International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Agentic AI-Enabled Cognitive Digital Twin for Urban Flood Prediction and Disaster Management: A Multi-Agent Framework for the Indian Urban Context

Dr.E.Gomathi[1], Maruthi Vasan K[2], Saikeerthika V[3], Jaya Vishakhan S[4], Dushyant R[5]

[1] Head of the Department,Department of Artificial Intelligence and Data Science,

Coimbatore Institute of Engineering and Technology, Coimbatore, India.

[2,3,4,5] Department of Artificial Intelligence and Data Science,

Coimbatore Institute of Engineering and Technology, Coimbatore, India.

Corresponding author: vasanmaruthi466@gmail.com

Abstract - Flood response in Indian cities still leans on rainfall forecasts and gauge readings that say little about which streets will go under. Existing flood digital twins can reproduce street-level inundation, but a design-storm run is reported to take hours, and a person must interpret the result and decide what follows. Learning-based flood mappers cannot tell when their own output is unreliable. This paper presents a cognitive digital twin for urban flood prediction, built from seven cooperating software agents for sensing, prediction, simulation, explanation, alerting, orchestration and conversation. The orchestrator treats prediction confidence as a control signal: below a threshold it re-acquires fused radar and optical data a bounded number of times, then hands the case to a human. Each prediction carries a spatial explanation, and alerts raised through chat require explicit confirmation. A browser client shows the twin in 3D and falls back to 2D maps on weaker devices. We demonstrate a prototype on zones in Bengaluru and Mumbai. Depth, confidence and alert state stayed consistent across the dashboard, the cognition feed and the alert log; one cycle that projected 1.05 m at 82% confidence raised a critical alert. Accuracy benchmarking on labelled flood events remains future work.

Key Words: digital twin, agentic AI, urban flood prediction, explainable AI, multi-agent system, remote sensing.

1. INTRODUCTION

Flooding is among the most costly natural hazards. A recent systematic review of digital twins for flood risk management reports that flood events affected more than 2.3 billion people between 1993 and 2024 and caused over USD 400 billion in damage between 2010 and 2024 [1]. The same review attributes urban vulnerability to dense populations, impervious surfaces and ageing or insufficient drainage [1].

India's exposure is large. The National Flood Commission (Rashtriya Barh Ayog) put the flood-liable area at about 40 million hectares in 1980, roughly 12% of the country's geographic area [2]. Cities add a second layer of risk. The Chennai floods of 2015 and the Bengaluru floods of 2022 brought urban flooding to national attention, and reporting on the Bengaluru event notes that several flooded districts along the Outer Ring Road stand on former lake areas [3]. Day-to-day response, however, still leans on rainfall forecasts and gauge readings. These indicate that heavy rain is coming but not which streets will flood or how deeply, and smaller cities often lack the dense sensing that finer predictions would need.

The digital twin concept originated in aerospace and manufacturing [4] and has since been applied to cities. For urban flooding, the most detailed example we found is FlowsDT-Galveston [5]. It couples a one- and two-dimensional hydrodynamic model with LiDAR terrain and storm-sewer geometry and presents the results in a four-dimensional interactive scene. Its authors position it as decision support for planners and emergency managers, and report that one 100-year, 24-hour scenario took 13 h 24 min on a workstation with an Intel i9 processor, an RTX 3090 graphics card and 64 GB of memory [5].

Three gaps motivate the present work. First, accurate physics-based twins are expensive to run, and the decision that follows each run stays with a person. Second, learning-based flood mappers can be accurate yet brittle: an attention U-Net on Sentinel-1 imagery reached an F1 score of 0.89 on the frame it was trained on but 0.61 on an unseen frame [6], and the pipeline as reported has no step that flags which predictions to distrust. Third, lifecycle frameworks such as the urban flooding digital twin framework of Ge and Qin [7] organise preparation, response and recovery around human stakeholders and, as far as their abstract describes, do not include software agents that act. Work on agentic AI for disaster management argues for such agents in principle [8], but we did not find a flood digital twin that implements a confidence-aware loop of this kind.

This paper makes six contributions.

(i) A seven-agent architecture in which sensing, prediction, simulation, explanation, alerting, orchestration and conversation are separate, independently callable modules coordinated through shared twin state.

(ii) An orchestration rule that treats prediction confidence as a control signal. Low-confidence outputs trigger bounded re-acquisition of fused radar and optical data and, if that fails, escalation to a human instead of an automatic alert.

(iii) A three-level explanation scheme (spatial, feature-level and confidence-level) whose confidence level is tied to the orchestrator's retry decision.

(iv) A data and rendering strategy for Indian conditions: ISRO products as primary sources with open international fallbacks, building heights tagged with the method that produced them, and a browser client that chooses 3D or 2D rendering from device capability.

(v) A domain-restricted conversational interface in which any action with real-world consequences, such as sending an alert, requires explicit confirmation.

(vi) A working prototype, CognitwinX, and a demonstration of its behaviour on zones in Bengaluru and Mumbai.

Sec. 2 reviews related work, Sec. 3 describes the framework, Sec. 4 gives implementation details, Sec. 5 reports the prototype demonstration, Sec. 6 discusses limitations, and Sec. 7 concludes.

2. RELATED WORK

2.1 Flood digital twins

FlowsDT-Galveston [5] is built from LiDAR-derived terrain, storm-sewer geometry and rain-gauge data, simulated with a coupled 1D-2D model and viewed in a four-dimensional platform. It was compared with flood locations harvested from news and social media for four historical events and then run for 2-, 10-, 25-, 50- and 100-year design storms. A radar-fed forecasting component is also described. The authors list the limits themselves: all inundated buildings are treated alike, storm surge is not coupled, clogged drains are not modelled, and computation is heavy [5].

