

Predictive ACO-ML Framework for Proactive Evacuation Route Adjustment in Dynamic Disaster Environments

Amoforitse, Fortune Ighotuweyin ¹, Omorogbe Osasu Harry (PhD)², Ozobialu Emmanuel Chukwuma ³, Prof. Kingsley Chiwuike Ukaoha ⁴, Obande, Bonnie Obeka, PhD (CLN) ⁵

¹ Department of Cybersecurity and ICT Unit, Igbinedion University, Okada. Edo State, Nigeria

Email: amoforitse.fortune@iuokada.edu.ng

² Department of Cybersecurity, Igbinedion University, Okada. Edo State, Nigeria

E-mail: omorogbe.harry@iuokada.edu.ng

³ Department of Software Engineering, Igbinedion University, Okada. Edo State, Nigeria

Email: ozobialu.emmanuel@iuokada.edu.ng

⁴ Dean, College of Science and Computing, WIGWE University, Rivers State, Nigeria

E-mail: kingsley.ukaoha@wigwe.edu.ng

⁵ Igbinedion University Library, Okada, Edo State, Nigeria

E-mail: unclebons@gmail.com, bonnie.obande@okada.edu.ng

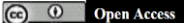
ARTICLE INFO

Keywords:

Ant Colony Optimization (ACO),
Machine Learning (ML),
Predictive Evacuation Routing,
Proactive Evacuation Planning,
Dynamic Disaster Environments,
Disaster Logistics.

Vol.01, No.03, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CCBY4.0).

<http://creativecommons.org/licenses/by/4.0/>


ABSTRACT

Disaster logistics and evacuation planning are increasingly challenged by dynamic environmental conditions, infrastructure failures, and uncertain resource demands. Conventional optimization methods remain largely reactive and are unable to adapt proactively to rapidly evolving disaster scenarios. This study developed a Predictive Ant Colony Optimization–Machine Learning (ACO–ML) framework for proactive evacuation route adjustment and disaster logistics optimization in dynamic disaster environments. The study adopted a Design Science Research (DSR) methodology integrated with simulation-based evaluation. The framework was developed by combining Machine Learning prediction models with an enhanced Ant Colony Optimization algorithm. Predictive models estimated road accessibility, traffic congestion, shelter demand, and infrastructure conditions using heterogeneous data sources, while the ACO algorithm generated adaptive evacuation routes. The proposed framework was evaluated through simulation and benchmarked against conventional optimization approaches. The findings showed that integrating predictive Machine Learning with Ant Colony Optimization significantly improved optimization performance by enhancing convergence speed, computational efficiency, route quality, predictive accuracy, scalability, robustness, and adaptability. The framework also improved situational awareness through the integration of multi-source disaster data, enabling proactive route adjustment and more effective disaster logistics and evacuation planning. The study concludes that the proposed Hybrid ACO–ML framework provides a predictive, adaptive, and scalable decision-support solution for disaster logistics and evacuation planning. The framework advances computational disaster management by combining predictive intelligence with swarm optimization and offers a practical foundation for next-generation intelligent disaster response systems, Smart Cities, and AI-enabled emergency management.

1. Introduction

The increasing frequency and intensity of disasters both natural and man-made pose significant challenges to humanitarian response, logistics coordination, and evacuation planning. Reports from the United Nations Office for Disaster Risk Reduction (UNDRR, 2023) indicate that disasters are becoming more complex, driven by climate change, rapid urbanization, and geopolitical instability. In such contexts, ensuring rapid and efficient decision-making is essential to allocate scarce resources, design reliable evacuation routes, and safeguard vulnerable populations. However, disaster environments remain inherently unpredictable, often characterized by rapidly changing road conditions, infrastructure disruptions, traffic surges, and fluctuating demands for aid (Pu et al., 2025). Traditional optimization

models, though mathematically rigorous, are frequently static and fail to capture this highly dynamic and uncertain nature (Chen et al., 2022).

The world has witnessed a breaking number of disasters in the last few years with a record of more than 300 natural disaster in the second half of 2020 and the first of 2025. This outpaces the recorded numbers from 2000-2019 with average of 185 disaster (Sreelakshmi and Vinod Chandra 2022). A number of 389 climate-related disasters only were recorded during 2020. As a result, the number of recorded disasters during the same year was greater than the average of the statistically recorder number in the previous years with 26% more storms and 23% more floods combined with a higher number of human and economic losses (Linardos et al. 2022a). These calamitous incidents significantly affect the infrastructure and properties resulting in homeless people, impacts the public mental health of the survivors who have lost everything due to this unforeseen and unpredicted natural hazard in the affected area and consequently causes socioeconomical losses.

Disaster prediction and detection contributes to reducing the number of casualties and the economic losses. Properties and natural resources can be protected in some cases such as hurricanes and floods if early warning systems are provided (Gupta and Kumar Rana n.d.). However, even with a short period of notice before disaster, people can take actions to protect themselves against harm and death. Therefore, vast investment needs to be done by governments to enhance the disaster detection and management systems using the recent technologies made by the advances in computer science that eased the access to large amount of data (Chamola et al. n.d.).

At the core of disaster logistics and evacuation planning lie combinatorial optimization challenges such as the Traveling Salesperson Problem (TSP) and the Vehicle Routing Problem (VRP). Both problems are NP-hard, meaning that exact solutions become computationally infeasible at scale, particularly under time-sensitive disaster conditions (Li et al., 2021). To address this, researchers have employed metaheuristic algorithms, which aim to provide near-optimal solutions within reasonable timeframes.

