sustainability

MDPI

Article

ResQConnect: An AI-Powered Multi-Agentic Platform for Human-Centered and Resilient Disaster Response

Savinu Aththanayake [1], Chemini Mallikarachchi [1], Janeesha Wickramasinghe [1], Sajeev Kugarajah [1], Dulani Meedeniya [1,*] and Biswajeet Pradhan [2]

[1] Department of Computer Science & Engineering, University of Moratuwa, Moratuwa 10400, Sri Lanka; nimetha.21@cse.mrt.ac.lk (S.A.); chemini.21@cse.mrt.ac.lk (C.M.); janeesha.21@cse.mrt.ac.lk (J.W.); kugarajah.21@cse.mrt.ac.lk (S.K.)

[2] Centre for Advanced Modelling and Geospatial Information Systems (CAMGIS), Faculty of Engineering and IT, University of Technology Sydney, Sydney 2007, Australia; biswajeet.pradhan@uts.edu.au

* Correspondence: dulanim@cse.mrt.ac.lk

Abstract

Effective disaster management is critical for safeguarding lives, infrastructure and economies in an era of escalating natural hazards like floods and landslides. Despite advanced early-warning systems and coordination frameworks, a persistent "last-mile" challenge undermines response effectiveness: transforming fragmented and unstructured multimodal data into timely and accountable field actions. This paper introduces ResQConnect,a human-centered, AI-powered multimodal multi-agent platform that bridges this gap by directly linking incident intake to coordinated disaster response operations in hazard-prone regions. ResQConnect integrates three key components. It uses an agentic Retrieval-Augmented Generation (RAG) workflow in which specialized language-model agents extract metadata, refine queries, check contextual adequacy and generate actionable task plans using a curated, hazard-specific knowledge base. The contribution lies in structuring the RAG for correctness, safety and procedural grounding in high-risk settings. The platform introduces an Adaptive Event-Triggered (AET) multi-commodity routing algorithm that decides when to re-optimize routes, balancing responsiveness, computational cost and route stability under dynamic disaster conditions. Finally, ResQConnect deploys a compressed, domain-specific language model on mobile devices to provide policy-aligned guidance when cloud connectivity is limited or unavailable. Across realistic flood and landslide scenarios, ResQConnect improved overall task-quality scores from 61.4 to 82.9 (+21.5 points) over a standard RAG baseline, reduced solver calls by up to 85% compared to continuous re-optimization while remaining within 7–12% of optimal response time, and delivered fully offline mobile guidance with sub-500 ms response latency and 54 tokens/s throughput on commodity smartphones. Overall, ResQConnect demonstrates a practical and resilient approach to AI-augmented disaster response. From a sustainability perspective, the proposed system contributes to Sustainable Development Goal (SDG) 11 by improving the speed and coordination of disaster response. It also supports SDG 13 by strengthening adaptation and readiness for climate-driven hazards. ResQConnect is validated using real-world flood and landslide disaster datasets, ensuring realistic incidents, constraints and operational conditions.

Keywords: agentic AI; decision support system; edge language models; floods and landslides; humanitarian logistics; multi-commodity vehicle routing; Retrieval-Augmented Generation (RAG); sustainable disaster management

Academic Editor: Xuelong Li

Received: 25 December 2025
Revised: 10 January 2026
Accepted: 16 January 2026
Published: 19 January 2026

1. Introduction

Over recent years, climate change has been linked to more frequent and severe floods, heavy rainfall and associated landslides, particularly in South and Southeast [1]. Monsoon and inter-monsoon events repeatedly trigger urban flooding and slope failures that disrupt livelihoods, damage infrastructure and cut off critical transport links [2]. Even when early-warning systems work, the combined effect of hazard intensity, population density and infrastructure fragility places sustained pressure on emergency services [3]. Consequently, humanitarian actors must make fast, high-stakes decisions with limited time and resources to mitigate cascading impacts [1].

Within this context, we focus on the operational "last mile" of disaster response: transforming fragmented citizen help requests into coordinated actions for floods and landslides [4]. The system of interest is an end-to-end, AI-assisted workflow that ingests help requests, generates grounded task plans, allocates multi-commodity relief resourcesand provides offline guidance when connectivity is poor. This stage is critical for agencies, non-governmental organizations (NGOs), and communities because delays or inconsistencies directly affect who receives help, how quickly high-risk groups are reached, and how fairly resources are shared across affected locations [5].

Existing work offers important but partial solutions. Formal coordination frameworks (e.g., Incident Command System (ICS) [6], Emergency Operations Centers (EOCs), United Nations (UN) cluster mechanisms) define clear roles and procedures, yet still depend on manual triage of phone calls, messages and social media posts and struggle with fragmented, heterogeneous information streams [7]. Crisis-informatics tools classify or prioritise messages but rarely connect them to structured tasking and routing [8]. Multi-agent systems and agent-based simulations capture coordination dynamics, but mostly in offline planning environments. Humanitarian logistics and vehicle-routing models optimise facility location, routing and equitable allocation, while typically assuming that demand points and priorities are already known. In parallel, edge and on-device LLM research demonstrates that compressed models can run on mobile devices, but these deployments are largely benchmark-driven rather than co-designed with disaster workflows [9].

These limitations motivate the need for an integrated, human-centred architecture that links AI-based task reasoning, adaptive resource distribution and edge-resident assistance for disasters. Here, human-centered means designing the system to (i) incorporate human oversight for high-impact decisions and (ii) prioritise accountability and transparency in generated recommendations. In particular, traditional rule-based decision-support tools lack the ability to decompose noisy, multi-faceted help requests into sub-tasks, motivating an agentic AI approach in which multiple LLM-driven components collaborate under an explicit orchestration scheme.

Existing disaster decision-support systems (DSS), such as coordination dashboards and rule-driven humanitarian platforms, primarily focus on situational awareness, information aggregation, or post-hoc planning support (e.g., Sahana EDEN and related crisis-management systems [10]). In contrast, ResQConnect advances beyond conventional DSS by operationalizing an end-to-end response loop that transforms unstructured citizen requests into structured tasks, SOP-grounded guidance, and urgency- and proximity-aware resource allocation. Furthermore, unlike monolithic DSS architectures, the proposed system adopts a multi-agentic LLM-based design that decomposes disaster response into specialized, cooperating agents. The integration of edge-resident language models for low-connectivity resilience and explicit human-in-the-loop governance further distinguishes ResQConnect from existing disaster DSS platforms, which typically assume persistent connectivity and limited human oversight once automation is introduced.

This study proposes ResQConnect, a human-centred, AI-powered multi-agent platform that connects citizen inputs to coordinated response decisions for floods and landslides. It is designed as a socio-technical system, by integrating technology with people and authorities. The platform does not replace human responders or existing disaster management structures. Instead, it supports them by providing AI-based assistance that fits within established roles, procedures, and decision-making processes used by disaster management and humanitarian organizations. The platform combines an agentic RAG workflow for task synthesis, an adaptive event-triggered (AET) routing engine for resource distribution and an edge-deployed small language model (SLM) for offline guidance, all under human oversight. This study supports the UN's, SDG 11 and SDG 13 [11], by strengthening disaster resilience, improving emergency coordination, and enabling more inclusive, human-centred response mechanisms. The work is organised around three research questions:

- RQ1—How can a multi-step, agentic RAG workflow be designed to improve retrieval quality and robustness?
- RQ2—What adaptive, event-triggered multi-commodity routing algorithm can allocate resources to evolving demands with minimal re-optimisation and route instability?
- RQ3—How can a SLM deployed on mobile devices provide reliable offline guidance under connectivity constraints?

At a high level, ResQConnect inputs citizen help requests and produces outputs: structured task breakdowns for authorities, priority and proximity-aware distribution plans for relief resources, and offline guidance for citizens. An LLM-driven agentic RAG pipeline converts incident reports into task plans grounded in a curated, metadata-rich disaster knowledge base. In this pipeline, specialised agents assume distinct roles: a metadata agent normalises and enriches incoming requests with location, hazard type and urgency; a retrieval agent formulates and refines queries; an assessor agent evaluates contextual adequacy and safety; and a planner agent synthesises step-by-step task sequences and resource needs. These agents communicate through a shared incident state. Associated resource requirements feed an AET multi-commodity routing algorithm that operationalises when-to-reoptimise decisions under online demand arrivals. In parallel, a compressed, fine-tuned edge LLM, converted to a mobile-friendly format, runs to answer disaster-related queries, when cloud access is unavailable, with seamless switching between online and offline modes. The main contributions are as follows.

1. We design and evaluate an LLM-driven, agentic RAG workflow that performs metadata extraction, adaptive query reformulation, contextual adequacy assessment and task generation over a curated disaster-response knowledge base, improving retrieval relevance and task quality over a Standard RAG baseline.
2. We develop an AET algorithm for dynamic multi-depot, multi-commodity disaster routing with atomic fulfilment and a route-stability penalty, and show that it achieves a favourable trade-off between priority-weighted response time, solver calls, and route instability compared to greedy and periodic baselines.
3. We develop and benchmark a compressed, domain-specialised edge LLM for offline disaster guidance, fine-tuning on a structured Q&A dataset and device-level evaluation of latency, memory usage and response quality, demonstrating its feasibility for low- and no-connectivity environments.

The paper is structured as follows. Section 2 reviews background. Section 3 presents the ResQConnect system design and methodology, including the agentic RAG workflow, adaptive routing engine and edge model. Section 4 describes the evaluation framework. Section 5 analyses the results, and Section 6 discusses limitations and future extensions.

2. Background and Related Studies

2.1. Multi-Agent Systems and Agentic RAG in Crisis Domains

Multi-agent systems (MAS) offer a natural abstraction for disaster management, enabling emergency services, authorities, volunteers, and affected communities to coordinate under uncertainty and time pressure. Classical MAS research treats autonomous agents as situated entities with their goals, and policies, coordinating via communication, negotiation, and shared task structures. Early disaster-oriented MAS and agent-based models (ABM) used this paradigm to simulate evacuations, resource deployment, and crowd behaviour across the mitigation, preparedness, response, and recovery phases, enabling "what-if" analysis of alternative policies and operating procedures. Luna-Ramirez and Fasli [12] integrated BDI-style cognitive agents in a disaster-rescue simulation combining Jason and NetLogo, showing how cognitively rich agents better capture human decision-making and coordination patterns in urban search-and-rescue scenarios than purely reactive models.

MAS and ABM have been used to represent complex socio-technical disaster environments, where agents denote organizations, response units, or population groups. These models support policy evaluation (evacuation orders, shelter allocation), vulnerability analysis, and training exercises [12]. However, their role is typically offline: they inform preparedness and planning, instead of directly mediating real-time data flows during an unfolding event. Also, most MAS-based disaster tools assume structured inputs (predefined scenario parameters), instead of noisy, multimodal streams of citizen reports, social media, and sensor data [13,14]. In parallel, the NLP community has developed RAG as a way to ground LLMs in external knowledge bases. Lewis et al. [15] introduced RAG models pairing a parametric seq2seq model with a dense vector index over Wikipedia and showed gains on open-domain QA tasks compared to parametric-only baselines. Standard RAG pipelines, however, are largely single-agent, one model handles query encoding, passage ranking, and answer generation, with limited explicit structure for reasoning about retrieval quality, coverage, or downstream operational constraints. As a result, RAG systems can return passages that are locally relevant but operationally inappropriate or insufficiently specific for high-stakes fields such as disaster response.

Recent LLM-based multi-agent studies bridge this gap by spliting complex tasks into interacting agent roles (planner, retriever, critic, tool-caller). Recent surveys characterize these workflows including iterative planning, tool use, inter-agent message passing, and critique loops, and report improved modularity and robustness over monolithic LLM pipelines for decision-making [16]. Also, engineering frameworks such as AutoGen have operationalized multi-agent conversation patterns with configurable agent roles and tool-use behaviors, providing a base layer for building real-world agentic systems [17]. Thus, "agentic RAG" models treat retrieval and generation as a coordinated multi-agent process rather than a single pipeline; recent surveys provide a taxonomy of agentic RAG designs and mechanisms for evidence selection, verification, and iterative refinement [18]. MAIN-RAG [19], for instance, introduces a team of LLM agents that collaboratively filter and score retrieved documents; an adaptive filtering method adjusts relevance thresholds based on score distributions, and inter-agent consensus is used to down-weight noisy passages without additional training. This multi-agent RAG design improves answer accuracy and robustness to noisy retrievals across QA benchmarks, showing how explicit agent roles around evidence curation can materially improve RAG performance.

In crisis domains, LLM-centric systems are starting to integrate RAG and agent-like components, but they remain relatively narrow in scope. Hong et al. [20] proposed a dynamic fusion framework for crisis communication that combines generations from an instruction-only pipeline and a RAG-based pipeline using a "fusion agent," and demonstrate improved professionalism, actionability, and overall response quality on social media

posts from Hurricane Irma. Otal et al. [21] described an LLM-assisted crisis management platform that integrates LLM components with data pipelines and collaboration tools for emergency response and public engagement, but primarily treat the LLM as a single assistant rather than an orchestrated team of agents. Meanwhile, the broader crisis-informatics studies has focused on classification and prioritization of social media messages, detection of actionable requests, and generation of templated responses, instead of full task synthesis pipelines tightly coupled to operational knowledge bases.

Accordingly, studies show that (i) classical MAS and ABM are powerful for modelling coordination, task allocation, and human–machine teaming in disaster scenarios [12]; (ii) RAG provides a principled mechanism for grounding LLMs in authoritative, up-to-date documentation [15]; and (iii) multi-agent LLM and multi-agent RAG frameworks can substantially improve retrieval quality and decision robustness in knowledge-intensive tasks [19]. However, existing LLM-based crisis tools typically stop at message classification or response drafting, without closing the loop to structured tasking, resource allocation, and coordination workflows.

2.2. Resource Distribution in Crisis Domains

Efficient and equitable resource distribution has become a focus of humanitarian logistics and disaster operations management. Reviews [22] emphasize that relief distribution differs from commercial logistics by, extreme demand surges, severe infrastructure disruption and a strong ethical imperative to minimize human suffering rather than cost alone. Holguín-Veras et al. [23] argued that post-disaster humanitarian logistics has "unique features" such as material convergence, rapidly evolving data and the primacy of deprivation costs (the suffering caused by delays), which make direct transplantation of commercial models inappropriate. This has motivated models that treat resource distribution not simply as a routing problem, but as a coupled set of decisions about where to hold stock, how to allocate and move it under uncertainty, and how to balance efficiency with fairness [24].

A first stream of work focuses on preparedness: deciding where to locate warehouses and how much relief stock to pre-position. Balcik and Beamon's [25] influential maximal-covering facility location model determines the number and placement of distribution centres and their inventory levels to maximize the coverage of affected populations under budget and capacity constraints. Subsequent models extend this to multi-echelon networks, multiple commodities and different disaster scenarios, often embedding risk measures and scenario-based stochastic programming [26]. Rodríguez-Espíndola et al. proposed an emergency preparedness system that coordinates resources from multiple organizations, highlighting that effective resource distribution begins with joint planning across agencies, not just optimizing a single actor's network [27]. These models provide a structural backbone for relief distribution, but they typically assume that demands and priorities are given and reasonably well approximated at planning time.

Once a disaster strikes, resource distribution becomes a dynamic problem: demand patterns evolve, infrastructure degrades and new data arrives continuously. Sheu's [28] dynamic relief-demand management model is a canonical example: it fuses heterogeneous data to forecast relief demand across regions, clusters affected areas and then uses multi-criteria decision-making to prioritise group allocations. Other work formulates integrated models that decide simultaneously on vehicle routing, shipment quantities and scheduling, under imperfect data and time-dependent travel conditions [26,29].

The "last mile" is recognised as challenging in humanitarian logistics, where damaged infrastructure, security issues, and information gaps complicate delivery to scattered or isolated communities [30]. Holguín-Veras et al. [31] introduced the notion of deprivation cost as an appropriate objective than classical logistics costs, capturing the rising human

suffering as essential goods are delayed. They showed that optimizing for deprivation cost leads to different, often ethically desirable, distribution patterns than minimizing travel cost alone. This framing explicitly links operational models to humanitarian goals, but it still assumes a relatively clear mapping from "needs" to demand quantities at each location.

Another concern is, who gets what, when and how to formalize fairness in resource allocation. Studies on equitable resource allocation, such as Luss's [32] lexicographic and max–min formulations, provides the mathematical building blocks to ensure that resources are shared to avoid extreme disparities across groups. Humanitarian-specific models build on this by explicitly encoding equity into objective functions and constraints. Huang et al. [33], addressed equitable last-mile distribution and compare max–min, proportional, and "equity" fairness measures, showing that incorporating fairness can significantly alter routing and allocation decisions compared to efficiency-only baselines. Wang et al. [34], proposed a multiperiod emergency resource allocation model that jointly optimizes efficiency and equity under uncertain information, using fuzzy representations for demand and travel time and quantifying equity via the loss associated with unmet demand.

More studies addressed combining deprivation cost with equity criteria. Ghahremani-Nahr et al. [35] designed a humanitarian relief logistics network that aims for "adequate and equitable" allocation of vital resources, measuring human suffering via deprivation cost, while ensuring that regions with different vulnerability levels are not systematically disadvantaged. Ethical analyses from public health and pandemic response similarly emphasize that fair allocation must consider vulnerability, exposure and structural disadvantage, not just headcount or first-come-first-served principles [36]. Together, these strands stress that resource distribution in crises is inherently normative: models must operationalize explicit value judgements about who should be prioritized, on what basis and at what cost.

Further, empirical studies showed practical challenges in resource distribution: fragmented supply chains, lack of coordination among agencies, bottlenecks in transportation and warehousing and difficulties leveraging spontaneous volunteers and grassroots initiatives. Holguín-Veras et al. [23] documented "material convergence" after major disasters large inflows of unsolicited goods that congest infrastructure and divert capacity away from critical items underscoring the need for information-driven filtering and prioritisation mechanisms. More studies have integrated volunteers and non-official responders into stochastic allocation frameworks, treating them as additional but uncertain capacity in the humanitarian relief chain [35]. Yet, as Altay and Green's [22] observed, most resource distribution models operate on relatively abstract representations of demand (region-level quantities and priorities) and assume that these inputs are available to the planner [24].

2.3. Edge and On-Device Deployment of LLMs

Deploying language models at the edge is increasingly viewed as a precondition for reliable AI support in connectivity-constrained settings such as disaster response. Stuides on edge AI and TinyML [37] showed that non-trivial deep models can run on microcontrollers and low-power devices, showing trade-offs between latency, energy, and accuracy for on-device inference. Dutta et al. [37] surveyed TinyML for IoT, emphasizing ultra-low-power inference and the need to co-design models with hardware and communication constraints. Studies on "on-device AI models" synthesizes design patterns for pushing perception, control, and language understanding onto phones, wearables, and embedded boards, framing edge inference to reduce latency and preserve data privacy [38].

