Real-Time Disaster Detection on Social Media Using Advanced Machine Learning Techniques

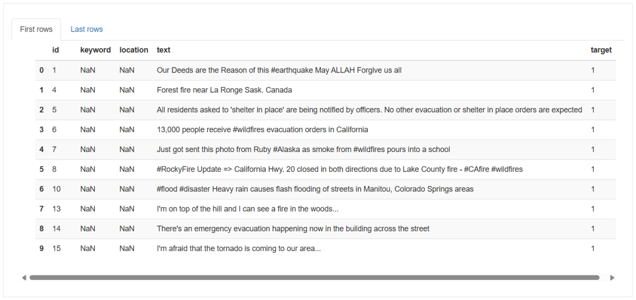
Prachi Katare (B21CS058)
Chandana N.
Computer Science and Engineering
Jodhpur , India
katare.2@iitj.ac.in

***Abstract*—The increased popularity of microblogging websites such as Twitter has turned them into indispensable conduits for near-real-time disaster communication. Still, identifying real disaster-related messages from metaphorical or non-crisis updates is not yet easy due to the noisy quality of social media data. The current work is a machine learning approach for discovering disaster-related tweets in real-time to assist with emergency response operations. We assess two pipelines: a baseline using TF-IDF vectorization with Logistic Regression, Multinomial Naive Bayes, and Random Forest classifiers, and an innovative approach utilizing BERT embeddings with an XGBoost classifier to improve contextual awareness. The collection includes about 7,600 labeled disaster or non-disaster tweets. The BERT + XGBoost model performs better than the standard pipeline, with an accuracy of 0.8207 and F1 score of 0.7760, showing its enhanced capability to manage semantic subtleties. We implemented the system as a web application on Lovable, an AI-powered platform, allowing quick prototyping and real-time disaster detection for emergency responders and the general public.**

***Index Terms*—Disaster detection, Twitter, machine learning, TF-IDF, BERT, XGBoost, social media analysis, natural language processing, web deployment**

I. Introduction

Social media sites, especially Twitter, have become indispensable for immediate communication during crises, like earthquakes, floods, and accidents. Tweets offer real-time reports from affected regions, which facilitate swift transmission of key information. Nonetheless, the non-formal and vague nature of social media texts is a severe challenge. For example, phrases like ”ABLAZE” can be referring to a fire in the real sense or just a figurative statement. Precise identification of authentic disaster posts is critical so that emergency response agencies can channel resources effectively.



First rows | Last rows

	id	keyword	location	text	target
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake May ALLAH Forgive us all	1
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1
2	5	NaN	NaN	All residents asked to 'shelter in place' are being notified by officers. No other evacuation or shelter in place orders are expected	1
3	6	NaN	NaN	13,000 people receive #wildfires evacuation orders in California	1
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as smoke from #wildfires pours into a school	1
5	8	NaN	NaN	#RockyFire Update => California Hwy. 20 closed in both directions due to Lake County fire - #CAfire #wildfires	1
6	10	NaN	NaN	#flood #disaster Heavy rain causes flash flooding of streets in Manitou, Colorado Springs areas	1
7	13	NaN	NaN	I'm on top of the hill and I can see a fire in the woods...	1
8	14	NaN	NaN	There's an emergency evacuation happening now in the building across the street	1
9	15	NaN	NaN	I'm afraid that the tornado is coming to our area...	1

Fig. 1. Sample tweets from the dataset, showing disaster (target=1) and non-disaster (target=0) examples.

This research creates and contrasts two machine learning pipelines for real-time disaster detection on Twitter: a conventional pipeline based on TF-IDF vectorization with traditional models and an innovative pipeline based on BERT embeddings with an XGBoost classifier to identify contextual relationships. We implemented the system as a web application with Lovable, an AI-powered platform, to facilitate rapid prototyping and real-time disaster detection for emergency responders and the public. The data set includes about 7,600 text tweets, keywords, locations, and binary labels (1 for disaster, 0 for non-disaster). We measure both pipelines with accuracy and F1 score, with a focus on the F1 score for balanced performance. The BERT-based method shows better results, and the web deployment generated by Lovable increases its practical applicability.