Ge and Qin [7] take a different angle. They propose a lifecycle framework with a conceptual model and a generative method for building twins at city, regional and street scale, intended to coordinate stakeholders across preparation, response and recovery. Fan et al. [9] earlier set out a vision of a disaster-city twin that combines artificial and human intelligence. Hlal et al. [1] reviewed 85 studies from 2018 to 2025. They observe that implementations emphasise technical capability, such as sensor integration, hydraulic simulation and learning-based forecasting, over policy coherence and equitable communication, that integration is fragmented, and that 57% of the studies came from Europe and high-income Asia against 26% from low- and middle-income regions [1].

2.2 Learning-based flood mapping

U-Net [10] and its attention variant [11] are standard tools for pixel-wise segmentation; attention gates on the skip connections suppress irrelevant regions. Fakhri and Gkanatsios [6] applied an attention U-Net to Sentinel-1 data over New South Wales and found VH polarisation the most effective channel. Their model scored F1 = 0.89 on the frame it was trained on, 0.61 on an unseen frame and 0.84 on a later event, and adding topographic features lowered accuracy [6]. ConvExNet [12] classifies images as flooded or not with a dilated convolutional network on water-index images derived from Sentinel-2, reporting 94.68% accuracy with far fewer parameters than transfer-learning baselines. Situ et al. [13] pair an LSTM with DeepLabv3+ and Bayesian optimisation for urban flood prediction. For India, Balamurugan et al. [14] apply machine-learning algorithms to predict rainfall-driven flooding with advance warning, and Kumar et al. [15] survey machine learning for flood forecasting and the challenges that remain.

2.3 Explainability and agentic systems

Shapley-value attribution [16] assigns each input feature a share of a prediction. ConvExNet [12] uses it to colour the pixels that pushed a classification towards flood or non-flood, and Dar et al. [17] propose an explainable network for flood prediction in cities. In the works we reviewed, explanations are produced for a human reader. We did not find one in which the explanation, or the confidence behind it, feeds back into what the system does next, which is the use made of it here. On the agentic side, Li et al. [8] argue for AI agents that work alongside people in disaster management; we did not find a flood-specific twin that implements such agents with confidence-based control.

Table −1 summarises how each line of work stops short of the closed loop targeted in this paper.

Table -1: Representative prior work and the issue each leaves open

Work	What it provides	What it leaves open
FlowsDT [5]	1D-2D simulation, 4D view; 13 h 24 min for a 100-year, 24 h storm	Decision support for people; heavy compute; no autonomous action
Attention U-Net [6]	SAR flood extent; F1 0.89 trained, 0.61 unseen	Single task; no flag for unreliable output
ConvExNet [12]	Image-level flood classification with SHAP	Not coupled to simulation or response
UFDT framework [7]	Lifecycle framework for stakeholders	Coordinates people; software agents not described
Review [1]	85 studies synthesised	Reports fragmented, technology-led practice

3. PROPOSED FRAMEWORK

3.1 Overview and shared twin state

The framework (Fig. 1) is a set of seven agents that share one twin state held in PostgreSQL with the PostGIS spatial extension. For each monitored zone the state records the zone's spatial identity and class, the time of the latest observation, the sources used, the prediction products, the confidence, the simulated future state, the explanation artefacts, the alert status and an audit trail. Each agent runs as a Celery task, and the orchestrator chains them. Every agent has one job and can be called on its own, which keeps failures local and lets each module be tested separately. Table −2 lists the responsibilities.

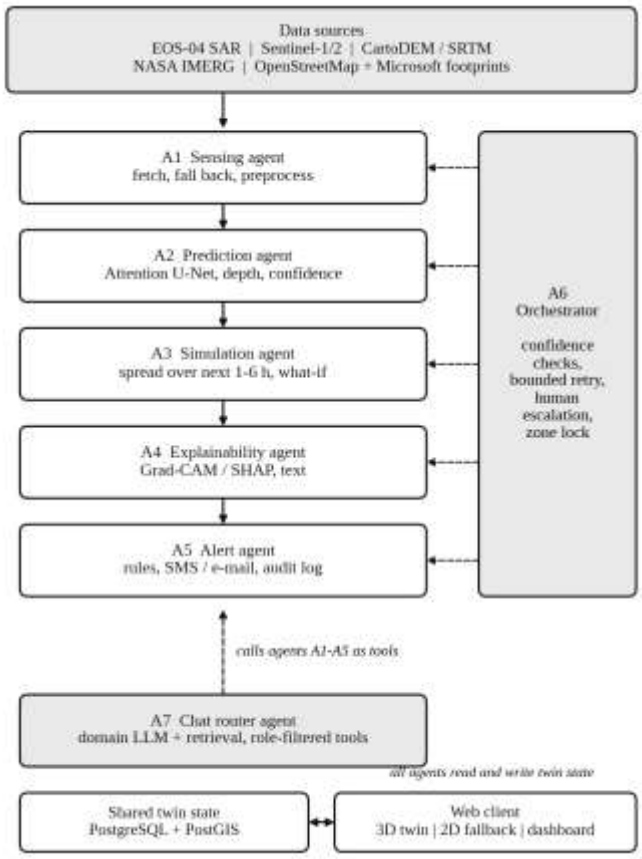


Fig -1: Architecture of the cognitive digital twin

Table -2: Agents, responsibilities and outputs

Agent	Responsibility	Output
A1 Sensing	Fetch SAR, optical, rainfall, elevation and building data; log every fallback	Calibrated tiles, rainfall, DEM
A2 Prediction	Attention U-Net extent, depth regression, confidence	Mask, depth, c in [0, 1]
A3 Simulation	Project spread for next 1-6 h; what-if queries	Depth-grid time series
A4 Explainability	Grad-CAM / SHAP, feature importance, confidence breakdown	Explanation record
A5 Alert	Depth, confidence and criticality rules; SMS and e-mail	Alert and logged decision
A6 Orchestrator	Chain agents, bounded retry, escalate, lock zone	Cognition-log entries
A7 Chat router	Map questions to agent calls; retrieval-grounded answers	Answers with sources