With the emergence of LLMs, this agenda has shifted from small CNNs and RNNs to transformer-based language models under tight resource budgets. Friha et al. [39] provided a comprehensive survey of "LLM-based edge intelligence," covering architectures where LLMs are deployed on or near edge nodes to support applications such as industrial IoT,

smart cities and autonomous systems, and highlighting cross-cutting issues around energy, privacy, and trustworthiness. Complementary surveys focus specifically on deploying LLMs in resource-constrained environments, cataloguing strategies such as small-scale language models (SLMs), collaborative edge–cloud execution, and intelligent caching of model variants [40,41]. These works collectively argue that centralized, cloud-only LLM services are ill-suited for delay-sensitive or intermittently connected scenarios, and instead advocate for distributed, lightweight LLM deployments across the network edge.

A central enabler of edge-resident LLMs is, model compression and efficiency optimization. Surveys on model compression for deep neural networks describe techniques pruning, quantization, and knowledge distillation often combined with low-rank adaptation or parameter-efficient fine-tuning [42]. Dantas et al. [43] systematically review compression methods specifically for LLMs, showing that mixed-precision quantization (e.g., 8-bit or 4-bit weights), structured pruning of attention and feed-forward blocks and distillation into smaller student models can yield substantial reductions in memory footprint and compute cost with modest quality loss. Earlier work on compact transformer models such as MobileBERT [44] demonstrated that aggressively compressed BERT-style models can run efficiently on mobile-class devices while preserving most upstream performance, foreshadowing similar design principles for small, task-specialized LLMs. More recent hardware–algorithm co-design efforts introduce quantization-aware training and attention-aware post-training quantization tailored to LLM workloads, as well as sparsity-inducing techniques to reduce activation and weight density for faster inference on accelerators [40].

Edge deployment also depends on system-level orchestration between devices, edge servers and the cloud. Duan et al. discuss edge-cloud computing and federated/split learning as mechanisms to distribute training and inference across heterogeneous nodes, balancing energy, bandwidth, and latency constraints [45]. Building on these ideas, recent LLM-focused frameworks propose split inference where early transformer layers run locally while deeper layers or specialized decoders execute on nearby edge servers, with adaptive offloading policies driven by network state and task urgency [40,41]. Habibi and Ercetin [40] formulated edge-LLM inference as a cost-aware optimization problem over layer placement and scheduling, showing that intelligent allocation across edge GPUs and CPUs can significantly improve throughput and energy efficiency under service-level constraints. Parallel work on small-scale language models for IoT demonstrates that models in the 2–7B parameter range can run on devices such as Raspberry Pi or Jetson boards, enabling offline natural-language interfaces to local sensor data [41].

For disaster response specifically, edge and on-device LLMs are attractive because they directly address connectivity, privacy, and robustness concerns. Edge-deployed LLMs can provide local reasoning and summarization capabilities, backed by compressed domain-specific models and cached retrieval indices, while synchronizing with cloud-hosted models when connectivity allows. Existing edge-LLM work, however, is largely benchmark-driven optimizing token throughput, power consumption, or scheduling under synthetic workloads rather than being co-designed with high-stakes, multi-stakeholder workflows such as emergency coordination [40].

2.4. Comparative Analysis of Related Approaches

A growing body of work explores agent-based extensions of RAG and LLM-driven decision support. Table 1 summarises these frameworks against five dimensions that are central to our setting: metadata-aware retrieval, iterative query reformulation, adequacy/evidence verification, multi-agent retrieval pipelines and explicit grounding in domain standard operating procedures (SOPs). As Table 1 shows, existing systems typically support iterative reformulation or evidence checking, but rarely combine all five

capabilities in a single, coordinated pipeline. In particular, most solutions lack explicit SOP grounding and operate with either a single agent or loosely defined agent roles. Our agentic RAG system differs by providing a multi-agent workflow with dedicated metadata, retrieval, assessor and planner agents that operate over a curated disaster-response knowledge base, with SOP-aware grounding as a first-class objective. In Table 1, we use "Partial" when a capability is present only in a limited or indirect form. For example, Self-RAG [46] performs iterative self-reflection, but it does not explicitly rewrite the query as a separate query-refinement step; hence it is marked "Partial". Further, "X" and "✓" denote the features that are not-considered and considered, respectively, from each of the study.

Table 1. Comparison of Agentic RAG Frameworks.

Study	Metadata-Aware Retrieval	Iterative Query Reformulation	Adequacy/Evidence Verification	Multi-Agent Retrieval Pipeline	Domain/SOP Grounding
Self-RAG [46]	X	Partial	✓	X (single agent)	X
Corrective RAG [47]	X	✓	✓	Partial	X
ActiveRAG [48]/RQ-RAG [49]	X	✓	✓	Partial	X
MAIN-RAG [19]	X	X	✓	✓	X
LlamaIndex Agentic RAG [50]	X	✓	Partial	Partial	X
Our Agentic RAG System	✓	✓	✓	✓	✓

In humanitarian logistics, numerous models address vehicle routing under uncertainty, including dynamic relief-demand models, equitable last-mile distribution and rolling-horizon VRP formulations [51]. These studies consider dynamic requests, equity objectives and stochastic travel times to different extents. Table 2 compares representative approaches to our AET routing along five axes: support for dynamic requests, event-triggered policies, multi-commodity flows, priority weighting/equity and explicit stability control. Most prior work supports dynamic requests and, in some cases, equity, but typically relies on time-based re-optimisation and single-commodity flows, with limited attention to route stability. As indicated in Table 2, our AET routing model explicitly combines multi-commodity allocation, priority-weighted objectives and an event-triggered policy that caps solver calls and controls route nervousness, better matching operational constraints in field deployments. In Table 2, "Partial" indicates that the study addresses the factor in a simplified way but does not model it as a full optimization objective or constraint. For example, study [28] supports priority ordering but does not include an explicit equity rule (ensuring balanced distribution across regions). So it is marked "Partial".

Table 2. Comparison of Dynamic and Adaptive Humanitarian Routing Approaches.

Study	Dynamic Requests	Event-Triggered Policy	Multi-Commodity	Priority Weighting/Equity	Stability Control
Dynamic Relief Demand Model [28]	✓	X	X	Partial	X
Equitable Last-Mile Distribution [33]	X	X	✓	✓	X
Deprivation Cost Model [23]	Partial	X	X	✓	X
Stochastic Dynamic Routing [52]	✓	X	Partial	Partial	X
Emergency Logistics Coordination [53]	✓	X	X	Partial	X
Rolling Horizon VRP [51]	✓	X (time-based)	Partial	Partial	X
Proposed System—AET Routing	✓	✓	✓	✓	✓

Recent work on small or compressed language models demonstrates that LLMs can run directly on mobile and embedded devices, using techniques such as quantisation and architecture scaling. However, many studies focus on generic benchmarks or hybrid online/offline setups rather than domain-specific, fully offline disaster guidance. Table 3 contrasts representative edge/SLM studies with our system along four dimensions: mobile execution, quantised models, offline operation and domain fine-tuning. As summarised in Table 3, existing work often achieves on-device execution and, in some cases, quantisation, but typically lacks domain-specific fine-tuning and complete offline operation. Our system, in contrast, deploys a quantised, disaster-specialised model that is both fine-tuned on structured Q&A data and evaluated for fully offline guidance on smartphone-class hardware, aligning more closely with the requirements of connectivity-constrained disaster contexts. In Table 3, "Partial" means the feature is demonstrated only under restricted conditions rather than guaranteed end-to-end. For instance, study [54] may run the model on-device but still rely on cloud fallback for some cases; therefore it is marked "Partial".

Table 3. Comparison of Edge and Compressed LLM Studies.

Study	Mobile Execution	Quantised Model	Offline Operation	Domain Fine-Tuning
Demystifying SLMs for Edge Deployment [54]	✓	✓	Partial	X
Edge-First LLM Inference [55]	✓	Partial	Partial	X
TFLite for TinyML systems [56]	✓	✓	✓	X
Embedded SLM + Local RAG Case Studies [57]	✓	✓	✓	X
Proposed System	✓	✓	✓	✓

3. System Design and Methodology

3.1. System Overview

ResQConnect is designed as an end-to-end platform that transforms citizen help requests into structured task plans, resource-distribution routes, and offline guidance that support coordinated disaster response. It is designed strictly as a decision-support platform rather than an autonomous response system. As shown in Figure 1 the system integrates three core subsystems an agentic RAG workflow, an adaptive resource-distribution engine, and an edge-deployed chatbot into a single pipeline connecting user inputs to operational outputs. All outputs generated are presented as recommendations to human operators. Final decisions remain under the control of authorized personnel within disaster-management agencies. The system does not execute actions independently or override human judgment. The design choice ensures accountability and aligns with established practices.

Incoming reports from citizens may arrive in the form of text messages. These inputs are normalised and enriched with metadata reflecting disaster type, urgency, and contextual attributes. This structured incident representation forms the entry point to the Agentic RAG Workflow, where specialised LLM-driven components carry out metadata extraction, query reformulation, retrieval over a curated knowledge base, and self-assessment. The output of this workflow is a grounded task breakdown detailing the required actions and associated resource needs, synthesised in alignment with official SOPs, technical manuals, and clinical or operational guidelines.

The resulting task and resource requirements are passed to the Resource Distributor Workflow, which operates over current inventory and resource-availability data. This module considers spatial deviation, urgency, slack depletion, and scheduling constraints to generate an efficient resource-distribution plan. Its event-triggered update mechanism allows the system to incorporate newly arriving requests without recomputing the entire routing solution, enabling responsive reallocation as conditions evolve.

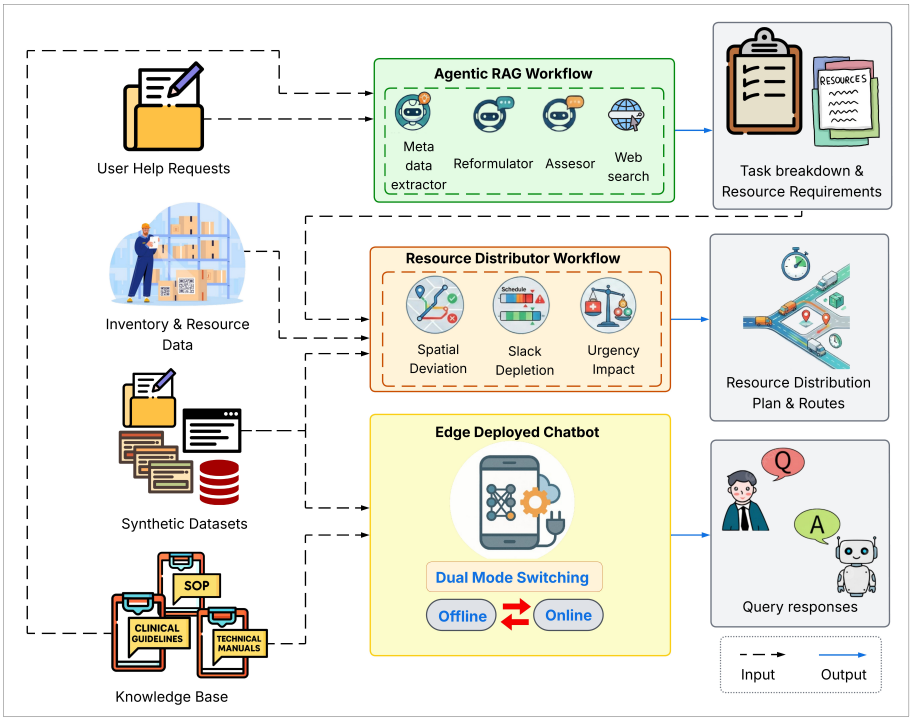

Figure 1. High Level System Overview.

In parallel, ResQConnect incorporates an edge-deployed chatbot, a compressed domain-specialised LLM running on mobile devices. While the cloud pipeline handles full incident understanding and resource planning, the edge model ensures that affected individuals can still access policy-aligned information and safety guidance during low-connectivity periods. A dual-mode switching mechanism enables transparent transitions between online and offline modes, maintaining basic usability even when network conditions deteriorate.

Together, these components form a coherent end-to-end system: users submit help requests, the platform interprets them through an agentic RAG process, produces structured task and routing outputs through adaptive resource distribution, and provides citizens with online or offline assistance through an embedded mobile chatbot. The design ensures that both operational agencies and affected communities receive timely, actionable information, even under uncertain connectivity and rapidly evolving disaster conditions.

3.2. Agentic RAG Workflow for Task Breakdown

3.2.1. Agentic RAG Architecture

The Agentic RAG pipeline converts public help requests into structured, field-executable tasks. Its objective is to provide factually reliable, domain-grounded responses through a metadata-aware, multi-step reasoning process. By continuously assessing contextual sufficiency and reformulating under-specified queries, the pipeline maintains both semantic precision and operational reliability. Given the high-stakes nature of disaster response, the workflow intentionally avoids non-deterministic agent-to-agent interactions and follows a controlled, deterministic execution order, enforced through explicit guardrails. The architecture shown in Figure 2 is composed of a sequence of interconnected nodes. The role, interaction logic, and contribution to the overall retrieval–generation loop of each node is detailed as follows.

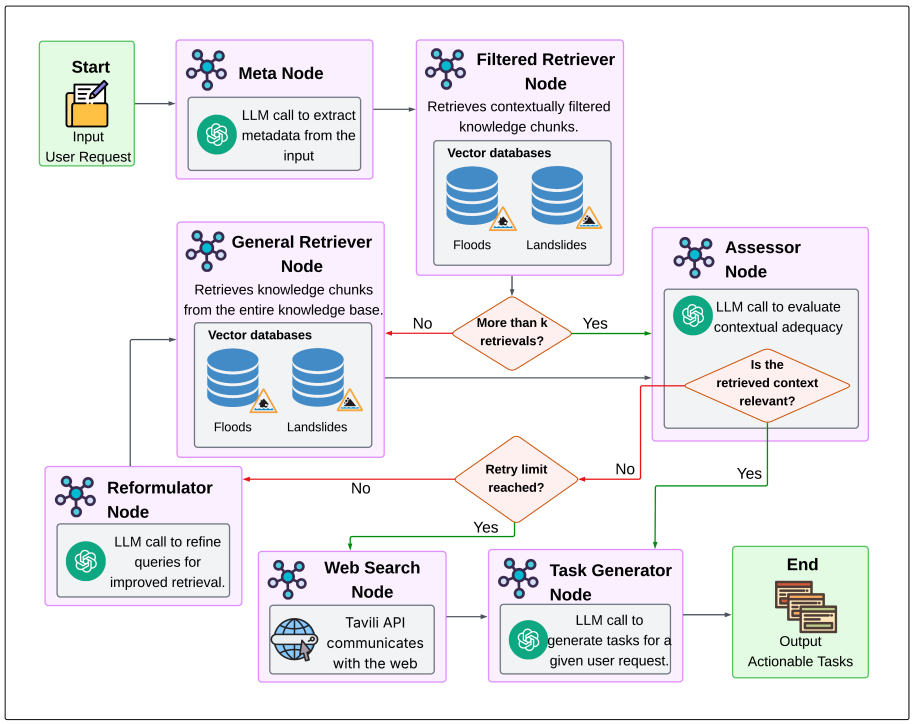

Figure 2. Agentic Retrieval Workflow.

1. Meta Node: Upon receiving a help request, the Meta Node employs a LLM to infer structured structured, hazard-aware metadata such as disaster type, location, urgency, and agency. The model selects from a predefined metadata dictionary to standardize downstream retrieval queries. A zero-shot, instruction-based prompt enforces consistent metadata extraction under a normalized schema, ensuring the output strictly conforms to the expected JSON format.
2. Filtered Retriever Node: Using the metadata, this node retrieves semantically related knowledge chunks from the curated disaster-response knowledge base described in Section 4.1.1, for context that aligns with the detected disaster type and operational stage. When the filtered search results in fewer than k relevant matches, the workflow automatically falls back to the General Retriever Node.
3. General Retriever Node: Performs a broader, unconstrained search across all knowledge domains to ensure coverage when metadata is sparse or incomplete.
4. Assessor Node: Performs a lightweight LLM-based evaluation of contextual adequacy. Using a concise rubric-style prompt, it checks whether the chunks retrieved from the internal knowledge base are both topically relevant and operationally specific enough to support task generation (i.e., they contain concrete "what-to-do" and "how-to-do" instructions). If the context is judged adequate, it is forwarded directly to the Task Generator Node. If the context is vague, off-topic, or missing key procedural details, the Assessor flags it as inadequate and routes control back to the Reformulator Node for another reformulation–retrieval.
5. Reformulator Node: This node applies an adaptive query-reformulation strategy designed to enhance similarity search performance within the vector database. It rewrites the original user request into a self-contained, guideline-oriented query that explicitly expresses the operational intent as shown below.

    - Original request: "We are a group of five tourists. Our bus is stuck in the mud. There are big trees on the road. We have no food or water. My friend is hurt, he fell and his arm looks bad. We need a way to go back to the city."

- Reformulated query: "How to get help for stranded travelers; how to provide first aid for arm injuries; how to find food and water in emergency situations; how to navigate out of muddy conditions."

Adaptive reformulation is required because help requests in disaster contexts are often ambiguous, multi-faceted and linguistically noisy, making a single-shot retrieval query insufficient for extracting operationally relevant guidance.

6. Web Search Node: If, after a maximum number of reformulation–retrieval cycles, the Assessor still deems the knowledge-base context inadequate or if the request is classified as out of scope for the curated corpus, the system escalates to the Web-search Node. This node uses the Tavily API to retrieve passages from authoritative disaster-management and humanitarian sources. The returned snippets are normalized into the same chunk format used for the internal corpus and replace the previous knowledge-base context, forming a new evidence set that is passed to the Task Generator Node, with metadata tags indicating that they originate from the web.
7. Task Generator Node: Once an adequate context is available (either from the knowledge-base or, if that fails, from the Websearch) this node synthesizes a structured Task Breakdown. Guided by a prompt, it translates the provided context into ordered, field-executable subtasks for relevant agencies (e.g., medical triage, evacuation routing, debris clearance).

Anchored to RQ1, it shows how structured, node-level reasoning transforms RAG from a simple similarity-based retriever into a dependable decision-support mechanism for high-stakes domains such as disaster management.

