafka

The scraped data is immediately streamed into Apache Kafka for real-time ingestion. Kafka acts as a message broker that handles high-throughput streams of tweets, partitioning the data into different topics based on keywords or geographic locations.

1. Kafka Producer API streams the tweets to respective topics (e.g., “#Flood”, “#Fire”).

2. Partitioning: Tweets are partitioned by disaster types (e.g., “earthquake”, “flood”) to allow for parallel processing.

This partitioning ensures scalable, real-time disaster monitoring.

4.4. Real-Time Data Processing with Apache Spark

Once the tweets are ingested, Apache Spark Streaming processes the data in micro-batches. This real-time processing pipeline performs the following tasks:

```python

from pyspark.sql import SparkSession

spark = SparkSession.builder.appName('DisasterTweetProcessing').getOrCreate()
df = spark.readStream.format("kafka").option("kafka.bootstrap.servers",
"localhost:9092").option("subscribe", "disaster_tweets").load()

# Perform sentiment analysis and location extraction
processed_tweets = df.withColumn('sentiment',
analyze_sentiment(df.value)).withColumn('location',
extract_location(df.value))

```

Code for sparksession

1. Sentiment Analysis: NLP techniques are used to assess the sentiment of tweets related to disaster events.

2. Location Extraction: Extracts disaster locations and severity from the tweet content using geospatial analysis and named entity recognition (NER).

By leveraging Spark’s distributed processing capabilities, the system can handle a massive influx of tweets, even during high-traffic periods, ensuring insights are delivered in real-time.

4.5. Multimodal Classification Model

The core of the system is a multimodal classification model designed to process both text and image data. For text classification, the system leverages BERT and XLNet, two powerful natural language processing models that have been fine-tuned on disaster-specific datasets. These models are able to capture the context and semantic meaning from the tweets, making them particularly effective for identifying and classifying disaster-related text content. In parallel, EfficientNet-B3 is used for image classification. This model has been trained on a curated dataset of disaster images, enabling it to accurately categorize visual data into classes such as flood, fire, and non-damage.

To combine the strengths of both text and image data, the system utilizes a model fusion approach. The embeddings produced by the BERT/XLNet models for text and the EfficientNet-B3 model for images are concatenated in a fusion layer. This unified representation of the tweet, which now contains both textual and visual information, is then passed through fully connected layers. The final output is a classification of the tweet, indicating whether it pertains to a flood, fire, or non-disaster event. By fusing these two modalities, the system is able to improve accuracy and leverage the complementary strengths of text and image data.

4.6. Training Procedure

In the training procedure, we utilize CrossEntropyLoss as the loss function, which is well-suited for multi-class classification tasks. This ensures that the model is optimized to correctly classify tweets into disaster categories such as flood, fire, or non-damage. For optimization, we use the AdamW optimizer with a learning rate of 1e-4. AdamW is known for its efficiency in handling large-scale datasets and includes built-in weight decay regularization, which helps prevent overfitting during training.

The model is trained over 20 epochs, with early stopping in place based on the validation accuracy. This ensures that training is halted when no significant improvement is observed in validation performance, reducing overfitting and unnecessary training time. During each epoch, the model is switched to training mode, and the loss is computed using both text and image inputs. The optimizer updates the model weights based on the calculated gradients. For each batch, the loss is computed, backpropagated, and accumulated, allowing us to track the running loss throughout the training process. The loss for each epoch is printed at the end to monitor the model's learning progress