3.2 Data sources and fallbacks

The primary radar source is the EOS-04 synthetic aperture radar (SAR) medium-resolution Level-2B product obtained through ISRO's Bhoonidhi portal. The water-spread product from the same mission serves as an independent reference where it is available for the zone and date. The fallback is Sentinel-1 ground-range detected imagery in VH polarisation, chosen because VH was the most effective channel in [6]. Sentinel-2 Level-2A imagery gives a normalised difference water index (NDWI) cross-check when skies are clear. Elevation comes from CartoDEM at 30 m with SRTM as fallback, and rainfall from NASA IMERG. Observations from the India Meteorological Department are used offline for comparison because we did not find an openly accessible programmatic feed. Sec. 4.3 states which of these sources were exercised in the demonstration.

When a primary source has no usable pass for a zone and time, the sensing agent switches to the fallback and writes the switch to the cognition log instead of failing silently. If the rainfall feed fails, the last good value is reused and the reuse is logged. Preprocessing comprises radiometric calibration, Lee speckle filtering, NDWI computation, grey-level co-occurrence texture features and tiling into 128 × 128 pixel patches. SAR and optical imagery are not interchangeable measurements, so the selected source is stored with every prediction.

3.3 Prediction

The prediction agent runs an attention U-Net [11] over the patches to produce a flood-extent mask. Depth comes from a lightweight regression on rainfall intensity, DEM-derived elevation (digital elevation model), a drainage-capacity proxy and, when available, soil moisture. This is deliberately a surrogate and not a hydrodynamic solver. We accept lower physical fidelity than a 1D-2D model [5] in return for run times that allow repeated cycles. The agent also returns a scalar confidence c in [0, 1] derived from the segmentation output. The orchestrator uses this value and nothing else from the model's internals.

3.4 Orchestration and the confidence-retry rule

The orchestrator runs one cognitive cycle per zone (Fig. 2). Three configurable defaults govern it: a confidence threshold of 0.6, a maximum of two retries, and an alert depth of 0.3 m. If c is at least 0.6, the cycle continues to simulation, explanation and the alert rules. Otherwise, while fewer than two retries have been used, the sensing agent re-acquires fused SAR and optical data and the prediction is repeated. When the retries are exhausted the zone is flagged for human review and no automatic alert is sent.

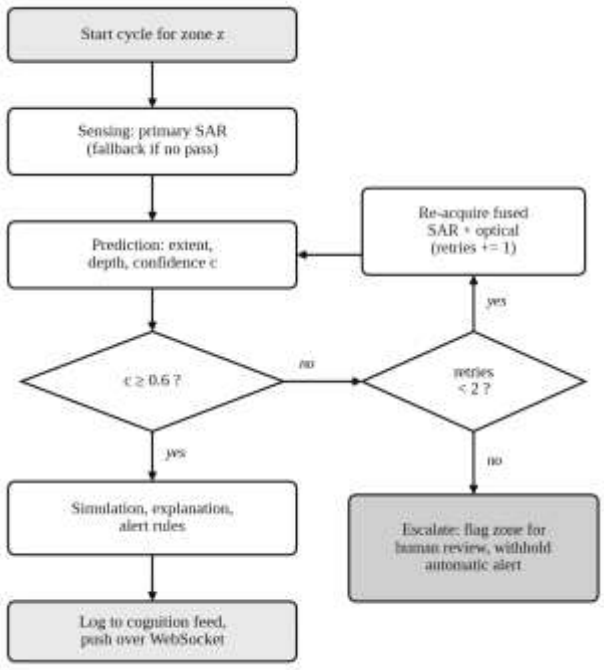


Fig -2: Orchestrator confidence-retry rule for one cognitive cycle

Two guards matter in practice. The retry cap prevents an ill-conditioned input from looping and delaying the alerts of other zones. A non-finite confidence value is treated as below threshold, because a comparison with NaN evaluates to false and would otherwise let a corrupt prediction pass as acceptable. A per-zone lock in Redis stops a scheduled cycle and an on-demand cycle from running at once. Each step is appended to a cognition log and pushed to the client over a WebSocket, so a user can see why the twin re-checked or escalated.

3.5 Three-level explanation

Explanations are generated at three levels. Level 1 is spatial: Gradient-weighted Class Activation Mapping (Grad-CAM) [18] is the default because it is cheap, and SHapley Additive exPlanations (SHAP) [16] are computed on request with a timeout so that a stalled computation never blocks a cycle. Level 2 is feature-level: SHAP values on the depth regression show how much rainfall, elevation, drainage capacity and soil moisture contributed to the estimated depth. Level 3 explains the confidence itself by listing the signals that raised or lowered it, such as missing optical corroboration, and links to the orchestrator's decision to retry.

The three levels are combined into a short plain-language summary produced from templates with conditional branches. The main driver named in the text is always the feature with the largest attribution, so the text cannot contradict the numbers beneath it. An explanation is evidence about what the model attended to; it is not proof that the model is right, and the interface does not present it as such.

