Melrick Paul, 2026, 14:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Multi-Source Disaster Monitoring and Early Warning System Using Risk Scoring

Melrick Paul, Vansh Bansod, Priya Pachgade, Sandhya Dhage
Department of Computer Engineering
St. Vincent Pallotti College of Engineering and Technology, Nagpur, Maharashtra, India

Abstract- Disaster information in Maharashtra is fragmented across government alerts, news reports and social media, which differ widely in speed and reliability, so District Disaster Management Authorities lack a single real-time view of an evolving event. This paper presents a multi-source disaster monitoring and early-warning system that ingests News APIs, social media and government alert feeds through an Apache Kafka producer–consumer pipeline, removes duplicate, spam and irrelevant content, and extracts the disaster type and affected city or village using keyword- and dictionary-based entity recognition. A risk-scoring engine weights each event by source reliability and recency and classifies every district as Low, Medium or High risk. A Safe Zone Identification module then applies disaster-specific safety criteria and Haversine distance to recommend the nearest suitable safe location, shown to the user as a photograph rather than text directions alone. Results are stored for retrieval and presented on a color-coded district map with a live news feed and threshold-based web and SMS alerts. The system is implemented as a working prototype; this paper describes its design and specifies the protocol for its quantitative evaluation, which is ongoing.

Keywords— disaster monitoring, early warning system, risk scoring, named entity recognition, safe zone identification, Kafka streaming, GIS dashboard, Maharashtra.

I. INTRODUCTION

Disaster monitoring is difficult because information about an emerging event rarely comes from a single trusted source. It arrives in fragments through government alerts, news reports and social-media posts, which differ in reliability and speed. Maharashtra is particularly exposed: its coastline is affected by cyclones, the Western Ghats by landslides, its river basins by seasonal floods, and industrial hubs such as Mumbai, Pune and Nagpur by fires and chemical accidents. District Disaster Management Authorities (DDMAs) still depend largely on manually compiled reports, which makes a rapid response difficult when large volumes of information are generated during a fast-moving event.

Each source has distinct strengths and limitations. Government alerts are reliable but are often released only after verification; news reports provide confirmation but may take time to publish; social media is the fastest but contains considerable noise and misinformation. Weighting sources by reliability and favoring recent evidence therefore gives a more realistic assessment of current risk, approximating the analysis that disaster-management officials would otherwise perform manually.

This paper presents a multi-source disaster monitoring and early-warning system for Maharashtra that continuously collects disaster-related information, scores district-level risk, and guides affected citizens to a nearby safe location.

The main contributions are:

- A Kafka-based ingestion pipeline that unifies News APIs, social media and government alert feeds, followed by duplicate, spam and irrelevance filtering.
- Keyword- and dictionary-based extraction of the disaster type and the affected city or village.
- A source-weighted, recency-adjusted risk-scoring engine that classifies each district as Low, Medium or High risk.
- A disaster-type-aware Safe Zone Identification module that returns a photograph of the nearest suitable safe location instead of map-reading instructions alone.

© 2026 Melrick Paul. his is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

- A live, authority-oriented dashboard with threshold-based web and SMS alerts, backed by storage of all processed data for retrieval and historical reference.

The remainder of this paper is organized as follows. Section II reviews related work and Section III states the research gap. Section IV presents the system architecture, and Section V describes the methodology, including the risk-scoring and Safe Zone Identification modules. Section VI sets out the evaluation plan, Section VII discusses limitations, Section VIII outlines future work, and Section IX concludes.

II. RELATED WORK

This section reviews the literature relevant to the system in five clusters and relates each to the pipeline described in Sections IV and V; Table I consolidates the comparison.