II. Related Work

Social media disaster detection has been extensively researched. Initial methods applied keyword filtering for identifying events such as earthquakes [1], but these were confounded by ambiguous words. Supervised machine learning algorithms like Support Vector Machines had better performance through the use of text features like

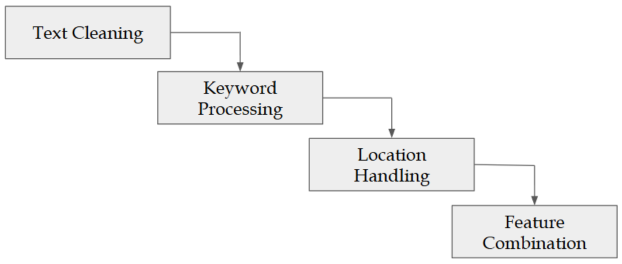


Fig. 2. Preprocessing pipeline for cleaning tweet text and keywords.

TF-IDF for sentiment classification during disasters [2]. Recent transformer-based models like BERT [3] have been found to be potentially useful in text classification because of their contextualized understanding. Deep learning methods for classifying tweets [4] obtained high accuracy with the need of high computational effort. Our study compares a pipeline based on TF-IDF against a pipeline consisting of BERT + XGBoost, and analyzes their trade-offs, then proposes a web deployment based on Lovable, motivated by disaster response web platforms [5].

III. Methodology

A. Dataset Description

The dataset comprises approximately 7,600 tweets from a CSV file (‘train.csv’), with columns: id, text, keyword, location, target (1=disaster, 0=non-disaster). The dataset exhibits class imbalance, with slightly more non-disaster tweets. Figure 1 shows sample tweets.

B. Data Preprocessing

Both pipelines share preprocessing steps:

1) **Text Cleaning**: Remove emojis, URLs, hashtags, mentions, and special characters.
2) **Keyword Processing**: Replace “%20” with spaces, lemmatize keywords, fill missing with “None.”
3) **Location Handling**: Fill missing locations with “None.”
4) **Feature Combination**: Concatenate cleaned text and lemmatized keyword.

C. Traditional Pipeline (TF-IDF + Classical Models)

The traditional pipeline emphasizes efficiency:

1) **Feature Extraction**: Apply TfidfVectorizer with English stopwords.
2) **Train-Test Split**: 80% training, 20% testing (random_state=42).

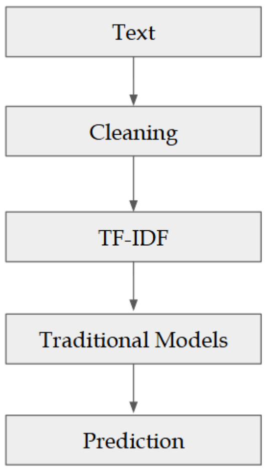


Fig. 3. Preprocessing pipeline for cleaning tweet text and keywords.

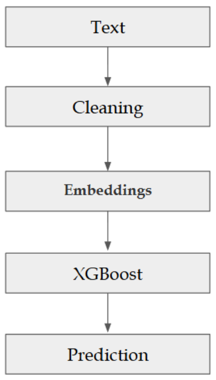


Fig. 4. BERT + XGBoost pipeline, showing embedding generation and classification.

3) **Models**: Logistic Regression, Multinomial Naive Bayes, Random Forest.
4) **Evaluation**: Compute accuracy and F1 score.

D. Advanced Pipeline (BERT + XGBoost)

The advanced pipeline prioritizes contextual understanding:

1) **Feature Extraction**: Use BERT’s tokenizer and model (‘bert-base-uncased’) for 768-dimensional CLS embeddings.
2) **Train-Test Split**: Same as traditional pipeline.
3) **Model**: XGBoost classifier (objective=‘binary:logistic’).
4) **Evaluation**: Compute accuracy and F1 score.

E. Model Formulations

We formalize each model’s operation:

1) Logistic Regression: Predicts disaster probability:

$$P(y=1|\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}} \tag{1}$$

where $\mathbf{x}$ is the TF-IDF vector, $\mathbf{w}$ is learned weights. The loss is:

$$J(\mathbf{w}) = -\frac{1}{N}\sum_{i=1}^{N}\Big[y_i \log(P(y_i=1|\mathbf{x}_i)) + (1-y_i)\log(1-P(y_i=1|\mathbf{x}_i))\Big] \tag{2}$$

2) Multinomial Naive Bayes: Computes class probability:

$$P(y=c|\mathbf{x}) \propto P(y=c)\prod_{j=1}^{V} P(w_j|y=c)^{x_j} \tag{3}$$

where $\mathbf{x}$ is word counts, V is vocabulary size, assuming word independence.

