WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

QuantumVision: Intelligent Disaster Navigation and Emergency Evacuation Management

Dr. V. Srinadh*, S. Sri Kavya, V. Sai, V. Guna Sekhar Reddy, K. Nandhu, G. Ganesh

*Supervisor, Department CSE – AI & ML, GMR Institute of Technology, Andhra Pradesh, India.
Students, Department CSE – AI & ML, GMR Institute of Technology, Andhra Pradesh, India.

ABSTRACT

Floods are among the most damaging hazards in coastal India. Timely estimates of rainfall, inundation and road accessibility can determine whether people reach safety. This paper presents QuantumVision, a decision-support framework integrating remote sensing, rainfall forecasting, classical and quantum machine learning, GIS analysis and flood-aware routing for Srikakulam district, Andhra Pradesh. Sentinel-1 radar backscatter, Copernicus DEM variables, multi-year station rainfall and reference flood records form the feature dataset for a 24-hour rainfall forecaster and binary flood classifier. Random Forest and XGBoost are compared with four-qubit Variational Quantum Classifier (VQC) and Quantum Neural Network (QNN) circuits using a PennyLane state-vector simulator. On a spatially separated test set of 982 samples, both tree ensembles achieved 99.80% accuracy and ROC-AUC above 0.999; VQC and QNN achieved 89.61% and 90.12% accuracy, with ROC-AUC values of 0.973 and 0.955. XGBoost gave the lowest rainfall error (MAE 4.115 mm; RMSE 8.552 mm). Predicted risk is overlaid on an OpenStreetMap road graph, where Dijkstra and A* use distance, flood and accessibility penalties. A QUBO model with simulated annealing supports selected routing and facility-allocation tasks. A React 19/FastAPI dashboard provides maps, alerts and reports. The study compares simulator-based quantum models with classical baselines and demonstrates an integrated prediction-to-navigation workflow for flood preparedness.

Keywords: Flood-Risk Assessment, Quantum Machine Learning, Sentinel-1 SAR, Digital Elevation Model, GIS, Rainfall Forecasting, Variational Quantum Classifier, Evacuation Routing, A* and Dijkstra Algorithms, Disaster Management.

1. INTRODUCTION

1.1 Background

Flooding results from hydrological triggers such as intense or prolonged rainfall interacting with terrain, drainage and land use. Its effects include damaged homes, disrupted transport and delayed medical care. Rapid urbanization, changing rainfall and settlement near rivers and coasts increase exposure. During heavy-rain events, responders need forecasts, likely inundation locations and usable routes to shelters or hospitals; most operational tools address only one of these needs.

1.2 Study Area and Motivation

Srikakulam, a coastal district in north-eastern Andhra Pradesh, is crossed by the Nagavali and Vamsadhara rivers, which flow through low-lying deltaic land to the Bay of Bengal. Monsoon rainfall and cyclone-driven storms expose the district to flooding. Low relief, drainage channels and settlements near waterways make it a relevant study area. Because monsoon cloud cover limits optical imagery, cloud-penetrating synthetic aperture radar (SAR) is useful for flood observation.

1.3 Problem Statement

Flood research often focuses on a single task, such as hazard mapping or damage assessment, while routing studies may assume static roads. Evacuation advice that ignores inundation can direct people onto flooded streets; flood maps without road-network analysis leave route selection to users. Quantum machine-learning studies also need stronger comparisons with classical baselines on identical geospatial data, and simulator results must not be overstated.

1.4 Objectives

The study has six objectives: (i) assemble district-scale Sentinel-1 SAR, Copernicus DEM, historical rainfall, flood-reference and OpenStreetMap data; (ii) forecast next-24-hour cumulative rainfall; (iii) compare Random Forest, XGBoost, VQC and QNN under identical partitions; (iv) map risk with roads,

waterways, shelters and hospitals; (v) compute flood-aware Dijkstra/A* routes and apply QUBO to selected allocation tasks; and (vi) deliver the modules through a web application and evaluate performance and limitations.

1.5 Contributions

- The framework contributes an integrated pipeline linking rainfall forecasts, SAR/terrain classification, spatial risk layers and graph routing; a controlled comparison of two tree ensembles and two four-qubit variational circuits using spatially grouped validation; and a district dashboard for forecasts, maps, simulation, alerts and reports.
- A controlled comparison of two tree ensembles and two four-qubit variational circuits using spatially grouped cross-validation to limit leakage between neighbouring samples.
- An operational dashboard offering forecasts, maps, simulation, alerts and reports for the study district.

2. LITERATURE SURVEY

Fifteen studies published between 2024 and 2026 form the core survey. They are reviewed by purpose, data, method and limitations, and grouped thematically.

2.1 Artificial Intelligence Across the Disaster-Management Cycle

Muthukumar et al. [1] reviewed AI frameworks across disaster preparedness, response and recovery, noting that real-deployment evidence remains limited. Whyte and Haupt [2] examined organizational readiness and advocated a “centaur” model in which AI supports responders. Dcruz et al. [7] proposed a structured decision scheme using enabling agents, decision levels and scenarios, reporting more consistent behaviour than unaided operators in tested situations. Visave [14] emphasized transparency, trust and accountability. These studies motivate showing data provenance and treating system outputs as decision support, not instructions.

2.3 Imagery, Damage Assessment and Detection

Mabina et al. [3] reviewed 162 studies on deep learning and geospatial post-flood damage assessment, noting that cascading hazards are rarely modelled together. Lagap et al. [9] used explainable AI to evaluate damage-assessment networks and showed that high accuracy can conceal poor severe-damage performance. Song et al. [11] studied landslide detection from remotely sensed images, while Yang et al. [15] addressed real-time urban fire detection. Image analysis can support response systems, but each hazard-detection module requires independent validation.

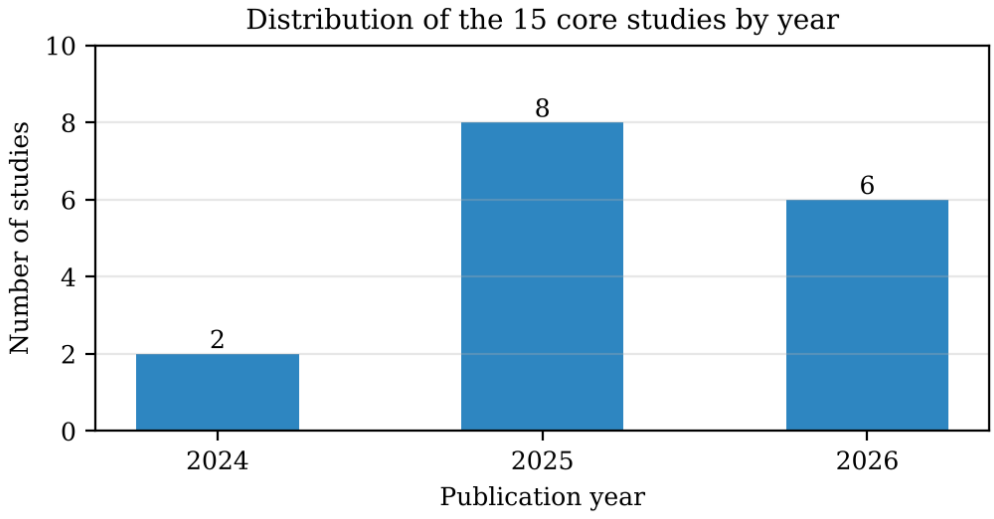


Figure 1: Distribution of the fifteen core studies by publication year

2.4 Research Gaps

Reading the studies together suggests the following gaps, which define the contribution of this work:

- Most studies address a single hazard or task; hazard detection is seldom linked to evacuation planning in the same system.
- Risk maps are often static, so routing built on them cannot react to changing rainfall or inundation.