1. Integrated Deep-Learning Risk Assessment

Nikam et al. [1] unify flood and landslide detection within one multi-hazard pipeline but rely solely on sensor and satellite imagery, with no mechanism for textual reports. Reddy et al. [2] show that lightweight CNNs (MobileNet) classify disaster imagery competitively with deeper networks, suggesting that efficient models suit near-real-time pipelines, although the study is limited to a single image modality. Venkatesh et al. [3] propose a big-data framework fusing satellite, sensor, social and IoT streams and argue for proactive management; the proposal remains architectural rather than a concrete, deployable pipeline and is not scoped to a specific administrative geography, which the present system addresses by combining news, social and alert sources ahead of a shared Kafka pipeline.

2. Social-Media Disaster Detection and NLP

Early work treated social-media users as sensors, for example detecting earthquakes from Twitter streams in near real time [4], and later surveys catalogue how social-media messages are processed in mass-emergency settings [5]. Shramenko and Shramenko [6] detect disasters from social-media text using NLP and report roughly 88.5% accuracy, but rely on a single source with no cross-channel corroboration. Bagde and Bhagat

[7] similarly improve social-media detection without fusing other channels. Islam et al. [8] benchmark transformer models (BERT-Base, DistilBERT, Twitter-RoBERTa, FinBERT) and CNNs on the CrisisMMD dataset and find that DistilBERT offers the best accuracy–latency trade-off for near-real-time use, which informs our preference for lightweight text classifiers in the extraction stage; their study, however, evaluates models in isolation rather than as part of a deployed system.

3. Named Entity Recognition and Misinformation Filtering

Li et al. [9] fine-tune a BERT-BiLSTM-CRF model for NER in geological disaster text, which motivates the city/village extraction module here, although it would require retraining on Maharashtra-specific vocabulary; the present system instead uses a lighter keyword- and location-dictionary-based approach suited to its real-time constraints. Nikam et al. [10] and Urane and Deshpande [11] filter fake or anomalous social content using statistical and deep-learning classifiers; both treat filtering as a standalone task disconnected from downstream risk computation, a gap this work addresses by placing spam and relevance filtering directly ahead of extraction and scoring.

4. GIS- and ML-Based Hazard Prediction

Kurniawati et al. [12] classify hazard levels from environmental sensor data using an SVM paired with GIS visualization, an approach that depends on dense physical sensor deployment unavailable across Maharashtra's districts. Zhang [13] combines statistical feature selection with K-means clustering and random-forest classification for flood-specific Low/Medium/High tiering, a three-tier scheme this work adopts but generalizes across hazard types using weighted, recency-adjusted source counts rather than a fixed historical dataset. Li and Wang [14] and Lin and Liang [15] confirm threshold-triggered alerting as a validated design pattern, and Das et al. [16] show that uncertainty-aware ensembling of weak signals improves robustness

under noisy data, a principle reflected in the source-weighted event counting used here.

5. Real-Time Data Pipelines and Notification Delivery

Because the system streams live data through an Apache Kafka cluster into a MongoDB store for continuous dashboard updates, streaming-infrastructure literature is directly relevant. Huang and Lu [17] show that fusing heterogeneous data streams improves situational coverage over any single modality. Kafka itself was designed as a low-latency distributed messaging system for log processing [18], and Dou and Wang [19] and Gogineni [20] respectively examine real-time big-data stream processing and Apache Kafka as a low-latency backbone for AI-enhanced pipelines, informing the producer–consumer Kafka-to-MongoDB architecture used here. None of these works, however, addresses how a downstream risk score or citizen-facing alert should be derived from the streamed data, which is the role of this system's risk-scoring, alerting and Safe Zone Identification stages.