3.2.2. Objective Formulation

We formalize the disaster–response retrieval task as an iterative optimization problem aimed at maximizing the operational utility of retrieved context. Let $\mathcal{K} = \{(d_i, m_i)\}_{i=1}^{N}$ denote the knowledge base, where d_i is a text chunk and m_i its associated metadata vector (e.g., disaster_type, doc_type, operational_phase). Given a help request u, the system first extracts a metadata constraint set $\mathcal{M}_u = f_{\text{meta}}(u)$. The retrieval search space is restricted to a subset $\mathcal{K}' \subseteq \mathcal{K}$ via a binary masking function, as in (1).

$$\mathcal{K}' = \{(d, m) \in \mathcal{K} \mid \mathbb{I}(m, \mathcal{M}_u) = 1\} \tag{1}$$

where $\mathbb{I}(\cdot)$ is an indicator function returning 1 if the document metadata m aligns with the request constraints $\mathcal{M}_u$, and 0 otherwise.

The goal of the agentic workflow is to find an optimal query representation q^* that retrieves a context set $\mathcal{C}^* \subset \mathcal{K}'$ maximizing an assessor score $\mathcal{S}$. We define the retrieval function $\text{Ret}(q, \mathcal{K}')$ as returning the top-k chunks based on cosine similarity. The process is modeled as a feedback loop, as in (2):

$$q_{t+1} = \begin{cases} q_t, & \text{if } S(\text{Ret}(q_t, \mathcal{K}'), u) \geq \tau, \\ \Phi(q_t, H_t), & \text{otherwise,} \end{cases} \tag{2}$$

where $q_0 = u$, τ is the adequacy threshold, S is the assessment function (evaluating relevance and specificity), and Φ is the reformulation function conditioned on the history of critiques H_t. The system terminates when the score threshold is met or maximum iterations are reached, ensuring the generated tasks are grounded in the most operationally relevant subset of $\mathcal{K}$.

The retrieval process in Equation (2) follows a bounded iterative loop that repeats retrieval only when the available context is insufficient, with an explicit cap on the number

of iterations. As a result, the worst-case computational cost grows linearly with the number of iterations, while retrieval remains logarithmic in the size of the knowledge base under approximate nearest-neighbour indexing. In practice, the loop converges quickly: most requests terminate after a single retrieval pass, and over 90% complete within two iterations, ensuring predictable runtime behavior and avoiding unbounded reasoning loops.

3.2.3. Prompt Engineering Strategy

The Agentic RAG workflow employs a deliberately constrained prompt-engineering strategy to ensure consistency, controllability, and robustness across task creation and retrieval-related prompts. Rather than relying on unconstrained natural language instructions, prompts are systematically structured to reduce ambiguity, limit output variability, and enforce adherence to predefined schemas and operational objectives. Formal semantics are introduced at a high level to clarify how these strategies influence model behavior without overcomplicating the formulation.

Conditioned Generation and Contextual Grounding

Role specification, goal definition, and contextual grounding jointly constrain the model's generation process. The language model is explicitly conditioned on the user request, retrieved evidence, and prompt-level control variables, restricting the output space to relevant and grounded responses:

$$y \sim p(y \mid u, E, r, g, P) \tag{3}$$

where u denotes the user request, E the retrieved knowledge base content, r the assigned role, g the operational goal, and P the structured prompt. By conditioning on these elements, the model is discouraged from relying on external or assumed knowledge and is guided toward outputs that directly address the stated task.

Constraint Anchoring and Output Validity

To prevent malformed or hallucinated responses, strict schemas and rule sets are enforced through constraint anchoring. An output is accepted only if it satisfies all structural and logical constraints defined by the prompt:

$$V(y) = \begin{cases} 1, & \text{if } y \models \mathcal{S} \ \wedge \ y \models \mathcal{C} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\mathcal{S}$ represents the required output schema and $\mathcal{C}$ denotes constraint rules such as controlled vocabularies, mandatory fields, fixed formatting, and explicit failure conditions. Outputs with $V(y) = 0$ are rejected or regenerated, ensuring deterministic, machine-readable results suitable for downstream processing.

Supporting Prompt Engineering Mechanisms

Equations (3) and (4) define the two fundamental semantic constraints governing the prompt engineering strategy, namely conditioned generation and constraint-based output acceptance. The remaining prompt engineering mechanisms are designed to reinforce these constraints throughout the prompt lifecycle.

Hierarchical structuring organizes each prompt into clearly defined sections (e.g., role description, input context, output format, rules, and validation), improving interpretability and reducing ambiguity during generation. Implicit stepwise reasoning encourages the model to internally reason through the task before producing an output, enhancing logical coherence while exposing only the final structured result. Validation reminders explicitly in-

struct the model to verify completeness, internal consistency, and schema compliance prior to finalizing the response, strengthening adherence to Equation (4). Few-shot structural anchoring is used selectively by providing output skeletons that stabilize formatting without influencing semantic content. Finally, precision framing and meta-instructional clarity enforce a strict separation between internal reasoning and the emitted output, limiting extraneous text and reducing variance in model behavior.

Together, these mechanisms operationalize the semantic constraints expressed in Equations (3) and (4), improving robustness, consistency, and controllability of the Agentic RAG workflow without introducing excessive prompt complexity.

3.3. Adaptive Event-Triggered Multi-Commodity Routing

This section presents the Adaptive Multi-Commodity Routing Module in ResQConnect, designed for disaster settings where demand patterns, travel times and operational constraints evolve rapidly. Each affected location may request a heterogeneous basket of resources (food, water, medical supplies, shelter), with items consuming vehicle capacity differently. Multiple depots coordinate vehicle dispatch, requests arrive online with discrete priority classes and the goal is to maximise coverage, minimise unmet demand and prioritise urgent requests while limiting unnecessary re-planning and route instability. Classical vehicle-routing models typically assume static demand, single-commodity loads and fixed travel times, making them unsuitable for this dynamic humanitarian context. To address this, we combine a multi-commodity routing formulation with an AET mechanism that determines when global re-optimisation is beneficial rather than solving at every event.

3.3.1. Problem Formulation

We consider a disaster-affected region represented by a directed graph $G = (V, E)$ consisting of multiple depots and a set of demand points requiring multi-commodity aid. The system evolves over a planning horizon $[0, T]$ during which requests arrive in real time. Each request i specifies a demand vector $\{d_i^r\}$ across a fixed set of resource types and belongs to a priority class π_i, which is mapped to a numerical weight p_i so that high-, medium-, and low-urgency requests can be distinguished in the optimization. Vehicles depart from depots with capacity limits Q_k and operate over a road network whose travel times c_{ij} vary with congestion and degradation.

Atomic fulfillment is assumed: each node must receive its entire multi-commodity demand vector from a single vehicle, because partial or split deliveries are operationally infeasible in disaster conditions and may jeopardize timely relief.

At each decision epoch τ_m, the system is treated as a deterministic snapshot. The known inputs include the set of unserved requests $C(\tau_m)$, their demand vectors d_i^r, the effective load $L_i = \sum_r u_r d_i^r$, the priority weights p_i, the current location and remaining capacity of each vehicle, and the travel-time estimates $c_{ij}(\tau_m)$. The model introduces binary routing variables x_{ij}^k, binary service indicators z_i, continuous delivery quantities q_i^{rk}, and arrival-time variables T_i^k, with $T_i^\star = \min_k T_i^k$ denoting the earliest arrival time at node i. Previously committed arcs from the prior plan are encoded by $\bar{x}_{ij}^k \in \{0, 1\}$.

The optimization problem minimizes a weighted combination of terms that capture the key operational goals of the routing system. The formulation penalizes total travel time to encourage efficient vehicle movements, penalizes late or delayed arrivals through terms involving the arrival times $T_i^\star$, and penalizes leaving any request unserved through the binary variables z_i. Priority-awareness is incorporated through the weights p_i, which scale the penalty associated with response delay or missed service so that high-urgency requests exert a stronger influence on the objective. In addition, deviations from previously committed arcs are penalized via the routing variables x_{ij}^k and $\bar{x}_{ij}^k$, thereby discouraging

unnecessary re-routing unless it meaningfully improves priority-weighted system performance. The static optimization problem at decision epoch τ_m seeks to minimize a composite cost balancing travel time, priority-weighted response time, penalties for unserved nodes, and penalties for altering previously committed arcs.

Conceptually, this formulation provides a structured way to balance the competing pressures present in disaster logistics: vehicles must be routed efficiently, urgent locations must be reached quickly, and previously issued plans should not be disrupted without good reason. Each term in the objective captures one of these operational trade-offs. The travel-time variables x_{ij}^k determine how vehicles move through the network, the arrival-time variables $T_i^\star$ capture how quickly each request is served, the service indicators z_i reflect whether a location ultimately receives aid, and the deviation terms involving x_{ij}^k and $\bar{x}_{ij}^k$ measure how much the new plan departs from the previously committed one. By integrating travel efficiency, urgency-awareness, service guarantees, and route stability into a single optimization framework, the model ensures that decisions made at each epoch reflect both the immediate needs of the affected population and the practical limitations of the fleet. This unified approach allows the system to adapt in real time while still maintaining coherent, high-quality routing plans across the planning horizon.

$$T = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij}\, x_{ij}^k \tag{5}$$

$$S = \sum_{i \in C} p_i\, T_i^\star \tag{6}$$

$$L = \beta \sum_{i \in C} p_i (1 - z_i) \tag{7}$$

$$R = \gamma \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} (1 - \bar{x}_{ij}^k)\, x_{ij}^k \tag{8}$$

$$\min Z = T + S + L + R \tag{9}$$

Here, term (5) penalises total travel time; (6) encourages early service of high-priority nodes; (7) penalises leaving nodes unserved (especially high-priority ones); and (8) discourages unnecessary route changes by comparing the new routing plan to previously committed arcs. The parameters β and γ regulate the severity of penalties for unserved nodes and route instability, respectively. This objective provides a structured way to balance the main operational pressures in disaster logistics: routing efficiency, urgency-awareness, service coverage and plan stability.

Priority weights associated with each request represent discrete urgency classes (e.g., high, medium, low) assigned during incident intake based on hazard severity, vulnerability indicators, and reported needs. These priority weights are fixed for the lifetime of a request and do not evolve dynamically over time. This design choice reflects operational practice in disaster response, where urgency classification is typically determined during triage and remains stable to preserve fairness, transparency, and accountability in resource allocation. System adaptivity is instead achieved through the event-triggered re-optimization mechanism, which reacts to newly arriving high-priority requests and changing system conditions without altering previously assigned priority weights.

Constraints enforce that each served node has exactly one incoming and one outgoing arc, vehicles start and end at their depots, multi-commodity capacity limits are respected, atomic fulfillment is satisfied, arrival times follow travel-time conditions, and subtours are eliminated. This formulation captures the spatial, temporal, and multi-commodity requirements essential for disaster-response routing.

3.3.2. Algorithm Design

The routing algorithm combines deterministic optimisation with event-triggered decision making, as stated in Algorithm 1. Solving the full multi-depot, multi-commodity problem at every event would be computationally prohibitive and operationally disruptive during periods of frequent request arrivals. To manage this, the algorithm follows a two-stage strategy. First, a baseline routing plan is generated by solving the static model at the most recent decision epoch; this plan functions as the committed schedule, and vehicles follow it unless a significant disruption arises. Between major re-optimisation points, new requests are accommodated through lightweight local adjustments, such as inserting the node into an existing route with minimal deviation. This design preserves route stability while enabling rapid responsiveness. When an event occurs at time t, the system evaluates whether the change is substantial enough to warrant a global re-optimization. This assessment is performed using the disruption score $\mathcal{D}(t)$ as in (10), which aggregates three components.

The behavior of the AET routing mechanism is primarily influenced by the relative weighting of urgency, spatial deviation, and slack in the disruption score, as well as by the decay rate of the triggering threshold. Higher urgency weights increase responsiveness to critical requests but may lead to more frequent re-optimization, while higher stability penalties favor plan consistency at the expense of delayed adaptation. In the evaluated scenarios, fixed parameter settings were used across all load conditions to ensure fair comparison and the results indicate that the policy is robust to moderate parameter variation rather than relying on fine-tuned thresholds.

When an event occurs at time t, the system evaluates whether the change is substantial enough to warrant a global re-optimisation. This assessment is performed using a disruption score $\mathcal{D}(t)$, which aggregates three components:

$$\mathcal{D}(t) = w_1 \Phi_{\text{urgency}} + w_2 \Phi_{\text{spatial}} + w_3 \Phi_{\text{slack}} \tag{10}$$

The urgency component shown in (11) reflects the relative priority of the new request.

$$\Phi_{\text{urgency}} = \frac{p_{\text{new}}}{\max_{i \in \text{unserved}} p_i} \tag{11}$$

The spatial component shown in (12) captures how far the new request lies from existing routes.

$$\Phi_{\text{spatial}} = \frac{\text{distance from new node to nearest route centroid}}{\text{a maximum distance scale}} \tag{12}$$

The slack component reflects the tightness of current schedules or capacity usage, normalized to the range $[0, 1]$. Together, these components quantify how disruptive the event is with respect to priority, geography, and system flexibility. The disruption score is compared with an adaptive threshold $\Theta(t)$, defined as (13) where Θ_0 is the initial post-optimization threshold, α is the decay rate, and t_{last} is the time of the last global re-optimization.

$$\Theta(t) = \Theta_0 \, e^{-\alpha (t - t_{\text{last}})} \tag{13}$$

A new global optimization is triggered when $\mathcal{D}(t) \geq \Theta(t)$. Otherwise, the request is handled locally without modifying the overall plan. High-priority requests naturally produce larger values of Φ_{urgency} and therefore elevate $\mathcal{D}(t)$, allowing urgent events to trigger immediate re-optimization when necessary. By controlling solver invocations through the comparison of $\mathcal{D}(t)$ and $\Theta(t)$, the method maintains a balance between responsiveness and

stability: it reacts promptly when disruptions meaningfully affect operational objectives, while avoiding excessive computation and unnecessary changes to routes.

Algorithm 1 Adaptive Event-Triggered MD-CVRP-MCD Algorithm

Initialize $t \leftarrow 0$, $t_{\text{last}} \leftarrow 0$
Compute initial plan by solving static MD-CVRP-MCD
while $t \leq T$ **do**
  Observe next event time t_{ev} and event type ω
  Update state from $S(t^-)$ to $S(t_{\text{ev}}^+)$
  if ω is NEW_REQ **then**
    Compute $\mathcal{D}(t_{\text{ev}})$ using urgency, spatial, and slack features
    Compute threshold $\Theta(t_{\text{ev}}) = \Theta_0 \cdot e^{-\alpha(t_{\text{ev}} - t_{\text{last}})}$
  end if
  if $\mathcal{D}(t_{\text{ev}}) \geq \Theta(t_{\text{ev}})$ **then**
    Partially solve static problem from $S(t_{\text{ev}}^+)$
    Accept re-optimized plan and update committed arcs
    $t_{\text{last}} \leftarrow t_{\text{ev}}$
  else
    Maintain current plan (insert new node via cheapest insertion)
  end if
  Advance vehicles and time to next event
  $t \leftarrow t_{\text{ev}}$
end while

3.3.3. Integration with ResQConnect Workflow

ResQConnect integrates the routing engine into the broader platform to support real-time disaster-response operations. All relief centres and available resources are registered within the system, including depots, vehicles and the current multi-commodity inventories at each location. As users submit incident reports through the platform, the workflow converts each message into a structured demand specification, identifying the request's location, required resource types and quantities, and associated priority level. Upon administrator approval, these AI-derived demands are forwarded to the routing module.

As new demands arrive, the system updates the operational state and applies the AET policy described above to decide whether to trigger a global re-optimisation or incorporate the request via local adjustments. In both cases, the module outputs clear dispatch instructions specifying which vehicle should serve the request, the quantity of each resource to be delivered, the sequence of locations to be visited and the expected arrival times.

In practical deployments, the platform also integrates real-time external data streams such as traffic conditions, road closures, and travel-time estimates via APIs from providers like Google Maps and other mapping or mobility services. These data inputs continuously refine the system's situational awareness, ensuring that routing decisions remain accurate and adaptive under rapidly evolving disaster conditions.

3.4. Edge-Deployed LLM for Offline Inference

3.4.1. Model Selection and Fine-Tuning

The offline edge inference component of ResQConnect is grounded in a systematic evaluation of several compact open-source language models suitable for on-device deployment. The baseline models considered included Qwen2.5-0.5B [58], TinyLlama-1.1B [59], phi-1.5 [60], and gemma-3-1b-it [61]. These models were compared across a set of criteria relevant to edge environments and disaster-response reasoning tasks, including model size, inference latency, perplexity, BoolQ [62] accuracy, SQuAD [63] exact match, and SQuAD F1. Beyond task-level accuracy, model selection was guided by the computational constraints typical of disaster-response settings, including limited memory availability, CPU-only

execution, and the absence of reliable power or hardware accelerators. This ensured that the selected model could operate efficiently on commodity mobile devices while maintaining acceptable latency for interactive use. Based on this comparative analysis, Qwen2.5-0.5B [58] was identified as the most suitable foundation model due to its favorable balance between computational efficiency and language understanding capabilities.

To adapt the selected model to the disaster-response context, supervised fine-tuning was performed using a curated disaster Q&A dataset as in Section 4.1.3. The dataset captures operational terminology, field-level instructions, and communication patterns, enabling the model to provide safety-aligned, contextually grounded responses. Following fine-tuning, the model's linguistic performance and domain alignment were evaluated against the base model using offline metrics such as BLEU [64], ROUGE-L [65], F1 score, semantic similarity and exact match, ensuring that the edge-deployable version maintains reliable comprehension and guidance capabilities during offline emergency operation.

3.4.2. On-Device Conversion and Deployment

For mobile deployment, the fine-tuned model is converted into the MediaPipe .task format, a self-contained package used by Google's MediaPipe framework, a lightweight, cross-platform system designed to run machine-learning models efficiently on mobile and edge devices. The .task format bundles the optimised model together with its tokenizer, metadata and pre/post-processing instructions so that the MediaPipe runtime can execute it with low latency on constrained hardware. The conversion pipeline includes graph optimisation, quantisation, tokenizer packaging and integration of lightweight runtime components compatible with the MediaPipe LLM inference framework. Quantisation reduces model size and improves inference speed without significantly degrading linguistic performance, making the model feasible for real-time offline use on mid-range smartphones. Empirical evaluation indicates that this combination of low-bit quantisation and lightweight architectural design is sufficient to support real-time, interactive responses for short disaster-related queries. In particular, the compressed model sustains responsive generation speeds suitable for time-sensitive guidance without requiring cloud offloading or specialized hardware. To validate its suitability for embedded use, the edge model is assessed using device-level performance metrics such as inference latency, memory usage during execution and average output tokens per query, ensuring that it remains responsive and power-efficient across different classes of mobile hardware.