- Fusing satellite, rainfall, terrain, road and facility data into one workflow remains difficult.

3. METHODOLOGY

3.1 Overall Framework

QuantumVision links flood-management stages (Figure 3): satellite, terrain, rainfall and infrastructure data are cleaned and aligned; features are derived; rainfall and flood models produce predictions; outputs become spatial layers; routes are computed on the road graph; and results are delivered through a web interface. Data coverage and provenance accompany outputs so predictions are not confused with observations.



Figure 3: Workflow overview of the QuantumVision flood-disaster prediction and evacuation-management framework

Inputs comprise SAR and DEM rasters, rainfall time series, reference records, boundary/road/facility vectors, and a user-selected origin and destination. Outputs include cleaned spatial layers, feature tables, forecasts, flood classes and probabilities, risk zones, edge costs and routes, presented as maps, charts and tables.

3.2 Preprocessing and Feature Extraction

3.2.1 Cleaning and Spatial Alignment

Duplicate or invalid records are removed, date/time formats standardized and coordinates validated. Missing rainfall is not replaced by zero, since missing observations differ from dry periods. GeoPandas and PyProj align all layers to a common coordinate reference system and clip them to the district boundary before overlays or distance calculations.

3.2.2 SAR Processing

SAR scene selection checks acquisition dates and polarization. Orbit information is applied, followed by radiometric calibration, speckle filtering and terrain correction. Backscatter is expressed in decibels and clipped to the study area; VV, VH, their relationship and pre-/post-event temporal differences are extracted.

3.2.3 Terrain Processing

The DEM is clipped, checked for gaps and used to compute slope from the local gradient of the elevation surface:

$$Slope = tan^{-1}(\sqrt{(\frac{\partial z}{\partial x})^2 + (\frac{\partial z}{\partial y})^2}) \quad (1)$$

where z is elevation. Slope is stored in degrees and elevation in metres. Distance to the nearest waterway is calculated from the vector drainage layer.

3.2.4 Feature Set and Scaling

Table 3: Features supplied to the flood classifiers

Feature	Unit	Description
sar_vv_db	dB	VV-polarization backscatter
sar_vh_db	dB	VH-polarization backscatter
sar_vv_vh_ratio	–	Defined relationship between VV and VH
sar_temporal_diff_db	dB	Change in backscatter between acquisitions
elevation_m	m	Terrain elevation from Copernicus DEM
slope_deg	°	Terrain slope derived from the DEM
distance_to_waterway_km	km	Distance to the nearest river or drain
rainfall_accum_24h_mm	mm	Rainfall accumulated over 24 hours
rainfall_intensity_mm_hr	mm/h	Rainfall intensity

Numerical features are rescaled with min–max normalization, which is especially important for the quantum circuits because the values are used as rotation angles:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2)$$

Minima and maxima are calculated from training data only, then applied unchanged to validation and test subsets to prevent leakage.

3.3 Twenty-Four-Hour Rainfall Forecasting

The target is cumulative rainfall over the next 24 hours. For hourly observations it is defined as the sum over the following 24 hours; other sampling intervals are handled accordingly.

$$Y(t) = R(t+1) + R(t+2) + \ldots + R(t+24) \quad (3)$$

Here, R denotes observed rainfall. Predictors include recent rainfall, 3-, 6-, 12- and 24-hour accumulations, intensity, available temperature and pressure, and seasonal indicators such as month and day of year.

A Random Forest regressor averages predictions from N trees trained on different samples and feature subsets:

$$\hat{Y} = \frac{1}{N}\sum_{i=1}^{N} T_i(X) \quad (4)$$

XGBoost builds trees sequentially, each correcting the ensemble residual; regularization controls model complexity:

$$Obj = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \quad (5)$$

Persistence, which repeats recent behaviour, is the baseline. Data are split chronologically (2021–2023 training; 2024–2025 testing), with a common target and test period. Each forecast stores location, generation time, horizon, predicted accumulation, model and data coverage; it enters flood assessment only when spatially and temporally aligned.

3.4 Flood Detection and Risk Classification

Each sample uses X = [VV, VH, SAR ratio, temporal difference, elevation, slope, waterway distance, 24-hour rainfall, rainfall intensity] and is labelled Class 0 or 1. Four classifiers are evaluated on identical partitions.

3.4.1 Random Forest and XGBoost Classifiers

Random Forest trains bootstrap-sampled trees and combines them by majority vote; the experiment uses 100 balanced trees. XGBoost adds trees sequentially and converts flood probability to a class using a threshold; positive-class weight 4.94 addresses the smaller flooded class.

3.4.2 Variational Quantum Classifier

A variational quantum classifier encodes the normalized features into the state of a small register and then applies trainable gates. Its state is written as:

$$|\psi(x,\theta)\rangle = U(\theta)\,U(x)\,|0\rangle \qquad (6)$$

The four-qubit circuit uses angle embedding and two variational layers (24 trainable parameters) with rotation and entangling gates. Measurement expectation gives the class score (Figure 4); a classical optimizer updates θ to minimize classification loss. Circuits run on PennyLane default.qubit state-vector simulation, not quantum hardware.

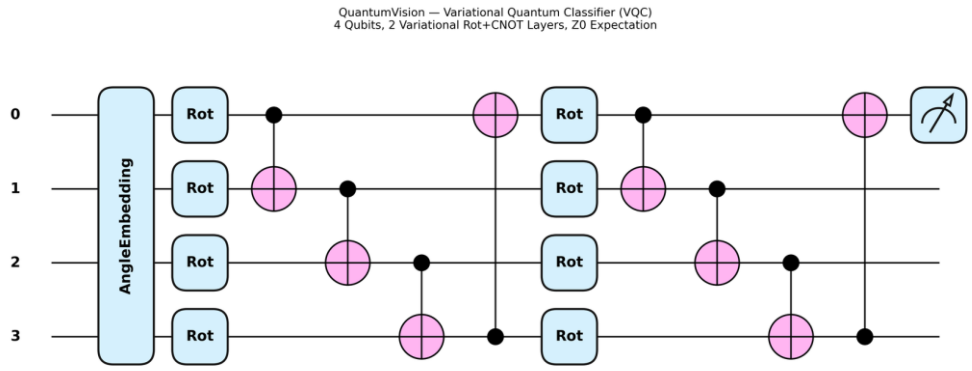


Figure 4: Four-qubit Variational Quantum Classifier circuit (angle embedding, two variational layers, measurement)

3.4.3 Quantum Neural Network

The QNN uses the same four-qubit register, with encoded inputs passing through trainable rotation and entanglement layers before measurement (Figure 5). The measured expectation M is mapped to class probability:

$$QNN(x,\theta) = M(\,U(\theta)\,U(x)\,|0\rangle\,) \qquad (7)$$

Training uses the Adam optimizer with a classification loss, and the measured outputs are passed through a classical post-processing step.