Among these, Ant Colony Optimization (ACO) has emerged as a widely adopted method. Inspired by the foraging behaviour of ants, ACO algorithms simulate how ants deposit and follow pheromone trails to reinforce successful paths, making it particularly suitable for routing problems in logistics (Dorigo & Stützle, 2004). Applications of ACO in disaster logistics have demonstrated its effectiveness in identifying viable routes and allocating resources under constrained conditions (Liu & Han, 2021; Zhang et al., 2023). Nevertheless, traditional ACO faces limitations:

- i. It is often reactive, generating solutions based on current states without anticipating future changes (Qin et al., 2022).
- ii. It suffers from slow convergence and the risk of getting trapped in local optima (Kaur & Singh, 2021).
- iii. Its performance deteriorates in highly dynamic and uncertain environments, such as rapidly evolving disasters (Wang et al., 2023).

In parallel, Machine Learning (ML) has emerged as a powerful tool in disaster management, offering predictive and adaptive capabilities beyond the scope of traditional metaheuristics. For example, deep learning models have been applied to flood forecasting (Shen et al., 2020), while satellite imagery analysis powered by convolutional neural networks has enhanced earthquake damage detection (Xu et al., 2021). More broadly, ML contributes through:

- i. Predictive analytics: Leveraging historical and real-time data to forecast disruptions, resource demands, and traffic bottlenecks (Zhao et al., 2022).
- ii. Dynamic decision-making: Reinforcement Learning (RL) allows adaptive policy learning in environments that evolve unpredictably (Alabsi et al., 2023).
- iii. Data integration: ML models can fuse heterogeneous data sources—including sensor readings, social media feeds, and satellite imagery—to provide comprehensive situational awareness (Rahman et al., 2021; Tang et al., 2023).

Statement of the Research Problem

Disaster logistics and evacuation planning require rapid and adaptive decision-making, yet existing optimization methods remain inadequate for dynamic disaster environments. Traditional models assume static conditions and cannot effectively respond to changing road conditions, infrastructure damage,

traffic congestion, or fluctuating resource demands. Although Ant Colony Optimization (ACO) efficiently identifies optimal routes, it is largely reactive and lacks the predictive capability to anticipate future disruptions. Furthermore, the computational complexity of large-scale disaster problems limits the scalability of existing approaches, while most optimization frameworks fail to integrate real-time data from IoT devices, satellite imagery, UAVs, and other emerging technologies. These limitations highlight the absence of a predictive, adaptive, and scalable disaster management framework. Therefore, this study proposes a hybrid Ant Colony Optimization–Machine Learning (ACO–ML) framework that combines predictive analytics with intelligent optimization to enable proactive evacuation route adjustment and improve disaster logistics and decision-making in dynamic disaster environments.

Research Questions and Corresponding Hypotheses

i. **RQ1:** How can a hybrid Ant Colony Optimization–Machine Learning (ACO–ML) framework improve the efficiency and adaptability of disaster logistics and evacuation planning compared to traditional methods?

a. **H1:** A hybrid ACO–ML framework will provide significantly better optimization performance (in terms of solution quality and convergence speed) compared to traditional ACO alone in disaster logistics and evacuation planning.

ii. **RQ2:** To what extent can Machine Learning models enhance the predictive capability of ACO in anticipating disruptions (e.g., road closures, demand surges, and resource constraints) during disaster response?

b. **H2:** The integration of Machine Learning predictive models within the ACO framework will enhance adaptability in dynamic disaster environments by enabling the system to anticipate disruptions and adjust evacuation routes proactively.

iii. **RQ3:** How scalable and robust is the proposed hybrid ACO–ML framework when applied to large-scale and uncertain disaster scenarios?

c. **H3:** The hybrid ACO–ML framework will demonstrate improved scalability and robustness, maintaining high performance under increasing problem sizes, higher uncertainty levels, and varying disaster scenarios.

Literature Review:

Disaster Prediction and Machine Learning Concepts

The increasing frequency and severity of natural and human-induced disasters have stimulated extensive research into the application of Artificial Intelligence (AI) and Machine Learning (ML) for disaster prediction and management. AI-driven techniques enable the analysis of large, heterogeneous datasets, allowing emergency managers to predict disaster occurrence, assess risks, optimize resource allocation, and improve evacuation planning. Compared with conventional approaches, ML models provide faster, more accurate, and adaptive predictions, making them valuable tools for proactive disaster risk reduction.

Disaster Prediction Techniques

Disaster prediction techniques are broadly classified into **traditional methods** and **artificial intelligence-based methods**. Traditional prediction approaches rely on statistical analyses, mathematical models, historical observations, and physical simulations to forecast disaster events. Although these methods have contributed significantly to hazard assessment, they often assume static environmental conditions and are less effective in modelling the complex, nonlinear, and uncertain nature of disasters.

The early development of Artificial Intelligence, commonly referred to as **Good Old-Fashioned Artificial Intelligence (GOFAI)**, emerged between the 1950s and 1980s. This paradigm emphasized symbolic reasoning, rule-based systems, and logical inference, where human knowledge was explicitly encoded into computer systems to solve structured problems (Haugeland, 1985; Russell & Norvig, 2010). While these systems offered transparency and interpretability, they lacked the ability to learn from data or adapt to changing environments, limiting their applicability in disaster prediction.

