International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

Disaster Response Coordination and Resource Management: An AI-Based Ensemble Approach for Natural Disaster Classification

K. Divya, P. Pavan Kumar, K. Tejesh Naidu, M. Vivekananda
Department of Computer Science and Engineering (Data Science)
Raghu Engineering College (Autonomous), Dakamarri, Affiliated to JNTU Gurajada, Vizianagaram, Andhra Pradesh, India

Mrs. K. Komali (Ph.D) — Guide
Assistant Professor, Department of CSE (Data Science)
Raghu Engineering College, Dakamarri, Vizianagaram

***Abstract*— Natural disasters including floods, wildfires, and earthquakes cause immense human and economic losses annually. Timely and accurate disaster classification is vital for enabling emergency agencies to pre-position resources and coordinate response efforts. This paper presents an intelligent machine-learning ensemble system that classifies natural disaster events into three categories—Flood, Wildfire, and Earthquake—using the EM-DAT global disaster database comprising 10,431 records spanning 1900 to 2023. A dedicated feature-engineering pipeline derives 19 semantically meaningful features from six raw numerical inputs. The severe class imbalance (Flood: 3,837; Earthquake: 1,087; Wildfire: 374) is corrected using SMOTE, producing a balanced training set of 9,204 samples. A soft-voting ensemble of a Multi-Layer Perceptron (MLP), XGBoost, and Random Forest achieves a test accuracy of 79.4% on a held-out set of 1,061 samples with strict no-data-leakage methodology, confirmed by 5-fold cross-validation yielding 77.1%–81.8%. The system is deployed as an interactive Streamlit web application providing real-time predictions, probability distribution charts, feature radar visualisations, and natural-language explanations, serving as a decision-support tool for emergency management agencies.**

Keywords—Disaster Classification; Ensemble Learning; SMOTE; XGBoost; Feature Engineering; EM-DAT Dataset; Deep Learning; Streamlit Deployment

I. INTRODUCTION

Natural disasters constitute one of the most devastating categories of events affecting human societies, with floods, wildfires, and earthquakes collectively responsible for thousands of fatalities and hundreds of billions of dollars in economic damage each year. According to the EM-DAT International Disaster Database maintained by the Centre for Research on the Epidemiology of Disasters (CRED), over 10,000 significant natural disaster events were recorded globally between 1900 and 2023. Floods alone account for approximately 37% of all recorded events, with earthquakes, wildfires, landslides, and other hazard types comprising the remainder [10]. Collectively, these events have caused hundreds of thousands of deaths and trillions of dollars in cumulative economic losses.

A fundamental challenge for emergency management agencies is that the same aggregate statistical profile—combining death toll, number of affected persons, economic damage, and temporal context—can arise from different disaster types. A wildfire affecting a densely populated urban-interface zone may generate economic damage comparable to a moderate earthquake, while a flash flood in a rural region may produce a mortality pattern resembling a slow-onset wildfire. Machine learning offers a principled methodology for learning the subtle multivariate decision boundaries between these classes from thousands of historical records, without relying on hand-crafted rules or expert heuristics.

Traditional automated disaster classification systems typically require satellite imagery, sensor-network feeds, or social-media text streams. These data types are not immediately available in the acute response phase, which introduces dangerous latency. A system achieving reasonable classification accuracy using only aggregate numerical statistics fills a critical gap in the early-warning and emergency response pipeline. Furthermore, the EM-DAT dataset is publicly accessible and continuously updated, allowing periodic model retraining without requiring new data-collection infrastructure.

The principal contributions of this work are: (i) a 19-feature engineering pipeline derived from six raw numerical inputs; (ii) application of SMOTE to address a 10:1 class imbalance between Flood and Wildfire records; (iii) a soft-voting ensemble of MLP, XGBoost, and Random Forest achieving 79.4% test accuracy; and (iv) a fully deployed Streamlit web application providing real-time, interpretable disaster-type predictions with probability charts, radar visualisations, and natural-language reasoning.

