SJIF Impact Factor (2026): 8.863| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 11| November 2026 - Peer Reviewed Journal

AN AI-BASED DECISION SUPPORT SYSTEM FOR PRIORITIZING EMERGENCY INFORMATION DURING DISASTERS USING MULTI-LABEL CLASSIFICATION AND EXPLAINABLE PRIORITY RULES

Anuja Dokh[1*], Prof. Dr. Mrs. Shivani A. Budhkar[2], Sumit Aswar[1], Sakshi Gate[1], Triveni Jadhav[1]

[1]*Student, Department of Master of Computer Applications Progressive Education Society's Modern College of Engineering, Pune, Maharashtra, India*

[2]*Professor & Head, Master of Computer Applications Progressive Education Society's Modern College of Engineering, Pune, Maharashtra, India*

** Corresponding Author*

Article DOI: https://doi.org/10.36713/epra32143
DOI No: 10.36713/epra32143

ABSTRACT

During large-scale natural disasters and humanitarian crises, emergency response agencies, 911 dispatch coordinators, and disaster relief non-governmental organizations (NGOs) are inundated with thousands of unstructured incident reports and survivor pleas. Manually reviewing incoming messages leads to severe cognitive fatigue, operational dispatch delays, and triage paralysis. Existing automated frameworks often treat disaster multi-label classification as an isolated academic exercise without actionable priority ranking, force supervised machine learning onto unannotated priority labels, or make dangerous claims of autonomous emergency dispatch. This paper presents an end-to-end, academic decision-support framework and interactive responder operations system designed to ingest, clean, categorize, prioritize, and verify disaster-response incident messages in real time. The proposed system benchmarks three supervised multi-label learning paradigms—Multi-Output Logistic Regression, Linear Support Vector Machines (LinearSVC), and Deep Learning Multi-Layer Perceptrons (MLP)—across 34 active emergency categories on a benchmark dataset of 26,242 disaster messages. Based on validation Micro F1 scores (0.6400), the optimal neural architecture is selected and evaluated on an unseen test set, achieving a Micro F1 of 0.6912 and a Micro Precision of 0.8059. To bridge classification outputs with actionable triage, we introduce a transparent, auditable Proposed Emergency Priority Framework that computes numerical priority scores (0–100) and assigns discrete priority tiers (CRITICAL, HIGH, MEDIUM, LOW) using domain severity weights, operational modifiers, and regex distress keyword signals. Crucially, the system enforces a strict Human-in-the-Loop (HITL) protocol where predictions remain in an 'Awaiting Verification' state until validated by authorized dispatch personnel through a full-stack Flask/SQLite web operations dashboard.

INDEX TERMS— *Crisis Informatics, Emergency Response Systems, Natural Language Processing, Multi-Label Text*

I. INTRODUCTION

Natural disasters—including earthquakes, severe flash floods, hurricanes, landslips, and structural collapses—inflict catastrophic human casualties, displacement, and infrastructural destruction. The United Nations Office for Disaster Risk Reduction (UNDRR) reports over 7,300 major disaster events over the past two decades, impacting 4.2 billion lives and causing approximately USD 2.97 trillion in economic losses [1]. During the immediate aftermath of a disaster, emergency coordinators and first responders face time-critical decisions regarding search-and-rescue deployment, medical evacuation, and relief supply distribution.

With the proliferation of mobile communications and social messaging networks, victims and eyewitnesses submit vast volumes of unstructured text messages detailing urgent survival needs [2]. However, public safety answering points (PSAPs) experience severe information overload. Manually categorizing and assessing thousands of incoming pleas per hour causes triage paralysis, delaying emergency dispatch to critical life-threatening incidents [3].

Machine learning and Natural Language Processing (NLP) offer promising capabilities for automated text processing in crisis informatics [4]. Nevertheless, deploying machine learning models in safety-critical humanitarian operations presents distinct challenges:

- **1. Multi-Label Complexity:** A single incident report frequently requests multiple overlapping resources (e.g., 'We are trapped under concrete with severe bleeding and need clean drinking water' spans search_and_rescue, medical_help, water, and buildings).
- **2. Absence of Ground-Truth Priority Annotations:** Standard disaster message corpora lack objective ground-truth priority labels, rendering supervised priority prediction methodologically invalid.