Modern AI has shifted toward data-driven learning approaches capable of modelling uncertainty and extracting complex patterns from large datasets. Machine learning algorithms including supervised, unsupervised, deep learning, and reinforcement learning techniques—have significantly improved the

prediction of floods, landslides, earthquakes, wildfires, and other disasters through continuous learning from historical and real-time data.

Machine Learning in Disaster Prediction

Machine learning has become a fundamental component of disaster prediction because of its ability to analyse multidimensional datasets from Geographic Information Systems (GIS), remote sensing, Internet of Things (IoT) sensors, weather stations, satellite imagery, and historical disaster records. These models identify hidden relationships among variables, enabling early warning systems and supporting timely decision-making.

Several empirical studies have demonstrated the effectiveness of ML algorithms in disaster prediction. Yuan and Moayed (2020) developed a multilayer perceptron (MLP) optimized with Ant Colony Optimization, Genetic Algorithms, Particle Swarm Optimization, and other optimization techniques to predict landslides, achieving approximately **85% prediction accuracy**. Similarly, Hafiz et al. (2021) applied Support Vector Machines (SVMs) combined with image processing techniques for flood detection, reporting prediction accuracies exceeding **90%** while reducing computational time.

Khalaf et al. (2015) developed an intelligent flood monitoring system integrating sensor networks and machine learning algorithms, including Random Forest, Bagging, Decision Tree, and HyperPipes. Among these methods, the Random Forest algorithm achieved the highest classification accuracy of **99.5%**, demonstrating the effectiveness of ensemble learning for flood prediction.

Machine Learning Applications in Disaster Management

Machine learning supports virtually every phase of disaster management, including disaster prediction, preparedness, evacuation planning, response, recovery, and post-disaster assessment. During emergency response, ML models analyse evolving environmental conditions, traffic flow, population movement, and infrastructure status to support rapid decision-making. In evacuation planning, predictive analytics improve route optimization by anticipating congestion, road closures, and shelter demand.

In post-disaster management, ML techniques assist in damage assessment and recovery planning using satellite imagery, unmanned aerial vehicles (UAVs), and computer vision technologies. Brighente et al. (2019) proposed an SVM-based framework integrated with UAVs to detect disaster threats and support emergency monitoring. Likewise, Cooner, Shao, and Campbell (2016) evaluated several machine learning algorithms—including multilayer feed-forward neural networks, radial basis function networks, and Random Forests—for earthquake damage detection, reporting accuracy levels above **90%**. Arefi and Behr (2018) further introduced deep learning models capable of automatically identifying damaged regions from remote sensing imagery, thereby improving post-disaster assessment.

Reinforcement Learning in Disaster Management

Among machine learning techniques, **Reinforcement Learning (RL)** has gained increasing attention because of its ability to learn optimal decision-making strategies through interaction with dynamic environments. RL enables intelligent agents to maximize cumulative rewards by receiving feedback in the form of rewards or penalties based on their actions (Mahesh, 2018). In disaster management, RL supports adaptive evacuation routing, resource allocation, traffic control, and emergency response planning under uncertain and rapidly changing conditions. Its ability to continuously learn from operational outcomes makes it particularly valuable for real-time disaster decision support.

Ant Colony Optimization and Related Concepts

Ant Colony Optimization (ACO) is a swarm intelligence metaheuristic inspired by the foraging behaviour of ants, where artificial ants communicate through pheromone trails to discover optimal paths in complex search spaces (Dorigo & Stützle, 2004). Owing to its effectiveness in solving NP-hard routing and scheduling problems, ACO has been widely applied to disaster logistics and evacuation planning.

The original **Ant System (AS)**, introduced by Dorigo et al. (1991), forms the foundation of ACO by simulating pheromone deposition and evaporation during path exploration. Later, the **Ant Colony System (ACS)** enhanced the original algorithm through local pheromone updates, candidate lists, and improved exploration strategies, resulting in faster convergence and better optimization performance (Dorigo & Gambardella, 1997).

Recent developments have focused on **adaptive ACO tuning**, where algorithm parameters such as pheromone evaporation rates, heuristic weights, and exploration strategies are dynamically adjusted using machine learning and real-time environmental feedback to improve optimization performance in uncertain disaster environments (Stützle & Hoos, 2000). This integration significantly enhances the responsiveness and robustness of ACO in dynamic evacuation planning.

Related Concepts

Several concepts underpin the integration of machine learning and optimization in disaster management. **Artificial Neural Networks (ANNs)** are computational models inspired by biological neural systems and are widely used for disaster prediction, image classification, and hazard susceptibility mapping (Pradhan & Lee, 2010). **Bio-inspired optimization** encompasses algorithms derived from natural phenomena, including ACO, Genetic Algorithms, and Particle Swarm Optimization, which efficiently solve complex optimization problems (Bonabeau et al., 1999).