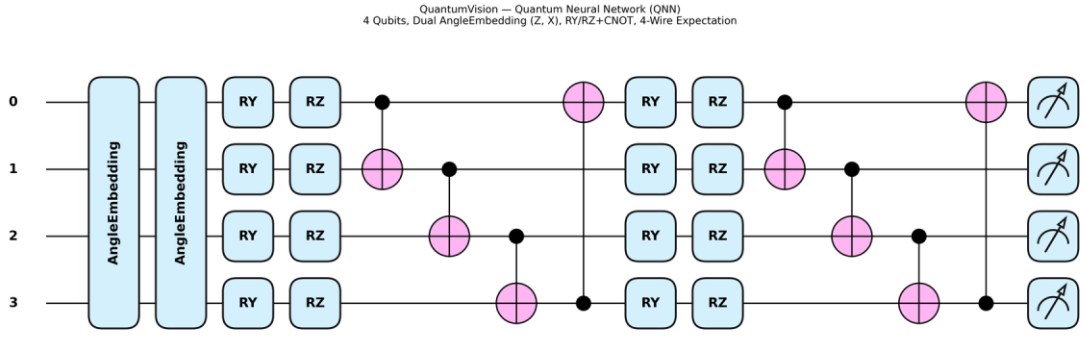


Figure 5: Four-qubit Quantum Neural Network circuit with encoding, variational rotations, entanglement and measurement

3.4.4 Risk Categories and Rainfall Linkage

The binary prediction is organized into four application-level categories (Low, Moderate, High and Severe) by a rule that combines the predicted flood probability with rainfall, elevation and slope:

$$Risk = f(P_{flood}, Rainfall, Elevation, Slope) \qquad (8)$$

The binary output is grouped into Low, Moderate, High and Severe categories using predicted flood probability, rainfall, elevation and slope. Thresholds are documented but are not claimed to be empirically validated beyond available evidence. Spatially and temporally aligned rainfall forecasts enter the classifier as rainfall features. Forecasts are not confirmed floods; risk outputs are labelled model-derived.

3.4.5 Classical–Quantum Comparison Protocol

All classifiers use the same labels, features and spatially grouped partitions. GroupKFold blocks of about 5.5 km prevent neighbouring correlated pixels from appearing in both training and test sets. Metrics are accuracy, precision, recall, F1, ROC-AUC, training time and inference latency.

3.5 GIS-Based Spatial Analysis

Predictions are joined to coordinates and converted to spatial layers. Terrain, rainfall and SAR observations identify low-lying areas; risk layers are overlaid with roads to flag intersecting segments, while shelters and hospitals are shown separately. GeoPandas, Rasterio, Shapely and PyProj support vector/raster processing, geometry operations and coordinate transformation. The 2-D map displays boundaries, risk, rainfall, terrain, roads, waterways, facilities and routes, reflecting actual observation coverage.

3.6 Flood-Aware Evacuation Routing

3.6.1 Graph Representation and Edge-Cost Model

The road network is a graph G = (V, E): nodes represent intersections or connected locations; edges represent road segments with length, class, connectivity and penalties. Edge cost is:

$$C(e) = D(e) + P_f(e) + P_a(e) \qquad (9)$$

D(e) is distance cost, Pf(e) the flood penalty and Pa(e) the accessibility penalty. Penalties use consistent distance units. Confirmed impassable segments receive a very high cost or are removed.

3.6.2 Dijkstra's Algorithm

Dijkstra finds a minimum-cost path for non-negative edge costs. The source is initialized to zero and other nodes to infinity; the algorithm repeatedly settles the lowest-cost unvisited node, relaxes its neighbours and stops at the destination or when no node remains. Path cost is the sum of edge costs:

$$C(P) = \sum_{e \in P} C(e) \quad (10)$$

3.6.3 A Algorithm*

A* search steers the exploration with a heuristic estimate of the remaining cost:

$$f(n) = g(n) + h(n) \qquad (11)$$

Here, g(n) is accumulated cost and h(n) estimates remaining cost. With an admissible heuristic, such as straight-line distance that does not exceed the true remaining cost, A* returns Dijkstra's optimal path while usually expanding fewer nodes (Figure 6).

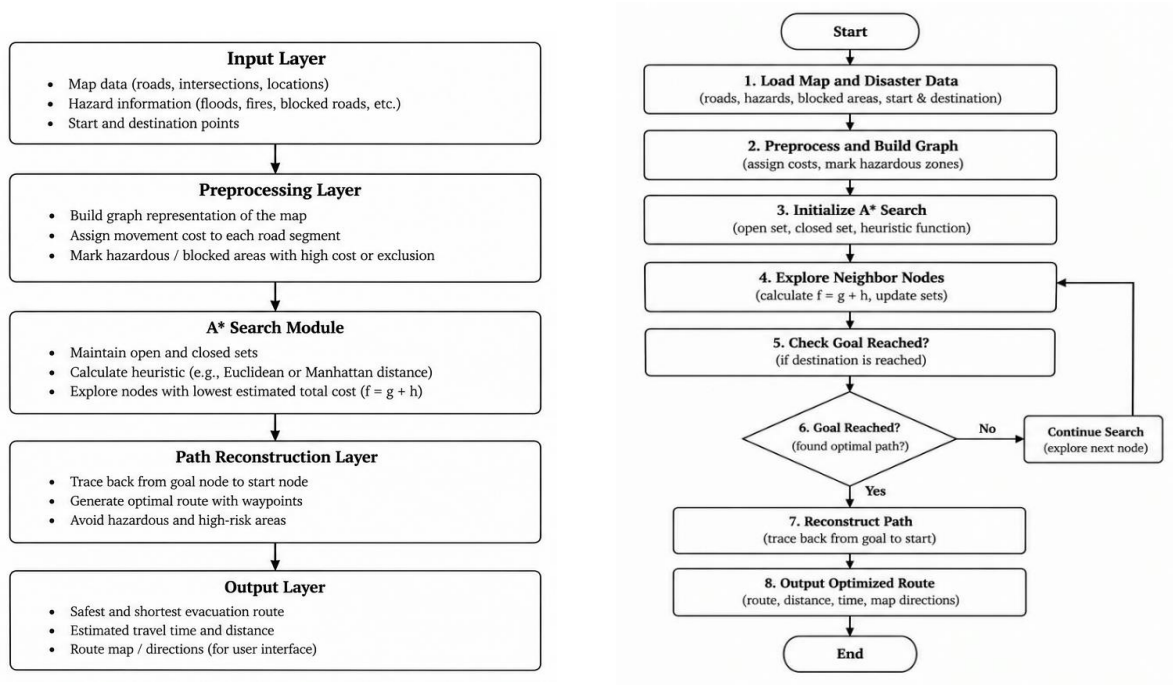


Figure 6: A route-optimization (a) layered architecture and (b) flowchart*

3.6.4 Routing Sequence and Dynamic Updates

Routing proceeds by loading the road graph and risk layer; finding road segments intersecting affected areas; assigning flood/accessibility penalties; updating edge costs; snapping origin and destination to the graph; running Dijkstra or A*; and displaying the route.