Table 1: Summary of Reviewed Literature by Cluster

Cluster	Papers	Technique(s)	Strength	Limitation	Gap Addressed
Integrated DL risk assessment	[1]–[3]	Multi-modal DL, CNN, big-data fusion	Multi-hazard, proactive framing	No unified multi-source text pipeline	Adds text + structured source fusion
Social-media detection and NLP	[6]–[8]	NLP, transformer, CNN benchmarking	High detection accuracy	Single source, no fusion	Adds cross-source weighting
NER and misinformation filtering	[9]–[11]	BERT-BiLSTM-CRF, anomaly/DL filters	Strong entity and fake-content detection	Standalone tasks	Embeds filtering + NER before scoring
GIS / ML hazard prediction	[12]–[16]	SVM, K-means + RF, ensembles	Validated tiering and alerting patterns	Single hazard/source scope	Generalized, recency-weighted scoring
Real-time pipelines and notification	[17], [19], [20]	Kafka, big-data stream fusion	Scalable, low-latency ingestion	No downstream scoring/alerting logic	Adds Kafka-based scoring + alerts

III. RESEARCH GAP

The reviewed literature is technically competent within its respective scopes but leaves four gaps relevant to Maharashtra's district administrations. First, most works process a single data modality or hazard type rather than fusing government alerts, news and social media in a unified pipeline. Second, few incorporate a mechanism that gives greater weight to recent evidence over outdated reports; risk scores in the cited systems are generally computed from static or historical datasets. Third, misinformation filtering [10], [11] and disaster detection [6]–[8] are usually treated as separate tasks rather than as integrated stages of one pipeline. Fourth, and most notably, none of the works we reviewed provides an actionable, citizen-facing safe-zone recommendation once a disaster is detected; existing systems generally stop at detection and mapping, leaving unanswered what an affected person should do next. To the best of our knowledge, the proposed system addresses all four gaps together by fusing three source types through a shared Kafka pipeline, applying a recency factor during scoring, integrating filtering before extraction and scoring, and introducing a disaster-type-aware Safe Zone Identification module.

IV. PROPOSED SYSTEM ARCHITECTURE

Fig. 1 illustrates the architecture as eight functional layers, numbered in the figure. The Sources layer (1) comprises News APIs, social media (Twitter/X) and government alert feeds, which are ingested through

a Kafka-based producer–consumer pipeline (2). Social-media posts are published to a Kafka cluster and consumed into a MongoDB store, enabling live dashboard updates without batch reprocessing. The Preprocessing layer (3) removes duplicate, spam and irrelevant content and normalizes text. Information Extraction (4) then detects the disaster type and resolves affected cities and villages through keyword matching against a location dictionary covering Maharashtra's districts and villages.

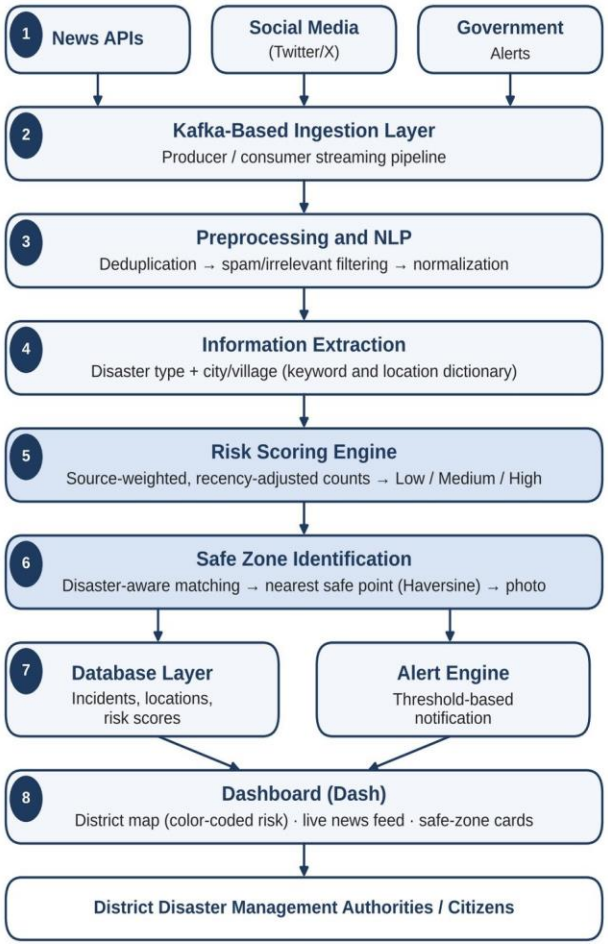


Fig. 1. Proposed system architecture (layers numbered 1–8).