3.6 Alert rules

Model inference and alert policy are kept apart. A prediction becomes an alert candidate only after the orchestrator accepts it. The alert agent then compares the projected depth with the 0.3 m default, raises the severity as depth increases, and escalates any zone classed as a hospital or

school to critical. Every alert record carries the zone, the time, the confidence, the depth and the action taken. When a zone is escalated for human review, the record states that automatic notification was withheld and why. SMS and e-mail are delivery channels behind the agent, so a failed delivery can be retried without rerunning the prediction, and an authorised user acknowledges each alert from the interface.

3.7 3D twin and responsive rendering

Building footprints come from OpenStreetMap and from Microsoft's machine-learning footprint dataset. The two are merged by spatial overlap, using real height or level tags where they exist and type-based height estimates tuned for Indian building types where they do not. Every building keeps a field recording how its height was obtained, geometries are validated and repaired, implausible heights are clipped, and one malformed record is skipped without aborting a zone. The scene is rendered in the browser with instanced meshes. The flood surface is a shader driven by the simulation agent's depth grid and interpolated between updates so that water visibly rises and recedes, and WebSocket updates are throttled to one every two to three seconds.

Before any renderer is mounted, the client checks WebGL support, processor cores and device memory and selects one of three tiers: the 3D scene, a vector-tile 2D map, or a raster 2D map for the weakest devices. The simulator panel gives an instant client-side preview of a what-if change and then morphs the scene to the backend projection when it arrives.

3.8 Conversational interface

The seventh agent answers questions by calling the other agents as tools. It is backed by a small open-source language model fine-tuned with low-rank adaptation (LoRA) [19] on flood-domain question-answer pairs, including examples that refuse unrelated requests, and by retrieval-augmented generation (RAG) [20] over live twin records. Numeric statements must come from retrieved records and not from the model's memory. The server filters the tool list by user role before the model sees it, so a public viewer has read-only tools. The alert tool never fires from a single message: the agent asks for confirmation and acts only on an explicit confirming reply, and the action is logged as chat-initiated.

3.9 Roles, security and audit

The twin exposes operational information, so access is separated by role (administrator, NDMA officer, public viewer) and enforced at the API, not only in the client. Officer accounts need administrator approval before they can acknowledge or trigger alerts. Every state-changing action writes an audit record with the actor, the time, the zone and the resulting state, and agent-generated actions also name the responsible agent. Credentials are read from environment variables and never stored in the code base.

4. IMPLEMENTATION

4.1 Software stack

The client is written in React with Three.js through React Three Fiber, and the 2D tiers use MapLibre GL and Leaflet. The server uses Django REST Framework with Django Channels for WebSockets, Celery with Redis for task chaining and scheduling, and PostgreSQL with PostGIS. Models are implemented in PyTorch, and the language model runs locally in a 4-bit quantised format through llama.cpp or Ollama. The services are packaged with Docker Compose.

4.2 Training resources

The development machines have no discrete GPU, so model training is designed to run on free cloud GPU notebooks (Google Colab or Kaggle), with only inference run on local processors.

4.3 Demonstration configuration

The prototype can run against live sources or against its demonstration data layer, chosen by one configuration flag. The results in Sec. 5 were captured in the demonstration configuration. In that mode a generator lays out street-grid building blocks, with density falling away from the zone centre, and supplies depth, confidence, elevation and drainage-efficiency values for each named zone. The generated buildings are not real structures, and the displayed values show the behaviour of the pipeline and the interface, not measured flooding. The zones are named after localities in Bengaluru and Mumbai: Bellandur - ORR Tech Corridor, Bellandur Demo Zone, Dharavi Demo Zone, Kurla West - Mithi River Confluence and Velachery Demo Zone.

4.4 Interface structure

The prototype is named CognitwinX (version 2.4 in the interface). Four views are reached from a top navigation bar: Portal Home, Mission Control, Zone Analysis and Alerts. A role badge in the corner shows the signed-in role, which is an NDMA officer in the captures reported here. A floating Chat with the Twin button opens the conversational panel from every view.

5. RESULTS: PROTOTYPE DEMONSTRATION

5.1 Scope

This section reports what the running prototype displayed during a demonstration session in which the authors were signed in with the NDMA-officer role. Four screens were captured: Mission Control (Fig. 3), Zone Analysis (Fig. 4), the alert log (Fig. 5) and the chat panel (Fig. 6). Values are quoted as they appeared. Because the session ran on the demonstration data layer (Sec. 4.3), the figures show how the framework behaves; they are not offered as measured flood accuracy, and no labelled flood event was used.



Fig -3: Mission Control: monitored basins (left), 3D twin with readouts and timeline (centre), cognition feed and what-if simulator (right), and a critical-alert banner on top