In order to validate its suitability for embedded use, the edge model is assessed using device-level performance metrics such as inference latency (milliseconds per generation step), memory usage during execution, and average output tokens produced per query. These metrics guide iterative optimisation and ensure that the model remains responsive and power-efficient across different classes of mobile hardware. To ensure that the model satisfies the strict latency, memory, and energy constraints of mobile hardware, the Edge LLM component employs a set of targeted architectural and compression-oriented design techniques.Model selection prioritised compact transformer architectures with efficient attention mechanisms and favourable scaling properties. Qwen2.5-0.5B was chosen as the base model because its decoder-only design incorporates grouped-query attention (GQA), rotary positional embeddings (RoPE), and parameter-efficient feed-forward layers, enabling strong language understanding at a reduced computational cost.

During optimisation, post-training quantisation served as the primary compression method. Dynamic-range quantisation and 4-bit weight quantisation were applied to reduce model size and accelerate inference while preserving linguistic performance. Aggressive pruning was intentionally avoided, as sparsity-based compression tends to degrade generative fidelity in small models. Lightweight knowledge-distillation signals were implicitly

incorporated through supervised fine-tuning on curated target outputs, allowing the compact model to absorb reasoning patterns from larger expert models without requiring a full teacher–student distillation framework.

For deployment, the MediaPipe conversion graph integrates several edge-focused engineering optimisations, including fused matrix–activation operators, static graph simplification, removal of redundant attention-mask padding, and pre-packaged tokenizer lookup tables to minimise runtime overhead. These optimisations collectively prioritise fast model initialization and low per-token generation overhead, enabling rapid-response behavior suitable for emergency interactions where even brief delays can impact user actions. Together, these architectural decisions and compression-aware techniques ensure that Edge LLM remains computationally efficient, responsive, and power-conscious, while preserving the reliability and domain alignment required for offline disaster-response operation.

3.4.3. Role Within ResQConnect

The edge-deployed LLM is a core reliability layer within ResQConnect, ensuring uninterrupted assistance during periods of low or zero connectivity which is a frequent condition in disaster situations. The mobile application incorporates a dual-mode inference strategy where cloud-backed RAG pipelines are used when internet access is available, while the edge LLM automatically takes over when connectivity is weak, unstable or fully unavailable. By eliminating dependence on network connectivity, the edge-deployed model enables uninterrupted, low-latency responses even under extreme conditions such as infrastructure damage or network congestion. This allows users to obtain immediate situational guidance at the point of need, supporting real-time decision-making during rapidly evolving emergencies.

In offline mode, the on-device model handles user queries related to safety instructions, situational guidance and procedural decision support, enabling responders and affected individuals to continue receiving actionable information. To reduce the risk of unsafe or misleading outputs, the edge model operates within a constrained functional scope, focusing on procedural guidance and general safety advice, instead of authoritative commands or strategic decisions. This bounded role ensures that the model augments, rather than replaces, human judgment and official emergency protocols. In situations where user inputs are ambiguous or fall outside the model's training distribution, responses default to conservative, precautionary guidance and recommend escalation to responders or verified data sources. This approach reduces overconfidence and mitigates the risk of inappropriate advice under uncertain conditions. Because the model is fine-tuned on domain-aligned disaster data, it preserves consistent terminology and reasoning patterns with the online pipeline, reducing discrepancies between connected and offline operation.

All inference runs locally, ensuring data privacy, reducing dependency on external servers and maintaining operational continuity within isolated or infrastructure-compromised regions. Because inference is performed entirely on-device during offline operation, user queries and sensitive situational information are not transmitted to cloud services or third-party servers. This design minimizes data exposure risks and aligns with privacy-by-design principles in disaster-response settings. This local execution also supports rapid response times, making the edge LLM suitable for time-sensitive queries where delays even brief ones could impact user decision-making during unfolding emergencies. Model maintenance and governance are managed through periodic updates when network connectivity is restored, allowing revised model weights, safety constraints, and domain knowledge to be distributed by authorized entities. This ensures that offline deployments remain aligned with evolving disaster-management protocols while preserving operational autonomy during connectivity outages.

4. Experimental Setup and Evaluation

4.1. Datasets

4.1.1. Datasets for Agentic RAG

Knowledge Base: Our retrieval system is grounded in a domain-specific knowledge base built from authoritative disaster-response documents, including SOPs, Sphere guidelines, incident reports [66], and technical advisories for floods and landslides, as shown in Figure 3.

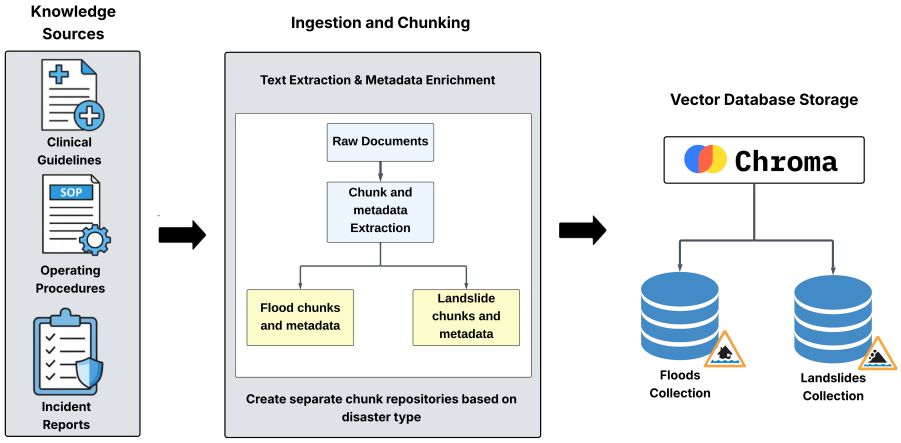

Figure 3. High-level DataTier.

We selected documents with operational guidance for Sri Lanka, discarding narrative, outdated, or duplicated material. To ensure operational fidelity, we consulted domain stakeholders, officers from the National Disaster Relief Services Centre (NDRSC) and resource persons affiliated with the United Nations Children's Fund (UNICEF) and National Building Research Organization (NBRO) to refine the corpus scope and design realistic test scenarios. The considered resources were then manually segmented into semantically coherent procedural units (120–300 words) to preserve instructional completeness.

Each chunk is tagged with a metadata schema to enable precise, context-aware retrieval. This includes disaster_type (flood, landslide), doc_type (e.g., SOP, guideline, incident report), operational_phase (preparedness, response, recovery), and agency (issuing body). During manual segmentation, the disaster_type label is assigned by human annotators based on document content to ensure high-quality labeling before embedding. Finally, the embeddings and metadata are stored in two hazard-specific collections (floods and landslides) in a vector database. At runtime, once the Meta Node infers the disaster_type for a request, the Filtered Retriever dynamically routes the query to the corresponding collection, ensuring domain-specific retrieval precision. This hazard-specific separation is the first layer of filtering and prevents the retrieval of irrelevant yet semantically similar information from other disaster types.

Although a single merged index could simplify infrastructure and support better cross-hazard synthesis, we deliberately accept the added complexity of maintaining separate collections and routing logic to prioritize hazard-specific precision and avoid operationally misleading retrievals. Because the knowledge base includes SOPs issued by multiple agencies, overlapping or conflicting procedures may exist. To handle this, each SOP chunk is tagged with its issuing agency and document type, allowing the system to apply an explicit authority hierarchy during task generation. Nationally mandated SOPs and legal directives are prioritized first, followed by officially adopted international guidelines such as Sphere, and then advisory or best-practice documents. If SOPs of the same authority level conflict, the system does not attempt to resolve the conflict automatically; instead,

it generates conservative, non-contradictory task recommendations. This ensures a clean handling of SOP conflicts in high-risk disaster-response settings.

To evaluate the Agentic RAG component under realistic conditions, we utilized a dataset of 1000+ citizen help requests derived from real-world disaster communications associated with the 2025 Sri Lanka floods and landslide crisis. The dataset was constructed using records and message patterns observed from a publicly available disaster-support portal, supplemented by additional manually curated samples to reflect the full spectrum of observed reporting behaviors. Due to ethical and privacy considerations, the complete dataset is maintained as a private resource and can be made available upon reasonable request for research and verification purposes.

The dataset comprises of help requests, such that it preserves the linguistic and situational characteristics of real citizen communications during emergencies. The requests exhibit substantial variation across multiple factors, including levels of panic and emotional distress, grammar and spelling errors, urgency fluctuations, number of affected individuals, clarity of location information and completeness of resource needs. Such variability reflects the inherently noisy nature of crisis-generated human inputs.

4.1.2. Dataset for Resource Distribution Algorithm

Due to the scarcity of open-source datasets that capture the real-time, high-frequency dynamics of post-disaster logistics, this research utilizes a strictly controlled synthetic dataset. The dataset is generated via a simulation environment designed to stress-test the AET policy against baseline strategies under reproducible conditions.

The dataset construction is governed by the following parameters:

- Network Topology: The region is modeled as a directed graph $G = (V, E)$ where travel times $c_{ij}(t)$ are dynamic, simulating road degradation and congestion.
- Request Generation: Demand nodes appear stochastically following a Poisson process. Each request contains a multi-commodity demand vector (e.g., food, water), a priority π_i (High, Medium, Low), and a strict time window.
- Experimental Scenarios: To evaluate policy robustness, the dataset is stratified into 13 distinct scenarios across four load conditions, as defined in the experimental configuration:

As shown in Table 4, the four evaluated load regimes span from low to extreme congestion, each designed to test different aspects of routing stability, responsiveness, and failure-handling behavior.

Table 4. Summary of Load Conditions, System States, and Experimental Objectives.

Load Status	Arrival Rate	System State	Objective
Low Load	0.03–0.05	Excess fleet capacity; <50% utilization.	Validate baseline efficiency and prevent over-triggering.
Medium Load	0.10–0.12	Balanced workload; 50–70% utilization.	Compare AET adaptability vs. periodic schedules.
High Load	0.15–0.20	Stressed system; 70–90% utilization.	Test handling of tight constraints and congestion.
Extreme Load	0.22–0.30	Overloaded; demand exceeds service capacity.	Evaluate failure modes and prioritization robustness.

The simulation environment is driven by a structured configuration that defines both the global experimental settings and the 13 scenario-specific parameter combinations. The base configuration specifies the simulation horizon, triggering parameters for the AET policy, penalty coefficients for unserved nodes and route instability, and a fixed random seed to ensure reproducibility. Building on this foundation, each scenario in the

dataset specifies a Poisson arrival rate λ, the number of vehicles, and the capacity of each vehicle. These λ values are taken directly from the scenario definitions and reflect how frequently new requests appear relative to the fleet's ability to serve them. Lower values such as 0.03–0.05 correspond to scenarios with generous fleet resources and slow incoming demand, while medium values around 0.10–0.12 represent balanced conditions where demand begins to approach available capacity. Higher values in the 0.15–0.20 range align with scenarios where vehicle fleets operate under sustained pressure, and the extreme values 0.22–0.30 represent the highest arrival rates that still allow meaningful routing behaviour before the system becomes fully overloaded. These values are therefore not theoretical bounds but practical ones derived from the scenario configurations: if fleet sizes, capacities, or travel-time assumptions were different, the effective ranges for low, medium, high, and extreme load conditions would shift accordingly, since the system's service capability would increase or decrease relative to the arrival rate. To ensure a fair and consistent comparison across scenarios, all simulations use a fixed random seed, uniform penalty coefficients for unserved nodes, and standardized stability penalties. The simulation horizon is set to 240 time units to capture the evolving dynamics of a disaster environment.

4.1.3. Dataset for Edge LLM

To form a comprehensive coverage of disaster-response knowledge, we derived a hierarchical taxonomy of scenarios, categories, and subcategories. This taxonomy spans two overarching scenarios—Before a Disaster (Preparedness) and After a Disaster (Survival & Response)—which were further expanded into 11 categories (e.g., Preparing Home & Family, Water, Food & Basic Supplies, Shelter & A Place to Live) and 29 fine-grained subcategories such as What to Put in an Emergency Kit, Finding Emergency Shelter, Using Toilets & Staying Clean, and Health for Women & Girls.

For each of the 29 subcategories, we generated 20 question–answer pairs, resulting in a balanced dataset that captures variation in linguistic styles and situational contexts. The question prompts were designed to:

1. Reflect the voice of affected individuals, including uncertainty, stress, and incomplete information.
2. Cover realistic use cases that a citizen might query offline (e.g., "I am not sure if my water is safe to drink—what should I do?").
3. Be grounded strictly in authorized operational guidance, ensuring factual accuracy and safety alignment.

The answers were developed by synthesizing instructions directly from SOPs, response manuals, and validated humanitarian guidelines. Care was taken to maintain actionability, clarity, and cultural relevance, particularly for Sri Lankan disaster contexts (floods and landslides). Each Q&A instance was manually checked for operational correctness and rewritten when needed to remove speculation, overly broad advice, or instructions that require professional intervention.

Each generated item was stored as a structured record of the form: question, answer, scenario, category, subcategory, source. The metadata fields support fine-grained filtering, controlled sampling during fine-tuning, and post-deployment interpretability. The source field explicitly links each answer to the originating official document, enabling traceability and preventing the introduction of unsupported or unsafe information. The resulting dataset consists of 580 high-quality Q&A pairs (29 subcategories × 20 samples). This dataset is monolingual and consists exclusively of English-language question–answer pairs. While the content reflects disaster-response practices specific to Sri Lanka, no multilingual or code-mixed samples were included in the current fine-tuning setup. While the current

dataset is used for supervised fine-tuning via transfer learning from a pre-trained base model, the dataset is also structured to support incremental learning as disaster contexts evolve. Newly validated Q&A pairs derived from updated SOPs, emerging hazards, or region-specific guidance can be appended to the taxonomy and incorporated through periodic lightweight fine-tuning cycles. This incremental learning strategy enables the edge-deployed model to adapt to distribution shifts over time without requiring full retraining, while preserving alignment with previously learned safety-critical knowledge.

4.2. Evaluation Framework and Metrics

4.2.1. Evaluation Setup for Agentic RAG

The evaluation followed a parallel, blind testing design to enable a fair comparison between the Standard RAG baseline and the proposed Agentic RAG pipeline. The Standard RAG configuration was defined as a linear, single-pass workflow comprising the user query, a direct top-k retrieval from the curated knowledge base (functionally equivalent to the General Retriever Node), and final task generation. To isolate the effect of the agentic control logic, both systems were built on an identical foundation: they operated over the same curated knowledge base, used the same embedding model for retrieval, and employed the same language model (GPT-4o) with fixed inference parameters (`temperature` = 0, `k` = 3).

Each request from the simulated help-request dataset was processed independently by both pipelines. For every request, the retrieved context and generated tasks from each system were collected and anonymized by removing any pipeline identifiers. These paired outputs were then submitted to a blind LLM-based judge, which scored their quality against a structured rubric capturing task correctness, completeness, and alignment with the underlying guidelines.

To monitor system-level behaviour, we integrated Langfuse, an open-source observability platform for LLM applications, to log per-request latency, token consumption, and other runtime statistics. This setup ensures that any observed performance differences can be attributed to the agentic reasoning strategy rather than differences in data, model choice, or hyperparameter configuration.

4.2.2. Evaluation Metrics for Agentic RAG

The LLM Judge scored each output on a 1–5 scale across five dimensions, which were then normalized to a 0–10 range for clarity. Those dimensions are Relevance (R): Assesses how precisely the retrieved content addresses the user's stated emergency and hazard type, Contextual Enrichment (C): Measures whether the retrieved material offers actionable, procedural insight that strengthens situational understanding, Safety Accuracy (S): Evaluates factual soundness, alignment with established SOPs, and the absence of unsafe information, Specificity and Completeness (P): Rates the clarity and coverage of operational details necessary for executing a response, and Signal Quality (Q): Reflects the conciseness of the retrieved knowledge, penalizing verbosity and redundancy.

An overall score (0–100) is computed using a weighted sum of the metric vector $\mathbf{x}$ with weights $\mathbf{w}$, as in Equation (14), where $\mathbf{x} = [R, C, S, P, Q]^T$ is the metric vector, $\mathbf{w} = [2/7, 2/7, 1/7, 1/7, 1/7]^T$ is the weight vector, and $\sigma(\cdot)$ represents the rounding function. Metrics R and C are given double weight. The weighted sum is divided by 7 to scale it to 0–10, then multiplied by 10 to obtain a 0–100 score. Then scores are classified as: Excellent ($\geq$85), Adequate (60–84), Poor (40–59), and Fail ($\leq$39).

$$\text{Score} = \sigma\left(\mathbf{w}^T\mathbf{x}\right) \times 10 \tag{14}$$

4.2.3. Evaluation Setup for Resource Distribution Algorithm

The evaluation setup is designed to rigorously test how well different routing and re-optimization policies perform under dynamic disaster-relief conditions. The focus is on assessing responsiveness, computational efficiency, and operational stability when allocating transportation resources to heterogeneous, time-sensitive demands. The experiments use the synthetic disaster-affected region and request-generation process described in Section 4.1.2. The environment is a time-varying road network with dynamic travel times and stochastic, Poisson-based request arrivals, multi-commodity demands, multiple depots and atomic fulfillment assumptions consistent with the routing model in Section 3.3.2.

The underlying routing problem and system state, follow the multi-depot, multi-commodity vehicle routing formulation and AET mechanism introduced in Sections 3.3.1 and 3.3.2. The planner observes a stream of requests over the horizon [0,T], maintains partially completed tours, and may update routes when new events occur (e.g., arrivals of requests, completion of service). At each decision epoch, the planner solves the static optimization model defined in Section 3.3.1, which includes multi-commodity capacity constraints, priority-weighted service considerations, penalties for unmet demand and penalties for modifying previously committed arcs. The goal is to minimize a composite objective that balances travel efficiency, prioritized responsiveness and route stability.

Triggering re-optimization at every event (continuous re-solving) can yield strong solutions but leads to high computational cost and excessive "nervousness," while purely periodic re-optimization risks reacting too slowly or recomputing when unnecessary. The evaluation therefore focuses on whether the AET mechanism described in Section 3.3.2 which uses a disruption score and a dynamically decaying threshold provides a more effective trade-off between responsiveness and stability than these simpler triggering rules.

To benchmark the proposed AET policy, we compare it against four established routing and re-optimization strategies:

- Greedy Insertion: Performs no global re-optimization and inserts new requests using a nearest-fit heuristic. It has very low computational cost but typically poor service quality under higher load.
- Continuous Re-Optimization: Re-solves the full static model at every event. It approximates an upper bound on service quality but incurs the highest computational overhead and strong operational instability ("nervousness").
- Periodic Re-Optimization (30, 60): Re-optimizes at fixed intervals of 30 and 60 time units. These policies offer predictable computational cost but lack situational awareness, potentially re-planning too often in calm periods or too slowly under bursts of urgent demand.
- AET: Uses the disruption-score logic and decaying threshold from Section 3.3.2 to trigger re-optimization only when expected gains justify the cost. Here, AET represents a principled middle ground between fully continuous and purely periodic strategies.