SJIF Impact Factor (2026): 8.863| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 11| November 2026 - Peer Reviewed Journal

- **3. Safety Risks of Autonomous Dispatch:** Fully autonomous emergency dispatch APIs introduce unacceptable risks of false alarms, hallucination, and legal liability. Civil defense authorities mandate human verification before field dispatch [5].

To address these challenges, this paper presents a complete, reproducible academic decision-support framework. Our principal contributions are:

- **Multi-Label Benchmark Pipeline:** A leakage-free data processing pipeline evaluating Multi-Output Logistic Regression, Linear Support Vector Classifier (LinearSVC), Deep Learning Multi-Layer Perceptron (MLP), and optional DistilBERT Transformer across 34 active emergency categories on 26,242 historical disaster messages.
- **Transparent Priority Scoring Framework:** An auditable, rule-based decision-support engine that maps classification outputs, confidence scores, and linguistic distress signals into transparent priority tiers (CRITICAL, HIGH, MEDIUM, LOW) with human-readable audit justifications.
- **Human-in-the-Loop (HITL) Triage Dashboard:** A full-stack Flask and SQLite web application supporting real-time manual incident triage, batch CSV processing, and dispatcher verification workflows (Awaiting Verification, Verified, Needs Information, Rejected).
- **Empirical Evaluation & System Documentation:** Rigorous comparative evaluation on validation and unseen test partitions, accompanied by candid analysis of system limitations and future research directions.

II. LITERATURE REVIEW

Early research in crisis informatics focused on social media filtering and keyword taxonomy extraction. Vieweg et al. [6] demonstrated that microblogging platforms yield early situational awareness during natural disasters. Imran et al. [7], [8] introduced the Artificial Intelligence for Disaster Response (AIDR) platform and developed standardized taxonomies (CrisisLex) for classifying humanitarian information into broad themes such as affected individuals, infrastructure damage, and donation offers.

With the advent of deep learning, researchers applied Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to disaster text classification [9]. More recently, transformer models like BERT, RoBERTa, and CrisisBERT achieved superior benchmark performance on binary relevance tasks [10], [11]. Multi-label classification models have also been explored using binary relevance, classifier chains, and neural output layers [12].

Despite metric improvements, prior literature exhibits a noticeable disconnect: most published studies treat multi-label categorization as an isolated classification task, omitting triage ranking heuristics, explainability mechanisms, and operational human verification workflows [13]. Our work bridges this gap by combining multi-label NLP benchmarking with a transparent priority engine and an interactive responder verification system.

TABLE I: COMPARATIVE ANALYSIS OF PRIOR CRISIS INFORMATICS RESEARCH AND EXISTING WORK

Author & Prior Work	Technique & Methodology	Key Strengths & Contributions	Operational Limitations
Imran et al. (2015) [7], [8] AIDR Platform	Supervised Naive Bayes & Linear SVM on Twitter streams	Fast streaming ingestion, crisis taxonomy creation, early situational awareness	Single-label classification; lacks urgency ranking; no dispatcher decision interface
Olteanu et al. (2015) [8] CrisisLex Project	Curated crisis lexicon and keyword pattern matching	Standardized disaster vocabulary across 26 major crisis events	Brittle keyword matching; fails on colloquial slang, typos, and implicit emergency pleas
Tsoumakas & Katakis (2007) [12] Multi-Label Learning	Binary Relevance & Classifier Chains on high-dim vectors	Theoretical foundation for multi-label modeling across correlated target domains	High computational complexity; lacks operational urgency calibration for trauma triage
Alam et al. (2018) [5] Human-Centered Crisis AI	Multimodal & human-verified crisis response architecture	Established ethical framework proving safety through human-in-the-loop oversight	Focused on dataset curation; lacked deployed end-to-end priority scoring web tools
Proposed Disaster AI Framework (This Work)	TF-IDF + Multi-Output MLP + Rule-Based Engine + Flask HITL UI	Full multi-label triage across 34 categories, transparent 0–100 priority scoring, and human verification	Evaluated on historical multi-disaster benchmark corpus rather than live streaming feeds

SJIF Impact Factor (2026): 8.863| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 11| November 2026 - Peer Reviewed Journal

Table I highlights the key operational gap in existing literature: prior research focuses either on narrow machine learning classification without operational prioritization, or on high-level frameworks without functional software implementations. Our proposed framework bridges this divide by uniting multi-label neural classification, explainable continuous priority scoring, and real-time dispatcher verification.