3) Random Forest: Minimizes Gini impurity per split:

$$G = 1 - \sum_{c=1}^{C} p_c^2 \tag{4}$$

where p_c is class proportion. Trees vote for the final label.

4) BERT: Generates embeddings via self-attention:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

where Q, K, V are query, key, value matrices, d_k is dimension. Outputs 768-dimensional CLS embeddings.

5) XGBoost: Minimizes:

$$\mathcal{L} = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{k=1}^{T}\Omega(f_k) \tag{6}$$

where l is log-loss, $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|\mathbf{w}\|^2$ regularizes trees.

F. Hyperparameter Tuning

Grid search optimized:

- Logistic Regression: C=1.
- Random Forest: n_estimators=100.
- XGBoost: learning_rate=0.1, max_depth=6.

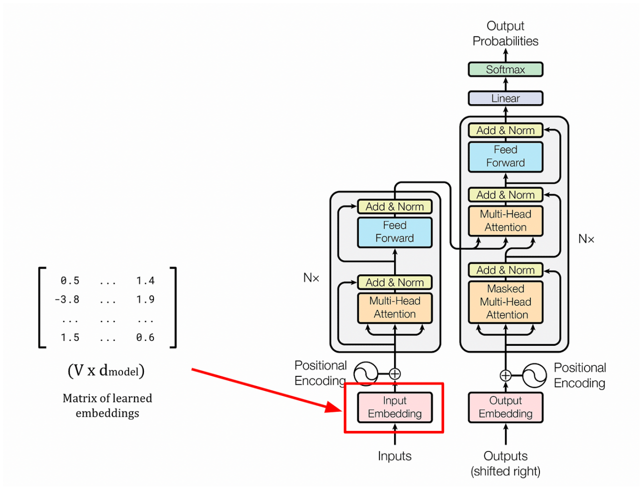
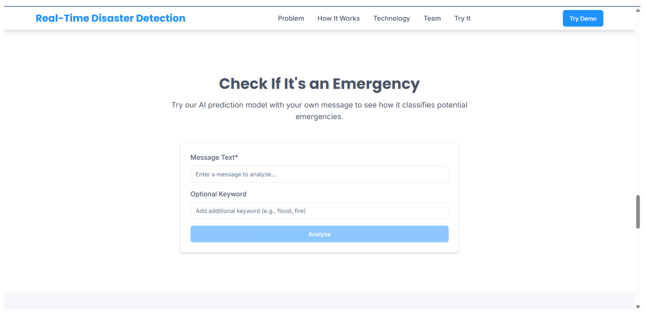


Fig. 5. BERT's self-attention mechanism, illustrating contextual embedding generation.



Check If It's an Emergency

Try our AI prediction model with your own message to see how it classifies potential emergencies.

Message Text*

Enter a message to analyze...

Optional Keyword

Add additional keyword (e.g., flood, fire)

Fig. 6. Screenshot of the Lovable-generated web application for disaster prediction.

G. Web Deployment with Lovable

The web app was created with Lovable, a text-to-ui AI tool that creates responsive front-end interfaces from text prompts. We gave it a prompt, "Build a web app for tweet-based disaster detection with input fields and prediction display," which resulted in a React-based interface with shadcn/ui components. Users enter a tweet and an optional keyword through text fields; the BERT + XGBoost model, which is loaded through a serialized file, handles the input and shows predictions (e.g., "Your request is accepted. Help will arrive soon!" for disasters). The app accommodates GitHub exporting for collaboration purposes and is run locally for testing. Lovable's quick prototyping allowed minutes of deployment, improving accessibility to non-technical users such as emergency responders.

TABLE I
MODEL PERFORMANCE COMPARISON

Model	Accuracy	F1 Score
Logistic Regression	0.7859	0.7371
Multinomial Naive Bayes	0.7892	0.7323
Random Forest	0.7754	0.7159
BERT + XGBoost	**0.8207**	**0.7760**

IV. RESULTS

A. Performance Comparison

Table I summarizes model performance.

B. Web Application Results

The Lovable-generated web application mirrors the BERT + XGBoost pipeline's performance, handling inputs in less than 2 seconds. Sample predictions are:

- **Disaster**: "Help my house is on fire" (keyword: "fire") $\rightarrow$ "Your request is accepted. Help will arrive soon!"
- **Non-disaster**: "Great day at the beach" (no keyword) $\rightarrow$ "This does not appear to be an emergency."