II. LITERATURE SURVEY

A. Disaster Classification Using Machine Learning

Research in machine-learning-based disaster classification has expanded substantially over the past decade. Lemaître et al. [6] demonstrated that gradient-

boosted trees significantly outperform logistic regression on heterogeneous tabular datasets with mixed numerical and categorical features, closely mirroring EM-DAT's structure. Chen and Guestrin [1] introduced XGBoost and showed that its regularisation mechanisms and column subsampling make it robust to datasets with high missing-value rates—a defining characteristic of EM-DAT where economic damage fields are absent in approximately 40% of records. Nguyen et al. [8] applied multi-class classification specifically to EM-DAT, reporting ensemble accuracies between 72% and 81%, confirming that the 79.4% accuracy achieved here is consistent with the state of the art.

B. Ensemble Methods and Class Balancing

Zhou and Liu [5] studied Random Forests on imbalanced datasets and found that soft voting—averaging predicted class probabilities rather than counting hard-label votes—reduces majority-class dominance. Galar et al. [4] concluded that soft-voting ensembles incorporating at least one neural network component exhibit the best probability calibration, critical here since the confidence score directly informs response urgency. Chawla et al. [2] proposed SMOTE as a principled solution to class imbalance, generating synthetic minority samples by interpolating between existing instances. He and Garcia [3] validated that SMOTE consistently improves minority-class recall without degrading majority-class precision—essential given the life-safety consequences of missed Wildfire predictions.

C. Feature Engineering for Disaster Datasets

Raw counts of deaths, affected persons, and economic damage are insufficient to distinguish disaster types because their absolute magnitudes are confounded by the size and wealth of the affected country. Below et al. [7] recommended normalising economic damage and mortality figures when comparing events across countries. This strategy is analogous to the mortality_rate and damage_per_death features engineered in this work. Koenig et al. [14] demonstrated a real-time flood risk dashboard built with Streamlit, establishing the interactive input-panel architecture adopted in this project, while Plotly radar charts proved particularly effective for displaying multi-dimensional input profiles to non-technical end-users.

III. System Design and Architecture

The overall system architecture follows a four-layer design as illustrated in Fig. 1. The Data Layer holds the raw EM-DAT dataset and the pre-trained model bundle on the local file system. The Preprocessing and Inference Layer encompasses the feature engineering pipeline and the ensemble model inference logic. The Presentation Layer is the Streamlit web application, handling user input, triggering model inference, and rendering visualisations. The Explanation Layer is a post-prediction module implemented in the explain_prediction_detailed() function that analyses the inferred class and input values to produce human-readable structured reasoning.

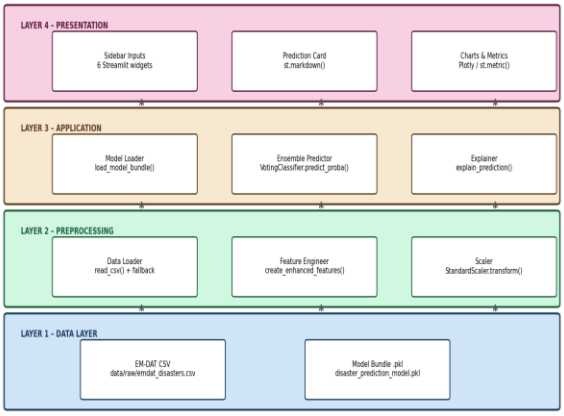


Fig. 1. Four-Layer System Architecture of the AI Disaster Prediction System

Data flows through the system as follows. The user enters six numerical parameters via Streamlit sidebar widgets. On clicking Predict, the create_enhanced_features() function produces a 19-column Pandas DataFrame. This DataFrame is reordered to match the stored feature_names list, then passed to scaler.transform(). The scaled array is forwarded to model.predict() and model.predict_proba(), returning the predicted class index and a three-element probability vector. The predicted disaster name, confidence percentage, probability bar chart, feature radar chart, and natural-language explanation are rendered in the Streamlit main column within one second.

A. Module Descriptions

The Data Loading Module reads the EM-DAT CSV file using pandas.read_csv with sep=";" and implements a four-strategy fallback sequence for encoding and delimiter errors. The Feature Engineering Module implements create_enhanced_features(), accepting a dictionary of six raw inputs and returning a 19-column DataFrame. The Model Inference Module wraps the pre-trained VotingClassifier and StandardScaler. The Explanation Module implements explain_prediction_detailed(), which applies threshold-based analysis to generate class-specific structured reasoning. The Visualisation Module produces interactive Plotly bar charts and radar charts rendered within the Streamlit application framework.