The extracted events pass to the Risk Scoring Engine (5), which assigns each source a fixed weight (government alerts highest, social media lowest), applies a recency factor, and maps the weighted totals for each district to a Low/Medium/High classification (Section V-A). The Safe Zone Identification module (6) uses the same disaster type and location to apply disaster-specific safety criteria (Table II), select the appropriate class of safe location, and compute the nearest instance by Haversine distance from the user's live location. Both the risk score and the safe-zone result are written to the Database Layer and read by the Alert Engine (7), which generates threshold-based notifications. The dashboard (8), developed using Dash, displays a color-coded Maharashtra district map, a live disaster news feed and safe-zone cards, delivering information to District Disaster Management Authorities and affected citizens.

V. METHODOLOGY AND IMPLEMENTATION

This section describes the end-to-end data flow and risk scoring (Section V-A), the Safe Zone Identification module (Section V-B), the model training and deployment pipeline (Section V-C), and the notification architecture (Section V-D).

1. Data Flow and Risk Scoring

Fig. 2 shows the pipeline as a continuous flow with an example artifact at each stage. Disaster-related text is collected at regular intervals (approximately every five to ten minutes) from News APIs, social-media keyword searches and government alert feeds where available, producing raw unstructured text, headlines and posts.

The raw text is cleaned by removing spam, duplicate news and irrelevant posts, and normalized through lower-casing and removal of extraneous symbols. Information extraction then matches keywords against a location dictionary of districts and villages, while a keyword-based classifier assigns the disaster type. For example, “flood in Kamptee village near Nagpur” resolves to the record {location: Nagpur, sub-location: Kamptee, type: Flood}.

The risk-scoring engine converts these structured records into a district-level score. Let S be the set of sources and Ed,s(t) the events for district d reported by source s up to time t, each with report time ti. The score of district d is

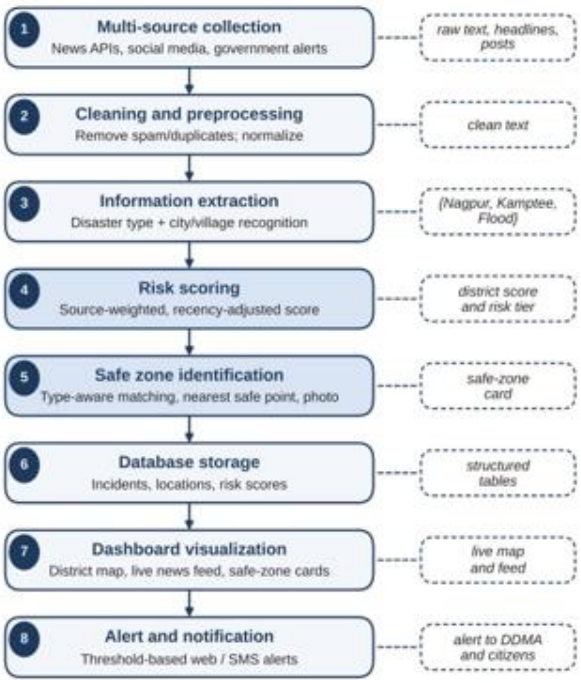


Fig. 2. End-to-end system data flow with an example artifact at each stage.

$$R_d(t) = \Sigma_{s\in S}\, w_s \,\Sigma_{i\in E(d,s)}\, f(t - t_i) \qquad (1)$$

where *ws* is the fixed reliability weight of source *s*, with *w*gov > *w*news > *w*social, and *f*(·) is a non-increasing recency factor of event age. An exponential form

$$f(\Delta t) = e^{-\lambda \Delta t}, \quad \lambda = \ln 2 / T_{½} \qquad (2)$$

where T½ is the half-life after which a report counts half as much, is a natural choice. A district is classified as Low if Rd < τ1, Medium if τ1 ≤ Rd < τ2, and High if Rd ≥ τ2.

