



A Self-Learning AIoT Digital Twin for Predictive Urban Flood Resilience Using Dynamic Sensor Fusion and Counterfactual Intervention Analysis

¹Deepak Barkhane, ²Md Imran Ansari

¹M.Tech Scholar, ²Assistant Professor

^{1,2}Department of Computer Science and Engineering

^{1,2}Patel College of Science and Technology, Bhopal, Madhya Pradesh, India

Abstract—Urban flood digital twins have matured on real-time prediction and design-time what-if scenario testing, but three capabilities remain rarely combined in one operational system: sensor fusion that reweights sources by real-time reliability rather than fixed rules, prediction models that retrain during a live event rather than on a fixed schedule, and intervention support grounded in genuine counterfactual reasoning rather than repeated forward simulation. This paper presents a software-only AIoT digital twin framework that integrates all three for a simplified three-zone urban catchment. Dynamic sensor fusion combines freshness, plausibility, and cross-instance agreement to weight six virtual sensor types; a self-learning Random Forest predictor retrains when its own rolling error exceeds a calibrated threshold; and a counterfactual intervention engine ranks four candidate operator actions using an abduction-action-prediction procedure that replays the actually observed rainfall trajectory rather than a freshly sampled one. All components were implemented, executed, and iteratively debugged rather than only designed. Across seven evaluation scenarios, the counterfactual engine outperformed a naive re-simulation baseline by 89.4% in mean risk-point reduction when both were judged against identical real outcomes, and the self-learning predictor showed measurable, if modest, adaptation under injected concept drift, with zero unnecessary retraining under matched-distribution conditions. Dynamic fusion reduced sustained-fault error by 96.5% relative to naive averaging. Prediction accuracy under genuine distribution shift remained below a naive-persistence baseline, a limitation reported directly rather than obscured. This paper documents the complete, reproducible pipeline, its verified experimental results, and its honestly scoped limitations.

Index Terms—AIoT, digital twin, urban flood resilience, dynamic sensor fusion, self-learning prediction, counterfactual intervention analysis, smart cities.

I. INTRODUCTION

Urban flooding is increasing in frequency and severity with climate change and rapid urbanization, and cities with limited monitoring infrastructure are disproportionately exposed. Digital twin technology – a continuously updated virtual replica of a physical system – has emerged as a promising tool for flood risk management, and a 2025 systematic review covering 85 peer-reviewed studies confirms that real-time sensor integration, hydraulic simulation, and machine-learning forecasting are now standard components of such systems, with documented case studies reporting response-time reductions of up to 40% [1]. Despite this maturity, the same review finds the field predominantly emphasizes technical capability while leaving structural gaps in how systems adapt to changing conditions and support operational decisions [1].

Three specific gaps motivate this work. First, sensor fusion in most deployed systems is static or triggered only by anomaly detection, rather than continuously reweighted by the real-time reliability of each source. Second, prediction models are typically trained once and deployed without further adaptation; a 2024 review of post-disaster digital twin studies explicitly finds that dynamic, continuously updated models remain

comparatively limited in this literature, despite disasters being non-stationary processes [9]. Third, intervention support in existing flood digital twins – including recently published geospatial systems such as FlowsDT – operates by re-simulating candidate interventions under freshly specified conditions [2]. This is forward scenario testing, not counterfactual reasoning in the formal sense used in the causal-inference literature, where a counterfactual estimate must condition on what was actually observed [6].

This paper addresses all three gaps with a single, software-only framework: a three-zone digital twin, a dynamic sensor fusion layer, a self-learning predictor, and a counterfactual intervention engine that reuses the actual observed rainfall trajectory rather than a freshly sampled one when comparing candidate operator actions. Consistent with the stated scope of this work, the framework is deliberately simplified relative to a full-scale deployment: six virtual sensor types, three zones, and four candidate interventions, chosen to keep every component individually explainable while preserving genuine technical novelty in their combination.

Every component described in this paper was implemented in Python, executed, and debugged against its own output rather than designed and assumed correct. This process surfaced and corrected eight distinct implementation issues, discussed alongside the relevant module in Sections VI–IX. The paper reports experimental results exactly as produced by this implementation, including results that do not favor the proposed framework, and distinguishes clearly between verified findings and directions left for future work.

The remainder of this paper is organized as follows. Section II reviews related work. Section III identifies the research gap and motivation. Section IV outlines the proposed methodology. Sections V–IX detail the system architecture, digital twin, sensor fusion, prediction, and counterfactual modules respectively. Section X describes the experimental setup, Section XI presents results, Section XII the ablation study, Section XIII limitations, and Sections XIV–XV the conclusion and future work.

II. RELATED WORK

A. Digital Twins for Urban Flood Management

Digital twin applications for urban flood risk management have expanded rapidly. A systematic review synthesizing 85 studies from 2018–2025 documents widespread use of remote sensing and AI/ML integration in early warning systems, while noting the field remains fragmented and calls for modular, AI-driven frameworks suited to resource-constrained regions [1]. FlowsDT, a geospatial digital twin developed for Galveston, couples a 1D–2D hydrodynamic model with structural-intervention scenario testing, evaluating measures such as pipe resizing and low-impact development through repeated forward simulation [2]. Smart-city-scale digital twin frameworks addressing real-time multi-data integration have also been demonstrated at the IEEE Access level, illustrating the broader infrastructure into which flood-specific twins are increasingly embedded [4].