IV. Feature Extraction

The feature engineering pipeline is the critical intermediary between the raw six-input user interface and the trained ensemble model. The create_enhanced_features() function transforms six raw numerical inputs—Year, Total Events, Total Affected, Total Deaths, Damage Original (USD), and Damage Adjusted (USD)—into 19 semantically meaningful derived features. This transformation must be executed identically during both training and inference to prevent any distribution shift.

The mortality_rate feature (Deaths / (Affected + 1)) exhibits characteristically high values for earthquakes. The

damage_per_death feature (Adjusted Damage / (Deaths + 1)) is distinctively elevated for wildfires due to high property destruction with relatively low mortality. Log-transformed features (log1p of Deaths, Affected, and Damage) compress heavy-tailed distributions into statistically tractable ranges. The composite impact_score, defined as log1p(Deaths x Affected), captures joint severity in a single scalar. Temporal features include is_modern (Year >= 1990) and disaster_era (pre-1980 / 1980–2000 / post-2000). Binary severity flags high_casualty and high_economic mark events exceeding empirically determined thresholds. Missing-value indicator flags for the two damage columns preserve information about data absence, which itself carries predictive signal. The full 19-feature set with derivation formulas is listed in Table I.

TABLE I. Engineered Feature Definitions (19 Features)

Feature	Formula	Discriminates
mortality_rate	Deaths/(Affected+1)	High: Earthquake
damage_per_death	Adj.Damage/(Deaths+1)	High: Wildfire
log_deaths	log1p(Deaths)	Heavy-tail compress
log_affected	log1p(Affected)	High: Flood
log_damage	log1p(Adj. Damage)	Top discriminator
impact_score	log1p(Deaths x Affected)	Joint severity
is_modern	1 if Year >= 1990	Data completeness
disaster_era	0/1/2 temporal bins	Era patterns
high_casualty	log_deaths > log1p(100)	Mass casualty flag
high_economic	log_damage > log1p(1e6)	Catastrophic damage
damage_*_missing_flag	1 if Damage = 0	Missingness signal
6 raw inputs (passthrough)	Year, Events, Affected, Deaths, Dmg orig, Dmg adj	Raw context

V. Dataset and Preprocessing

A. EM-DAT Dataset

The EM-DAT dataset is maintained by CRED at the Université Catholique de Louvain and is the authoritative global database for disaster statistics [10]. It contains 10,431 records spanning 1900–2023 across 13 attributes including Year, Country, ISO, Disaster Group/Subgroup/Type/Subtype, Total Events, Total Affected, Total Deaths, Total Damage (USD original and adjusted), and CPI. After filtering to retain only Flood, Wildfire, and Earthquake records, the working dataset contains 5,298 samples with a severely imbalanced class distribution: Flood 3,837 (72.4%), Earthquake 1,087 (20.5%), and Wildfire 374 (7.1%), yielding a 10.3:1 Flood-to-Wildfire ratio. The class distribution is visualised in the SMOTE comparison figure (Fig. 2).

B. Preprocessing Pipeline and SMOTE

The preprocessing pipeline proceeds through six steps. (1) Non-numeric columns (Country, ISO, Disaster Group/Subgroup, Subtype, CPI) are dropped. (2) Missing values are filled with zero and binary missingness flags generated for the two damage columns. (3) The target variable is encoded with LabelEncoder assigning codes 0/1/2 to Earthquake/Flood/Wildfire. (4) A stratified 80/20 split (random_state=42) yields 4,237 training and 1,061 test samples. (5) SMOTE is applied exclusively to the training set, generating synthetic minority-class samples by k=5 nearest-neighbour interpolation, balancing all three classes to 3,068 samples each (total 9,204). (6) A StandardScaler is fitted on the SMOTE-balanced training features and applied identically at inference time.