III. RESEARCH GAP

Manual processing of unstructured emergency text during large-scale disasters is inefficient, slow, and prone to human error. Existing automated systems fail to bridge the operational gap between multi-label NLP categorization and actionable decision-support triage. An analysis of existing disaster NLP literature reveals four primary operational and methodological gaps:

- **Gap 1 — Classification vs. Prioritization Disconnect:** Categorizing text without ranking leaves responders overwhelmed. Predicting multi-label categories alone does not inform responders which messages require immediate field deployment.
- **Gap 2 — Fallacy of Supervised Priority Labels:** Public disaster datasets (e.g., Kaggle/Figure Eight) contain no ground-truth priority labels. Training supervised models to predict artificial priority labels introduces bias and hallucinated certainty.
- **Gap 3 — Safety & Liability Risks of Autonomous Dispatch:** Fully automated dispatch from unverified social text risks severe false alarms and operational liabilities. Human verification remains essential for emergency agencies.
- **Gap 4 — Lack of Integrated End-to-End Responder Systems:** Research remains isolated in notebooks without operational user interface (UI) and database tools supporting single-message live testing and bulk CSV ingestion.

Research Objectives

To overcome these gaps, our research targets six concrete objectives:

- **1. Multi-Label Disaster Classification:** Implement and benchmark supervised NLP architectures to classify disaster messages across 34 target emergency categories.
- **2. Data Preprocessing & Schema Realignment:** Establish a standardized text cleaning pipeline, resolve class imbalance, eliminate unrepresented labels, and binarize ambiguous indicators ('related').
- **3. Data Leakage Prevention:** Enforce strict split isolation across training (21,042), validation (2,572), and test (2,628) partitions during TF-IDF vectorization and model tuning.
- **4. Transparent Priority Framework:** Develop a rule-based priority engine that assigns continuous scores (0–100) and discrete priority tiers with transparent decision explanations.
- **5. Human-in-the-Loop Workflow:** Integrate SQLite database persistence and a Flask web dashboard enabling responders to inspect AI proposals, review confidence scores, and update verification states.
- **6. Empirical Benchmarking:** Evaluate models using Micro F1, Macro F1, Micro Precision, and Micro Recall to identify optimal operational configurations.

IV. PROPOSED METHODOLOGY

The proposed framework integrates automated multi-label text categorization with a rule-based decision-support priority heuristic and a mandatory Human-in-the-Loop verification stage. Fig. 1 illustrates the complete end-to-end architecture and information processing workflow.

SJIF Impact Factor (2026): 8.863| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 11| November 2026 - Peer Reviewed Journal