Case-Based Reasoning (CBR) improves disaster decision support by retrieving and adapting solutions from previous disaster events to similar situations (Zhu et al., 2019). **Dynamic routing** enables continuous adjustment of evacuation routes based on changing disaster conditions, traffic congestion, and infrastructure availability (Kaur & Singh, 2021). **Evacuation planning** focuses on optimizing safe and efficient movement of affected populations using multi-objective optimization techniques (Murray-Tuite & Wolshon, 2013), while **disaster logistics** involves coordinating emergency supplies, transportation, and relief operations under resource constraints (Altay & Green, 2006). These concepts collectively contribute to **Disaster Risk Reduction (DRR)** by improving preparedness, response, and resilience in line with the Sendai Framework and the United Nations Sustainable Development Goals (UNDRR, 2023).

Overall, the literature demonstrates that integrating predictive machine learning with adaptive optimization algorithms, particularly Ant Colony Optimization, offers significant potential for developing intelligent decision-support systems capable of proactive evacuation planning and efficient disaster logistics in dynamic environments.

Learning Theory and Predictive Intelligence in Disaster Contexts

Learning theory underpins machine learning (ML), focusing on how systems improve performance through experience derived from data, observation, or environmental interaction (Mitchell, 1997; Shalev-Shwartz & Ben-David, 2014). In disaster contexts, ML enables a paradigm shift from reactive optimization to predictive, anticipatory planning. By learning patterns from historical and real-time data, ML models can forecast demand surges, hazard propagation, and population movements, thereby informing optimization algorithms with foresight rather than hindsight (Russell & Norvig, 2021).

Research Methodology

This study adopts a **Design Science Research (DSR)** methodology combined with **simulation-based evaluation** to design, develop, and validate a hybrid **Ant Colony Optimization–Machine Learning (ACO–ML)** framework for disaster logistics and evacuation planning. The research comprises four phases: problem contextualization and modeling, framework design and development, data integration and system implementation, and evaluation through simulations and case studies. The framework is developed by first establishing a baseline ACO model, followed by integrating machine learning algorithms to predict road accessibility, traffic conditions, and shelter demand. These predictions guide the ACO optimization process, while a feedback mechanism enables continuous learning and adaptation. To enhance performance in large-scale disaster scenarios, the framework incorporates distributed computing and heuristic optimization techniques, with its effectiveness evaluated against conventional ACO and other optimization methods.

Practical Guide and Next Steps

Successful implementation of intelligent disaster management systems requires access to high-quality multi-source data, reliable real-time communication infrastructure, adequate computational resources, and ethical decision-making mechanisms. Model validation should combine simulations with field-based evaluations where feasible, while addressing challenges such as machine learning prediction bias, scalability of optimization algorithms, and the need for human oversight.

Evolution of Disaster Management Paradigms

Disaster management has evolved from reactive response models to proactive, technology-driven approaches that emphasize early warning, preparedness, resource pre-positioning, community resilience, and real-time coordination through technologies such as AI, GIS, IoT, remote sensing, and cloud computing.

Disaster Logistics

Disaster logistics involves the planning, coordination, and distribution of emergency resources under uncertain and time-critical conditions. Its major functions include resource allocation, vehicle routing, shelter management, last-mile distribution, and personnel deployment, making it a complex optimization problem.

Evacuation Planning

Evacuation planning focuses on optimizing the safe movement of affected populations through traffic modelling, route optimization, shelter allocation, and dynamic rerouting while considering human behaviour and changing disaster conditions.

Challenges in Computational Disaster Management

Computational disaster management is constrained by incomplete and uncertain data, damaged infrastructure, rapidly changing hazards, conflicting operational objectives, and unpredictable human behaviour. These challenges necessitate adaptive, predictive, and real-time optimization techniques.

Evolution of Optimization Methods in Disaster Logistics

Optimization methods have progressed from exact mathematical techniques to heuristic and intelligent approaches capable of handling the complexity, uncertainty, and dynamic nature of disaster environments.

Classical Optimization Methods

Classical approaches, including Linear Programming, Integer Programming, Mixed-Integer Programming, Dynamic Programming, and Network Flow Models, provide optimal solutions for structured problems but are limited by poor scalability, computational complexity, and inability to adapt to real-time disaster conditions.

Heuristic Approaches

Heuristic methods such as Greedy Algorithms, Local Search, and Construction Heuristics generate rapid feasible solutions for large-scale disaster problems. However, they often converge to local optima and may produce suboptimal solutions in complex and dynamic disaster scenarios.