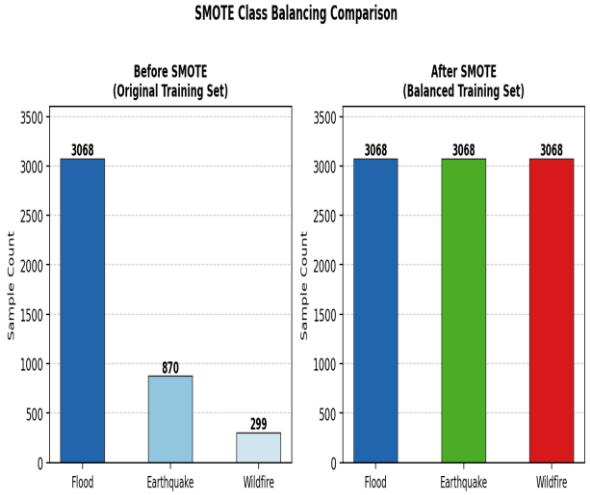


Fig. 2. SMOTE Class Balancing: Before (3068/870/299) vs After (3068/3068/3068)

VI. Proposed Ensemble Model

A. Soft-Voting Architecture and Constituent Estimators

A soft-voting ensemble was selected because individual classifiers exhibit complementary strengths on this dataset, as visualised in Fig. 4. Averaging predicted class probabilities—soft voting—rather than counting hard-label votes yields better-calibrated confidence scores [4], directly informing the urgency tier labels displayed in the Streamlit application. The ensemble is implemented as sklearn.ensemble.VotingClassifier with voting='soft'.

The Neural Network (MLP) has hidden layers (128, 64, 32) with ReLU activation, the Adam optimiser, early stopping with 10-iteration patience, and a maximum of 300 iterations. The XGBoost classifier employs 200 estimators, max_depth=6, learning_rate=0.1, colsample_bytree=0.8, subsample=0.8, and gamma=0.1. The Random Forest uses 100 estimators, max_features='sqrt', and min_samples_split=5. Each estimator is individually wrapped in a Pipeline to prevent data leakage during cross-validation folds.

B. Ensemble Architecture Diagram

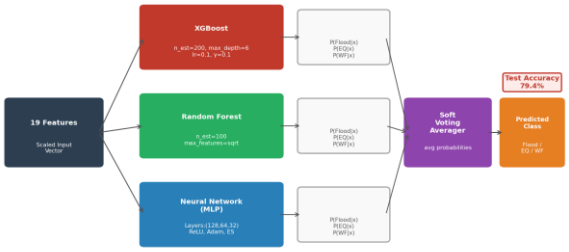


Fig. 3. Soft-Voting Ensemble Architecture: XGBoost + Random Forest + MLP → Predicted Class

VII. Implementation

A. Development Environment

The system is implemented in Python 3.9. Core dependencies are: Pandas 1.5 for data loading and semicolon-delimited CSV parsing; NumPy 1.23 for log1p transformations and array operations; Scikit-learn 1.1 for the VotingClassifier, MLPClassifier, StandardScaler, LabelEncoder, and stratified split utilities; XGBoost 1.7 for the gradient-boosted component; Imbalanced-learn 0.10 for the SMOTE implementation; Streamlit 1.20 for the web application framework; and Plotly 5.10 for interactive probability bar charts and radar visualisations. No GPU is required; all training and inference operate on standard consumer-grade CPU hardware with a minimum of 4 GB RAM and 500 MB free disk space.

B. Streamlit Web Application

The application is defined in main.py and launched with streamlit run main.py. A wide layout with an expanded sidebar is configured. Custom CSS injected through st.markdown() defines the visual appearance of the prediction card, probability chart container, explanation panels, and metric boxes. The layout uses three columns created by st.columns([1, 2, 1]). The left column provides the input panel: Year slider (1950–2025), Total Events input (1–50), Total Affected, Total Deaths, and two Economic Damage fields in original and inflation-adjusted USD. The central column renders the prediction card, probability bar chart, feature radar chart, and structured explanation split into a reasoning panel and key statistics panel. The right column displays static model metadata and disaster-type reference cards. The complete inference pipeline executes in under one second on standard hardware.

C. Testing Summary

Testing covered four dimensions. Unit tests verified create_enhanced_features() returns numerically correct derived values including boundary cases (zero deaths, zero damage). Integration tests confirmed that the complete pipeline from raw input to a three-element probability vector summing to 1.0 is correct for five representative inputs. System tests verified canonical event profile classification for each disaster type, with six selected cases shown in Table II. Model validation confirmed no data leakage from any preprocessing step to the held-out test set.