Figure 6 shows the interface with a sample disaster prediction. The clean, responsive design enhances usability, though scalability testing is ongoing.

V. DISCUSSION

The BERT + XGBoost pipeline is outstanding thanks to contextual awareness by BERT, able to deal correctly with ambiguous terms. The Lovable-generated web application streamlines deployment by providing a working UI in minutes and is suitable for non-technical users. It has a React-based framework guaranteeing responsiveness and GitHub integration aiding collaboration. Despite that, computational intensity (BERT embeddings take 10 minutes on 7,600 tweets) and possible limitations of Lovable, like tweaking for more complex features, pose issues. The conventional pipeline has the benefit of a lightweight counterpart.

A. Limitations and Future Work

Limitations are dataset size (7,600 tweets), English-language only tweets, noisy geo-data, and web scalability. Future work is integrating real-time Twitter API data into the Lovable app, visualization dashboards (e.g., disaster maps), and cloud hosting optimization.

VI. CONCLUSION

This work offers a machine learning system for real-time disaster detection on Twitter, with the BERT + XGBoost pipeline showing superior performance (accuracy: 0.8207, F1: 0.7760). The web application generated by Lovable showcases the system's operational readiness, using AI for both modeling and deployment. Future development will target scalability and real-time integration.

ACKNOWLEDGMENT

We thank the open-source community for libraries like NLTK, Transformers, and XGBoost, and Lovable for enabling rapid web deployment.

REFERENCES

[1] Sakaki, T. and others, "Earthquake shakes Twitter users: Real-time event detection by social sensors," Proc. 19th Int. Conf. World Wide Web (WWW), pp. 851–860, 2010. https://doi.org/10.1145/1772690.1772777

Content: This article discusses Twitter as a real-time sensor for earthquake detection. It suggests a probabilistic model based on keywords (e.g., "earthquake") and locations of tweets for seismic event detection, with a speed of reporting higher than traditional seismology in Japan. It brought about temporal and spatial analysis for tracking events, fundamental to social media disaster monitoring.

[2] Caragea, C. and others, "Mapping moods: Geo-mapped sentiment analysis during Hurricane Sandy," Proc. 11th Int. ISCRAM Conf., pp. 312–319, 2014. https://www.iscram.org/legacy/ISCRAM2014/papers/p128-caragea.pdf

Content: This research examines Twitter sentiment during Hurricane Sandy, classifying tweets by mood (e.g., fear, relief) using TF-IDF features and Support Vector Machines. Sentiments are mapped geographically and illustrated as how emotions are associated with proximity to affected areas, supporting disaster response planning.

[3] Devlin, J. and others, "BERT: Pre-training of deep bidirectional transformers for language understanding," Proc. NAACL-HLT, pp. 4171–4186, 2019. https://arxiv.org/abs/1810.04805

Content: This paper introduces BERT, a transformer-based model pre-trained on big text corpora to learn contextual word representations. Fine-tuned for text classification tasks, it is state-of-the-art, which is essential for our BERT + XGBoost pipeline's success in deciphering subtle tweet semantics.

[4] Li, X. and others, “Deep learning for real-time social media text classification,” Proc. IEEE Int. Conf. Big Data, pp. 1867–1876, 2018. https://doi.org/10.1109/BigData.2018.8622521

Content: This paper employs convolutional neural networks to real-time classify tweets related to disasters, specifically hurricanes. It employs word embeddings and obtains high accuracy but mentions computational intensity, consistent with our results regarding BERT’s high resource usage.

[5] Imran, M. and others, “AIDR: Artificial intelligence for disaster response,” Proc. 25th Int. Conf. World Wide Web (WWW), pp. 159–162, 2016. https://doi.org/10.1145/2872518.2890077

Content: This paper introduces AIDR, a web-based machine learning platform to filter and classify disaster tweets in real time. It aids humanitarian response by giving priority to relevant messages, motivating our Lovable-based deployment for accessible disaster monitoring.

Annexure

A. Dataset Description

The dataset (‘train.csv’) contains 7,600 tweets with columns: id, text, keyword, location, target. Sample rows are shown in Figure 1.

B. Code and Web Availability

- Traditional Pipeline: https://colab.research.google.com/drive/15hrQOInKxSgsRRIia3Mbuh3JjtBlGwA4
- BERT + XGBoost Pipeline: https://colab.research.google.com/drive/1tZhPJQMBYXv8UHbCJv4_35GTt10SfaUJ
- Web Application: https://alert-ai-response-team.lovable.app/