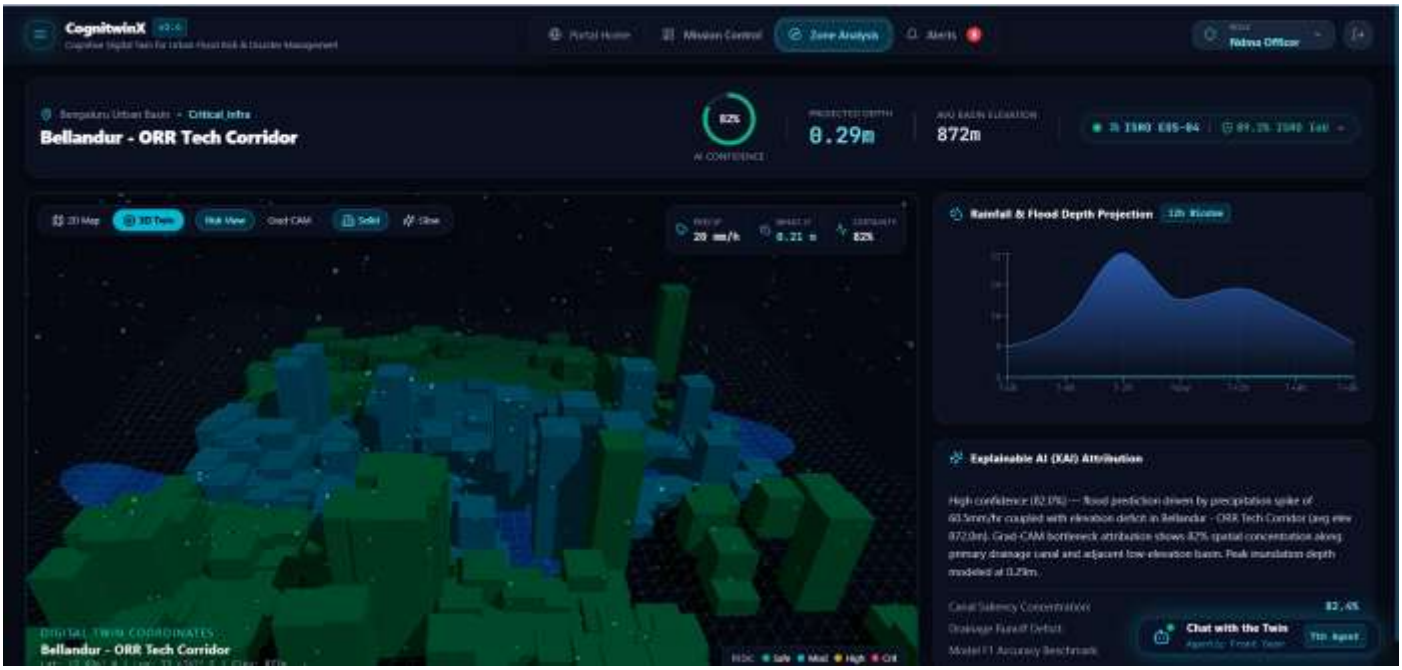


Fig -4: Zone Analysis for Bellandur - ORR Tech Corridor: headline readouts, 3D twin, 12-hour rainfall and depth projection, and explainable-AI attribution

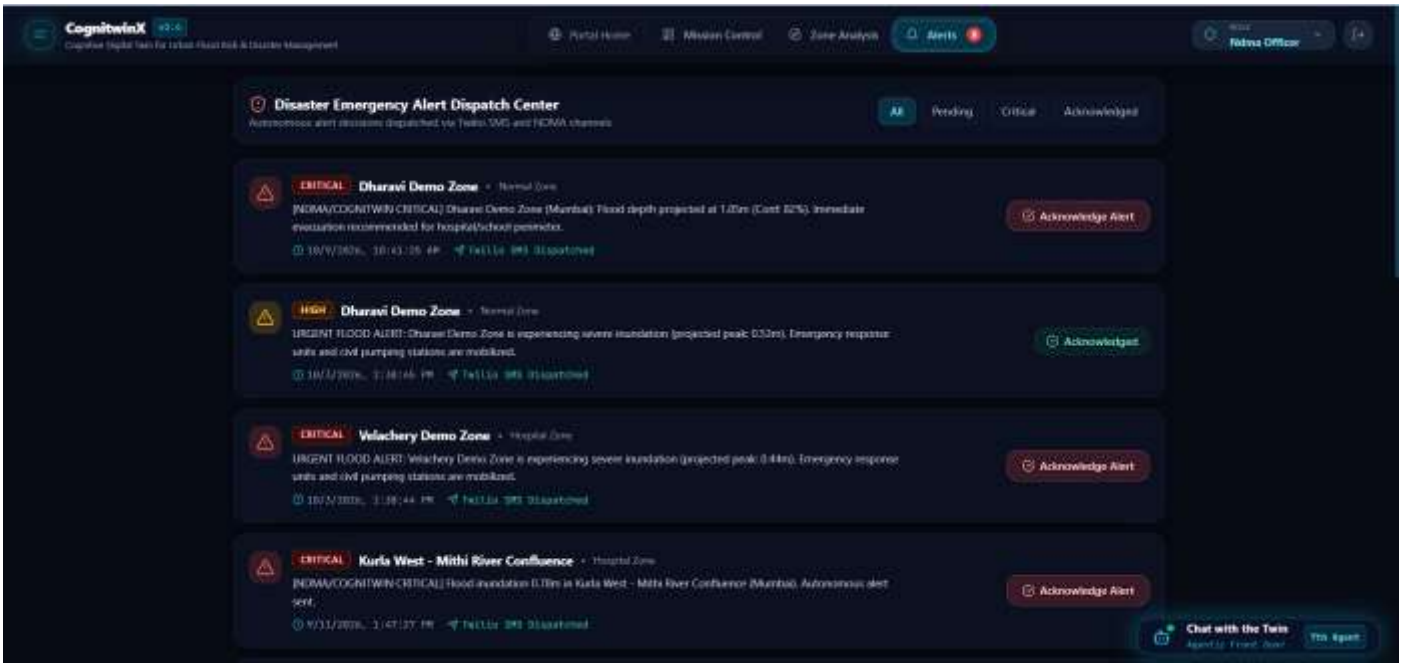


Fig -5: Alert dispatch centre as seen by an NDMA officer: severity, zone class, message, time, dispatch status and acknowledge control

5.2 Zone state

The left panel of Mission Control lists the monitored basins, each with a confidence ring, a projected depth, an elevation and a drain efficiency (Table −3). The two Bellandur zones in Bengaluru, both of critical-infrastructure class, show a projected depth of 0.29 m at 82% and 80% confidence, with elevations of 872 m and 868 m and drain efficiencies of 50% and 52%. The Dharavi Demo Zone in Mumbai, classed as normal, shows 1.05 m at 82% confidence, an elevation of 7.5 m and the lowest drain efficiency of the three, 38%. The ordering is what the inputs would suggest: the zone with the least drainage capacity and the lowest ground carries the deepest projected water. With three zones this is an observation about the demonstration data, not evidence of model validity.