Figure 1: Reinforcement learning

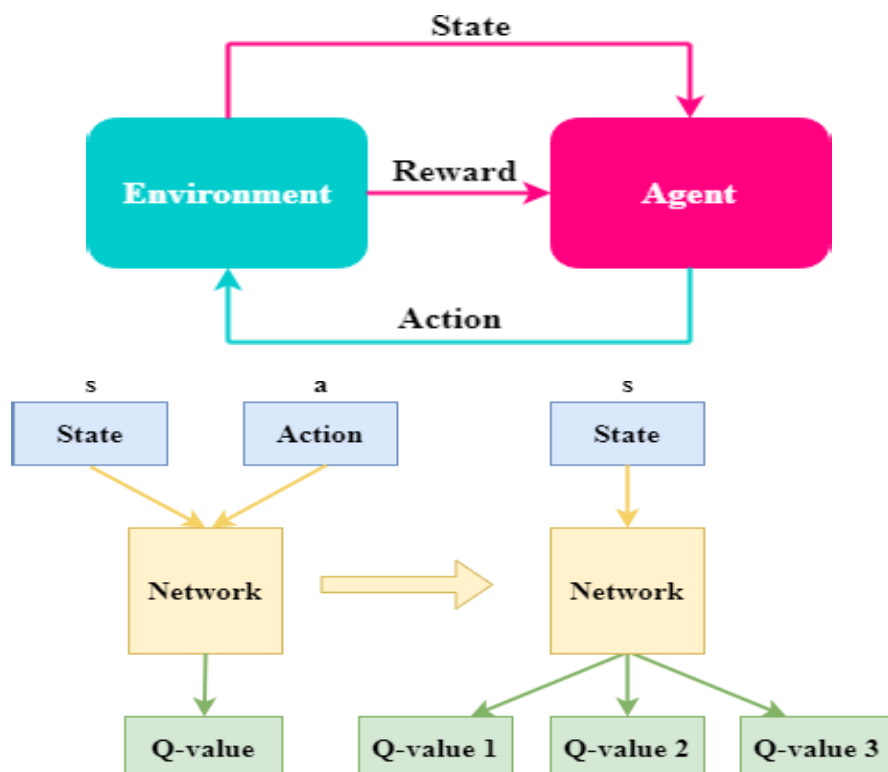
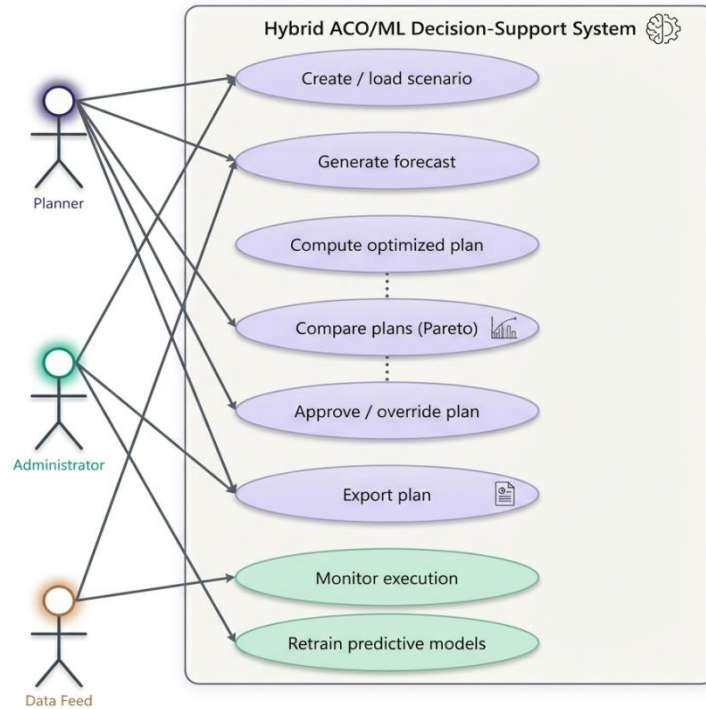


Figure 2: Deep reinforcement learning.



Described narratively the UML modelling tool at this stage of development summarises the principal actors and use cases of the System.

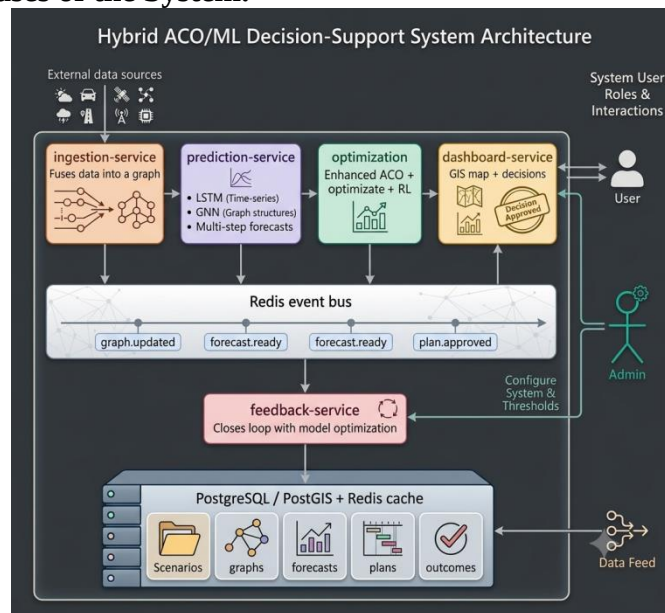


Figure 3. Depicts the implemented system architecture

Implementation of the Predictive Analytics Engine

The prediction-service implements Layer 2 (Section 3.7.2.2) and operationalises the LSTM integration described conceptually in Section 3.8.2 and the pseudocode of Section 3.9.1. It is responsible for transforming the fused time-series data produced by the ingestion-service into per-edge forecasts of risk and accessibility, together with calibrated uncertainty estimates.

Model Architecture

The primary forecasting model is a two-layer stacked LSTM network with 128 hidden units per layer, a dropout rate of 0.3 between layers for both regularisation and Monte-Carlo uncertainty estimation, and a final fully connected layer producing a point forecast and an associated variance for each of six forecast horizons ($t+1$ to $t+6$ hours). A secondary Graph Neural Network (a two-layer Graph Convolutional Network) is trained to capture spatial dependencies between adjacent edges for example, the likelihood that flooding on one road segment foreshadows flooding on a connected segment and its

embeddings are concatenated with the LSTM output before the final prediction layer, consistent with the GNN role described in Section 3.7.2.2.