1. Load the processed road network and the flood-risk spatial layer.
2. Find road segments that intersect potentially affected regions.
3. Assign flood and accessibility penalties to those edges.
4. Update the edge costs, using consistent units.
5. Snap the user's start point and chosen destination to the graph.
6. Run Dijkstra or A* on the updated graph.
7. Return the route and draw it on the map.

Nearby shelters and hospitals are ranked by geographic or network distance and available information. New risk data or a changed origin/destination triggers cost updates and route recomputation. If no acceptable route exists, the application reports this rather than showing an unverified path.

3.7 Evaluation Metrics

Rainfall forecasts are judged by the mean absolute error, the root mean square error and the coefficient of determination:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (14)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (15)$$

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \quad (16)$$

Heavy-rain MAE, computed on samples with at least 20 mm of rainfall, is reported separately. Flood classifiers are evaluated using:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (17)$$

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \quad (18)$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (19)$$

ROC-AUC is also reported. Recall is important because a missed flooded location (false negative) is more serious for evacuation than a false alarm. Routing measures include distance, accumulated cost, unsafe-segment avoidance and computation time. End-to-end tests cover processing, forecasting, classification, layer generation, routing, facility lookup and frontend–backend communication.

3.8 Comparison with Existing Approaches

Table 4: Existing approaches versus the proposed framework

Aspect	Typical existing approaches	QuantumVision
Scope	A single task such as hazard mapping or detection	Forecast, classification, mapping, routing and alerts in one pipeline
Data	Single source or independent features	SAR, DEM, rainfall, labels, roads and facilities fused
Models	One family of learners	Random Forest, XGBoost, VQC and QNN on identical data
Risk maps	Static susceptibility layers	Risk layer feeds route costs and can be refreshed
Routing	Predefined or shortest-distance routes	Dijkstra and A* with flood and accessibility penalties
Optimization	Distance-based facility choice	QUBO with simulated annealing for selected tasks
Delivery	Offline analysis	React and FastAPI dashboard with alerts and reports
Reporting	Accuracy only	Recall, AUC, latency, leakage control and stated limitations

4. RESULTS AND DISCUSSION

4.1 Experimental Setup

Rainfall models used observations from 1 January 2021–31 December 2023 for training and 1 January 2024–31 December 2025 for testing, with 70,080 samples per period across eight stations. Flood classifiers used a spatially separated held-out set of 982 samples (820 non-flooded; 162 flooded). Quantum circuits ran on the PennyLane state-vector simulator. Dashboard figures are the values displayed by the implemented system for these runs.

4.2 Rainfall Forecasting Results

Table 5: Held-out chronological evaluation of 24-hour rainfall forecasting (2024–2025)

Model	MAE (mm)	RMSE (mm)	R^2	Heavy-rain MAE (mm)
Persistence (baseline)	4.803	10.806	−0.1486	22.796
Random Forest Regressor	4.192	8.710	0.2537	22.769
XGBoost Regressor (selected)	4.115	8.552	0.2806	22.791

Both learned models outperform persistence, whose negative R^2 explains less variance than predicting the mean. XGBoost has the lowest MAE/RMSE and highest R^2, so it was selected. Its explanatory power remains modest ($R^2 = 0.28$); heavy-rain MAE is about 22.8 mm for all models, indicating difficulty predicting extremes from rainfall history alone. The forecast is therefore a coarse indicator; numerical weather-model inputs may improve it.

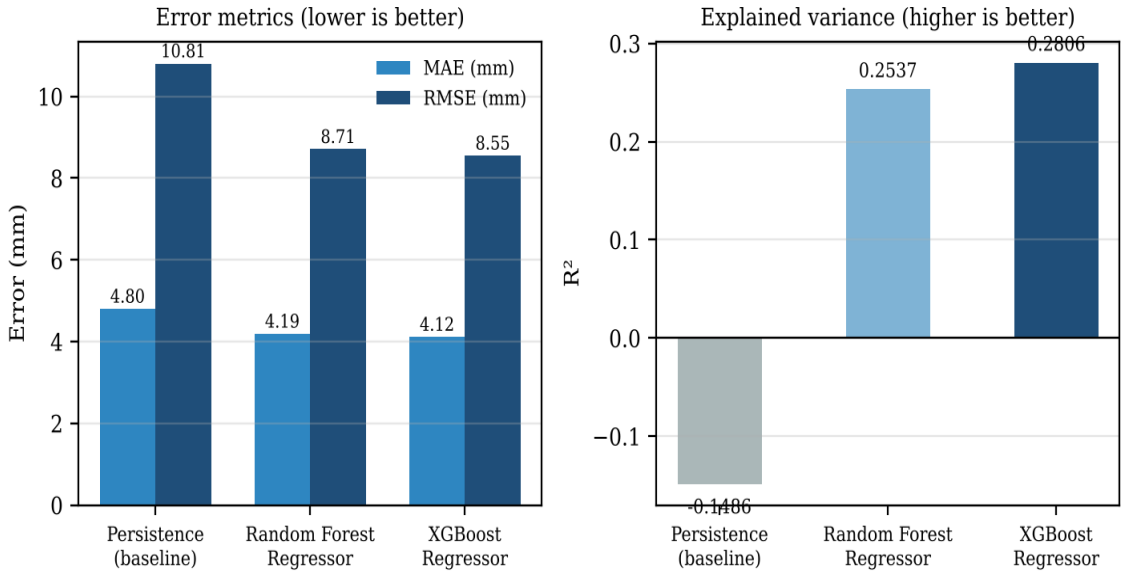


Figure 7: Rainfall-forecast errors and explained variance of the three models

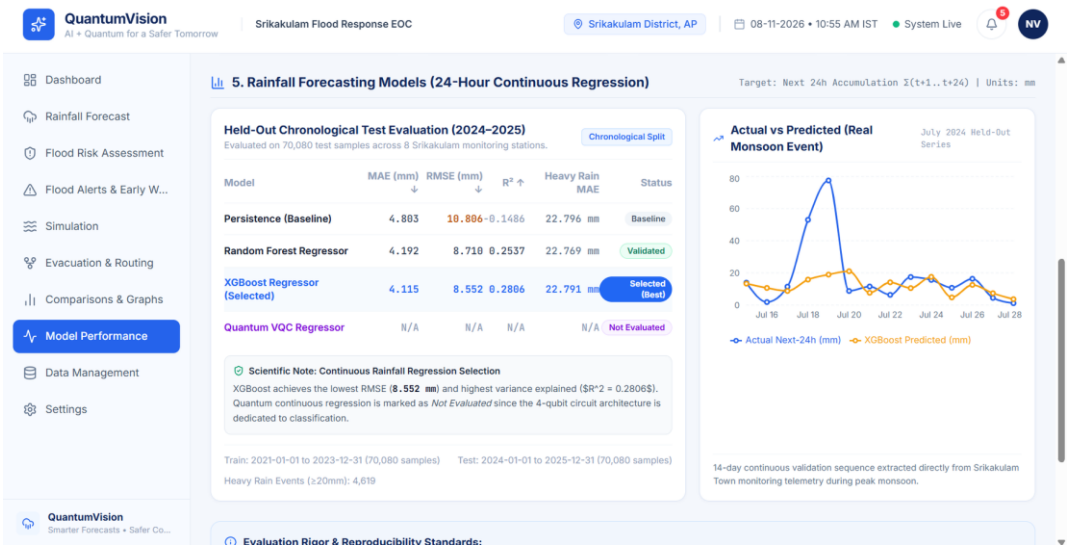


Figure 8: Rainfall-forecasting performance dashboard, including the actual-versus-predicted series for a July 2024 monsoon event

Figure 8 shows that the selected model follows general rainfall levels but underestimates the sharp peak around 18–19 July, consistent with the heavy-rain error in Table 5.