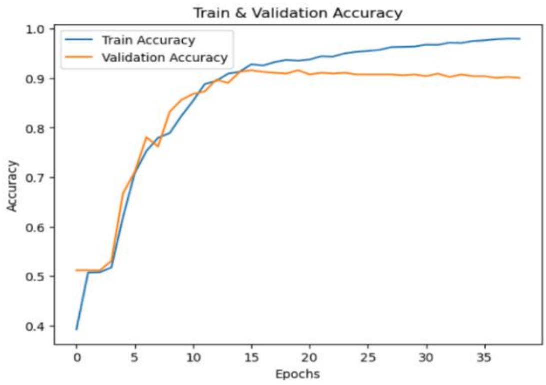

Fig 3. Train & Validation Accuracy Over Epochs

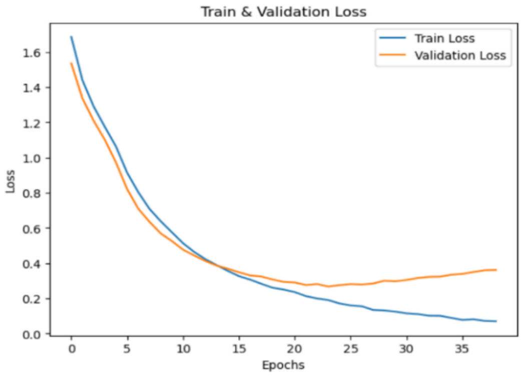

Fig 4. Train & Validation Accuracy Over Epochs

4.7. Evaluation Metrics

The evaluation of the model's performance is carried out using several key metrics. Accuracy provides a basic measure of overall correctness by determining the proportion of correct predictions. To further assess the model's effectiveness in distinguishing between different disaster categories, Precision, Recall, and the F1-Score are used. These metrics provide deeper insights, especially in handling imbalanced data by focusing on the balance between correctly identified disasters and the false positives or negatives.

Additionally, the Confusion Matrix is used to offer a more detailed view of the model's classification performance. It visualizes how well the model is distinguishing between the three classes—flood, fire, and non-damage—by showing the counts of true positives, false positives, true negatives, and false negatives for each class.

Moreover, throughout the training process, we track training loss, validation loss, training accuracy, and validation accuracy. These metrics, plotted over the epochs, help in analyzing the model's convergence behavior, indicating whether the model is improving, overfitting, or underfitting. By monitoring these metrics, we can adjust the model training to optimize its performance for disaster tweet classification.

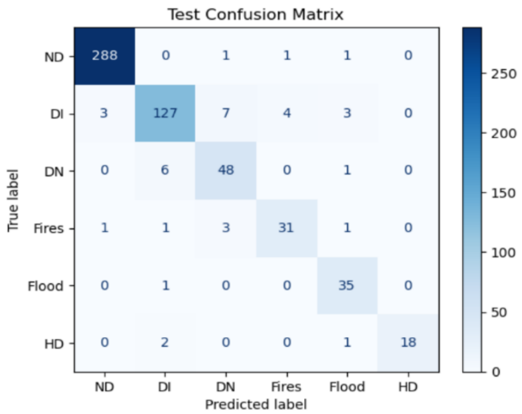

Fig 5. Confusion Matrix

This implementation outlines the full pipeline of our disaster tweet processing system, from web scraping and real-time streaming with Kafka, to multimodal classification using BERT and EfficientNet, tweet filtering with Gemini LLM, and data storage with MongoDB and Elasticsearch. The system delivers real-time disaster insights through a combination of advanced data processing techniques, machine learning models, and visualization tools.

V. RESULTS AND DISCUSSION

The results of the comparative analysis between the EfficientNet-BERT and ResNet-BERT models demonstrate distinct performance characteristics in classifying disaster-related tweets. Both models achieved high accuracy rates, but EfficientNet-BERT outperformed ResNet-BERT across all key metrics, indicating its superior ability to differentiate between relevant and irrelevant tweets in the context of disaster management.

Metrics	EfficientNet-BERT	ResNet-BERT
Accuracy	0.9366	0.9250
Precision	0.9385	0.9200
Recall	0.9366	0.9180
F1 Score	0.9368	0.9190

Table 2. Summary of performance metrics

5.1. Model Performance

The EfficientNet-BERT model achieved an impressive accuracy of **93.66%**, indicating that it correctly classified a high percentage of tweets across the flood, fire, and non-damage categories. This suggests that the model has learned to differentiate effectively between relevant and irrelevant tweets in the context of disaster management. In contrast, the ResNet-BERT model performed well with an accuracy of **92.50%**, but demonstrated slightly lower scores across all metrics compared to

EfficientNet-BERT. This indicates that, while ResNet-BERT is a strong model, it may not be as finely tuned for this specific classification task.