Table -3: Zone state shown in Mission Control (Fig. 3)

Zone	Class	Conf.	Depth (m)	Elev. (m)	Drain eff.
Bellandur - ORR Tech Corridor (Bengaluru)	Critical infra.	82%	0.29	872	50%
Bellandur Demo Zone (Bengaluru)	Critical infra.	80%	0.29	868	52%
Dharavi Demo Zone (Mumbai)	Normal	82%	1.05	7.5	38%

5.3 One cycle seen from three views

The latest orchestrator entry in the cognition feed (cycle f9028ff7, Dharavi Demo Zone, 10:41:25 AM) reads: depth 1.05 m, confidence 82.0%, ISRO IoU score 72.0%, alert critical. The red banner above Mission Control and the first entry of the alert log (Fig. 5) quote the same depth and confidence and the same time. The agreement follows from the shared twin state of Sec. 3.1: each view reads one record, and none keeps a private copy. The Alert agent's Emergency Threshold Evaluation entry carries the same timestamp as the orchestrator's completion entry.

The source badge above the 3D view reads ISRO EOS-04 with an IoU of 89.2% for the selected Bellandur zone, while the 72.0% in the feed belongs to the Dharavi cycle. The agreement score is therefore carried per zone and per cycle and not as one global figure.

5.4 3D twin and timeline

The central panel renders the selected zone in 3D (Fig. 3 and Fig. 4). Six toggles control the view: 2D Map, 3D Twin, Risk View, Grad-CAM, Solid and Glow. Buildings are coloured on a four-step legend of safe, moderate, high and critical. In the Bellandur scenes, where the projected depth is 0.29 m, the buildings around the flooded depression are coloured moderate and the rest safe, and the flood surface appears as a blue patch on the terrain. Small readouts on the viewport show precipitation, a what-if value and the certainty. A timeline under the scene runs from 24 hours before Now, with a play control and a historical-replay button.

5.5 Projection and explanation

Zone Analysis for Bellandur - ORR Tech Corridor repeats the headline readouts (confidence 82%, projected depth 0.29 m, average basin elevation 872 m) and adds a 12-hour rainfall and depth projection from T−6 h to T+6 h. Read from the plot, the rainfall curve rises to a peak of about 32 on its axis near T−2 h, dips, forms a second, lower peak of about 22 near T+2 h and falls to about 9 at T+6 h, while the depth curve stays close to zero on the same axis.

The explainable-AI panel puts the result in words: high confidence (82.0%), a precipitation spike of 60.5 mm/h combined with an elevation deficit, a Grad-CAM attribution concentrated 82% along the primary drainage canal and the adjacent low-lying basin, and a peak inundation depth of 0.29 m. A separate readout gives the canal saliency concentration as 82.4%. The text therefore combines the three levels of Sec. 3.5: a spatial statement, the feature drivers and the confidence.

5.6 What-if simulator

The What-If Inundation Simulator in the right-hand column of Mission Control offers a monsoon precipitation anomaly slider that runs from −50% (drought) to +100% (cloudburst), captured at +25%, with a stormwater drainage control below it. The panel describes the behaviour of Sec. 3.7: an instant client-side 3D preview followed by a smooth morph to the backend projection.

5.7 Alert behaviour

The alert log lists each decision with a severity badge, the zone and its class, the message, a timestamp and the dispatch status (Fig. 5 and Table −4). Filters for All, Pending, Critical and Acknowledged sit at the top, and the navigation bar shows a badge count of eight. Four entries are visible. A depth of 0.53 m in a normal-class zone is logged as high, whereas 1.05 m in the same zone is critical, so severity rises with depth even without a sensitive facility. A depth of 0.44 m in a hospital-class zone is already critical, as the escalation rule of Sec. 3.6 prescribes. Every entry reads Twilio SMS Dispatched; one is marked Acknowledged and the others offer an acknowledge button. Both Bellandur zones sit at 0.29 m, just under the 0.3 m default alert depth, and neither appears among the four visible entries.

Table -4: Alert log entries visible in Fig. 5 (dates as displayed, month/day/year)

Zone (class)	Severity	Depth (m)	Logged	Status
Dharavi Demo Zone (normal)	Critical	1.05	10/9/2026, 10:41 AM	SMS dispatched; pending
Dharavi Demo Zone (normal)	High	0.53	10/3/2026, 2:38 PM	SMS dispatched; acknowledged
Velachery Demo Zone (hospital)	Critical	0.44	10/3/2026, 2:38 PM	SMS dispatched; pending
Kurla West - Mithi River Confluence (hospital)	Critical	0.78	9/11/2026, 1:47 PM	SMS dispatched; pending

5.8 Conversational interface

The Cognitive Twin Chat panel (Fig. 6) opens with the selected zone as context, an indicator of six agents online and a language selector set to English. Four shortcut chips offer typical questions: water level, compare zones, what-if +30% rain and explain risk. The opening message says the assistant can reach the six twin agents, forecast what-if scenarios, compare zone vulnerability and look up historical events. The footer names two rules in force: answers cite actual sensor or model values, and an alert needs confirmation before it is sent. A microphone button supports voice input. The captured screen shows no question-and-answer exchange, so answer quality is not assessed here.