4.3 Flood Classification Results

Table 6: Classification performance on the held-out set (N = 982; 820 non-flooded, 162 flooded)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	ROC-AUC	Train time (s)	Latency (ms)
Random Forest	99.80	100.00	98.77	99.38	0.9998	2.6	0.007
XGBoost	99.80	100.00	98.77	99.38	0.9999	1.9	0.005
VQC (4 qubits)	89.61	62.30	93.83	74.88	0.9729	32.5	1.42
QNN (4 qubits)	90.12	64.84	87.65	74.54	0.9550	29.8	1.38

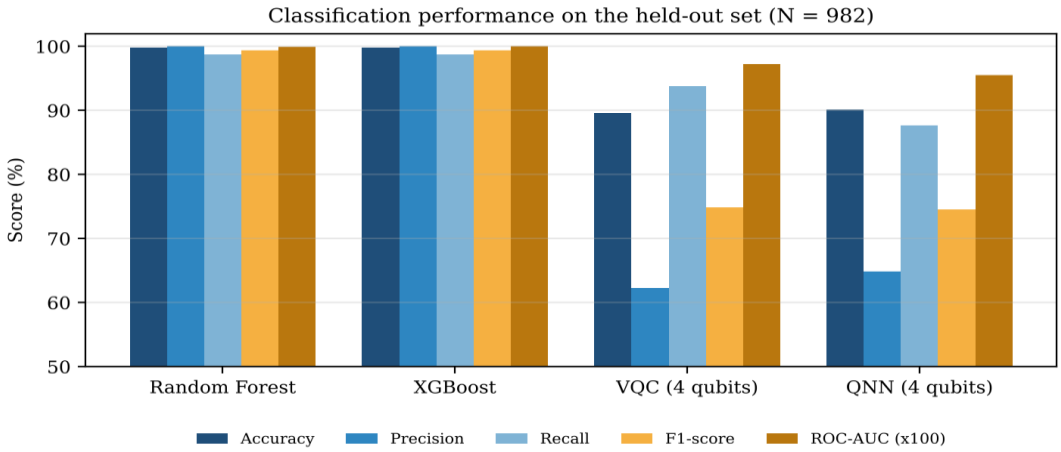


Figure 9: Comparison of Random Forest, XGBoost, VQC and QNN on the held-out set

Random Forest and XGBoost perform almost identically: each misses two of 162 flooded samples (1.23%) and produces no false alarms. XGBoost trains slightly faster and is the primary deployed model. VQC detects 152 flooded samples (recall 93.83%), the highest quantum-model recall, but precision is 62.30%, indicating many false alarms. QNN has 90.12% accuracy, 87.65% recall and 64.84% precision.

Approximate confusion-matrix counts are back-calculated from the reported recall, precision and test-set sizes; they make the error profile explicit.

Table 7: Confusion-matrix counts implied by the reported metrics

Model	**TP**	**FN**	**FP**	**TN**
Random Forest	160	2	0	820
XGBoost	160	2	0	820
VQC (4 qubits)	152	10	92	728
QNN (4 qubits)	142	20	77	743

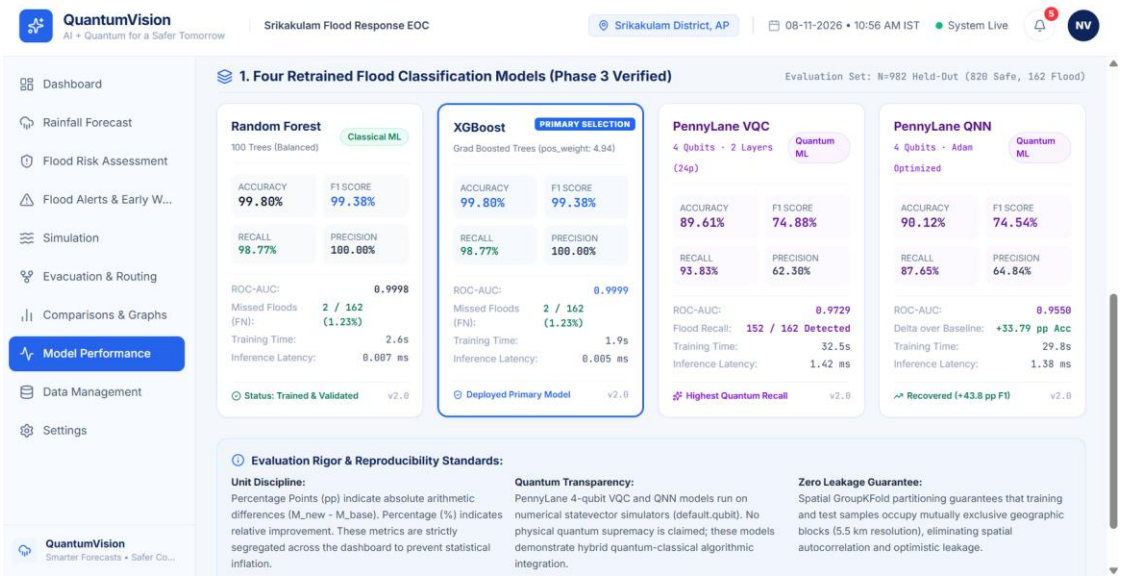


Figure 10: Model-performance dashboard of the system showing the four retrained flood classifiers

4.3.1 Interpreting the Classical–Quantum Gap

Tree ensembles use all nine features at full numerical resolution, while a four-qubit circuit compresses them into four rotation angles, losing information before training. The circuits have 24 trainable parameters and gradient-based optimization may encounter flat loss regions, limiting fit to imbalanced data. Classical simulation is slower: quantum training takes about 30 s and inference 1.4 ms, versus 2–3 s and a few microseconds for trees. VQC and QNN AUCs (0.973 and 0.955) still indicate useful ranking; threshold and class-imbalance effects account for much of the accuracy gap, suggesting threshold tuning and class weighting.

4.3.2 Why Near-Perfect Scores Need Cautious Reading

Although 99.80% accuracy and AUC near 1.0 are high, grouped geographic validation only addresses spatial autocorrelation. Labels derive from historical records and SAR water evidence, so SAR features may partly encode label information. The test set has only 162 flooded samples; two misses alter recall by over one percentage point. Validation on other events, years and districts is needed before operational claims.