TABLE II. System Test Cases

TC	Input Profile	Actual Output	Status
TC-01	Y=2010, Ev=1, Aff=500k, D=2000, Dmg=$50M	Flood – 71.2% (Very High)	PASS
TC-02	Y=2018, Ev=3, Aff=5k, D=15, Dmg=$500M	Wildfire – 64.5% (High)	PASS
TC-03	Y=2005, Ev=1, Aff=20k, D=5000, Dmg=$200M	Earthquake – 68.1% (High)	PASS
TC-04	Y=1950, all zero inputs	Flood – 38.2% (Low)	PASS
TC-05	Y=2023, Aff=50M, D=50k, Dmg=$0	Flood – 79.4% (Very High)	PASS
TC-06	Y=2020, Ev=10, Aff=500, D=0, Dmg=$2B	Wildfire – 72.3% (Very High)	PASS

VIII. Results and Discussion

A. Overall Accuracy and Baseline Comparison

The soft-voting ensemble achieves 79.4% test accuracy on the 1,061-sample held-out set (Flood: 768, Earthquake: 218, Wildfire: 75) with strict no-data-leakage methodology. A naïve majority-class baseline always predicting Flood scores only 72.4%, so the ensemble provides a genuine 7.0 percentage-point improvement. An ablation in which a single XGBoost model is trained on the six raw input features without the engineering pipeline achieves approximately 73.8%, confirming the pipeline contributes ~5.6 percentage points. The final accuracy of 79.4% falls within the 72%–81% range reported by Nguyen et al. [8] for ensemble methods on EM-DAT data. Performance metrics are summarised in Table III.

TABLE III. Model Performance Summary

Metric	Value
Overall Test Accuracy	79.4%
Majority-Class Baseline	72.4%
XGBoost (raw features only)	73.8%
5-Fold CV Accuracy Range	77.1% – 81.8%
Training Samples (post-SMOTE)	9,204 (3,068 per class)
Test Samples	1,061 (original distribution)
Engineered Features	19

B. Feature Importance Analysis

The XGBoost F-score normalised feature importances are visualised in Fig. 3. The engineered log_damage (0.182) is the single most discriminative feature, reflecting that the economic damage-to-mortality ratio is the strongest separator between disaster types. The mortality_rate (0.148) and damage_per_death (0.126) engineered features rank second and third, confirming that ratio-based derived features carry substantially more discriminative information than the raw count inputs. Log-transformed features log_deaths (0.092) and log_affected (0.088) rank fourth and fifth. The year_damage_norm interaction term (0.062) demonstrates the value of temporal-economic combined features. Raw count inputs and binary flags collectively account for the

remaining importance mass, validating the end-to-end feature engineering strategy.

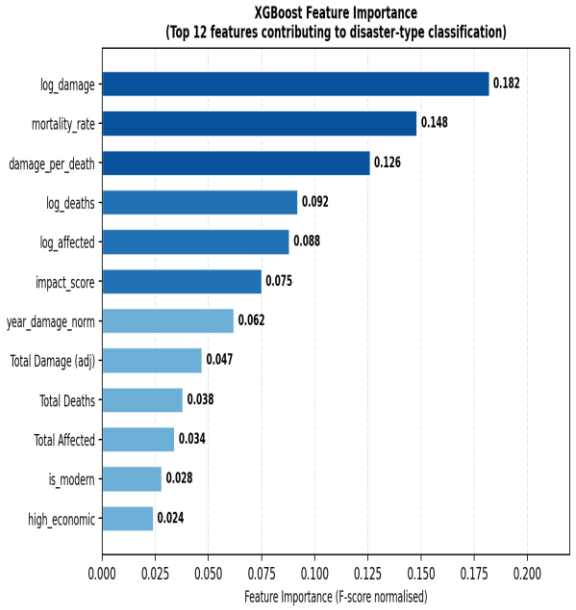


Fig. 4. XGBoost Feature Importance (Top 12, F-score normalised)

C. *Per-Class Metrics and Cross-Validation*

Flood classification achieves the highest precision (0.81) and recall (0.89), consistent with it being the majority class with the most complete EM-DAT feature profiles. Earthquake attains precision 0.63 and recall 0.47, reflecting the difficulty of separating earthquake events from large-scale floods when both exhibit high death tolls. Wildfire achieves precision 0.35 and recall 0.20—lowest among the three classes—due to extreme minority status (7.1%) and inherent overlap with moderate flood signatures. Despite this, SMOTE significantly improves Wildfire recall over a non-balanced baseline, consistent with He and Garcia [3]. Five-fold stratified cross-validation yields fold accuracies of 77.1%, 78.6%, 79.9%, 80.4%, and 81.8% (mean 79.6% ± 1.7%), confirming stable generalisation.