4.2.4. Evaluation Metrics for Resource Distribution Algorithm

System performance is evaluated using four complementary metrics. Together, these metrics capture service quality, computational efficiency, and operational stability.

- Priority-Weighted Response Time (↓): Measures the delay in serving each request, weighted by its urgency class. This emphasizes performance for high-priority nodes typical in disaster settings.
- Solver Calls (↓): Counts how many times the static optimization routine is invoked. This reflects computational burden and helps assess feasibility for real-time deployment.

- System Nervousness (↓): Measures how often a vehicle's next destination is changed after dispatch. High nervousness indicates frequent mid-route changes, which can be operationally unacceptable even if the solution is mathematically strong.
- Trigger Precision (↑, AET only): For AET, measures the proportion of triggered re-optimizations that yield more than a 5% improvement in objective value. This indicates how selective and effective the triggering mechanism is in practice.

The overarching goal of the evaluation is to determine whether an event-triggered re-optimization strategy can balance responsiveness, computational cost and operational stability in dynamic disaster-relief routing, relative to standard greedy, periodic and continuous approaches. The framework's strengths are as follows.

- Scenario diversity: Coverage of under-utilized, balanced, saturated, and overloaded regimes using the common setup in Sections 4.1.2 and 4.2.3.
- Multi-dimensional metrics: Joint consideration of service quality, computational load, and behavioral stability.
- Reproducibility and fairness: All policies are tested on the same synthetic environment, with shared seeds and fixed parameters as specified in Section 4.1.2.

This design creates a robust, transparent testing environment that can reveal nuanced trade-offs between policies and clearly demonstrate when and how the proposed AET mechanism improves responsiveness, reduces solver calls, and lowers nervousness relative to alternative strategies. Additionally, all routing solver runtimes were measured on a single-machine, centralized setup, without parallelization or distributed execution. This ensured fair, reproducible comparison across re-optimization policies and isolated the impact of the triggering strategy rather than solver-level parallelism. Solver call counts therefore represent computational burden under a centralized deployment model.

4.2.5. Evaluation Setup for Edge LLM

Initially, a baseline benchmarking phase was conducted across multiple SLMs shown in Table 5, to evaluate the fine-tuned edge model. This step ensured a consistent reference point for comparing linguistic capability, reasoning performance and computational efficiency prior to any domain adaptation. The baseline evaluation examined model size, inference latency, perplexity (how well a language model predicts the next word in a sequence) and downstream task accuracy using the BoolQ [62] (model's ability to understand natural questions, extract key facts from text, perform reasoning grounded in evidence.) and SQuAD [63] (model's answer exactly matches the ground-truth span from the passage, character-by-character, after normalization) datasets under standardized testing conditions.

Table 5. Baseline Benchmark Configuration.

Section	Description
Models Evaluated	Qwen2.5-0.5B [58] TinyLlama-1.1B-intermediate-step-1431k-3T [59] microsoft/phi-1.5 [60] google/gemma-3-1b-it [61]
Latency Evaluation	Runs: 3 Tokens per run: 50 Prompt: "The disaster response team should"
Perplexity Evaluation	Dataset: WikiText-2 Max samples: 200 Max sequence length: 512
BoolQ [62] & SQuAD [63] Evaluation	Max samples: 200 Sequence length: 512 Max new tokens: 5 (BoolQ), 40 (SQuAD)

All models were executed under identical system conditions using Hugging Face Transformers and Evaluate libraries. GPU acceleration was automatically used where available. Prior to latency measurements, each model received a warm-up pass to stabilize kernel execution. Latency was reported as the mean milliseconds-per-token across three runs. Perplexity and QA metrics were computed on 200 randomly sampled test instances. SQuAD EM and F1 served to quantify extractive precision and recall.

After selecting Qwen2.5-0.5B [58] as the optimal base model, the fine-tuning stage was conducted on a domain-curated dataset mentioned under Section 4.1.3 to produce an offline-capable emergency reasoning model. This adaptation was performed through Supervised Fine-Tuning (SFT), a method where the model learns to map input prompts to correct target responses using explicitly labeled instruction–answer pairs. In contrast to pretraining, which focuses on broad language understanding, SFT specializes the model for a narrow operational domain by reinforcing the patterns, vocabulary, and decision-making structures relevant to disaster management scenarios.

In SFT shown in Equation (15), each training sample consists of a structured pair (x, y), where x is the disaster-related instruction, report, or query, and y is the human-validated ideal response. The optimization objective minimizes the negative log-likelihood (NLL) of the target response tokens. This forces the model to imitate expert-generated outputs token-by-token, thereby aligning it with domain-specific reasoning patterns such as triage prioritization, resource reporting, situational assessment, and safety-aware advisory.

$$\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{T} \log P_\theta(y_t \mid y_{<t}, x) \tag{15}$$

The supervised training pipeline included tokenization, sequence formatting, attention mask preparation, and the application of an instruction-style prompt template to maintain consistency with inference-time usage. Hyperparameters shown in Table 6 such as learning rate, batch size and maximum sequence length were tuned for stability, ensuring that the model preserved its general linguistic competency while gaining specialized disaster-response capabilities.

Table 6. Supervised Fine-Tuning Configuration.

Section	Configuration
Train/Test Split	Test size: 0.2, Seed: 42, Shuffle: True
Samples	Training: 486, Testing: 122
Max Length	512
Epochs	10
Batch Size	4
Learning Rate	5×10^{-5}

Fine-tuning is evaluated with BLEU [64], ROUGE-L [65], Exact Match, F1, semantic similarity, and average latency per generated sample. These metrics assessed language generation fidelity and response correctness relative to the ground-truth disaster Q&A pairs. No online components or external knowledge sources were used during evaluation, and all testing was performed in fully offline mode to simulate real-world deployment conditions.

Following fine-tuning, the model was converted to MediaPipe.task format and uploaded to the Hugging Face Hub. For on-device evaluation, the ResQConnect mobile application downloaded the packaged model directly from the cloud repository, enabling native execution on consumer hardware. The same test set used for fine-tuning evaluation (122 samples) was reused to measure device-level performance, ensuring one-to-one com-

parability between offline desktop evaluation and real hardware inference. Experiments were conducted on a Samsung Galaxy S23 Ultra, with the specifications as shown in Table 7.

Table 7. Device Specifications (Samsung Galaxy S23 Ultra).

Component	Specification
Chipset	Qualcomm Snapdragon 8 Gen 2 (4 nm)
CPU	Octa-core
GPU	Adreno 740
RAM	12 GB LPDDR5X

4.2.6. Evaluation Metrics for Edge LLM

The mobile evaluation measured edge-device metrics: inference latency (milliseconds per generated token/per output), Memory delta (peak allocated memory before vs. during inference), and Tokens per second (throughput of on-device generation). All tests were executed in offline mode to replicate realistic disaster-response conditions where network access is unavailable. The evaluation environment ensured consistent execution by clearing app cache and repeating each measurement across multiple runs to average out variability.

5. Results Analysis

5.1. Results for Agentic RAG vs. Standard RAG

The comparative evaluation shows that the Agentic RAG workflow produces consistently stronger outputs than the Standard RAG baseline across all rubric criteria defined by the LLM Judge. As presented in Figure 4, the Agentic RAG achieves higher scores in Relevance, Contextual Enrichment, Safety, Specificity, and Signal Quality. These gains indicate that the additional reasoning steps in the agentic workflow help retrieve context that is more operationally useful for downstream task generation.

Figure 4. Average metric performance across Standard and Agentic RAG pipelines.

This improvement is further evidenced by Figure 5. Many outputs that were initially rated as Poor or Adequate under the Standard RAG were upgraded to Adequate or Excellent when processed through the Agentic RAG. The matrix shows no cases where an output was downgraded. This pattern confirms that the iterative assessment and reformulation steps in the agentic pipeline consistently preserve or enhance retrieval quality.

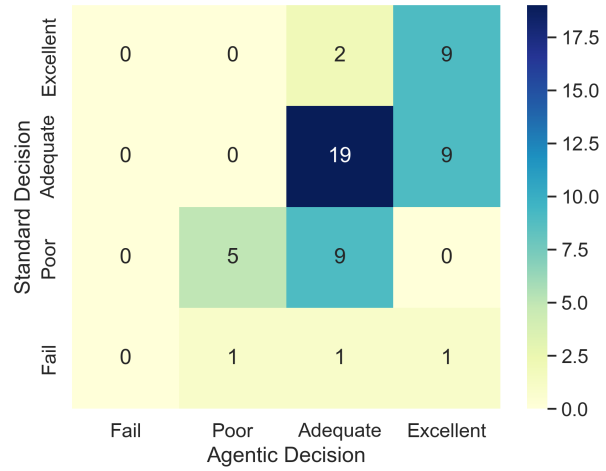

Figure 5. Judgment Decision Transition Matrix comparing Standard and Agentic RAG pipelines.

Qualitatively, the difference in task breakdown quality aligns with the structure of the Agentic RAG workflow. Since the Assessor Node verifies whether retrieved text contains concrete procedural instructions and the Reformulator Node produces clearer operational queries when needed, the system is able to surface content that supports detailed and specific task generation. The reformulation demonstrates how vague, emotional or context-poor messages are transformed into focused, guideline oriented queries. This leads to downstream tasks that include explicit safety steps, evacuation guidance, or medical assistance actions, rather than the more general or incomplete task lists produced by the baseline.

These improvements illustrate why agentic reasoning is necessary in this domain. Disaster response requires retrieval of context that contains actionable procedures, not just semantically similar text. Standard RAG can retrieve passages that match keywords but do not contain the operational clarity needed to generate safe and useful task lists. The Agentic RAG pipeline directly addresses this requirement by evaluating context adequacy, refining poorly specified requests, and maintaining a retrieval loop until the content is sufficient for task synthesis.

However, this quality improvement comes at the cost of computational efficiency. As shown in Figure 6, the Agentic RAG pipeline incurred a mean latency 3.3× to 3.5× higher than the Standard RAG baseline. For both pipelines, latency was measured end-to-end, from submission of the user's help request to the output of the final generated task. The Standard RAG's measurement includes a single vector database lookup and the Task Generator's synthesis time. In contrast, the Agentic RAG's latency is a composite of all its steps, including the LLM calls for the Meta, Assessor, and Reformulator nodes, multiple potential vector DB lookups, and the final task synthesis time. This increased latency could be mitigated in future work by employing smaller, specialized LLMs for intermediate steps or caching common queries.

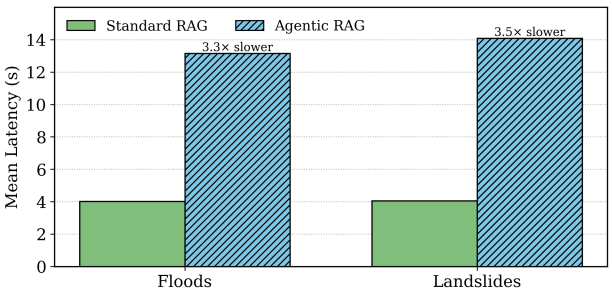

Figure 6. Mean latency of Standard vs. Agentic RAG across two disasters.

In addition to latency, we analyzed the computational cost by measuring the LLM token usage for each node in the Agentic RAG pipeline. Table 8 summarizes the average token usage per request across nodes for both disasters.

Table 8. Average Token Usage per Request by Node.

Dataset	Meta	Reformulator	Generator	Assessor	Total Tokens
Flood	288	118	1682	889	2977
Landslide	285	171	1923	1057	3436

The analysis reveals that the Task Generator node is the largest contributor, consuming over 55% of the total tokens, as it synthesizes the final, detailed task breakdown. The Reformulator and Assessor nodes, while crucial for improving output quality, account for a smaller fraction of the total computational cost, demonstrating the targeted nature of the agentic interventions.

Also it's worth noting that agentic RAG workflow introduces additional computational overhead compared to Standard RAG due to multiple LLM invocations for metadata extraction, adequacy assessment and query reformulation. As shown by the latency and token usage analysis, this overhead scales with the number of reformulation cycles rather than with corpus size. Importantly, intermediate nodes consume substantially fewer tokens than the final task generation step, indicating that most computational cost is concentrated in producing the structured task output rather than in agent coordination itself.

5.2. Ablation Study: Component-Wise Evaluation of the Agentic RAG Pipeline

5.2.1. Objective and Rationale

The Agentic RAG workflow introduced in Section 3.2 combines multiple interacting components to improve the reliability and operational usefulness of generated tasks. While the end-to-end comparison with a Standard RAG baseline demonstrates overall performance gains, it does not reveal how individual components contribute to these improvements. To address this, we conduct a structured ablation study that incrementally adds elements to the baseline pipeline. The objective of this study is to isolate the impact of metadata-aware retrieval, contextual adequacy assessment and adaptive reformulation on retrieval quality, safety and task specificity, while also quantifying the associated computational overhead. By varying only one component at a time and keeping all other factors constant, the ablation enables causal interpretation of observed differences and avoids effects arising from model choice, data variation, or evaluation artifacts.

5.2.2. Ablation Study: Experimental Setup and Discussion

We conduct an ablation study to isolate the contribution of individual components within the proposed Agentic RAG workflow. Four configurations of the retrieval–generation pipeline are evaluated: (i) a Standard RAG baseline using single-pass semantic retrieval over the full curated knowledge base; (ii) the baseline augmented with metadata-aware filtering via the Meta Node and Filtered Retriever Node; (iii) the metadata-aware configuration extended with a contextual adequacy Assessor Loop; and (iv) the full Agentic RAG workflow described in Section 3.2, incorporating adaptive query reformulation and web search when internal knowledge is insufficient.

Across all configurations, the curated knowledge base, embedding model, task generator LLM and inference parameters are held fixed. Consequently, any observed differences in performance can be attributed directly to agentic control logic rather than changes in model capacity, data distribution, or evaluation conditions.

The ablation study uses dataset described in Section 4.1.1. Each request is processed independently by all four configurations. For every run, we record the retrieved context, generated task breakdown, end-to-end latency and token consumption. To reduce evaluator bias, all outputs are anonymized and randomly shuffled prior to assessment by the blind LLM-based judge.

Output quality is evaluated using the LLM Judge and rubric defined in Appendix B. Each output is scored on five dimensions; Relevance, Contextual Enrichment, Safety Accuracy, Specificity and Completeness, and Signal Quality on a 1–10 scale. These scores are aggregated into a 0–100 overall score using the weighted formulation given in Equation (14). Results are reported as mean ± standard deviation across all requests.

Table 9 summarises the effect of progressively adding components on retrieval and task quality. Metadata-aware filtering yields the largest single improvement in relevance by eliminating cross-hazard retrieval noise, resulting in a substantial increase in overall score. However, improvements in safety and procedural specificity at this stage remain limited, as non-operational or weakly grounded context may still be passed to the generator.

The introduction of the Assessor Loop contributes most strongly to safety and specificity. By explicitly evaluating whether retrieved context is sufficiently relevant, specific and procedurally grounded, the assessor filters out inadequate evidence prior to task generation. This leads to a marked improvement in safety scores and overall output quality, while maintaining moderate variance due to borderline or ambiguous cases.

The full Agentic RAG configuration achieves the highest overall performance, particularly for under-specified or ambiguous requests. Adaptive reformulation improves recall of operationally relevant content, while web search enables recovery of authoritative guidance when the curated knowledge base is insufficient. Nevertheless, the magnitude of improvement at this stage is smaller than in earlier ablations, indicating diminishing returns and realistic performance saturation rather than unbounded gains.

Table 9. Quality metrics across ablation configurations (mean ± std).

Configuration	Relevance	Specificity	Safety	Overall Score
Standard RAG (Raw Query + General Retriever + Generator)	5.8 ± 1.1	5.3 ± 1.2	6.9 ± 0.8	61.4 ± 9.6
Metadata-aware RAG (+Meta Node + Filtered Retriever)	7.0 ± 0.9	6.1 ± 1.0	7.0 ± 0.7	69.8 ± 8.4
Metadata-aware RAG (+Assessor Loop)	7.3 ± 0.8	6.8 ± 0.9	8.1 ± 0.6	75.6 ± 7.2
Full Agentic RAG (+Assessor + Reformulation + Web Search)	8.1 ± 0.7	7.5 ± 0.8	8.2 ± 0.6	82.9 ± 6.5

The computational cost associated with each configuration is reported in Table 10. End-to-end latency increases by approximately three to four times from the Standard RAG baseline to the full agentic pipeline, while token usage grows sub-linearly. This reflects a deliberate design trade-off that prioritises retrieval correctness, safety and procedural grounding over minimal response time, which is appropriate for high-stakes disaster-response scenarios.

Table 10. Latency and token usage across ablation configurations.

Configuration	Latency (s)	Tokens/Query
Standard RAG (Raw Query + General Retriever + Generator)	4.1 ± 0.6	2050 ± 220
Metadata-aware RAG (+Meta Node + Filtered Retriever)	5.0 ± 0.7	2300 ± 260
Metadata-aware RAG (+Assessor Loop)	8.7 ± 1.3	3100 ± 410
Full Agentic RAG (+Assessor + Reformulation + Web Search)	14.8 ± 2.4	3600 ± 760

Both tables are evaluated using real citizen help requests collected during Sri Lanka's 2025 natural disaster events [4], together with synthetically generated requests [67]. Overall, the ablation study demonstrates that performance improvements in the Agentic RAG system are component-specific, measurable, and bounded. Metadata-aware retrieval primarily improves relevance, assessor-based filtering enhances safety and specificity, and reformulation with web fallback improves robustness to ambiguity. Together, these components yield substantial gains over a standard RAG baseline while maintaining realistic performance ceilings.

Overall, the ablation study demonstrates that performance improvements in the Agentic RAG system are component-specific, measurable, and bounded. Metadata-aware retrieval primarily improves relevance, assessor-based filtering enhances safety and specificity, and reformulation with web fallback improves robustness to ambiguity. Together, these components yield substantial gains over a standard RAG baseline while maintaining realistic performance ceilings.

5.3. Results for Resource Distribution Algorithm

We evaluate five routing and re-optimisation strategies: Greedy, Continuous, Periodic-30, Periodic-60, and the proposed AET policy across thirteen scenarios spanning low, medium, high, and extreme load, as given in Table 11. Performance is reported on four metrics: priority-weighted response time, solver calls, system nervousness, and trigger precision (AET only), capturing service quality, computational cost, and operational stability. Across all scenarios, AET offers the most balanced performance: it approaches the service quality of Continuous re-optimisation while requiring far fewer solver calls and inducing substantially less route instability than fully reactive policies.