5.2. Precision and Recall

EfficientNet-BERT's precision of **0.9385** signifies a robust ability to minimize false positives, meaning that when it predicts a tweet as relevant to disasters, it is correct **93.85%** of the time. This is crucial in disaster response contexts, where false alarms can lead to unnecessary resource allocation. The recall score of **0.9366** highlights that the model effectively captures the majority of actual disaster-related tweets. This balance between precision and recall is vital for ensuring that significant tweets are not missed, which can be critical during emergencies.

5.3. F1 Score

The F1 Score of **0.9368** for EfficientNet-BERT reflects a harmonious trade-off between precision and recall. It indicates that the model is robust in classifying tweets without being overly sensitive to either false positives or false negatives. On the other hand, the F1 Score of **0.9190** for ResNet-BERT, while commendable, suggests that there may be a slight trade-off in performance. This potentially makes EfficientNet-BERT the more reliable option in scenarios where both precision and recall are equally important.

VI. CONCLUSION AND FUTURE WORK

This research paper presents the development of an AI-driven disaster management system that harnesses real-time social media data to enhance disaster response. By integrating data from platforms like Twitter, Instagram, and Facebook, the system collects both textual and image-based information critical for situational awareness during disasters. The system employs advanced machine learning models, such as BERT and XLNet for text analysis, and EfficientNet and ResNet50 for image classification, to accurately categorize and assess the severity of disaster events. This multimodal approach significantly improves the precision of disaster detection, providing actionable insights that help responders prioritize and allocate resources effectively.

The system’s real-time data ingestion and processing pipeline, built using Apache Kafka and Apache Spark, ensures scalability and efficiency even during high-traffic disaster periods. By fusing textual and visual data, the system achieves high classification accuracy, ensuring that relevant disaster information is quickly identified. The architecture’s scalability allows for processing large volumes of social media data without bottlenecks, making it a robust solution for managing disaster-related data in real time. Ultimately, this system demonstrates the potential of social media analytics in improving disaster response, helping save lives by providing timely and actionable insights.

Looking forward, the system will be expanded into a web portal aimed at providing disaster response agencies in India with a powerful tool for analysing and categorizing disaster-related data from significant repositories, particularly social media. This portal will automate the collection, processing, and analysis of data, delivering real-time actionable insights that allow agencies to respond more quickly and efficiently. By reducing response times, the portal will help agencies save valuable time, money, and resources, while also enhancing the accuracy and relevance of the data used for decision-making. This future development will streamline the process of disaster monitoring and management, offering a centralized platform where agencies can access critical information, ultimately improving disaster preparedness and response.

References

[1] Wiegmann, M., Kersten, J., Senaratne, H., Potthast, M., Klan, F., and Stein, B.: *"Opportunities and risks of disaster data from social media: a systematic review of incident information,"* Nat. Hazards Earth Syst. Sci., 21, 1431–1444, https://doi.org/10.5194/nhess-21-1431-2021, 2021.

[2] A. Pal, J. Wang, Y. Wu, K. Kant, Z. Liu, and K. Sato, *"Social Media Driven Big Data Analysis for Disaster Situation Awareness: A Tutorial,"* IEEE Transactions on Big Data, vol. 9, no. 1, pp. 1-21, Feb. 2023, doi: 10.1109/TBDATA.2022.3158431.