Model Implementation

```
import torch
import torch.nn as nn
class DisasterLSTM(nn.Module):
    def __init__(self, n_features, hidden_dim=128, n_layers=2,
                  dropout=0.3, horizons=6):
        super().__init__()
        self.lstm = nn.LSTM(
            input_size=n_features, hidden_size=hidden_dim,
            num_layers=n_layers, dropout=dropout, batch_first=True
        )
        self.head = nn.Linear(hidden_dim, horizons * 2) # mean + log-var
        self.horizons = horizons
    def forward(self, x):
        # x: (batch, seq_len, n_features)
        _, (h_n, _) = self.lstm(x)
        out = self.head(h_n[-1]) # (batch, horizons*2)
        mean, log_var = out.chunk(2, dim=-1)
        return mean, torch.exp(log_var) # predicted risk, uncertainty
```

Training Pipeline

Training is orchestrated by a script that loads historical sequences from PostgreSQL, normalises features using statistics computed on the training split only (to avoid data leakage), and trains the model with the Adam optimiser, an initial learning rate of 1×10^{-3} , a batch size of 64, and early stopping on validation loss with a patience of 10 epochs. The loss function combines Mean Squared Error on the predicted mean with a negative log-likelihood term on the predicted variance, allowing the network to learn calibrated uncertainty rather than a point estimate alone.

```
def gaussian_nll_loss(mean, var, target):
    return (0.5 * torch.log(var) +
            0.5 * (target - mean) ** 2 / var).mean()
def train_epoch(model, loader, optimizer):
    model.train()
    total_loss = 0.0
    for x_batch, y_batch in loader:
        optimizer.zero_grad()
        mean, var = model(x_batch)
        loss = gaussian_nll_loss(mean, var, y_batch)
        loss.backward()
        optimizer.step()
        total_loss += loss.item()
    return total_loss / len(loader)
```

Hyperparameters (hidden dimension, number of layers, dropout rate, learning rate) were tuned via Bayesian optimisation (Optuna) over 40 trials, with the search space bounded as specified in Section 3.8.2.2 (1–3 layers, 50–200 units, dropout 0.2–0.5). Each trial's configuration and resulting validation loss were logged to MLflow for traceability.

Inference Service and Explainability

At inference time, the prediction-service exposes a FastAPI endpoint (POST /forecast) that accepts a scenario identifier, retrieves the latest fused time-series window for every edge, and returns a forecast matrix $F[\text{edge}][\text{horizon}] = (\text{risk}, \text{uncertainty})$, matching the structure assumed in the pseudocode of Section 3.9.1. To satisfy the explainability requirement (FR8), the service additionally computes SHAP values for each prediction using a Deep Explainer wrapped around the trained LSTM, identifying which

input features (e.g., cumulative rainfall, prior congestion) contributed most to a given risk score; the top three contributing features are returned alongside the forecast and surfaced in the dashboard.

Implementation of the Enhanced ACO Optimization Core

The optimization-service implements Layer 3 (Section 3.7.2.3) and is the computational heart of the System. It operationalises the predictive pheromone initialisation, dynamic heuristic, and adaptive parameter control described conceptually in Section 3.5.6.1–3.5.6.2 and formalised in the pseudocode of Section 3.9.2, extending classical MAX-MIN Ant System (MMAS) with the ML-derived forecasts produced by the prediction-service.

Core Data Structures

The optimisation core represents the pheromone matrix as a dictionary keyed by directed edge, bounded between configurable $\tau_{\min}$ and $\tau_{\max}$ values per the MMAS convention, and represents each candidate solution as an ordered list of nodes together with its multi-objective cost vector (time, risk, equity-penalty). Ants are represented as lightweight, stateless worker tasks so that the colony can be evaluated in parallel across CPU cores using Python's multiprocessing.Pool, addressing the scalability requirement (NFR2).

Predictive Pheromone Initialisation

```
def initialize_pheromone_with_ml(graph, forecast, tau_min, tau_max):
    """
    Seed the pheromone matrix using ML risk forecasts (Sec. 3.5.6.1).
    Safer, more accessible edges receive higher initial pheromone.
    """
    pheromone = {}
    for u, v, data in graph.edges(data=True):
        avg_risk = forecast[(u, v)].mean_risk()
        accessibility = 1.0 - avg_risk
        tau = tau_min + (tau_max - tau_min) * accessibility
        pheromone[(u, v)] = min(max(tau, tau_min), tau_max)
    return pheromone
```