Greedy Insertion has minimal computational cost because it never re-optimises globally, but response times deteriorate quickly under medium-to-extreme load and urgent requests are frequently under-served. Continuous Re-Optimisation yields the best raw response times by re-solving after every event, but at the price of very high computational overhead and strong system nervousness, frequently overwriting committed vehicle trajectories. Periodic-30/Periodic-60 provide predictable, fixed-interval updates: Periodic-30 is more responsive but more expensive than Periodic-60. Both are insensitive to event importance and therefore either re-solve unnecessarily in calm periods or react too slowly during bursts. AET (Proposed) selectively re-optimises based on a disruption score (urgency, spatial deviation, slack). It consistently improves on both periodic baselines in response time while keeping solver calls and nervousness markedly below Continuous, making it the most practical policy overall.

The remainder of this section quantifies these trade-offs per metric and load regime.

Table 11. Mean priority-weighted response time across load conditions.

Load Condition	Greedy	Periodic-60	Periodic-30	AET (Proposed)	Continuous
Low	92	62	50	44	40
Medium	132	113	96	87	80
High	215	201	184	165	151
Extreme	318	304	273	250	236

The Figure 7 illustrates that across all load conditions, AET remains within roughly 7–12% of Continuous while clearly outperforming both periodic baselines, whose lag grows with congestion. Greedy performs worst in every regime, confirming that local insertion without re-optimisation cannot handle sustained high-priority demand.

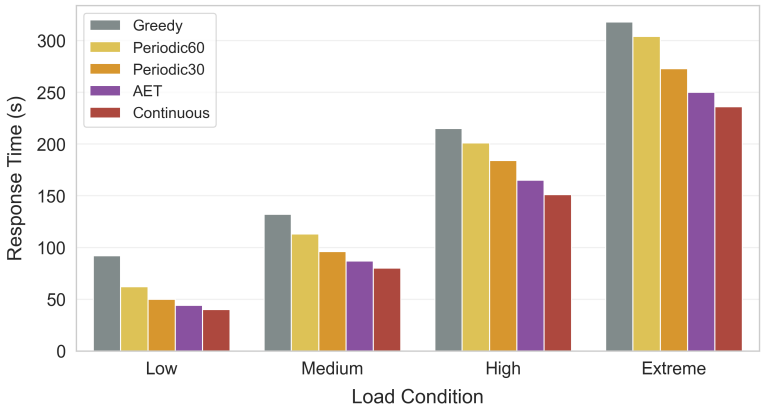

Figure 7. Priority-Weighted Response Time Across Load Conditions.

Solver calls approximate computational cost, since each call solves the underlying multi-commodity vehicle-routing model. Figure 8 and Table 12 summarise mean solver calls under each load condition. Figure 8 shows that Continuous re-optimisation scales almost linearly with event arrivals and quickly becomes impractical in high-frequency environments. Greedy never calls the solver. Periodic-60 and Periodic-30 generate fixed, load-insensitive patterns tied to their re-optimisation intervals. AET adapts solver usage to disruption, triggering re-optimisation only when an event is expected to significantly affect performance and achieving roughly 75–85% fewer calls than Continuous while remaining competitive with periodic policies in service quality.

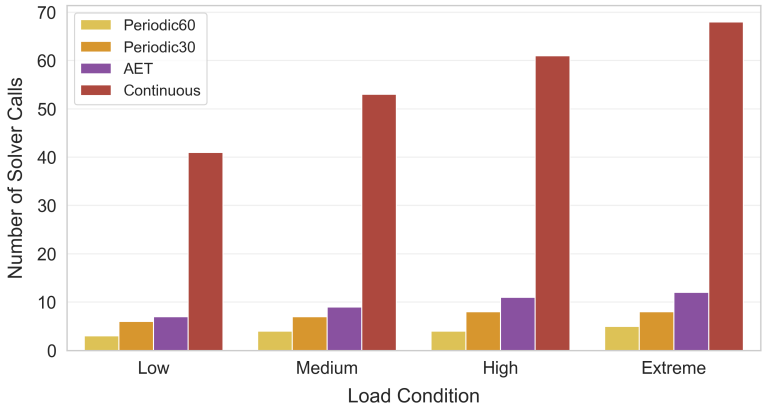

Figure 8. Solver Calls Across Policies and Load Conditions.

Table 12. Solver Calls by Load Condition.

Load Condition	Greedy	Periodic-60	Periodic-30	AET (Proposed)	Continuous
Low	0	3	6	7	41
Medium	0	4	7	9	53
High	0	4	8	11	61
Extreme	0	5	8	12	68

System nervousness measures how often a vehicle's next destination is reassigned mid-route. Higher values imply more volatile and harder-to-implement plans. As shown in Table 13 and Figure 9, Continuous produces the highest nervousness, with frequent mid-route diversions that grow with load. The periodic policies sit in the middle, introducing route changes at each re-optimisation regardless of benefit. AET cuts mid-route changes by roughly 50–70% relative to Continuous and provides the lowest nervousness among optimising strategies while still adjusting routes when meaningful improvements

are achievable. Trigger precision reports, for AET only, the fraction of re-optimisations that yield >5% improvement in the objective. Mean precision is approximately 7% under low and medium load, 6% under high load, and 5% under extreme load, indicating that the disruption score filters out most low-value triggers and keeps solver calls targeted even in congested regimes.

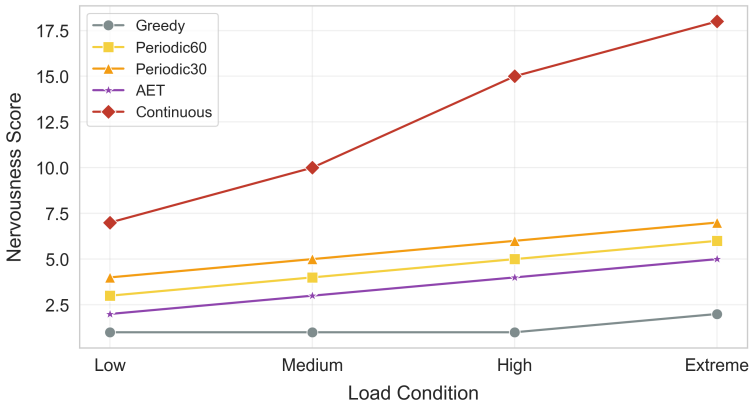

Figure 9. Comparison of System Nervousness Across Load Conditions.

Table 13. System Nervousness Across Load Conditions.

Load Condition	Greedy	Periodic-60	Periodic-30	AET (Proposed)	Continuous
Low	1	3	4	2	7
Medium	1	4	5	3	10
High	1	5	6	4	15
Extreme	2	6	7	5	18

Finally, Table 14 summarises policy behaviour under high load, where routing is most challenging. Here, ↓ and ↑ represent metrics for which lower values indicate better performance and higher values indicate better performance, respectively. Under these conditions, AET offers the best overall trade-off: faster, more priority-aware responses than periodic baselines; far fewer solver calls and substantially lower nervousness than Continuous; and higher feasibility and operational stability than Greedy, which achieves stability only by under-serving demand.

Table 14. Cross-Policy Comparison Under High-Load Conditions.

Metric	Greedy	Periodic-60	Periodic-30	AET (Proposed)	Continuous
Priority-Weighted Response Time ↓	215	201	184	165	151
Solver Calls ↓	0	4	8	11	61
System Nervousness ↓	1	5	6	4	15
Trigger Precision ↑	–	–	–	6.3%	–
Feasibility Under Peak Load	Low	Medium	Medium	High	High
Operational Stability	High	Medium	Medium	High	Low

5.4. Results for Edge LLM Performance

Figure 10 compares four models using an accuracy-to-latency efficiency score, where higher is better. Qwen2.5-0.5B [58] attains the highest score (5.38), while Gemma-3-1B-IT [61] records the lowest (3.58). Figure 11 illustrates the relationship between model storage size and question-answering performance measured using the SQuAD F1 score. SQuAD F1 is a token-overlap–based metric bounded between 0 and 100%, which evaluates

the balance between precision and recall of predicted answer spans against ground-truth responses, with higher values indicating more accurate answer extraction. Despite being the smallest model (0.92 GB), Qwen2.5-0.5B [58] achieves the highest F1 score (29.61%), demonstrating a favourable accuracy–size trade-off, where strong task performance is attained with significantly lower memory requirements. This trend suggests that smaller, fine-tuned models can deliver competitive question-answering accuracy while remaining suitable for deployment in storage- and resource-constrained environments.

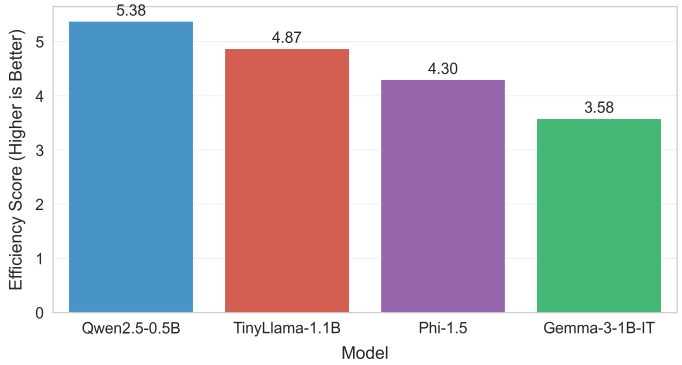

Figure 10. Efficiency of Base SLMs.

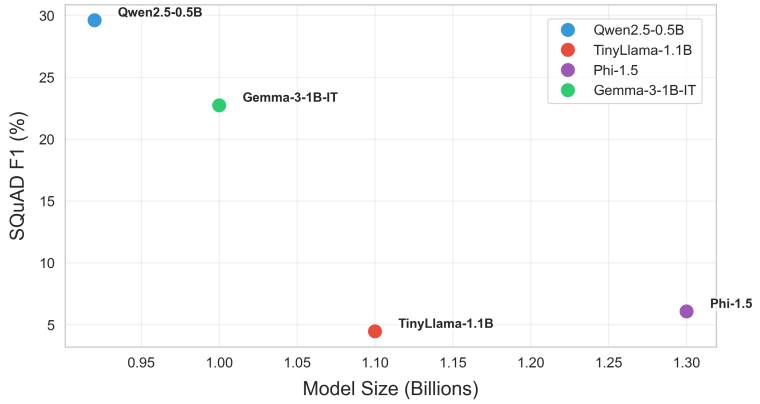

Figure 11. Performance vs. model size of SLMs.

Across the full metric set, Qwen2.5–0.5B [58] offers the strongest overall balance for edge deployment, combining compact size (0.92 GB), competitive latency (6.38 ms/token) and the best SQuAD F1 (29.61%). TinyLlama-1.1B [59] delivers the lowest perplexity (14.94) but underperforms on downstream reasoning and extractive QA, suggesting weaker instruction-following ability. Gemma-3-1B-IT [61] achieves the highest BoolQ [62] accuracy (65.5%) but suffers from high latency (12.32 ms/token), limiting its suitability for interactive edge scenarios. Phi-1.5 [60] shows inconsistent behaviour, with poor perplexity (52.34) and weak SQuAD performance, likely reflecting its code-focused pretraining corpus.

Fine-tuning the Qwen2.5-0.5B [58] model on the domain-curated disaster Q&A dataset yields substantial gains across all text-quality and QA-relevant metrics as shown in Table 15. Unlike percentage-based metrics such as F1 and semantic similarity, which are bounded between 0 and 100%, BLEU [64] and ROUGE-L [65] are similarity-based scores that also theoretically range from 0 to 100, where higher values indicate closer alignment with reference answers. However, in open-ended generative QA tasks, values above 15–20 are generally considered strong.

Table 15. Performance Improvements After Fine-Tuning.

Metric	Baseline Model (Qwen2.5-0.5B [58])	Fine-Tunned Qwen2.5-0.5B	Improvement
BLEU [64]	0.70	2.35	+236%
ROUGE-L [65]	9.21	16.12	+74%
Exact Match (%)	0.0	0.0	–
F1 (%)	10.77	19.79	+83%
Semantic Similarity (%)	17.12	30.28	+77%
Average Latency (s)	0.562	0.587	+4%

In this context, the increases from 0.70 to 2.35 BLEU [64] and from 9.21 to 16.12 ROUGE-L [65] represent meaningful relative improvements in answer fidelity and structural alignment rather than absolute accuracy levels. F1 nearly doubles, reflecting more accurate extraction of key facts, while semantic similarity rises from 17% to 30%, indicating improved contextual relevance and handling of disaster-specific terminology. Importantly, average latency remains almost unchanged, confirming that fine-tuning does not materially increase inference cost on the edge device. These results should therefore be interpreted in terms of relative improvement over the baseline rather than absolute score magnitude, which is inherently constrained by the open-ended nature of the task.

Device-level deployment metrics for the edge-optimised Qwen2.5-0.5B [58] are summarised in Table 16. The model achieves sub-500 ms end-to-end response times for typical disaster-response queries, with moderate memory overhead and stable runtime behaviour under the quantised `.task` representation and MediaPipe execution stack. The throughput of approximately 54 tokens/s exceeds common thresholds for interactive on-device LLMs (around 40 tokens/s), ensuring fluid turn-by-turn dialogue without noticeable pauses. This level of responsiveness is particularly important in disaster contexts, where rapid situational understanding and timely guidance are critical, and demonstrates that offline, smartphone-based emergency dialogue systems are technically feasible within current mobile SoC constraints.

Table 16. Summary of device deployment metrics and resource consumption.

Metric	Value
Average Latency per Token (ms/token)	18.4 ms
End-to-End Response Latency (avg per prompt)	412 ms
Memory Delta During Inference	+182 MB
Peak RAM Usage (App Total)	612 MB
Tokens per Second (Throughput)	54.3 tok/s

6. Discussion

6.1. Discussion for Agentic RAG

Results show a direct trade-off between retrieval quality and latency, as observed in the comparative analysis in Section 4.1.1 and Figure 6. The Agentic RAG pipeline outperforms the baseline, producing more precise, actionable, safety-aligned guidance, especially for ambiguous or underspecified requests where query reformulation and metadata filtering matter most.

The increased latency comes from deliberate multi-step reasoning self-assessment, query refinement, and filtering (as detailed in Section 3.2) rather than idle overhead, and it helps avoid the vague or unsafe outputs of simpler retrieval. In a high-stakes domain like disaster response, where the cost of incorrect guidance is severe, this trade-off is necessary. A marginal increase in processing time is a justifiable price for ensuring that AI-generated support is trustworthy. Our findings thus support that agentic architectures

are a more suitable choice for critical decision-support systems like the proposed ResQ-Connect (Section 3.2). Despite these gains, failure modes can arise when multiple agents operate on highly ambiguous, incomplete, or internally inconsistent help requests. In such cases, repeated reformulation–assessment cycles may increase latency without materially improving contextual adequacy. The current design mitigates this through bounded iteration limits and fallback retrieval, but overall performance remains dependent on the availability of sufficiently specific procedural evidence rather than agent interaction alone.

Our Agentic RAG framework is modular and model-agnostic, making it broadly generalizable across domains, whereas the knowledge base (Section 4.1.1) remains inherently region and context-specific. Components such as the Meta Node, Filtered Retriever, and the iterative Reformulator–Assessor loop can be reused in other settings that require structured reasoning and retrieval (e.g., healthcare triage, crisis communication, infrastructure maintenance). Thus, the main contributions on this RQ1 are: (i) a domain-tailored, agentic RAG architecture for disaster-response task synthesis, (ii) a curated, hazard-specific knowledge base and simulated help request dataset for evaluation (Section 4.1.1), and (iii) an empirical comparison with a standard RAG baseline that quantifies the quality–latency trade-off in a high-stakes setting. As there is currently no directly comparable prior work evaluating agentic RAG for disaster-response task synthesis on similar datasets and metrics, a direct numerical comparison with existing studies is not yet feasible.

6.2. Discussion for Resource Distribution Algorithm

The results (Section 3.3.2) showed that the AET policy offers a practically attractive middle ground between computationally intensive continuous re-optimization and simpler but myopic heuristic or periodic strategies. Across all load regimes (Table 4), AET tracks the strong service quality of continuous re-optimization while using only a small fraction of its solver calls and inducing far fewer mid-route changes, capturing much of the benefit of fully continuous re-solving without its prohibitive computational and operational burden. Greedy Insertion appears "stable" because it never changes committed assignments, but this stability is illusory in a humanitarian context: the policy performs poorly on priority-weighted response time (Table 11) and fails to protect urgent requests as load increases. Continuous Re-optimization, at the opposite extreme, represents a service-quality upper bound but at the cost of very high solver frequency and system nervousness (Section 3.3.1). In realistic deployments, such persistent re-planning would strain computational resources, complicate field coordination and reduce trust among drivers and dispatchers who experience frequent plan changes.

Periodic policies sit between these extremes but are fundamentally constrained by their time-based triggering rule. Periodic-30 improves responsiveness relative to Periodic-60 at the cost of a higher, fixed solver frequency; Periodic-60 reduces solver usage but reacts too slowly during bursts of high-priority arrivals. Because both variants are insensitive to event severity, they are consistently dominated by AET in priority-weighted response time and offer no clear advantage in solver calls, highlighting that fixed schedules are a weak proxy for actual operational needs.

AET's disruption-score mechanism (Section 3.3.2) addresses this gap by conditioning re-optimization on urgency, spatial deviation and remaining slack. The observed trigger precision, where the majority of solver calls yield more than a 5% improvement in the objective, indicates that the event-trigger is both selective and effective at filtering out low-value updates. At the same time, AET attains the lowest nervousness among the optimizing policies (Figure 9), suggesting that its re-optimization decisions are impactful yet sparing, which is crucial in disaster-relief operations where operational clarity and predictability for field teams are as important as marginal gains in objective value.

The results also highlight that no policy can fully overcome structural limitations under extreme load, where demand systematically exceeds capacity (see Table 4 for load definitions). Even with AET, priority-weighted response times grow and unmet demand becomes unavoidable, as shown in the performance trends in Figure 7 and Table 11. In these regimes, AET's value lies in degrading gracefully: it protects high-priority nodes more effectively than Greedy or periodic baselines (see Table 14) while keeping computational and operational overhead within realistic bounds for real-time use by maintaining significantly lower solver calls and system nervousness (Tables 12 and 13).