4.4 Dashboards and GIS Outputs

The Overview dashboard (Figure 11) summarizes forecast, current risk, route status and data source, with links to the other modules.

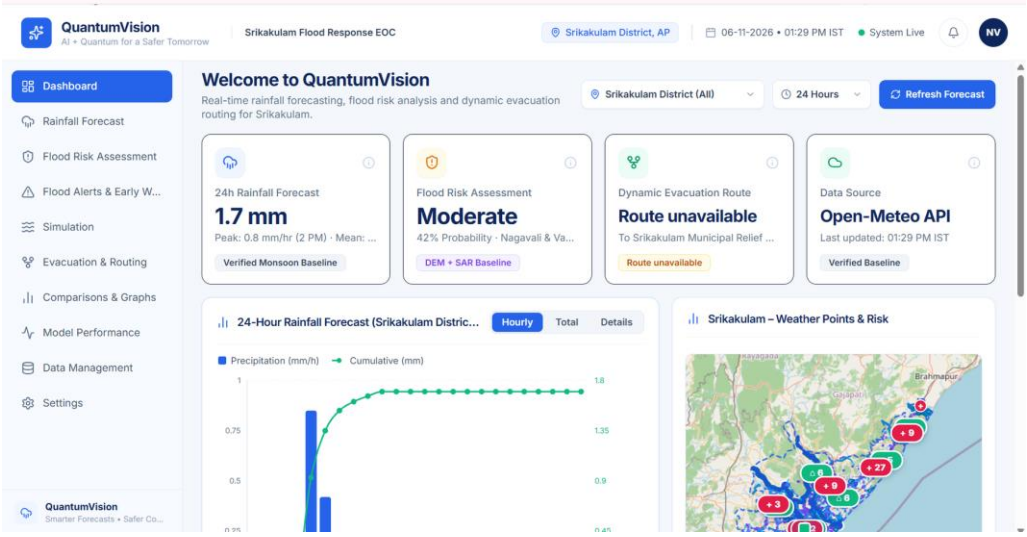


Figure 11: Main dashboard providing an overview of rainfall, flood risk and evacuation status

The Rainfall Forecast dashboard (Figure 12) allows location selection and displays 24-hour accumulation charts and weather-point markers. These values feed flood-risk assessment.

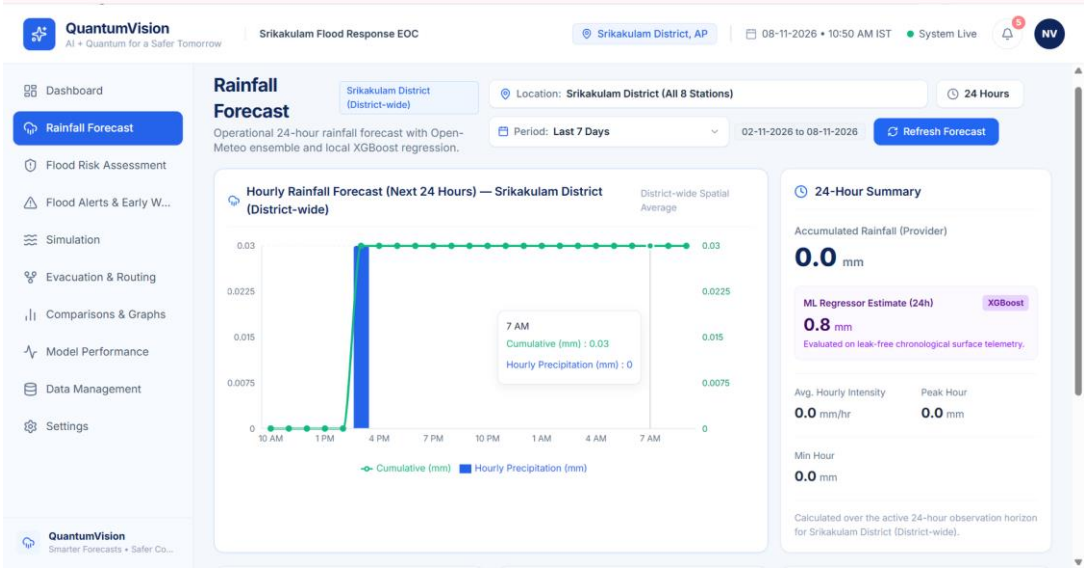


Figure 12: Rainfall forecast dashboard with 24-hour summary and location selection

The Flood Risk Assessment dashboard (Figure 13) displays model-derived risk score/category, contributing factors (including rainfall intensity, elevation and drainage proximity) and inundation mapping with hospital and shelter clusters. The example reports Low risk for the selected period.

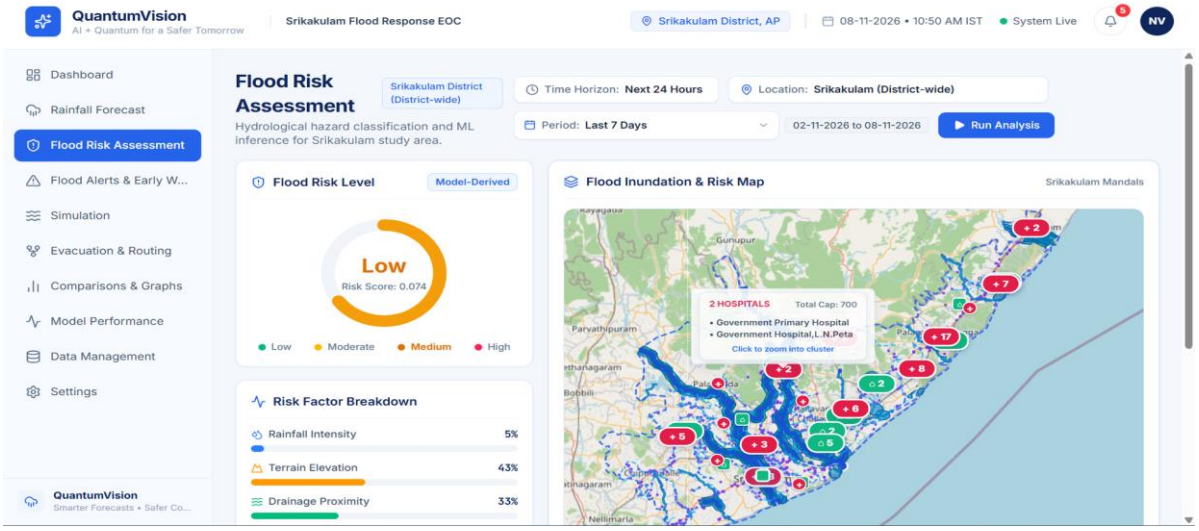


Figure 13: Flood-risk assessment dashboard with risk level, factor breakdown and inundation map

4.5 Evacuation Routing Results

The routing module accepts a start and shelter, offers flood-aware A* or shortest-distance Dijkstra, and displays the route on an OpenStreetMap-based map (Figure 14). The example route is 0.56 km, with an estimated one-minute travel time and moderate exposure to mapped flood zones, plus turn-by-turn waypoints.