To illustrate the intended behavior, take *w*gov = 1.0, *w*social = 0.2 and *T*½ = 6 h (illustrative values for exposition only, not tuned). Two government alerts that are 1 h old contribute 2 × 1.0 × 0.89 ≈ 1.78 to *R*d, whereas 30 social-media posts that are 24 h old contribute 30 × 0.2 × 0.0625 ≈ 0.38. A short burst of recent official alerts therefore outweighs a much larger volume of older social-media activity, as intended.

Processed incidents, resolved locations and risk scores are stored in structured tables (incidents, locations, risk_scores) for retrieval and historical reference. The dashboard presents this data through a live district map with color-coded risk levels (green/amber/red), an auto-refreshing disaster news feed covering both cities and villages, and safe-zone cards showing a photo, name and distance for the nearest recommended location.

2. Safe Zone Identification

Once the disaster type and location are known, the module applies disaster-specific criteria (Table II) to select the appropriate class of safe location, finds the nearest candidate to the user's live location, and retrieves a stored photograph of it. Candidate safe points are obtained from OpenStreetMap (parks, open grounds, rivers and water bodies), SRTM 30-m elevation data for identifying high ground, and government shelter lists where available. The distance between two points is computed with the Haversine formula,

$$a = \sin^2(\Delta\varphi/2) + \cos\varphi_1 \cos\varphi_2 \sin^2(\Delta\lambda/2) \qquad (3)$$

$$d = 2R_E \arcsin(\sqrt{a}) \qquad (4)$$

Table 2: Safe Zone Criteria by Disaster Type

Disaster Type	Safe Zone Criteria	Avoid
Flood	High elevation, far from rivers/lakes	Low-lying areas, riverbanks
Earthquake	Open grounds, parks, wide open spaces	Tall buildings, glass structures
Fire	Open areas, upwind direction, away from dry vegetation	Downwind areas
Cyclone	Concrete shelters, inland areas	Coastal areas, weak structures

3. Model Training and Deployment Pipeline

Alongside the keyword-based classifier of Section V-A, the system supports a learned disaster-type classifier that follows an offline-training and online-deployment approach, separating periodic retraining from continuous inference. In the training pipeline, historical disaster-related text is preprocessed and used to train the classifier, which is evaluated and refined through hyperparameter tuning until it meets an internal quality check; the best-performing version is then saved. In the deployment pipeline, the saved model is loaded onto a cloud server and applied to live, preprocessed social-media and location data as it arrives. Its predictions, together with the resolved city/village and disaster type, are converted into map markers and pushed to the dashboard in real time. This separation allows the model to be retrained and re-validated offline as labelled data accumulate, without interrupting the live ingestion-to-dashboard pipeline of Section V-A.

4. Notification Delivery Architecture

Alert delivery follows a notification-gateway pattern. Upstream services, primarily the risk-scoring and Safe Zone Identification modules, submit individual or batched notifications to a central Notification Gateway. The gateway forwards each notification to a distribution stage that validates its content, applies scheduling and priority rules, and selects a message template before placing it on a notification router backed by message queues. The deployed system delivers web and SMS alerts; the gateway design also accommodates email, in-app and social-media delivery according to each recipient's channel preference. Delivery outcomes are recorded by a tracing and analytics component for operational monitoring. This design lets DDMAs and citizens receive alerts through the channel most reliable for them, without coupling the risk-scoring logic to any single delivery mechanism.