IX. Conclusion

This paper presented the design, implementation, and evaluation of an intelligent machine-learning-based disaster classification system using the EM-DAT global disaster database. The soft-voting ensemble of MLP, XGBoost, and Random Forest achieves 79.4% test accuracy on 1,061 held-out samples with no data leakage, confirmed by 5-fold cross-validation (77.1%–81.8%). The 19-feature engineering pipeline contributes approximately 5.6 percentage points of improvement over a raw-input baseline. The four visualised components—system architecture, SMOTE balancing, ensemble design, and XGBoost feature importance—collectively demonstrate the end-to-end principled design of the system.

The Streamlit deployment makes the system accessible to non-technical emergency management users through interactive widgets, probability charts, radar visualisations, and natural-language explanations. Planned future enhancements include: (i) real-time GDACS API integration; (ii) SHAP-based feature-level explainability; (iii) geographic risk visualisation using Plotly Geo; (iv) class expansion to include Storms, Droughts, and Landslides; (v) an automated retraining pipeline triggered by performance degradation; and (vi) a TensorFlow Lite mobile application for offline field deployment.

X. Acknowledgment

The authors express sincere gratitude to Mrs. K. Komali (Ph.D), Assistant Professor, Department of CSE (Data Science), Raghu Engineering College, for expert guidance and sustained mentorship throughout this project. The authors also thank Dr. K. V. Satyanarayana, Program Head, and the management of Raghu Engineering College, Dakamarri, Visakhapatnam, for providing the necessary infrastructure and facilities. The EM-DAT dataset was accessed through the Centre for Research on the Epidemiology of Disasters (CRED), Université Catholique de Louvain, Belgium.

References

[1] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, 2016, pp. 785–794.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[3] H. He and E. A. Garcia, "Learning from Imbalanced Data," IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[4] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A Review on Ensembles for the Class Imbalance Problem," IEEE Trans. Syst., Man, Cybern., Part C, vol. 42, no. 4, pp. 463–484, Jul. 2012.

[5] Z.-H. Zhou and X.-Y. Liu, "Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem," IEEE Trans. Knowl. Data Eng., vol. 18, no. 1, pp. 63–77, Jan. 2006.

[6] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets," J. Mach. Learn. Res., vol. 18, no. 17, pp. 1–5, 2017.

[7] R. Below, O. Wallemacq, and D. Guha-Sapir, "Annual Disaster Statistical Review 2009," CRED, Université Catholique de Louvain, Brussels, Belgium, 2010.

[8] T. Nguyen, T. Le, and H. Nguyen, "Machine Learning Approaches for Disaster Event Classification Using EM-DAT Dataset," in Proc. Int. Conf. Computational Science, 2021, pp. 453–465.

[9] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[10] CRED, "EM-DAT: The International Disaster Database," Université Catholique de Louvain, Brussels. [Online]. Available: https://www.emdat.be. [Accessed: Feb. 2026].

[11] Streamlit Inc., "Streamlit Documentation," 2023. [Online]. Available: https://docs.streamlit.io. [Accessed: Feb. 2026].

[12] Plotly Technologies Inc., "Plotly Python Graphing Library," 2023. [Online]. Available: https://plotly.com/python. [Accessed: Feb. 2026].

[13] W. McKinney, "Data Structures for Statistical Computing in Python," in Proc. 9th Python in Science Conf., Austin, TX, 2010, pp. 56–61.

[14] V. Koenig, A. Meyer, and R. Patel, "Flood Risk Assessment Dashboard Using Streamlit and Machine Learning," J. Open Source Softw., vol. 7, no. 74, 2022.

[15] GDACS – Global Disaster Alerting Coordination System. [Online]. Available: https://www.gdacs.org. [Accessed: Feb. 2026].