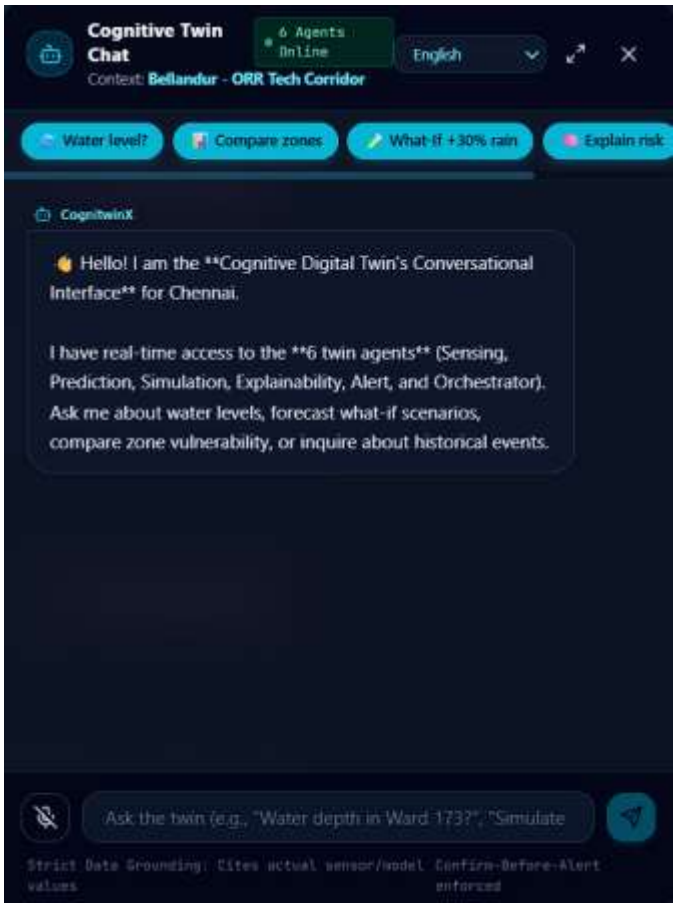


Fig -6: Cognitive Twin Chat panel with zone context, shortcut questions and language selector

5.9 Run time

Mission Control carries a card that compares run times. It shows the attention U-Net running in 1.2 s against the 13-hour PCSWMM run of FlowsDT-Galveston. The reference value is 13 h 24 min for a 100-year, 24-hour storm on an Intel i9 workstation with an RTX 3090 and 64 GB of memory [5] (Table −5). The two numbers are not like-for-like: one is neural-network inference on an image patch, the other a physics-based simulation of a full design storm. The comparison explains why a screening layer can run repeatedly; it does not show that the surrogate is as accurate. The 1.2 s value is the figure shown by the dashboard.

Table -5: Run-time comparison

Configuration	Time	Source
Full 1D-2D simulation, 100-year 24 h storm (Intel i9, RTX 3090, 64 GB)	13 h 24 min	[5]
This prototype: attention U-Net inference, as shown on the dashboard	1.2 s	Dashboard readout
This prototype: observation-to-alert time, end to end	Not reported	-

5.10 Interface functions observed

Table −6 lists the functions visible in the captures and where each appears.

Table -6: Interface functions observed in the captures

Function	View	Figure
Zone list with confidence ring, depth, elevation, drain efficiency; Monitored Basins and Risk Ranking tabs	Mission Control	Fig. 3

Function	View	Figure
3D twin with 2D Map, 3D Twin, Risk View, Grad-CAM, Solid and Glow modes	Mission Control, Zone Analysis	Figs. 3, 4
Cognition feed with Run Cycle control; Upgrade Priorities tab	Mission Control	Fig. 3
What-if simulator: precipitation anomaly and stormwater drainage controls	Mission Control	Fig. 3
Timeline playback from −24 h to Now; historical replay	Mission Control	Fig. 3
Source and agreement badge (ISRO EOS-04, IoU)	Mission Control, Zone Analysis	Figs. 3, 4
12-hour rainfall and flood-depth projection	Zone Analysis	Fig. 4
Explainable-AI attribution panel	Zone Analysis	Fig. 4
Alert log with filters, severity, dispatch status and acknowledge	Alerts	Fig. 5
Role badge (NDMA officer)	All views	Figs. 3-5
Chat with context, shortcut questions, language selector and voice input	All views	Fig. 6

6. DISCUSSION AND LIMITATIONS

What the demonstration shows is integration and traceability. A single cycle can be followed from the orchestrator entry, through the explanation text, to the alert record, and the values agree because they come from shared state. The confidence rule, the depth-dependent severity, the hospital escalation and the role badge are all visible in working form. These are properties of the design, and they are what the captures can support.

What it does not show is accuracy. No labelled flood event was used, so the paper reports no precision, recall, F1 or measured IoU for the segmentation model, and the agreement scores in the interface come from the demonstration layer. The retry rule has not been counted over a test set, so its success rate is unknown, and the chat panel's answers have not been scored for grounding or refusal. The 1.2 s figure is a single readout, and an end-to-end observation-to-alert time has not been reported.