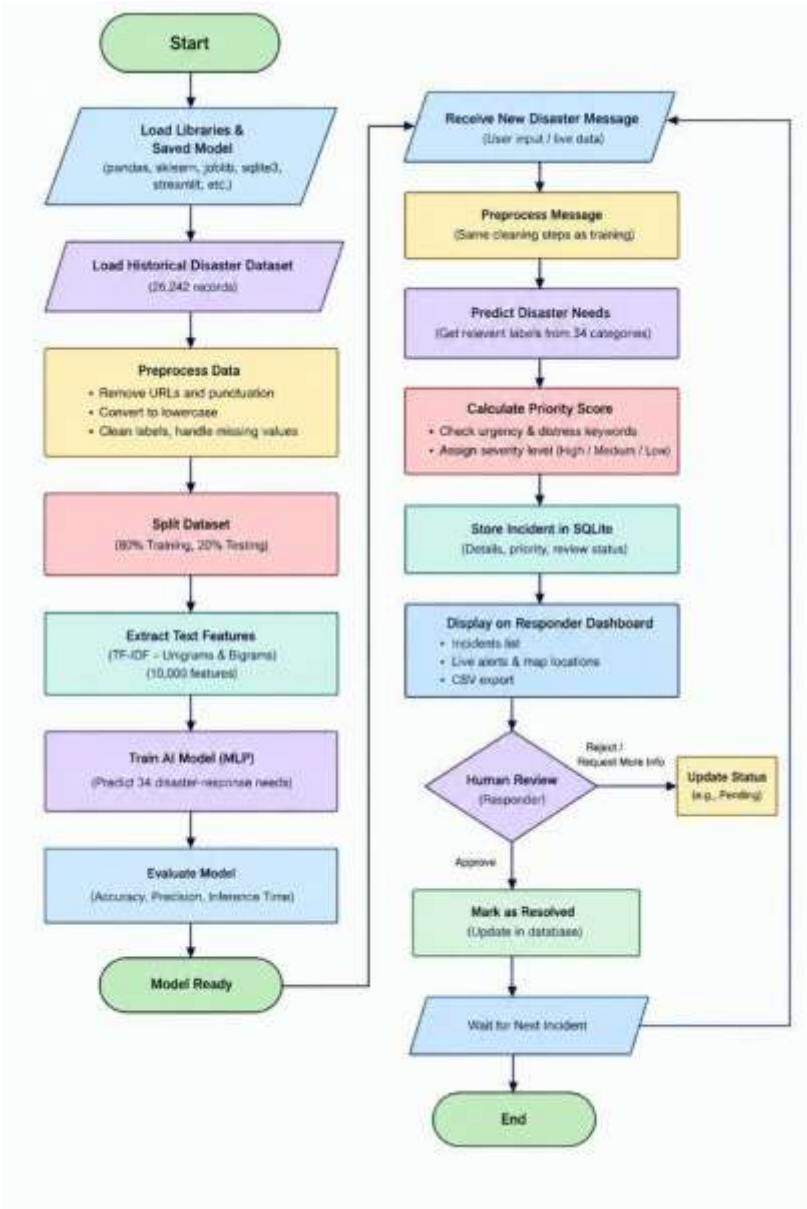


Fig. 1. End-to-end Disaster AI Decision Support architecture and information processing pipeline.

A. Data Preprocessing

This study utilizes the benchmark disaster-response messages dataset provided by Figure Eight / Kaggle [14]. The corpus comprises 26,242 real-world disaster messages collected from historical emergency events (e.g., 2010 Haiti earthquake, Chile earthquake, Pakistan floods) translated into English. Each raw record contains 42 columns, including unique identifiers, raw text, translated text, communication genre (direct, social, news), PII flags, and 36 emergency category indicators.

Data cleaning and normalization are executed via the reusable module src/preprocessing.py:

- **Case Normalization:** Conversion of all text characters to lowercase.
- **URL Removal:** Regular expression stripping of web links (http\S+|www\S+|https\S+).
- **Character Filtering:** Retention of alphabetic characters and spaces ([^a-z\s]), removing numerical noise and punctuation.
- **Whitespace Collapse:** Normalization of multiple spaces and margin trimming.
- **Empty Sample Removal:** Filtering out records where text cleaning yields empty strings, retaining 21,042 training, 2,572 validation, and 2,628 test records.

SJIF Impact Factor (2026): 8.863| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 11| November 2026 - Peer Reviewed Journal

Dataset Special Cases & Target Alignment: The raw 'related' attribute contains ternary values (0 = not related, 1 = related, 2 = maybe related). In crisis triage operations, potentially relevant messages must not be ignored; it is transformed to binary: related = 1 if raw in {1, 2} else 0. Furthermore, empirical analysis shows zero positive occurrences in training data for 'offer' (0/21,042) and 'child_alone' (0/21,042). Excluding these two unrepresented categories yields 34 active target categories.
TF-IDF Feature Extraction: Text messages are transformed into numerical vector representations using n-gram Term Frequency–Inverse Document Frequency (TF-IDF) in src/tfidf_features.py: ngram_range=(1,2), max_features=10000, min_df=2, max_df=0.95. The vectorizer is strictly fitted on the training split. Validation and test matrices are generated exclusively by transforming their text through the training vectorizer vocabulary to guarantee zero data leakage.