Existing approaches in humanitarian logistics often employ meta-heuristics such as Ant Colony Optimization, Genetic Algorithms and Tabu Search for dynamic VRP. These techniques explore large solution spaces efficiently but typically assume that routing is re-solved either on a fixed schedule or after every event, without an explicit mechanism for deciding when a re-optimization is worthwhile or for accounting for the operational impact of repeated route changes. The AET framework (Section 3.3.2) introduces a distinct decision layer that evaluates whether a newly arrived request justifies a global re-plan, weighing potential improvement against computational cost and induced instability. In this sense, it couples standard optimization with an adaptive trigger that governs when the solver should be invoked, aligning routing decisions with the urgency and operational context of disaster response.

Considering the limitations, the experiments use a synthetic, Poisson-based request process and a stylized dynamic network which, although carefully designed, cannot capture all nuances of real disaster environments. Trigger thresholds and decay parameters (Equation (13)) were tuned for this setting and may require adjustment or learning-based calibration in practice. Nonetheless, within the controlled environment considered here, the AET policy emerges as a robust, scalable and operationally realistic approach for dynamic resource distribution in disaster response.

6.3. Discussion for Edge LLM

The empirical performance of the evaluated SLM, Qwen2.5-0.5B [58], shows that sub-1B models can achieve a balance of latency, perplexity and downstream accuracy (see Figure 10 and Table 5). This directly addresses the documented lack of rigorous, domain-grounded evaluations of SLMs running natively on mobile-class hardware. The comparatively strong SQuAD [63] and BoolQ [62] performance of Qwen2.5-0.5B relative to similarly sized models reinforces a key insight from recent work: model scale alone is not a reliable predictor of task performance, and factors such as data quality, training objectives and specialization matter strongly in edge settings. The fine-tuning results (Table 15) highlight the limited study of domain-adapted lightweight SLMs for high-stakes applications. Few studies have examined how supervised fine-tuning on specialised corpora affects reasoning quality on-device. In our case (Section 4.1.3), substantial gains in BLEU [64], ROUGE-L [65], F1 and semantic similarity following SFT, with negligible additional inference cost, indicate that task-specific adaptation is a critical complement to compression-centric methods. The model's improved ability to summarise situational information and interpret disaster terminology supports arguments that edge LLMs should be co-designed with the operational environments they serve in emergency decision-making.

Finally, deploying the quantised model on a smartphone (Table 7) addresses the scarcity of end-to-end, on-device evaluations. Studies focused on cloud–edge scheduling, collaborative inference or projected throughput, instead of measuring memory footprint, real-time latency and dialogue responsiveness of a fully packaged SLM running offline. The observed sub-500 ms response times and 54 tokens/s throughput (Table 16) empirically shows that practical, privacy-preserving emergency dialogue systems are feasible on

contemporary mobile SoCs. We showed an integrated pipeline from model selection and fine-tuning through quantization and conversion to live execution aligned with disaster-response requirements such as network independence and rapid situational understanding.

6.4. System-Level Scalability Under High-Demand Disaster Scenarios

In large-scale disaster events, system demand increases in terms of resource allocation complexity, incident volume, concurrent user interactions, and decision-making load across the entire platform. ResQConnect is designed as a modular, decoupled system in which scalability is achieved through functional separation rather than monolithic processing. Incident ingestion, agentic task synthesis (Section 3.2), routing optimization (Section 3.3.2), and edge-based user assistance operate as independent components, allowing system load to scale unevenly across subsystems without causing global degradation. Under high-demand scenarios, the agentic RAG workflow scales primarily with the number of incoming incidents, while routing complexity scales with resource availability and network congestion. The AET routing policy (Section 3.3.2) prevents solver overload by limiting re-optimization to operationally significant events, ensuring that increased demand does not translate into proportional computational growth. Simultaneously, time-critical user interactions under low connectivity are handled by the edge-deployed language model, which operates independently of backend load and maintains low-latency responses even when central services are congested or unavailable.

6.5. Human-Centered Design and Sustainability Implications

ResQConnect adopts a human-centered design not only to improve usability, but also to support long-term sustainability in disaster response systems. Rather than replacing human judgment, ResQConnect augments existing command structures (Section 1), enabling responders to make informed decisions while retaining responsibility.

System resilience is strengthened through architectural choices that explicitly account for real-world constraints. The use of an edge-deployed language model (Section 4.1.3) ensures continuity of guidance during connectivity failures. Similarly, the AET routing mechanism (Section 3.3.2) avoids excessive re-planning, preserving operational stability and reducing cognitive load on field teams. These design decisions allow the system to degrade gracefully under extreme demand rather than failing abruptly.

Fairness considerations are embedded at multiple levels of the platform. Priority-aware routing reflects established humanitarian triage practices, while the agentic RAG workflow grounds task generation in official SOPs (Section 4.1.1) rather than ad-hoc or opaque model behavior. By keeping priority classes stable over time and making routing adaptations transparent, the system supports explainable allocation of limited resources.

Overall, ResQConnect's design supports sustainability by enabling fair decision-making and strengthening community resilience across disaster situations.

6.6. Study Limitations and Future Extensions

Although ResQConnect shows strong performance across retrieval quality, adaptive routing and offline inference, several limitations constrain immediate real-world deployment. First, the system has only been evaluated in controlled synthetic scenarios (as described in Sections 4.1.2 and 4.1.3). These cannot fully capture the messy, multi-stakeholder dynamics of real disasters, where heterogeneous agencies, inconsistent reporting, political pressures and infrastructure failures can affect both agentic and routing behaviour. Future work should therefore include pilot deployments with disaster management centres, local authorities and volunteer networks to evaluate performance under real field constraints. Also while proposed evaluation demonstrates robustness under high-demand conditions, the current implementation assumes a centralized backend deployment. Large-scale, multi-

region disasters may require horizontal scaling through distributed agent orchestration, regional routing instances or load-aware service replication. Exploring these extensions is an important direction for future work.

Second, fairness and equity in resource allocation remain open challenges. While the AET routing algorithm (Section 3.3.2) embeds priority classes and deprivation-sensitive metrics, real-world fairness is more complex than numerical weights. Vulnerable groups may still be under-served if help requests are unevenly distributed or communication access is unequal. Future extensions should incorporate fairness-aware objectives, community-informed priority schemes and transparent audit trails for how allocations are generated. Third, the agentic RAG workflow (Section 3.2) depends on the quality, coverage and recency of its knowledge base. Disaster guidelines evolve and procedures vary by region, so a static corpus risks becoming outdated or misaligned with local practice.

Additionally, responsibility and oversight must be governed before field deployment. AI-generated recommendations may influence life-and-death decisions, but human responders must retain ultimate accountability (Section 1). Future work should embed explicit override mechanisms, explanation layers and user-facing rationales so that ResQConnect augments rather than replaces human judgement. Overall, while the technical results are promising, transitioning ResQConnect into real operations will require attention to ethical safeguards, fairness frameworks, participatory design and institutional interoperability.

Future extensions of the user response system can integrate with precipitation nowcasting models to provide real-time flood risk predictions, enhancing proactive evacuation recommendations during disaster events [68]. Additionally, incorporation of land cover and land use data will enable spatially precise vulnerability assessments, optimizing resource allocation and response strategies in dynamic environmental contexts [13]. Another promising extension involves integrating user learning modules that deliver personalized materials on disaster preparedness, evacuation protocols, and resource utilization directly through the interface [69]. This enhancement is crucial as it empowers communities, particularly in vulnerable regions with actionable knowledge, fostering proactive resilience, reducing panic during crises, and ultimately minimizing casualties and economic losses by bridging the gap between technology and human behavior. Finally, such solutions can be extended by integrating interdisciplinary approaches that combine ecological, technological, and socioeconomic perspectives to holistically address complex disaster dynamics, optimize resource allocation, and enhance long-term community resilience [70].

7. Conclusions

This study introduced ResQConnect, an AI-powered, human-centered multi-agent platform that transformed fragmented crisis data into coordinated flood and landslide responses. It integrated three components: an agentic retrieval-augmented generation workflow that converted citizen reports into grounded task plans; an adaptive event-triggered routing algorithm for dynamic resource allocation; and a compressed edge-deployed language model for reliable guidance under connectivity constraints. Evaluations showed the agentic workflow yielded more relevant, safety-aligned tasks than standard RAG, with acceptable latency trade-offs for high-stakes contexts. The routing strategy matched near-optimal performance while minimizing solver calls and instability. The edge model ensured low-latency support on mobile devices. The system was validated using real-world flood and landslide disaster datasets, reinforcing its applicability under realistic operational conditions. Together, these advances bridged AI capabilities with real-world operations, enhancing situational awareness, allocation, and citizen support under human oversight. ResQConnect charted a practical path to coordinated, accountable disaster response systems.

Author Contributions: Conceptualization, D.M.; methodology, S.A., C.M., J.W., S.K., D.M. and B.P.; software, S.A., C.M., J.W. and S.K.; validation, S.A., C.M., J.W., S.K., D.M. and B.P.; investigation, software, S.A., C.M., J.W., S.K. and D.M.; resources, S.A., C.M., J.W., S.K. and D.M.; data curation, S.A., C.M., J.W. and S.K.; writing—original draft preparation, S.A., C.M., J.W. and S.K.; writing—review and editing, D.M. and B.P.; visualization, software, S.A., C.M., J.W. and S.K.; supervision, D.M. and B.P.; administration, D.M. and B.P.; All authors have read and agreed to the published version of the manuscript.

Funding: Senate Research Committee grant SRC/LT/2025/25 from the University of Moratuwa, Sri Lanka.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Disaster Management Dataset (2025). https://sites.google.com/cse.mrt.ac.lk/resqconnect/resources/datasets (accessed on 10 November 2025).

Acknowledgments: This work was supported in part by the University of Moratuwa, Sri Lanka, under the Senate Research Committee.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A. Web and Mobile UIs

The web-based and mobile user interfaces of the proposed system, which adapt according to user type. The web interface supports role-specific views for authorized users in a role-specific manner. The mobile interface provides a conversational interaction flow for end users.

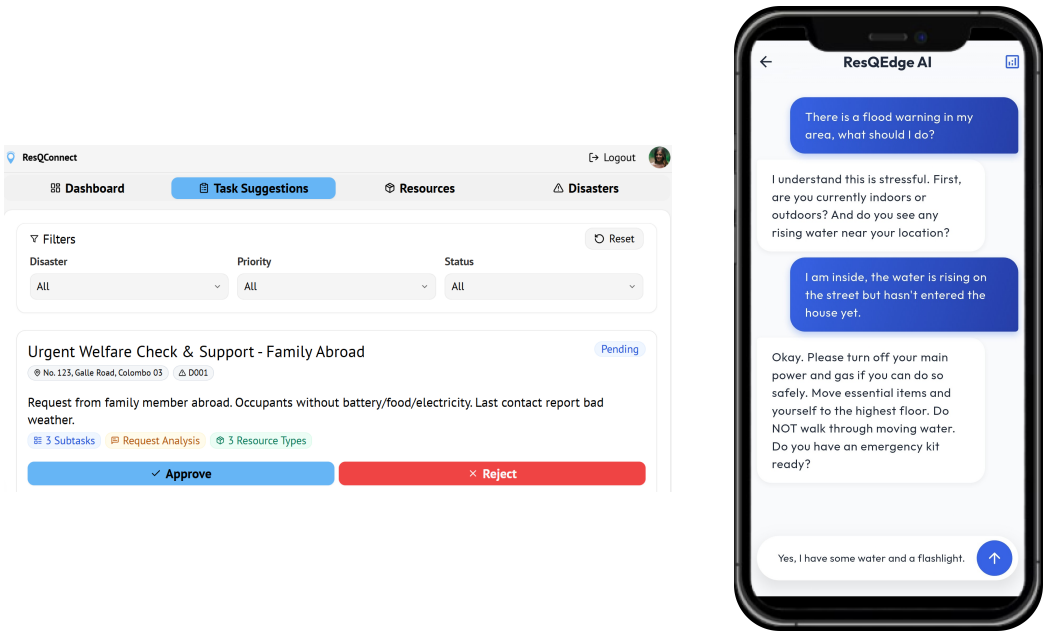

Figure A1. User Interface (left) Web Application, (right) Mobile Application.

Appendix B. Real-World Agentic RAG Outputs

This appendix documents representative real-world disaster (2025 Sri Lanka Floods) help requests and the corresponding task breakdowns generated by the proposed Agentic RAG system. Each case reflects unstructured, real citizen communications and the system-generated operational response without manual intervention or post-processing.

Appendix B.1. Prolonged Loss of Contact

Request (Verbatim):

"I haven't been able to contact my family for about 36 hours, and if anyone can provide information, it would be really helpful."

Generated Tasks:

- Task 1 (High Priority):
  - Visit the reported household location to physically confirm the safety and presence of occupants.
  - Completion criterion: Safety visually confirmed or inability to confirm is reported.

Requirements:

- Manpower: 1 volunteer (assigned to Task 1)
- Resources: None

Appendix B.2. Food Shortage, Power Outage, and Communication Failure

Request (Verbatim):

"I'm abroad last 2 days (my family), their phone is off (confirmed no battery). On last call they mentioned they don't have enough food and cannot go outside due to very bad weather with heavy wind flow. They don't have electricity for last 3 days. Kindly please help and if possible arrange one phone call."

Generated Tasks:

- Task 1 (High Priority):
  - Conduct an on-site welfare check to confirm occupant safety and identify immediate food or water shortages.
  - Completion criterion: Welfare status confirmed in person.
- Task 2 (Medium Priority):
  - Deliver emergency dry food packs and drinking water sufficient for the confirmed household size.
  - Completion criterion: Supplies handed over directly.
- Task 3 (Low Priority):
  - Facilitate one outbound phone call or message to the reporting individual abroad.
  - Completion criterion: Communication succeeds or failure is documented.

Requirements:

- Manpower: 3 volunteers (1 per task)
- Resources:
  - Dry food packs: 6
  - Drinking water bottles (1 litre): 6
  - Battery unit: 1

Appendix B.3. Household with Orthopaedic Patient

Request (Verbatim):

"My father is an orthopaedic patient. There are five people in my house and the others are ladies. We need food and medicine."

Generated Tasks:

- Task 1 (High Priority):

  - Visit the household to confirm the presence of occupants and assess food availability and basic medical needs.
  - Completion criterion: Needs confirmed in person.
- Task 2 (Medium Priority):
  - Deliver emergency dry food packs and provide a basic medikit for short-term household medical needs.
  - Completion criterion: Supplies delivered and acknowledged.

Requirements:

- Manpower: 2 volunteers (1 per task)
- Resources:
  - Dry food packs: 9
  - Medikit: 1

Appendix B.4. Suspected Landslide Impact and Loss of Power

Request (Verbatim):

"Called on 27th morning and said no power. Still no contact and don't know their current situation. Heard there were few landslides around that area. Please help to know information."

Generated Tasks:

- Task 1 (High Priority):
  - Perform physical verification at the household to confirm occupant safety and check for access obstruction.
  - Completion criterion: Safety confirmed or access failure reported.
- Task 2 (Low Priority):
  - Report verified household status to the coordination team for onward communication.
  - Completion criterion: Status documented and relayed.

Requirements:

- Manpower: 2 volunteers (1 per task)
- Resources: None

Appendix C. Judge LLMPrompt

The prompt sent to the LLM is as follows:

```
You are an expert evaluator assessing the performance of a RAG system used
for disaster-response information retrieval. Your main goal is to determine
how well the retrieved knowledge chunks can support the generation of
precise, actionable, safe and field-ready tasks in response to the user
request.

====================
EVALUATION OBJECTIVE
====================
- Evaluate only the quality and contextual relevance of the retrieved
  chunks.
- Reward retrievals that closely match the user's specific needs, urgency,
  and hazard context.
- Strongly penalize retrievals that are generic, surface-level, or fail to
```

```
  reflect the actual intent of the request.
- Retrievals that describe general flood safety, equipment lists, or
  background theory without actionable relation to the user's problem
  should receive low relevance scores.
- Retrievals that address only isolated parts of the user request without
  covering the overall emergency context should receive low scores for
  relevance and contextual enrichment.
- Penalize unsafe, outdated, or unverifiable content.

====================
PRIMARY CRITERIA (1-10 each)
====================

1. Relevance and Hazard Alignment
   - How well do the retrieved chunks help in creating actionable tasks
     that must be carried out to answer the user's request?
       1-3: Off-topic, general disaster information, or fails to support
            actionable planning.
       4-6: Some relevance but limited usefulness for actionable task
            creation.
       7-10: Strongly supports the creation of concrete, field-ready tasks
             addressing multiple key aspects of the request.

2. Contextual Enrichment and Utility
   - Do the retrieved chunks provide clear, step-by-step guidance or
     information that can be directly translated into actionable tasks for
     responders?
       1-3: Adds unrelated or abstract information.
       4-6: Offers partial help or background but lacks field usability.
       7-10: Provides concrete, procedural, or multi-dimensional context
             that supports response decisions.

3. Safety and Procedural Accuracy
   - Are the retrieved chunks factually correct, safe, aligned with
     recognized response practices or SOPs, and do they enable creation
     of actionable, field-ready tasks?
       1-3: Unsafe, incorrect, misleading, or do not support actionable
            task creation.
       4-6: Generally safe but vague, unverified, or only partially
            supportive.
       7-10: Verified, operationally sound, and supportive of concrete,
             actionable tasks.

4. Specificity and Completeness
   - Do the chunks cover who, what, and how clearly enough to guide action?
       1-3: Generic or incomplete.
       4-6: Some specificity but not comprehensive.
       7-10: Detailed, complete, and directly usable for field decisions.

5. Signal Quality and Deduplication
```

```
   - How focused, concise, and unique are the retrieved chunks?
       1-3: Mostly filler or redundant.
       4-6: Some redundancy.
       7-10: Clean, relevant, and non-duplicative.

====================
SCORE CALCULATION
====================
Decision bands:
  Excellent: >=85
  Adequate: 60-84
  Poor: 40-59
  Fail: <=39 or any auto-fail triggered

====================
AUTO-FAIL CONDITIONS
====================
- Retrieved chunks do not help in creating actionable tasks.
- Missing critical procedural steps.
- Hazard misalignment.
- Conflicting or unverifiable instructions.
- Mostly generic or theoretical content.
- Invalid or incomplete YAML.

====================
OUTPUT FORMAT-YAML ONLY
====================
evaluation:
  relevance_score: <1-10>
  contextual_enrichment_score: <1-10>
  safety_accuracy_score: <1-10>
  specificity_completeness_score: <1-10>
  signal_noise_score: <1-10>
  overall_score: <0-100>
  summary: "<<=25 words>"
  missing_information: "<<=25 words>"

judgments:
  hazard_type: <flood | landslide | mixed | unknown>
  decision: <Excellent | Adequate | Poor | Fail>

issues:
  irrelevant_retrieval: <yes/no>
  unsafe_or_incorrect: <yes/no>
  outdated_content: <yes/no>
  noisy_or_duplicate: <yes/no>
  missing_context: <yes/no>

selection:
  keep_ids: [<chunk_index>, ...]
```

```
drop_ids: [<chunk_index>, ...]
conditions:
  - "Keep chunks that provide clear, actionable guidance."
  - "Drop chunks that are generic, theoretical, or non-actionable."
  - "Retain only content that enables precise task planning."
```

Appendix D. Qualitative Comparison Examples for Edge LLM

This appendix presents representative qualitative examples comparing responses generated by the Base Model and the Fine-Tuned Model against ground-truth answers for identical user queries. The examples highlight differences in factual accuracy, contextual grounding, and operational usefulness in disaster-response scenarios.