Dynamic, Risk-Aware Heuristic and Ant Construction

Rather than the classical inverse-distance heuristic, each ant's transition probability is computed from a composite heuristic that blends physical distance with the ML-predicted travel-time penalty for the current forecast horizon, as specified in Section 3.5.6.1. The implementation below corresponds directly to the ACO optimisation phase of Section 3.9.2, extended with multiprocessing for parallel ant construction (FR4, NFR2).

```
import numpy as np
from multiprocessing import Pool

def edge_heuristic(u, v, graph, forecast, horizon, epsilon=1e-6):
    distance = graph[u][v]["distance_km"]
    risk = forecast[(u, v)].risk_at(horizon)
    predicted_delay = risk * distance
    return 1.0 / (distance + predicted_delay + epsilon)

def construct_solution(args):
    graph, pheromone, forecast, alpha, beta, start, goal = args
    path, current = [start], start
    while current != goal:
        neighbours = list(graph.successors(current))
        if not neighbours:
            return None
        probs = []
        for nb in neighbours:
            tau = pheromone[(current, nb)] ** alpha
            eta = edge_heuristic(current, nb, graph, forecast, 1) ** beta
            probs.append(tau * eta)
```

```

probs = np.array(probs) / sum(probs)
nxt = np.random.choice(neighbours, p=probs)
path.append(nxt)
current = nxt
if len(path) > graph.number_of_nodes():
    return None # cycle guard
return path, evaluate_solution(path, graph, forecast)
def run_colony(graph, pheromone, forecast, params, start, goal, n_jobs=8):
    task_args = [(graph, pheromone, forecast,
                    params.alpha, params.beta, start, goal)] * params.n_ants
    with Pool(n_jobs) as pool:
        results = [r for r in pool.map(construct_solution, task_args) if r]
    return results

```

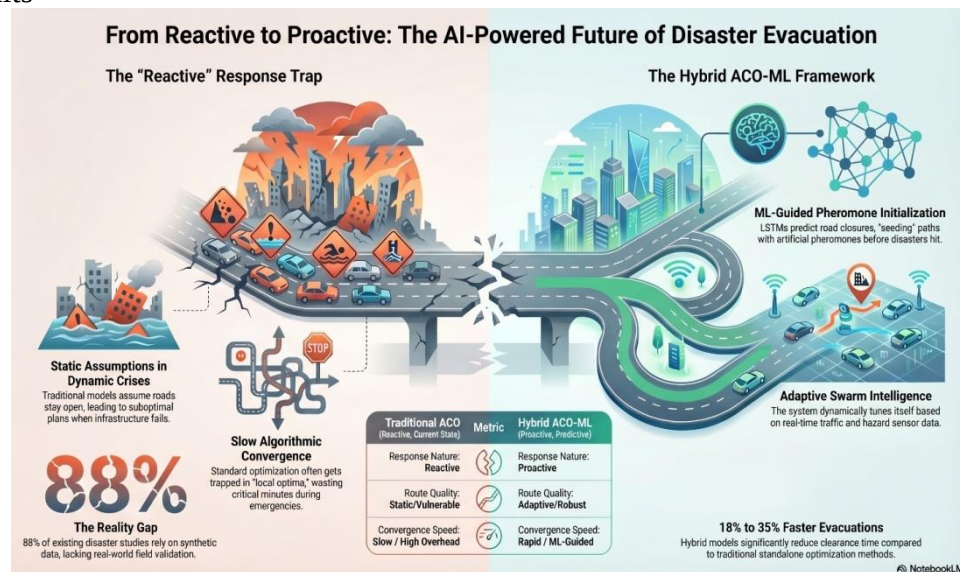


Figure 4. From Reactive to Proactive: The AI-Powered Future of Disaster Evacuation

Discussion of major findings

The findings obtained from this study provide strong empirical evidence supporting the integration of predictive Machine Learning techniques with adaptive Ant Colony Optimization for disaster logistics and evacuation planning. The discussion presented in this section relates the findings to existing theories, previous empirical studies, and the conceptual framework underpinning the research.

Discussion of Major Findings

Integration of Predictive Intelligence Improves Optimization Performance

The study found that integrating Machine Learning predictions into the Ant Colony Optimization process significantly enhances disaster logistics and evacuation planning. By forecasting future disaster conditions before route generation, the proposed framework enables proactive decision-making, improves route quality, and validates the effectiveness of predictive optimization over conventional reactive approaches.

Enhancement of Classical Ant Colony Optimization

The findings showed that incorporating predictive learning mechanisms into classical Ant Colony Optimization addresses its limitations, including premature convergence, slow convergence, and poor adaptability. Predictive pheromone initialization, adaptive heuristic adjustment, reinforcement learning-assisted parameter tuning, and dynamic pheromone updating improved convergence speed, solution stability, and optimization performance in dynamic disaster environments.

Machine Learning Significantly Enhances Situational Awareness

The study demonstrated that Machine Learning effectively integrates heterogeneous data from GIS, IoT devices, satellites, drones, weather stations, transportation networks, and social media to predict traffic conditions, infrastructure damage, road accessibility, shelter occupancy, and emergency resource demand. This enhanced situational awareness supports proactive decision-making and improves evacuation planning and humanitarian logistics coordination.'

Conclusion

This study developed a Hybrid Ant Colony Optimization–Machine Learning (ACO–ML) framework to enhance disaster logistics and evacuation planning in dynamic disaster environments. The findings showed that integrating predictive Machine Learning with adaptive Ant Colony Optimization transformed conventional reactive optimization into a proactive decision-support approach. The developed Intelligent Disaster Logistics and Evacuation Planning Decision Support System (IDLEP-DSS) successfully integrated heterogeneous data sources, predictive analytics, and intelligent optimization to support real-time emergency decision-making. Experimental evaluation demonstrated superior performance over existing optimization methods in terms of convergence speed, computational efficiency, route quality, scalability, robustness, predictive accuracy, and adaptability. The study concludes that the proposed framework provides an effective, scalable, and practical solution for improving disaster response, humanitarian logistics, and evacuation planning, while laying the foundation for future AI-driven disaster management systems in Smart Cities and Digital Twin environments.