The design also carries trade-offs. The depth surrogate has not been validated against a hydrodynamic model, so its depths are screening estimates. The retry rule helps only when the second acquisition adds information; if radar and optical data are both obscured or the viewing geometry is poor, the zone is escalated, which is the intended outcome. The rule limits the harm of the generalisation gap seen in [6] by flagging uncertain predictions, but it does not close that gap, and neural-network confidence is not automatically calibrated, so the 0.6 threshold needs tuning on held-out events.

Data availability is a practical constraint. EOS-04 coverage and revisit for a given area are not guaranteed, so fallback sources add heterogeneity. Heights obtained from type-based estimates carry error, which is why each is labelled with its source, and they are not suitable for damage estimation. Only a handful of zones in two cities were shown, and the chat model is small and trained on a limited domain set.

Automatic alerts raise accountability questions that a technical design cannot settle. A false alarm has a cost, and Hlal et al. [1] point to governance and equity concerns around digital twins. We therefore keep a human acknowledgement step, withhold alerts on low confidence, and keep an audit log of every decision. Threats to validity include label noise in the reference water-spread product, timing mismatch between satellite passes and flood peaks, and small test sets.

7. CONCLUSIONS AND FUTURE WORK

We described a cognitive digital twin for urban flood prediction in which seven agents sense, predict, simulate, explain, alert, coordinate and converse. Its distinguishing feature is that prediction confidence controls what the system does next, with bounded retries and escalation to a human. The framework is built around ISRO data products with open fallbacks, and the prototype presents the twin in a browser client that adapts to the device. A demonstration on zones in Bengaluru and Mumbai showed the views, the explanation text and the alert log agreeing with one another.

The next step is a quantitative evaluation: running the pipeline on live sources for labelled flood events, reporting precision, recall, F1 and IoU, and counting what the retry rule changes against a version without it. Further work includes a field trial with disaster-management staff, calibration of the confidence score and alert depth, adversarial testing of the chat refusal behaviour, additional cities, and coupling with coastal surge models, as the authors of [5] suggest.

CONFLICT OF INTEREST

The authors declare no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

DATA AND CODE AVAILABILITY

The prototype source code and the demonstration data generator are held by the authors and are available from the corresponding author on reasonable request.

REFERENCES

1. Hlal, M., Baraka Munyaka, J.-C., Chenal, J., Azmi, R., Diop, E.B., Bounabi, M., Ebnou Abdem, S.A., Almouctar, M.A.S., Adraoui, M.: Digital twin technology for urban flood risk management: A systematic review of remote sensing applications and early warning systems. Remote Sens. 17(17), 3104 (2025)
2. Roy, P.: It's been 40 years! Update India's flood map now. Down To Earth. https://www.downtoearth.org.in/news/climate-change/it-s-been-40-years-update-india-s-flood-map-now-78166 (accessed 11 Oct 2026)
3. The Quint: As Bengaluru loses lakes, green cover, impact of urban floods takes a toll. https://www.thequint.com/climate-change/as-bengaluru-loses-lakes-green-cover-impact-of-urban-floods-takes-a-toll (accessed 11 Oct 2026)
4. Grieves, M.: Origins of the digital twin concept. White paper (2016)
5. Mandal, D., Zou, L., Wadhwa, A., Wilkho, R.S., Cai, Z., Zhou, B., Ye, X., Newman, G., Gharaibeh, N., Güneralp, B.: FlowsDT: A geospatial digital twin for navigating urban flood dynamics. Comput. Environ. Urban Syst. 126, 102414 (2026)
6. Fakhri, F., Gkanatsios, I.: Quantitative evaluation of flood extent detection using attention U-Net case studies from Eastern South Wales Australia in March 2021 and July 2022. Sci. Rep. 15, 12377 (2025)
7. Ge, C., Qin, S.: Urban flooding digital twin system framework. Syst. Sci. Control Eng. 13(1), 2460432 (2025)
8. Li, B., et al.: Disaster management in the era of agentic AI systems: A vision for collective human-machine intelligence for augmented resilience. arXiv:2510.16034 (2025)
9. Fan, C., Zhang, C., Yahja, A., Mostafavi, A.: Disaster City Digital Twin: A vision for integrating artificial and human intelligence for disaster management. Int. J. Inf. Manag. 56, 102049 (2019)

10. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Proc. MICCAI, pp. 234–241 (2015)
11. Oktay, O., et al.: Attention U-Net: Learning where to look for the pancreas. arXiv:1804.03999 (2018)
12. Bathe, K.D., Patil, N.S.: ConvExNet: Deep learning-based flood detection utilizing Shapley additive explanations. J. Earth Syst. Sci. 134, 99 (2025)
13. Situ, Z., et al.: Improving urban flood prediction using LSTM-DeepLabv3+ and Bayesian optimization with spatiotemporal feature fusion. arXiv:2304.09994 (2023)
14. Balamurugan, R., Choudhary, K., Raja, S.: Prediction of flooding due to heavy rainfall in India using machine learning algorithms: Providing advanced warning. IEEE Syst. Man Cybern. Mag. 8(4), 26–33 (2022)
15. Kumar, V., et al.: Machine learning applications in flood forecasting and predictions, challenges, and way-out in the perspective of changing environment. AIMS Environ. Sci. 12(1), 72–105 (2025)
16. Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. In: Proc. NeurIPS, pp. 4765–4774 (2017)
17. Dar, S.S., et al.: An explainable deep neural network with frequency-aware channel and spatial refinement for flood prediction in sustainable cities. arXiv:2509.08003 (2025)
18. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: Proc. IEEE Int. Conf. Comput. Vis. (ICCV), pp. 618–626 (2017)
19. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. arXiv:2106.09685 (2021)
20. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Advances in Neural Information Processing Systems 33 (NeurIPS 2020)