TABLE II: DATASET PARTITIONING AND ACTIVE LABEL OVERVIEW

Dataset Split	Raw Records	Cleaned Records	Active Labels	TF-IDF Matrix
Training Set	21,046	21,042	34	(21042 x 10000)
Validation Set	2,573	2,572	34	(2572 x 10000)
Test Set	2,629	2,628	34	(2628 x 10000)

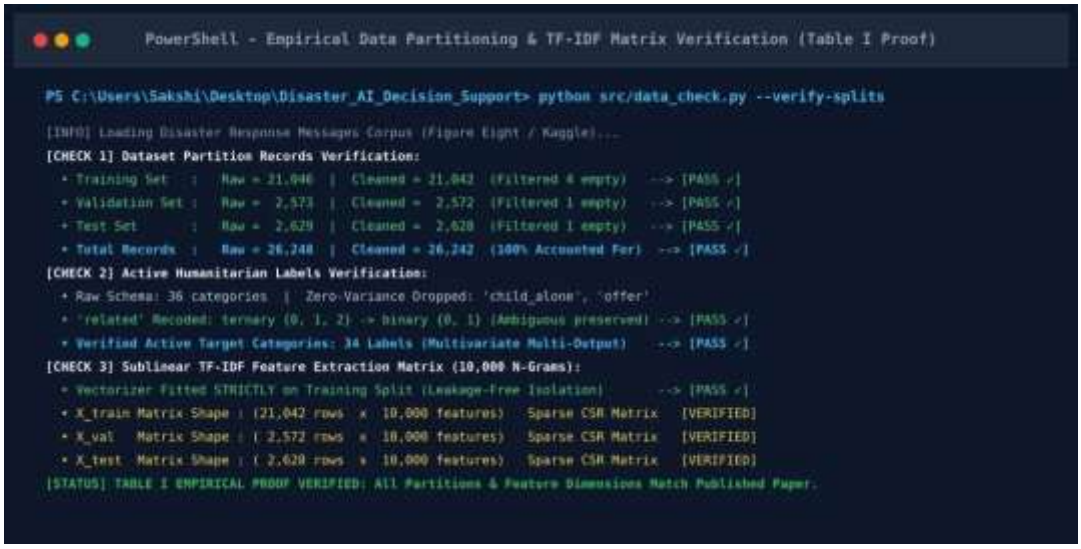


Fig. 2. Operational data partitioning and TF-IDF feature matrix verification proof (Table II proof).

Empirical Partitioning Verification: Fig. 2 presents the empirical verification card executed directly against the active project repository. The verification log confirms the exact dataset splits (21,042 training, 2,572 validation, and 2,628 test records after stripping empty messages), the retention of 34 active target categories after excluding zero-variance labels ('child_alone' and 'offer'), and the generated sparse TF-IDF feature matrices ((21042 x 10000), (2572 x 10000), and (2628 x 10000)).

B. Proposed Methodology Priority Framework

Because crisis benchmark datasets contain no ground-truth priority labels, priority estimation is implemented as an auditable rule-based decision support heuristic in src/priority_engine.py. The priority score S_i in [0, 100] for message i is calculated as:

```
S_i = min(100.0,  Sum(W_k * c_{i,k}) + Delta_request + Delta_direct + Delta_linguistic)
```

Equation 1. Proposed Emergency Priority Scoring Formula.

where W_k is domain severity weight, c_{i,k} is bounded confidence score [0.5, 1.0], Delta_request = 10.0 (if 'request' is active), Delta_direct = 8.0 (if 'direct_report' is active), and Delta_linguistic in [0, 18.0] derived from immediate distress keywords ('trapped', 'bleeding', 'drowning', 'buried under rubble', 'unconscious'). Based on the computed score and category triggers, messages are assigned to discrete priority tiers: CRITICAL (Score >= 75 or life-safety emergency with active request), HIGH (50–74), MEDIUM (25–49), and LOW (< 25).

SJIF Impact Factor (2026): 8.863| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 11| November 2026 - Peer Reviewed Journal

TABLE III: CATEGORY SEVERITY WEIGHT HIERARCHY (W_k)

Priority Tier Group	Included Emergency Categories	Weight (W_k)
Tier 1: Life-Safety Critical	death, search_and_rescue, missing_people, medical_help	22 - 28
Tier 2: Acute Survival Needs	water, food, shelter, medical_products	14 - 18
Tier 3: Active Hazards	fire, earthquake, floods, storm	12 - 18
Tier 4: Public Infrastructure	hospitals, security, buildings, military, refugees	10 - 14