[3] H. Hao and Y. Wang, "Leveraging multimodal social media data for rapid disaster damage assessment," *International Journal of Disaster Risk Reduction*, vol. 51, 2020, Art. no. 101760. https://doi.org/10.1016/j.ijdrr.2020.101760

[4] Jaebeom You, Kisung Lee, Hyuk-Yoon Kwon, "*DeepScraper: A complete and efficient tweet scraping method using authenticated multiprocessing*," Data & Knowledge Engineering, Volume 149, 2024, 102260, ISSN 0169-023X, https://doi.org/10.1016/j.datak.2023.102260.

[5] Domala, Jayashree & Dogra, Manmohan & Masrani, Vinit & Fernandes, Dwayne & D'souza, Kevin & Fernandes, Delicia & Carvalho, Tejal. (2020). *"Automated Identification of Disaster News for Crisis Management using Machine Learning and Natural Language Processing."* 503-508. 10.1109/ICESC48915.2020.9156031.

[6] Mishra, Sameer & Bisen, Atharva & Mundhada, Soham & Singh, Utkarsh & Bongirwar, Vrushali. (2022). *"Disaster Information Verification and Validation Application Using Machine Learning. International Journal of Next-Generation Computing".* 10.47164/ijngc.v13i5.907.

[7] Ningsih, A & Id Hadiana, Asep. (2021). Disaster Tweets Classification in Disaster Response using Bidirectional Encoder Representations from Transformer (BERT). IOP Conference Series: Materials Science and Engineering. 1115. 012032. 10.1088/1757-899X/1115/1/012032.

[8] Kusumasari, B., Prabowo, N.P.A. *"Scraping social media data for disaster communication: how the pattern of Twitter users affects disasters in Asia and the Pacific."Nat Hazards* **103**, (2020). https://doi.org/10.1007/s11069-020-04136-z

[9] M. Laya, M. K. Delimayanti, A. Mardiyono, F. Setianingrum, A. Mahmudah and D. Anggraini, "Classification of Natural Disaster on Online News Data Using Machine Learning," *2021 5th International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*, Medan, Indonesia, 2021, pp. 42-46, doi: 10.1109/ELTICOM53303.2021.9590125.

[10] A., Aswathy., Rekha, Prabha., Lakshmi, Gopal., Divya, Pullarkatt., Maneesha, Vinodini, Ramesh. (2022). *"An efficient twitter data collection and analytics framework for effective disaster management."* doi: 0.1109/DELCON54057.2022.9753627

[11] Meghna, Jayan., Lija, Jacob. (2023). 1. "*Categorizing Disaster Tweets Using Learning Based Models for Emergency Crisis Management."* doi: 10.1109/ICACCS57279.2023.10113105

[12] P., Gnanasekaran. (2023). 1. *"Juncture of Text Preprocessing Techniques & Extracting Sentiment Analyzing of Micro-Blog Based on Machine Learning Algorithms."* doi: 10.1109/icses60034.2023.10465450

[13] Parth, Mahajan., Pranshu, Raghuwanshi., Hardik, Setia., Princy, Randhawa. (2024). 1. *"A Multi-Model Approach for Disaster-Related Tweets."* doi:10.57159/gadl.jcmm.3.2.240125

[14] Pardeep, Singh., Monika, Singh., P., Das., Satish, Chand. (2024). 2. *"Leveraging domain-adapted transformer based models for disaster-specific tweet classification."* doi: 10.3233/jifs-219419

[15] Wei, Chen., Janet, Fang. (2024). 1. *"Optimizing AI-Driven Disaster Management through LLMs."* doi: 10.20944/preprints202407.1446.v1

[16] Frankie, Frankie., Yeremia, Alfa, Susetyo. (2023). 2. *"Implementation of text indexing system in web-based document search application using mongodb."* doi: 10.52436/1.jutif.2023.4.5.959

[17] Siva, Krishna, Dasari., G., Srinivas., P., Jayarama, Reddy. (2024). 2. *"A deep parallel hybrid fusion model for disaster tweet classification on Twitter data."* Decision Analytics Journal, doi: 10.1016/j.dajour.2024.100453