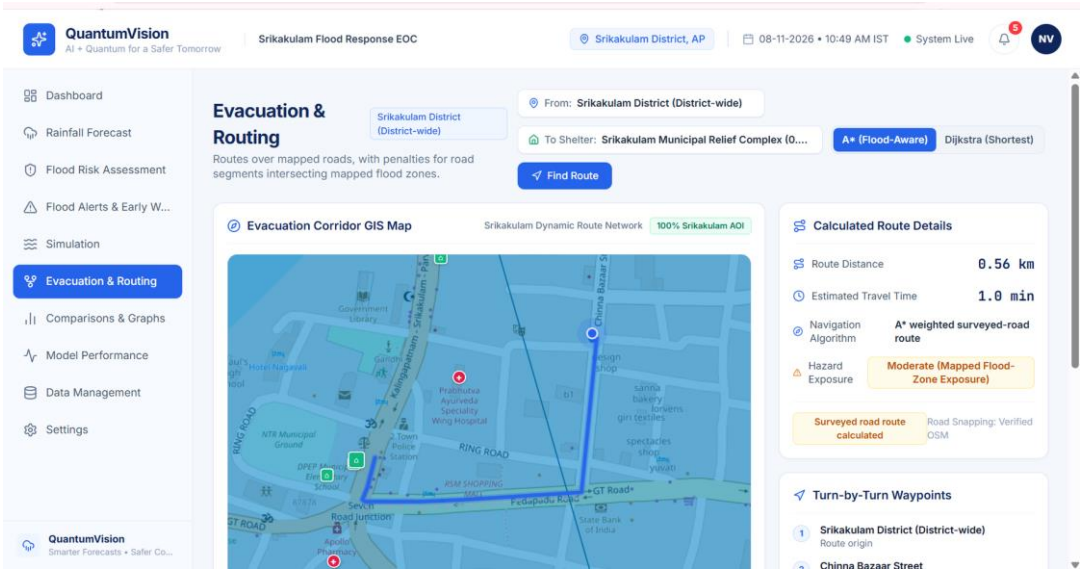


Figure 14: Dynamic evacuation-routing dashboard with A flood-aware route, distance, travel time and hazard exposure*

With an admissible heuristic, Dijkstra and A* return equal-cost routes but differ in search effort: Dijkstra expands by cost in all directions, while A* focuses toward the destination and usually visits fewer nodes. Flood penalties make affected segments more expensive, so a slightly longer safer route may be preferred. If no acceptable path exists, the interface reports none. A multi-route quantitative benchmark is not included and remains future work.

4.6 Flood Simulation and Scenario Analysis

The Simulation dashboard (Figure 15) lets users select normal, heavy or extreme rainfall and view affected area, exposure, risk level and rainfall-impact forecasts. It helps planners compare changes in flooded footprint and affected roads across scenarios.

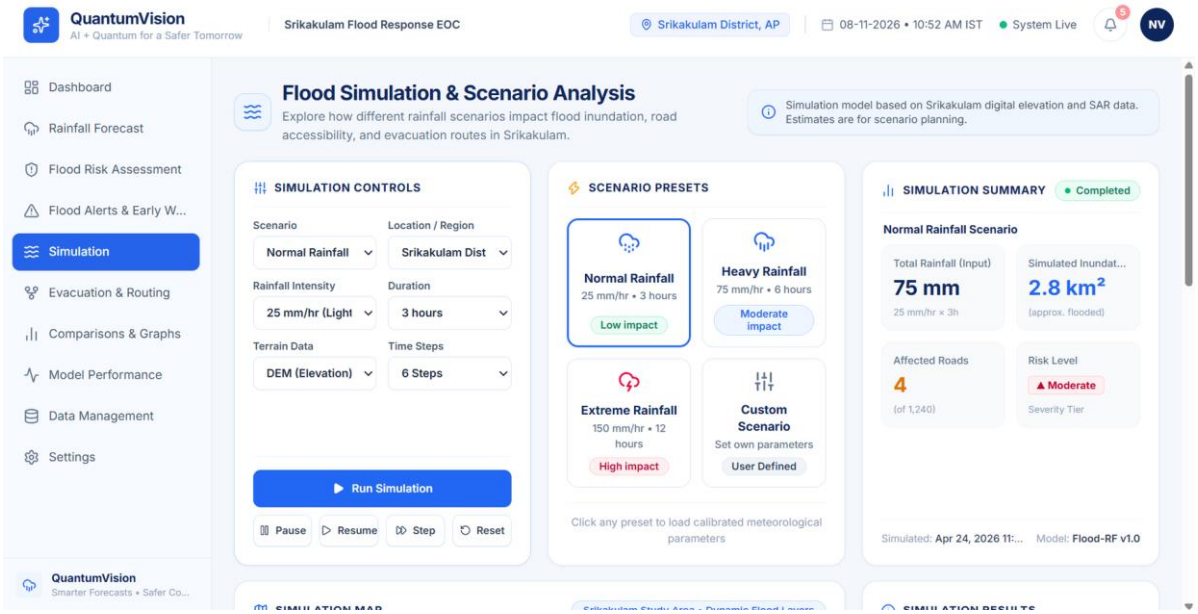


Figure 15: Flood simulation and scenario-analysis dashboard

4.7 Alerts and Reports

The Alerts dashboard (Figure 16) lists forecast- and risk-based warnings with location, risk level, description and timestamp; a roadmap panel highlights affected arterial roads. Reports combine forecasts, classifications, routes and performance measures.

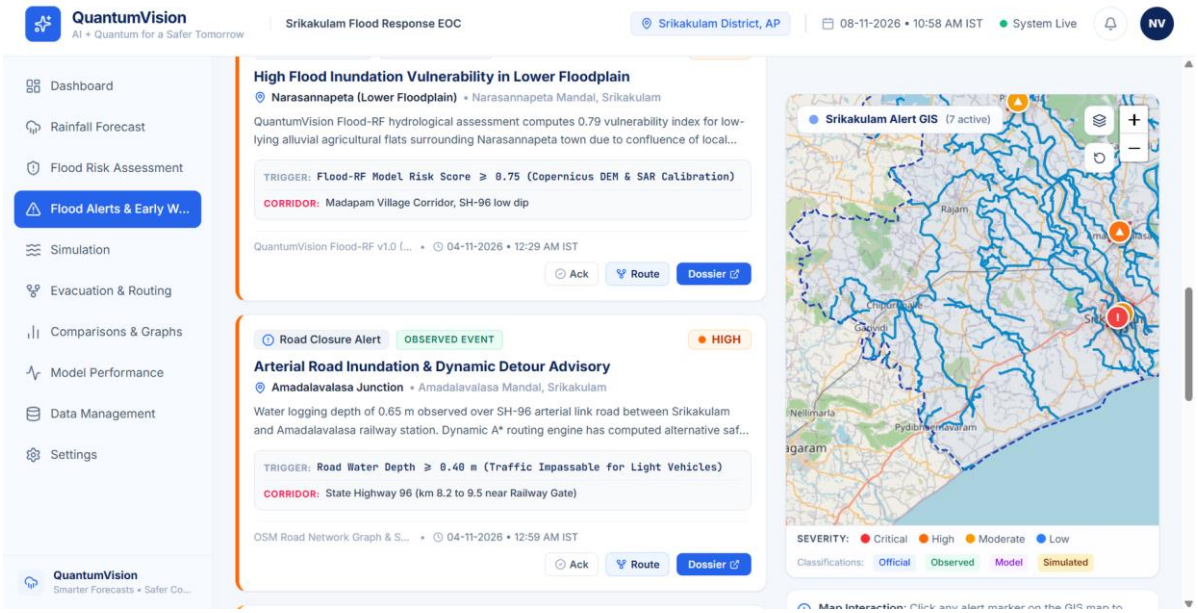


Figure 16: Flood alerts and early-warning dashboard with affected-road information

4.8 Discussion

The results indicate that tree ensembles are efficient baselines for tabular SAR-terrain-rainfall features and are the practical default. Simulated quantum classifiers are feasible and achieve useful recall, but currently require more computation and have lower precision, so their role is exploratory. Integrating forecasting, classification, spatial layers and routing makes stage dependencies explicit and lets one risk layer drive both maps and route costs.