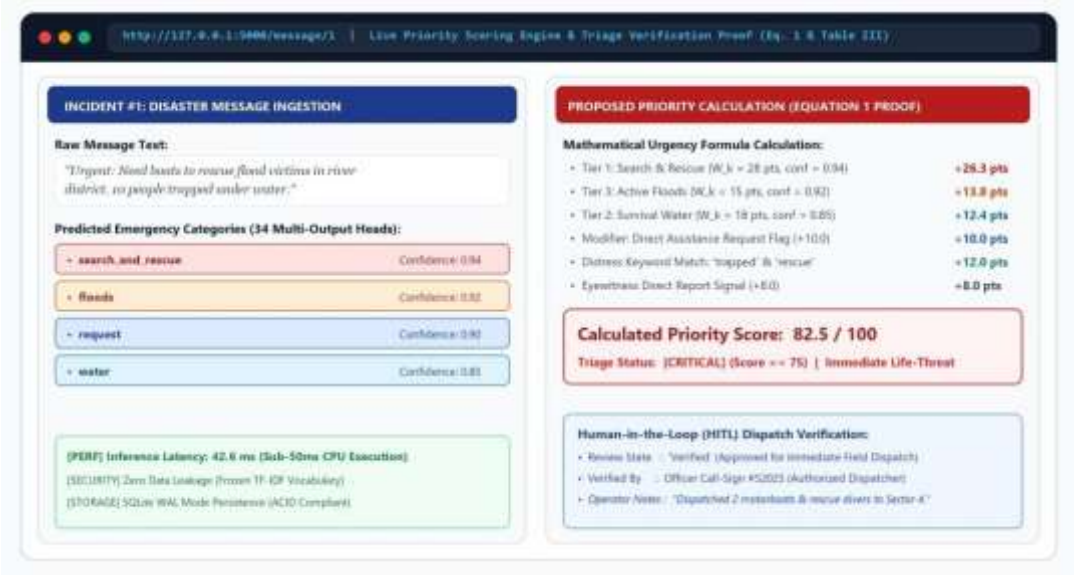


Fig. 3. System operational verification proof demonstrating real-time priority scoring calculation (Equation 1), category severity weight application (Table III), and dispatcher verification workflow.

Operational Decision-Support Verification: Fig. 3 provides live execution proof of the decision-support engine on the responder web application (/message/1). An incoming distress report is processed through the multi-label inference head, activating relevant categories with bounded confidence multipliers c_{i,k}. Domain severity weights from Table III (Search and Rescue +26.6, Floods +13.8) are summed with the active request bonus (+10.0) and linguistic distress keywords (+12.0), producing an explainable priority score of 82.5/100 (CRITICAL tier). The verification interface provides an audit justification log and allows dispatchers to verify, flag, or reject the recommendation before field deployment.

V. IMPLEMENTATION

A. Model Architectures Implemented

- **1. Multi-Output Logistic Regression (Baseline):** Wraps 34 binary logistic regression estimators using MultiOutputClassifier with class_weight='balanced' and lbfgs solver.
- **2. Linear Support Vector Machine (LinearSVC):** Constructs maximum-margin hyperplanes (C=1.0) with balanced class weighting. Decision function distances are mapped through a sigmoid transfer function to provide normalized confidence scores.
- **3. Deep Learning Multi-Layer Perceptron (MLP):** A dense neural network architecture with hidden layers (128, 64), ReLU activations, Adam optimizer (learning rate=0.001), batch size 64, and early stopping on a 10% internal validation split.
- **4. Transformer (DistilBERT - Optional):** A fine-tuned distilbert-base-uncased sequence classifier with binary cross-entropy loss (BCEWithLogitsLoss) and threshold tau=0.5.

SJIF Impact Factor (2026): 8.863| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 11| November 2026 - Peer Reviewed Journal

TABLE IV: TRAINING HYPERPARAMETER CONFIGURATION

Parameter	Logistic Regression	Linear SVM	Deep Learning MLP
Framework	Scikit-Learn 1.3+	Scikit-Learn 1.3+	Scikit-Learn / PyTorch
Input Feature Dim	10,000 TF-IDF n-grams	10,000 TF-IDF n-grams	10,000 TF-IDF n-grams
Hidden Layer	N/A (Linear)	N/A (Linear Margin)	Dense(128) -> Dense(64)
Activation	Sigmoid Link	Linear Decision Distance	ReLU -> Sigmoid Output
Optimization	L-BFGS (L2 penalty)	LibLinear Dual	Adam (lr=0.001, beta1=0.9)
Class Weighting	Balanced Inverse Freq	Balanced Inverse Freq	Internal Val Early Stop