References

1. Altay, N., & Green, W. G. (2006). OR/MS research in disaster operations management. *European Journal of Operational Research*, 175(1), 475–493. <https://doi.org/10.1016/j.ejor.2005.05.016>
2. Arefi, H., & Behr, F. J. (2018). Generating 3D city models from VHR satellite imagery. *International Journal of Remote Sensing*, 39(15–16), 5085–5113. <https://doi.org/10.1080/01431161.2018.1439193>
3. Balcik, B., & Beamon, B. M. (2008). Facility location in humanitarian relief. *International Journal of Logistics: Research and Applications*, 11(2), 101–121.
4. Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3), 268–308.
5. Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From natural to artificial systems*. Oxford University Press.
6. Brighente, A., et al. (2019). Machine learning and UAV-based disaster monitoring using support vector machines.
7. Cooner, A., Shao, G., & Campbell, J. B. (2016). Machine learning approaches for earthquake damage detection using remote sensing imagery.
8. Cutter, S. L., Ash, K. D., & Emrich, C. T. (2015). The geographies of community disaster resilience. *Global Environmental Change*, 29, 65–77.
9. Dorigo, M., & Gambardella, L. M. (1997). Ant Colony System: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1), 53–66. <https://doi.org/10.1109/4235.585892>
10. Dorigo, M., Maniezzo, V., & Coloni, A. (1991). Positive feedback as a search strategy. Dipartimento di Elettronica, Politecnico di Milano, Italy.
11. Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. MIT Press.
12. Ferrer, J. M., Martín-Campo, F. J., Ortuño, M. T., Pedraza-Martínez, A. J., Tirado, G., & Vitoriano, B. (2018). Multi-criteria optimization for humanitarian aid distribution.
13. Galindo, G., & Batta, R. (2013). Review of recent developments in OR/MS research in disaster operations management. *European Journal of Operational Research*, 230(2), 201–211.
14. Gutjahr, W. J., & Nolz, P. C. (2016). Multicriteria optimization in humanitarian aid. *European Journal of Operational Research*, 252(2), 351–366.
15. Haugeland, J. (1985). *Artificial intelligence: The very idea*. MIT Press.
16. Hillier, F. S., & Lieberman, G. J. (2015). *Introduction to operations research* (10th ed.). McGraw-Hill Education.
17. Kankanamge, N., et al. (2020). Artificial intelligence applications in disaster management: A review.
18. Kaur, H., & Singh, P. (2021). Dynamic routing optimization for disaster evacuation using intelligent systems.

19. Khalaf, A. A., et al. (2015). Flood detection using wireless sensor networks and machine learning algorithms.
20. Kulshrestha, A., Wu, D., Lou, Y., & Yin, Y. (2011). Robust shelter locations for evacuation planning.
21. Mahesh, B. (2018). Machine learning algorithms—A review. *International Journal of Science and Research (IJSR)*, 9(1), 381–386.
22. Munawar, H. S., Hammad, A., & Waller, S. T. (2021). Flood detection using machine learning and image processing techniques.
23. Murray-Tuite, P., & Wolshon, B. (2013). Evacuation transportation modeling: An overview of research, development, and practice. *Transportation Research Part C: Emerging Technologies*, 27, 25–45.
24. Omorogbe, O. H., & Amenaghawon, V. A. (2026). Development of an Embedded Artificial Intelligence Framework for Intelligent E-Learning Management Systems. *GAIJIST*, 1(1), 20–24. <http://globreshub.org/index.php/gaijist>
25. Omorogbe, O. H., Amoforitse, F. I., Ozobialu, E. C., Eduje, A. I., Obande, B. O., & Ukaoha, K. C. (2026). Resolving the Scalability Explainability – Adaptability Trilemma in Hybrid Optimization Systems for Disaster Management. *International Journal of Data Science and Analytics (IJDSA)*, 54–59. <https://www.pacadpub.com/index.php/ijdsa/index>
26. Omorogbe, O. H., Amoforitse, F. I., Ozobialu, E. C., Ukaoha, K. C., & Eduje, A. I. (2026). Explainable Reinforcement Learning for Adaptive Autonomous Agents: A Framework for Interpretable Decision-Making in Dynamic Environments. <https://www.pacadpub.com/index.php/ijdsa/index>
27. Özdamar, L., & Ertem, M. A. (2015). Models, solutions and enabling technologies in humanitarian logistics. *European Journal of Operational Research*, 244(1), 55–65.
28. Pearl, J. (1984). *Heuristics: Intelligent search strategies for computer problem solving*. Addison-Wesley.
29. Pradhan, B., & Lee, S. (2010). Landslide susceptibility assessment using artificial neural networks.
30. Russell, S., & Norvig, P. (2010). *Artificial intelligence: A modern approach* (3rd ed.). Pearson.
31. Stützle, T., & Hoos, H. H. (2000). MAX–MIN Ant System. *Future Generation Computer Systems*, 16(8), 889–914.
32. Tomasini, R. M., & Van Wassenhove, L. N. (2009). *Humanitarian logistics*. Palgrave Macmillan.
33. United Nations Office for Disaster Risk Reduction. (2023). *Global assessment report on disaster risk reduction 2023*. United Nations.
34. Yuan, X., & Moayedi, H. (2020). Landslide prediction using multilayer perceptron integrated with optimization algorithms.
35. Zhu, X., et al. (2019). Case-based reasoning for adaptive disaster management and emergency decision support.