4.9 Limitations

- Flood labels are drawn from historical records and satellite evidence, and their accuracy limits every classifier; the number of flooded samples in the test set is small.
- The quantum models are four-qubit, noise-free simulations with limited tuning; no hardware advantage is claimed.
- Rainfall forecasting uses station observations only, and extreme-rain errors remain large.

5. FUTURE SCOPE

Future work includes live weather, river-level and gauge feeds plus numerical weather prediction to improve heavy-rain forecasts and enable continuous alerts; larger, independently verified flood labels and testing across districts/years; larger quantum circuits, alternative encodings, noise-aware simulation, threshold calibration and eventual hardware execution; and systematic routing benchmarks with verified shelter capacity and live road closures. Explainable AI, uncertainty estimates, digital-twin simulation, offline mobile apps with location notifications and extension to cyclones and fires may broaden the platform.

6. CONCLUSION

QuantumVision integrates Sentinel-1 SAR, terrain, rainfall and infrastructure data with rainfall forecasting, classical/quantum flood classification, GIS mapping, flood-aware Dijkstra/A* routing and QUBO optimization in a web dashboard for Srikakulam. On held-out data, Random Forest and XGBoost achieved 99.80% accuracy and 98.77% recall; simulator-based four-qubit VQC and QNN achieved about 90% accuracy, with recalls of 93.83% and 87.65%. XGBoost gave the best 24-hour forecast (MAE 4.115 mm; RMSE 8.552 mm), although heavy-rain explanatory power remains limited. The integrated data-to-guidance pipeline is practical, but simulator results do not establish quantum advantage. Predictions and routes remain decision support requiring further validation before operational use.

ACKNOWLEDGEMENT

The authors thank Dr. V. Srinadh for his guidance and the Department of Computer Science and Engineering (Artificial Intelligence & Machine Learning), GMR Institute of Technology, Rajam, for providing the facilities and support for this work.

REFERENCES

1. S. Muthukumar, S. Rajesh, A. Talpallikar, and T. K. Mishra, “A systematic review of artificial intelligence frameworks for holistic disaster management,” Discover Artificial Intelligence, vol. 6, no. 1, Art. no. 308, Dec. 2026, doi: 10.1007/s44163-026-01020-w.
2. C. Whyte and B. B. Haupt, “Building centaur responders: Is emergency management ready for artificial intelligence?” Disasters, vol. 50, no. 3, Art. no. e70054, Jul. 2026, doi: 10.1111/disa.70054.
3. L. M. Mabina, F. M. Mashao, F. I. Mathivha, M. V. Singo, and H. Chikoore, “Deep learning and geospatial approaches for post-flood disaster damage assessment: A systematic review,” Applied Geomatics, vol. 18, no. 2, Art. no. 95, Jun. 2026, doi: 10.1007/s12518-026-00745-8.
4. M. Mishra et al., “Integrating machine learning and geospatial analysis for flood hazard, vulnerability, and risk assessment in Odisha, India,” Water Resources Management, vol. 40, no. 4, Art. no. 348, Mar. 2026, doi: 10.1007/s11269-026-04709-w.
5. A. Kumar, G. Pandey, and R. V. Kale, “Ensemble machine learning and deep learning framework for flood susceptibility mapping in the transboundary Rapti River Basin,” Environmental Earth Sciences, vol. 85, no. 3, Art. no. 168, Feb. 2026, doi: 10.1007/s12665-026-12912-6.
6. J. Zhao, W. Feng, and X. Yi, “A spatial prediction architecture utilizing machine learning models for diverse forecasting demands in coseismic landslide scenarios,” International Journal of Disaster Risk Science, vol. 17, no. 1, pp. 45–58, Feb. 2026, doi: 10.1007/s13753-026-00749-3.
7. J. G. Dcruz, A. Zolotas, N. R. Greenwood, and M. Arana-Catania, “Structured AI decision-making in disaster management,” Scientific Reports, vol. 15, no. 1, Art. no. 32093, Dec. 2025, doi: 10.1038/s41598-025-15317-w.
8. D. Bao et al., “A machine learning-based prediction-to-map framework for rapid and accurate spatial flood prediction,” npj Natural Hazards, vol. 2, no. 1, Art. no. 71, Oct. 2025, doi: 10.1038/s44304-025-00122-2.
9. U. Lagap et al., “Towards reliable deep learning for post-disaster damage assessment: An XAI-based evaluation,” International Journal of Disaster Risk Reduction, vol. 130, Art. no. 105839, Sep. 2025.
10. F. Özen and A. Souri, “Cloud-based disaster management architecture using hybrid machine learning approach in IoT,” Multimedia Tools and Applications, vol. 83, no. 14, pp. 24105–24128, May 2024, doi: 10.1007/s11042-024-18333-6.
11. Y. Song, L. Hao, and W. Li, “Landslide detection using deep learning on remotely sensed images,” Applied Computing and Geosciences, vol. 27, Art. no. 100278, Sep. 2025, doi: 10.1016/j.acags.2025.100278.

12. K. Ouyang et al., “Escape: An optimization method based on crowd evacuation behaviors,” Artificial Intelligence Review, vol. 58, no. 1, Art. no. 19, Jan. 2025, doi: 10.1007/s10462-024-11008-6.

13. A. Sevim, Q. Guo, and E. E. Ozguven, “A simulation-based framework for leveraging shared autonomous vehicles to enhance disaster evacuations in rural regions with a focus on vulnerable populations,” Journal of Infrastructure Preservation and Resilience, vol. 6, no. 1, Art. no. 10, Mar. 2025, doi: 10.1186/s43065-025-00122-6.

14. J. Visave, “Transparency in AI for emergency management: Building trust and accountability,” AI and Ethics, vol. 5, no. 4, pp. 3967–3980, Nov. 2025, doi: 10.1007/s43681-025-00692-x.

15. W. Yang, Y. Wu, and S. K. K. Chow, “Deep learning method for real-time fire detection system for urban fire monitoring and control,” International Journal of Computational Intelligence Systems, vol. 17, no. 1, Art. no. 216, Aug. 2024, doi: 10.1007/s44196-024-00592-8.