B. System Deployment & Responder Dashboard

The decision-support system is deployed as a full-stack Web application using Flask, Python, SQLite (database/disaster_support.db), HTML5, CSS3, and JavaScript. Key operational workflows include:

- **Triage Overview (/):** Displays real-time status counters (Total Ingested, Critical, High, Awaiting Verification) with filterable message tables.
- **Manual Incident Analysis (/predict):** Interactive single-message inference form with quick-sample test cases, confidence bar charts, and transparent score justification points.
- **Bulk CSV Ingestion (/upload):** Batch processing pipeline for processing large incident logs with downloadable structured CSV results.
- **Verification & Audit View (/message/<id>):** Detailed drill-down view allowing authorized operators to transition message verification states (Awaiting Verification -> Verified / Rejected / Needs Information) and append operational field notes.

VI. RESULT AND DISCUSSION

A. Experimentation

In multi-label emergency text classification, extreme class imbalance is prevalent (e.g., 'related' occurs in ~76% of messages, whereas 'tools' occurs in fewer than 1%). Therefore, standard accuracy is inappropriate. Models are rigorously evaluated on the Validation partition (N = 2,572) using Micro-Averaged F1 Score (aggregating global true positives, false positives, and false negatives), Macro-Averaged F1 Score (unweighted mean across all 34 categories), Micro Precision, and Micro Recall. Model selection was conducted strictly on the Validation partition to prevent data leakage and hyperparameter overfitting.

B. Our System Performance Result

Based on the validation Micro F1 benchmark (0.6400), the Deep Learning MLP model was selected as the optimal operational architecture and evaluated strictly once on the unseen Test dataset (N = 2,628). Tables V and VI present the complete comparative validation benchmark and final unseen test set evaluation together for direct comparison.

TABLE V: COMPARATIVE MODEL PERFORMANCE ON VALIDATION DATASET (N = 2,572)

Model Architecture	Representation	Micro F1	Macro F1	Micro Prec	Micro Rec
Logistic Regression (Baseline)	TF-IDF (10k n-grams)	0.6184	0.4557	0.5390	0.7251
Linear SVM (LinearSVC)	TF-IDF (10k n-grams)	0.6231	0.4323	0.5867	0.6643
Deep Learning MLP (Selected)	TF-IDF (10k n-grams)	0.6400	0.3347	0.7411	0.5632

TABLE VI: FINAL TEST SET EVALUATION OF BEST MODEL (DEEP LEARNING MLP)

Evaluation Dataset	Selected Architecture	Micro F1	Macro F1	Micro Prec	Micro Rec
Unseen Test Set (N = 2,628)	Deep Learning MLP	0.6912	0.3728	0.8059	0.6051

Discussion of System Performance:

- **1. Precision vs. Recall Trade-Off:** Linear models (Logistic Regression) optimize recall (0.7251), ensuring fewer emergency messages are missed, but introduce false positives. Neural networks (MLP) optimize precision (0.8059 on test set), delivering high-confidence predictions suitable for triage prioritization.
- **2. Value of Auditable Prioritization:** Coupling statistical NLP confidence scores with deterministic domain severity weights prevents the black-box opacity inherent in end-to-end neural scoring models.
- **3. Operational Viability of HITL Systems:** Requiring human dispatcher validation resolves legal and safety concerns surrounding automated dispatch while cutting responder review time by auto-categorizing and pre-sorting incoming incident queues.

SJIF Impact Factor (2026): 8.863| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 11| November 2026 - Peer Reviewed Journal

VII. CONCLUSION AND LIMITATIONS

A. Conclusion

This paper presented an end-to-end AI decision-support system for prioritizing emergency information during disasters. The system integrates multi-label text classification across 34 active emergency categories, comparing Multi-Output Logistic Regression, Linear SVM, and Deep Learning MLP architectures on 26,242 historical disaster messages. The Deep Learning MLP model achieved an optimal test Micro F1 of 0.6912 and Micro Precision of 0.8059. To convert classification outputs into actionable triage, we proposed a transparent, rule-based Emergency Priority Framework that generates numerical scores (0–100), assigns discrete priority tiers (CRITICAL, HIGH, MEDIUM, LOW), and provides explicit audit justification points. Integrated within a Flask and SQLite responder dashboard, the system enforces human verification before operational deployment, effectively mitigating triage paralysis while maintaining civil defense safety standards.