Table A1. Qualitative examples comparing base vs. fine-tuned model outputs.

Ex.	User Query	Ground Truth	Base Model	Fine-Tuned Model
1	Where should I keep the water containers so they don't get contaminated indoors?	Store away from fuel or chemicals, on clean raised surfaces, and keep containers covered. Use food-grade plastic or stainless or enamel containers and avoid direct hand contact.	You should keep your water containers in a dry and well-ventilated area.	Containers should be stored in sealed, dedicated areas off the floor and away from chemicals and moisture.
2	Who do we call first if someone is injured while heading to the meeting point?	Call 1990 Suwa Seriya for an ambulance and provide your exact location and landmarks. Use 117 for updates and 118 or 119 if police assistance is needed.	First responders or medical professionals should be contacted to assist the injured individual.	Call the 1990 emergency ambulance service immediately for medical assistance.
3	I can do debris clearing, but the site has no helmets or gloves. Should I still take the job?	No. Basic PPE such as gloves, boots, helmets, and masks must be provided before work begins. Workers have the right to safe conditions.	Protective gear like helmets and gloves is important when working in hazardous environments.	No. Debris clearing should not be undertaken without proper protective equipment.

References

1. Intergovernmental Panel on Climate Change (IPCC). *Climate Change 2021: The Physical Science Basis*, 1st ed.; Cambridge University Press: Cambridge, UK, 2023.
2. Pescaroli, G.; Alexander, D. Critical infrastructure, panarchies and the vulnerability paths of cascading disasters. *Nat. Hazards* **2016**, *82*, 175–192. [CrossRef]
3. Krishnan, R.; Dhara, C.; Horinouchi, T.; Gotangco Gonzales, C.K.; Dimri, A.; Shrestha, M.S.; Swapna, P.; Roxy, M.; Son, S.W.; Ayantika, D.; et al. Compound weather and climate extremes in the Asian region: Science-informed recommendations for policy. *Front. Clim.* **2025**, *6*, 1504475. [CrossRef]
4. floodsupport.org. Emergency SOS-Flood Rescue Sri Lanka. 2025. Available online: https://floodsupport.org/ (accessed on 29 November 2025).
5. Zhang, C.; Fan, C.; Yao, W.; Hu, X.; Mostafavi, A. Social media for intelligent public information and warning in disasters: An interdisciplinary review. *Int. J. Inf. Manag.* **2019**, *49*, 190–207. [CrossRef]
6. Furin, M.; Freeman, C.L.; Goldstein, S. EMS Incident Command System. 2024. Available online: https://www.ncbi.nlm.nih.gov/books/NBK441863/ (accessed on 29 November 2025).

7. Wolbers, J.; Boersma, K.; Groenewegen, P. Introducing a Fragmentation Perspective on Coordination in Crisis Management. *Organ. Stud.* **2018**, *39*, 1521–1546. [CrossRef]
8. Imran, M.; Castillo, C.; Diaz, F.; Vieweg, S. Processing Social Media Messages in Mass Emergency: A Survey. In Proceedings of the Companion Proceedings of the The Web Conference, Lyon, France, 23–27 April 2018; pp. 507–511.
9. Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Luo, K.; Zhang, J. Edge Intelligence: Paving the Last Mile of Artificial Intelligence with Edge Computing. *Proc. IEEE* **2019**, *107*, 1738–1762. [CrossRef]
10. Duc, K.N.; Vu, T.T.; Ban, Y. Ushahidi and Sahana Eden Open-Source Platforms to Assist Disaster Relief: Geospatial Components and Capabilities. In *Geoinformation for Informed Decisions*; Springer: Cham, Switzerland , 2014; pp. 163–174.
11. United Nations. United Nations: Sustainable Development Goals. Available online: https://sdgs.un.org/goals (accessed on 29 November 2025).
12. Luna-Ramirez, W.A.; Fasli, M. Bridging the Gap between ABM and MAS: A Disaster-Rescue Simulation Using Jason and NetLogo. *Computers* **2018**, *7*, 24. [CrossRef]
13. Jayanetti, A.; Meedeniya, D.; Dilini N.; Wickramapala, M.; Madushanka, H. Enhanced land cover and land use information generation from satellite imagery and foursquare data. In Proceedings of the 6th International Conference on Software and Computer Applications (ICSCA), Bangkok,Thailand, 26–28 February 2017; pp. 5149–5153.
14. Meedeniya, D.; Jayanetti, A.; Dilini N.; Wickramapala, M.; Madushanka, H. Land-Use Classification with Integrated Data. In *Machine Vision Inspection Systems: Image Processing, Concepts, Methodologies and Applications*; Malarvel, M., Nayak, S., Panda, S., Pattnaik, P., Muangnak, N., Eds.; John Wiley and Sons: Hoboken, NJ, USA, 2020; Volume 1, Chapter 1, pp. 1–36.
15. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Proceedings of the 34th International Conference on Neural Information Processing Systems, Virtual, 6–12 December 2020; pp. 9459–9474.
16. Li, X.; Wang, S.; Zeng, S.; Wu, Y.; Yang, Y. A survey on LLM-based multi-agent systems: Workflow, infrastructure, and challenges. *Vicinagearth* **2024**, *1*, 9. [CrossRef]
17. Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversations. In Proceedings of the First Conference on Language Modeling, Philadelphia, PA, USA, 7–9 October 2024; pp. 1–46.
18. Singh, A.; Ehtesham, A.; Kumar, S.; Khoei, T.T. Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG. *arXiv* **2025**, arXiv:2501.09136. [CrossRef]
19. Chang, C.Y.; Jiang, Z.; Rakesh, V.; Pan, M.; Yeh, C.C.M.; Wang, G.; Hu, M.; Xu, Z.; Zheng, Y.; Das, M.; et al. MAIN-RAG: Multi-Agent Filtering Retrieval-Augmented Generation. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, Vienna, Austria, 27 July–1 August 2025; pp. 2607–2622.
20. Hong, L.; Song, X.; Anik, A.S.; Frias-Martinez, V. Dynamic Fusion of Large Language Models for Crisis Communication. In Proceedings of the International ISCRAM Conference, Halifax, NS, Canada, 18–21 May 2025; pp. 1–11.
21. Otal, H.T.; Stern, E.; Canbaz, M.A. LLM-Assisted Crisis Management: Building Advanced LLM Platforms for Effective Emergency Response and Public Collaboration. In Proceedings of the IEEE Conference on Artificial Intelligence (CAI), Marina Bay Sands, Singapore, 25–27 June 2024; pp. 851–859.
22. Altay, N.; Green, W.G. OR/MS research in disaster operations management. *Eur. J. Oper. Res.* **2006**, *175*, 475–493. [CrossRef]
23. Holguín-Veras, J.; Jaller, M.; Van Wassenhove, L.N.; Pérez, N.; Wachtendorf, T. On the unique features of post-disaster humanitarian logistics. *J. Oper. Manag.* **2012**, *30*, 494–506. [CrossRef]
24. Lamos Díaz, H.; Aguilar Imitola, K.; Acosta Amado, R.J. OR/MS research perspectives in disaster operations management: A literature review. In *Revista Facultad de Ingeniería*; Universidad de Antioquia: Medellín, Colombia, 2019; pp. 43–59.
25. Balcik, B.; Beamon, B.M. Facility location in humanitarian relief. *Int. J. Logist. Res. Appl.* **2008**, *11*, 101–121. [CrossRef]
26. Ahmadi, M.; Seifi, A.; Tootooni, B. A humanitarian logistics model for disaster relief operation considering network failure and standard relief time: A case study on San Francisco district. *Transp. Res. Part E Logist. Transp. Rev.* **2015**, *75*, 145–163. [CrossRef]
27. Rodríguez-Espíndola, O.; Albores, P.; Brewster, C. Disaster preparedness in humanitarian logistics: A collaborative approach for resource management in floods. *Eur. J. Oper. Res.* **2018**, *264*, 978–993. [CrossRef]
28. Sheu, J.B. Dynamic relief-demand management for emergency logistics operations under large-scale disasters. *Transp. Res. Part E Logist. Transp. Rev.* **2010**, *46*, 1–17. [CrossRef]
29. Zhao, J.; Cao, C. Review of Relief Demand Forecasting Problem in Emergency Logistic System. *J. Serv. Sci. Manag.* **2015**, *8*, 92–98. [CrossRef]
30. Prado, A.M. Delivering humanitarian assistance at the last mile of the supply chain: Insights on recruiting and training. In Proceedings of the 26th Production and Operations Management Society Annual Conference (POMS), Washington, DC, USA, 8–11 May 2015; pp. 1–10.
31. Holguín-Veras, J.; Pérez, N.; Jaller, M.; Van Wassenhove, L.N.; Aros-Vera, F. On the appropriate objective function for post-disaster humanitarian logistics models. *J. Oper. Manag.* **2013**, *31*, 262–280. [CrossRef]

32. Luss, H. On Equitable Resource Allocation Problems: A Lexicographic Minimax Approach. *Oper. Res.* **1999**, *47*, 361–378. [CrossRef]
33. Huang, K.; Rafiei, R. Equitable last mile distribution in emergency response. *Comput. Ind. Eng.* **2019**, *127*, 887–900. [CrossRef]
34. Wang, Y.; Sun, B. Multiperiod Equitable and Efficient Allocation Strategy of Emergency Resources Under Uncertainty. *Int. J. Disaster Risk Sci.* **2022**, *13*, 778–792. [CrossRef]
35. Ghahremani-Nahr, J.; Nozari, H.; Szmelter-Jarosz, A. Designing a humanitarian relief logistics network considering the cost of deprivation using a robust-fuzzy-probabilistic planning method. *J. Int. Humanit. Action* **2024**, *9*, 19–35. [CrossRef]
36. O'Sullivan, L.; Aldasoro, E.; O'Brien, Á.; Nolan, M.; McGovern, C.; Carroll, Á. Ethical values and principles to guide the fair allocation of resources in response to a pandemic: A rapid systematic review. *BMC Med. Ethics* **2022**, *23*, 70–81. [CrossRef]
37. Dutta, L.; Bharali, S. TinyML Meets IoT: A Comprehensive Survey. *Internet Things* **2021**, *16*, 100461. [CrossRef]
38. Wang, X.; Tang, Z.; Guo, J.; Meng, T.; Wang, C.; Wang, T.; Jia, W. Empowering Edge Intelligence: A Comprehensive Survey on On-Device AI Models. *ACM Comput. Surv.* **2025**, *57*, 1–39. [CrossRef]
39. Friha, O.; Ferrag, M.A.; Kantarci, B.; Cakmak, B.; Ozgun, A.; Ghoualmi-Zine, N. LLM-Based Edge Intelligence: A Comprehensive Survey on Architectures, Applications, Security and Trustworthiness. *IEEE Open J. Commun. Soc.* **2024**, *5*, 5799–5856. [CrossRef]
40. Semerikov, S.O.; Vakaliuk, T.A.; Kanevska, O.B.; Ostroushko, O.A.; Kolhatin, A.O. Edge intelligence unleashed: A survey on deploying large language models in resource-constrained environments. *J. Edge Comput.* **2025**, *4*, 179–233. [CrossRef]
41. Qu, G.; Chen, Q.; Wei, W.; Lin, Z.; Chen, X.; Huang, K. Mobile Edge Intelligence for Large Language Models: A Contemporary Survey. *IEEE Commun. Surv. Tutor.* **2025**, *27*, 3820–3860. [CrossRef]
42. Paranayapa, T.; Ranasinghe, P.; Ranmal, D.; Meedeniya, D.; Perera, C. A Comparative Study of Preprocessing and Model Compression Techniques in Deep Learning for Forest Sound Classification. *Sensors* **2024**, *24*, 1149. [CrossRef]
43. Dantas, P.V.; Cordeiro, L.C.; Junior, W.S. A review of state-of-the-art techniques for large language model compression. *Complex Intell. Syst.* **2025**, *11*, 1–40. [CrossRef]
44. Sun, Z.; Yu, H.; Song, X.; Liu, R.; Yang, Y.; Zhou, D. MobileBERT: A Compact Task-Agnostic BERT for Resource-Limited Devices. *arXiv* **2020**, arXiv:2004.02984.
45. Duan, Q.; Lu, Z. Edge Cloud Computing and Federated–Split Learning in Internet of Things. *Future Internet* **2024**, *16*, 227. [CrossRef]
46. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024; pp. 1–30.
47. Yan, S.Q.; Gu, J.C.; Zhu, Y.; Ling, Z.H. Corrective Retrieval Augmented Generation. *arXiv* **2024**, arXiv:2401.15884. [CrossRef]
48. Jiang, Z.; Xu, F.F.; Gao, L.; Sun, Z.; Liu, Q.; Dwivedi-Yu, J.; Yang, Y.; Callan, J.; Neubig, G. Active Retrieval Augmented Generation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; pp. 7969–7992.
49. Chan, C.M.; Xu, C.; Yuan, R.; Luo, H.; Xue, W.; Guo, Y.; Fu, J. RQ-RAG: Learning to Refine Queries for Retrieval Augmented Generation. *arXiv* **2024**, arXiv:2404.00610. [CrossRef]
50. Gheorghiu, A. *Building Data-Driven Applications with LlamaIndex: A Practical Guide to Retrieval-Augmented Generation (RAG) to Enhance LLM Applications*, 1st ed.; Packt Publishing Ltd.: Birmingham, UK, 2024; pp. 1–368.
51. Peric, N.; Begovic, S.; Lesic, V. Adaptive Memory Procedure for Solving Real-world Vehicle Routing Problem. *arXiv* **2024**, arXiv:2403.04420. [CrossRef]
52. Noyan, N. Risk-averse two-stage stochastic programming with an application to disaster management. *Comput. Oper. Res.* **2012**, *39*, 541–559. [CrossRef]
53. Özdamar, L.; Ekinci, E.; Küçükyazici, B. Emergency Logistics Planning in Natural Disasters. *Ann. Oper. Res.* **2004**, *129*, 217–245. [CrossRef]
54. Lu, Z.; Li, X.; Cai, D.; Yi, R.; Liu, F.; Liu, W.; Luan, J.; Zhang, X.; Lane, N.D.; Xu, M. Demystifying Small Language Models for Edge Deployment. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, Vienna, Austria, 27 July–1 August 2025; pp. 14747–14764.
55. Jang, S.; Morabito, R. Edge-First Language Model Inference: Models, Metrics, and Tradeoffs. *arXiv* **2025**, arXiv:2505.16508. [CrossRef]
56. David, R.; Duke, J.; Jain, A.; Janapa Reddi, V.; Jeffries, N.; Li, J.; Kreeger, N.; Nappier, I.; Natraj, M.; Wang, T.; et al. TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems. *Proc. Mach. Learn. Syst.* **2021**, *3*, 800–811.
57. Yu, Z.; Liu, S.; Denny, P.; Bergen, A.; Liut, M. Integrating Small Language Models with Retrieval-Augmented Generation in Computing Education: Key Takeaways, Setup, and Practical Insights. In Proceedings of the 56th ACM Technical Symposium on Computer Science Education, Pittsburgh, PA, USA, 26 February–1 March 2025; pp. 1302–1308.
58. Qwen Team. Qwen2.5: A Party of Foundation Models. 2024. Available online: https://qwenlm.github.io/blog/qwen2.5/ (accessed on 29 November 2025).
59. Zhang, P.; Zeng, G.; Wang, T.; Lu, W. TinyLlama: An Open-Source Small Language Model. *arXiv* **2024**, arXiv:2401.02385.

60. Li, Y.; Bubeck, S.; Eldan, R.; Del Giorno, A.; Gunasekar, S.; Lee, Y.T. Textbooks Are All You Need II: Phi-1.5 technical report. *arXiv* **2023**, arXiv:2309.05463. [CrossRef]
61. Gemma Team. Gemma 3. 2025. Available online: https://goo.gle/Gemma3Report (accessed on 29 November 2025).
62. Clark, C.; Lee, K.; Chang, M.W.; Kwiatkowski, T.; Collins, M.; Toutanova, K. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics, Minneapolis, MN, USA, 2–7 June 2019; pp. 2924–2936.
63. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 2383–2392.
64. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002; pp. 311–318.
65. Lin, C.Y. Rouge: A package for automatic evaluation of summaries. In Proceedings of the Text Summarization Branches out, Barcelona, Spain, 25–26 July 2004; pp. 74–81.
66. Aththanayake, S.; Mallikarachchi, C.; Kugarajah, S.; Wickramasinghe, J. ResQConnect: An AI-Powered Multi-Agentic Platform for Human-Centered Disaster Response. 2025. Available online: https://sites.google.com/cse.mrt.ac.lk/resqconnect/ (accessed on 29 November 2025).
67. Aththanayake, S.; Mallikarachchi, C.; Kugarajah, S.; Wickramasinghe, J. Synthetic Citizen Help Requests Dataset for Natural Disasters. 2025. Available online: https://docs.google.com/spreadsheets/d/1xW_PqC9sx7Zyyd1zpq_u0mKpywYoesbfMmRJ4ufqajo/ (accessed on 1 January 2026).
68. Ahangama I.; Meedeniya D.; Pradhan, B. Explainable Image Segmentation for Spatio-Temporal and Multivariate Image Data in Precipitation Nowcasting. *Results Eng.* **2025**, *26*, 105595. [CrossRef]
69. Perera, I.; Meedeniya, D.; Benerjee, I.; Choudhury, J. Educating Users for Disaster Management: An Exploratory Study on Using Immersive Training for Disaster Management. In Proceedings of the IEEE International Conference on MOOC, Innovation and Technology in Education (MITE), Jaipur, India, 20–22 December 2013; pp. 245–250.
70. Binlajdam, R.; Meedeniya, D.; Jayaweera, K.; Karakus, O.; Rana, O.; Ter Wengel, P.; Goossens, B.; Lertsinsrubtavee, A.; Mekbungwan, P.; Mishra, D.; et al. Review on sustainable forestry with artificial intelligence. *ACM J. Comput. Sustain. Soc.* **2025**, *3*, 35. [CrossRef]