VI. EVALUATION PLAN

Several components of the system are implemented while others remain under active refinement, so this paper does not claim final accuracy figures. Table III specifies the metrics and reference data that will be used to evaluate the system. Information-extraction quality will be benchmarked against a manually annotated sample of Maharashtra disaster-related text, and system-level measures will be taken under realistic ingestion loads rather than idealized single-document tests. Where feasible, a small user study with sample disaster scenarios will assess whether the photograph-based safe-zone card can be recognized and acted upon more quickly than equivalent text-based directions, which directly motivates the module's design.

Table 3: Planned Evaluation Metrics

Component	Metrics	Reference data
Disaster-type classification	Accuracy, precision, recall, F1-score	Annotated Maharashtra text sample
City/village recognition	Accuracy, precision, recall, F1-score	Same annotated sample
End-to-end system	Collection-to-dashboard latency; dashboard update time	Realistic ingestion load
Alerting	False-positive rate; alert precision	Logged alerts vs. known events
Risk classification	Tier accuracy	Known disaster timelines, selected districts
Safe-zone module	Recommendation relevance (Table II criteria); detection-to-card latency	Detected events
Safe-zone card usability	Recognition and response time vs. text directions	Small user study

Future Scope

Planned extensions include a Marathi-language NLP pipeline for hyper-local social-media posts and citizen reports, direct API integration with IMD and NDMA feeds in place of bulletin scraping, and an operator feedback loop to recalibrate source weights and thresholds over time. On the citizen-facing side, WhatsApp-based alerts and a mobile application could extend reach beyond the web dashboard, while crowdsourced contributions could expand the safe-zone photograph database in villages that are underrepresented in mapping services.

On the modeling side, replacing the keyword- and dictionary-based extraction with a fine-tuned transformer model, informed by the NER approaches surveyed in Section II-C, could improve robustness to informal or code-mixed phrasing once enough labelled Maharashtra-specific data are available. Extending district-level granularity to the taluka/tehsil level and replicating the architecture, including its Kafka-based ingestion and Safe Zone Identification modules, in other Indian states are natural directions once the current implementation has been thoroughly evaluated.

Limitations

The following limitations apply to the current design and should be considered when interpreting its outputs.

- Extraction: dictionary matching cannot resolve ambiguous, renamed or misspelled place names and handles English text only; Marathi and code-mixed posts are not yet supported.
- Scoring: source weights, the recency function and tier thresholds are set by hand rather than calibrated against observed events, so tier boundaries remain heuristic until the evaluation of Section VI is complete.
- Data sources: social-media volume depends on platform API access and terms, rule-based filtering cannot guarantee removal of coordinated or subtle misinformation, and government alerts may appear only after verification.
- Safe zones: Haversine distance is a straight-line measure that ignores roads, rivers and barriers; OpenStreetMap, SRTM and shelter data may be incomplete for villages; the criteria in Table II are general guidance rather than formally validated instructions; and photograph retrieval depends on the licensing terms of the image provider. Recommendations should be treated as advisory and confirmed by the authorities before public use.
- Validation: the evaluation in Section VI is not yet complete, so no claim is made about accuracy, latency or user benefit.

VII. CONCLUSION

This paper presented a multi-source disaster monitoring and early-warning system that streams News APIs, social media and government alerts through a Kafka-based pipeline, extracts the disaster type and location through keyword- and dictionary-based recognition, computes a source-weighted, recency-adjusted district risk score, and identifies and displays a photograph of the nearest disaster-appropriate safe zone to affected citizens. The review of related work indicates that existing systems address detection, classification or alerting in isolation, and that none of the systems reviewed extends to actionable, citizen-facing safe-zone recommendation. The pipeline described in Sections IV and V addresses this gap for Maharashtra's district administrations. Section VI specifies how the system will be evaluated as implementation continues, and Sections VII and VIII identify its limitations and concrete extensions, including multilingual support and expanded safe-zone coverage.