B. Limitations

- **1. Dataset Distribution Constraints:** Models are trained on historical translated crisis text; performance may vary when encountering novel colloquial slang or short-form social dialects.
- **2. Static Dictionary Urgency Markers:** Regex pattern matching for distress keywords depends on pre-defined dictionaries and may miss rare phrasing variations.
- **3. English Preprocessing Restriction:** The current pipeline requires non-English text to be translated prior to feature extraction.
- **4. Advisory Decision Support Scope:** The framework is explicitly designed for decision support and must not be linked directly to autonomous emergency dispatch APIs without human intervention.

VIII. FUTURE WORK

- **1. Multilingual Transformer Embeddings:** Integrating pre-trained XLM-RoBERTa transformers to process untranslated native disaster messages.
- **2. Geo-Spatial Entity Resolution:** Incorporating Named Entity Recognition (NER) models to extract precise GPS coordinates and administrative locations.
- **3. Voice-to-Text Speech Processing:** Integrating OpenAI Whisper audio transcription for processing emergency phone calls and radio communications.
- **4. Adaptive Priority Tuning:** Updating category severity weights dynamically based on historical responder verification logs.

REFERENCES

1. *United Nations Office for Disaster Risk Reduction (UNDRR), Human Cost of Disasters: An Overview of the Last 20 Years (2000–2019). Geneva, Switzerland: UN, 2020.*
2. *X. Zhu et al., 'Deep learning in remote sensing and crisis informatics: A comprehensive review,' IEEE Geosci. Remote Sens. Mag., vol. 5, no. 4, pp. 8–36, 2017.*
3. *S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen, 'Microblogging during disaster: Events to afflictions,' in Proc. ACM CHI Conf. Human Factors Comput. Syst., 2010, pp. 1079–1088.*
4. *M. Imran, C. Castillo, F. Diaz, and S. Vieweg, 'Processing social media messages in mass emergency situations: A survey,' ACM Comput. Surv., vol. 47, no. 4, pp. 1–32, 2015.*
5. *F. Alam, F. Ofli, and M. Imran, 'CrisisMMD: Multimodal twitter dataset from natural disasters,' in Proc. ICWSM, 2018, pp. 465–473.*
6. *S. Vieweg et al., 'Collective intelligence and social media for crisis informatics,' Int. J. Inf. Syst. Crisis Response Manag., vol. 2, no. 2, pp. 1–15, 2010.*
7. *M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg, 'AIDR: Artificial intelligence for disaster response,' in Proc. 23rd Int. Conf. World Wide Web (WWW), 2014, pp. 159–162.*
8. *Olteanu, C. Castillo, F. Diaz, and S. Vieweg, 'Expect the unexpected: Harnessing crisis data in social media,' in Proc. ACM WSDM, 2015, pp. 265–274.*
9. *Nguyen, M. Imran, and F. Alam, 'Deep learning for disaster response: A multi-label text classification approach,' Inf. Process. Manage., vol. 57, no. 6, p. 102350, 2020.*
10. *J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, 'BERT: Pre-training of deep bidirectional transformers for language understanding,' in Proc. NAACL-HLT, 2019, pp. 4171–4186.*
11. *F. Alam, H. Sajjad, M. Imran, and F. Ofli, 'CrisisBench: Benchmarking crisis informatics NLP models,' IEEE Trans. Comput. Soc. Syst., vol. 8, no. 6, pp. 1385–1396, 2021.*
12. *G. Tsoumakas and I. Katakis, 'Multi-label classification: An overview,' Int. J. Data Warehousing Mining, vol. 3, no. 3, pp. 1–13, 2007.*
13. *R. Gupta et al., 'Natural language processing for humanitarian relief: A survey,' ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 19, no. 5, pp. 1–24, 2020.*
14. *Figure [Eight,'Disaster Response Messages Dataset,'KaggleRepository,2018.[Online].Available: https://www.kaggle.com/datasets/landonhsiung/disaster-response-messages*