REFERENCES

1. T. Nikam, S. Mahangade, P. More, and R. Kumar, "An integrated deep learning framework for multi-hazard risk assessment: Flood prediction and landslide detection," Int. J. Adv. Comput. Theory Eng., vol. 15, no. 2S, pp. 83–90, 2026.
2. G. M. V. Reddy, P. Bharatwaj Teja, K. T. Devi, and V. P. K. Sistla, "Natural disaster prediction using deep learning," in Proc. IEEE Conf., Feb. 2025, IEEE Xplore Doc. 10928127.
3. N. Venkatesh, M. K. Kaarthik, M. Ashad, K. Vignesh, B. Mamatha, and B. Venkataramana, "Big data-driven disaster management and early

warning system for natural calamities," Int. J. Res. Appl. Sci. Eng. Technol., vol. 14, 2026, doi: 10.22214/ijraset.2026.77458.
4. T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes Twitter users: Real-time event detection by social sensors," in Proc. 19th Int. Conf. World Wide Web (WWW), Raleigh, NC, USA, 2010, pp. 851–860.
5. M. Imran, C. Castillo, F. Diaz, and S. Vieweg, "Processing social media messages in mass emergency: A survey," ACM Comput. Surv., vol. 47, no. 4, Art. no. 67, 2015.
6. V. Shramenko and N. Shramenko, "Disaster detection using natural language processing in social networks," WSEAS Trans. Inf. Sci. Appl., vol. 22, pp. 731–742, 2025.
7. A. Bagde and S. Bhagat, "Enhancing disaster detection on social media," preprint, Semantic Scholar, 2024.
8. M. A. Islam, F. Rabbi, and N. U. I. Hossain, "Performance evaluation of NLP and CNN models for disaster detection using social media data," Soc. Netw. Anal. Min., vol. 15, 2025, doi: 10.1007/s13278-024-01374-y.
9. Y. Li, L. Luo, X. Zeng, et al., "Fine-tuned BERT-BiLSTM-CRF approach for named entity recognition in geological disaster texts," Earth Sci. Inform., vol. 18, Art. no. 368, 2025, doi: 10.1007/s12145-025-01870-5.
10. S. Nikam, N. Shitole, S. Gaikwad, A. Radde, and P. U. Thakare, "Identifying fake news and tweets using anomaly detection," Int. J. Innov. Res. Multidiscip. Field, vol. 5, no. 1, 2019.
11. S. Urane and A. Deshpande, "Deep learning based fake news detection," preprint, Semantic Scholar, 2024.
12. A. D. Kurniawati, P. Kristalina, Z. Hadi, and D. Widodo, "GIS-based disaster management using support vector machine model for hazard level classification in disaster areas," in Proc. 2023 Int. Electron. Symp. (IES), 2023, doi: 10.1109/IES59143.2023.10242594.
13. Zhang, "A flood hazard prediction and risk assessment model based on machine learning approach," 2025.
14. Y. Li and S. Wang, "Research on natural flood disaster prediction and early warning," preprint, Semantic Scholar, 2024.
15. Y. Lin and X. Liang, "Application of machine learning in early warning of natural hazards," preprint, Semantic Scholar, 2024.
16. S. D. Das, A. Basak, and S. Dutta, "A heuristic-driven uncertainty based ensemble framework for fake news detection in tweets and news articles," Neurocomputing, 2021, arXiv:2104.01791.
17. Q. Huang and X. Lu, "Research on deep learning based multi-source data fusion," preprint, Semantic Scholar, 2024.
18. J. Kreps, N. Narkhede, and J. Rao, "Kafka: A distributed messaging system for log processing," in Proc. NetDB Workshop, 2011.
19. H. Dou and Y. Wang, "The application of big data real-time stream processing in disaster monitoring," preprint, Semantic Scholar, 2024.
20. N. Gogineni, "Apache Kafka for low-latency AI-enhanced data pipelines," preprint, Semantic Scholar, 2024.