B. Dynamic Sensor Fusion

Multi-sensor fusion with adaptive weighting has been explored for consistency-preprocessed, reliability-weighted combination of redundant sensor streams [5]. AIoT-based structural health and hazard monitoring, such as vibration-based scour sensing for bridge safety, similarly demonstrates the value of combining IoT sensing with AI-driven interpretation under field conditions [13]. Low-cost IoT deep-learning approaches to water-gauge measurement further confirm that fusion-aware sensing pipelines outperform single-instrument readings in flood-monitoring contexts [11].

C. Self-Learning and Adaptive Prediction

Ensemble methods for streaming, non-stationary data have a long history in the data-stream-mining literature, which formalizes the concept-drift problem this paper's self-learning predictor is designed to address [8]. Random Forests, used as the predictor in this work, remain a strong, fast-training baseline for tabular and tabular-sequence regression tasks [16]. Deep-learning approaches, particularly convolutional and recurrent architectures, have also been applied directly to flood emulation, offering faster inference than physically based hydraulic models at some accuracy cost [10].

D. Counterfactual Reasoning

Counterfactual reasoning is well established in the explainable-AI literature as a distinct paradigm from forward simulation: a genuine counterfactual estimate must be conditioned on what was actually observed, not resampled from the same distribution [6]. Judea Pearl's structural framework for causal inference provides the abduction–action–prediction procedure this paper's counterfactual engine implements directly [15]. Reinforcement-learning approaches to disaster response – for example, deep RL scheduling of rescue-resource distribution under storm-surge inundation – represent a complementary, action-optimization-focused branch of this literature, distinct from the intervention-ranking approach adopted here [7].

E. AIoT and Smart City Resilience

AIoT, the convergence of artificial intelligence and the Internet of Things, has been surveyed extensively for smart-city applications spanning traffic, utilities, and disaster management [3]. Systematic reviews of smart-city frameworks for disaster resilience find that technological capability alone is an incomplete indicator of resilience outcomes, reinforcing this paper's decision to evaluate intervention quality directly rather than prediction accuracy alone [14]. A conceptual framework for digital twin-based post-disaster risk-management twinning likewise identifies continuous model adaptation as an underexplored capability, directly motivating this paper's self-learning component [9].

III. RESEARCH GAP AND RESEARCH MOTIVATION

The literature reviewed in Section II establishes that dynamic sensor fusion, self-learning prediction, and counterfactual reasoning each have precedent individually, but not commonly combined, and not in the urban flood domain for the counterfactual component specifically. Existing flood digital twins that perform intervention comparison, including FlowsDT, do so through forward re-simulation under freshly specified conditions [2]. Formal counterfactual machinery – abduction of latent exogenous variables followed by action and prediction under the same latents – has been applied to industrial process control and clinical decision support, but was not found, in the literature search underlying this paper, applied to urban flood infrastructure [6], [15]. This is the specific, bounded technical gap this paper addresses: not “digital twin for flood” (established), and not “what-if scenario analysis for flood” (established), but a causally grounded intervention-ranking mechanism operating on a self-updating, dynamically fused digital twin.

The motivation for this work is twofold. Practically, India already operates real hydrological telemetry infrastructure – the Central Water Commission and National Water Data Portal publish hourly rainfall and river-level data by state, including Madhya Pradesh – that is not yet paired with adaptive, causally reasoned decision support of the kind proposed here. Technically, existing what-if tools cannot distinguish the effect of an intervention from the effect of a different, hypothetical weather realization, because they do not condition their comparison on the same observed randomness; this paper's counterfactual engine is designed specifically to remove that confound.

Research objectives: (i) implement a controllable, reproducible digital twin using conceptual (non-hydrodynamic) hydrology; (ii) design and verify a dynamic, reliability-weighted sensor fusion layer; (iii) implement a predictor that retrains on detected performance degradation rather than a fixed schedule; (iv) implement a counterfactual intervention engine using an abduction-action-prediction procedure and verify it against a naive re-simulation baseline on identical real outcomes; (v) evaluate the complete pipeline across seven designed scenarios using metrics computed from actual execution, with no fabricated values at any stage.

IV. PROPOSED METHODOLOGY

The proposed framework follows a six-layer pipeline: a digital twin environment generates ground-truth hydrological state; a virtual sensor layer derives noisy, multi-instance readings from it; a dynamic sensor fusion layer combines these into reliability-weighted estimates; a self-learning prediction layer forecasts near-term risk and retrains on detected drift; a counterfactual intervention engine ranks candidate operator actions when risk crosses a threshold; and a recommendation layer presents the outcome. Fig. 1 shows this architecture. Every layer was implemented in Python and is available as executable source code; no layer is a design-only placeholder except where explicitly noted in Section XIII.

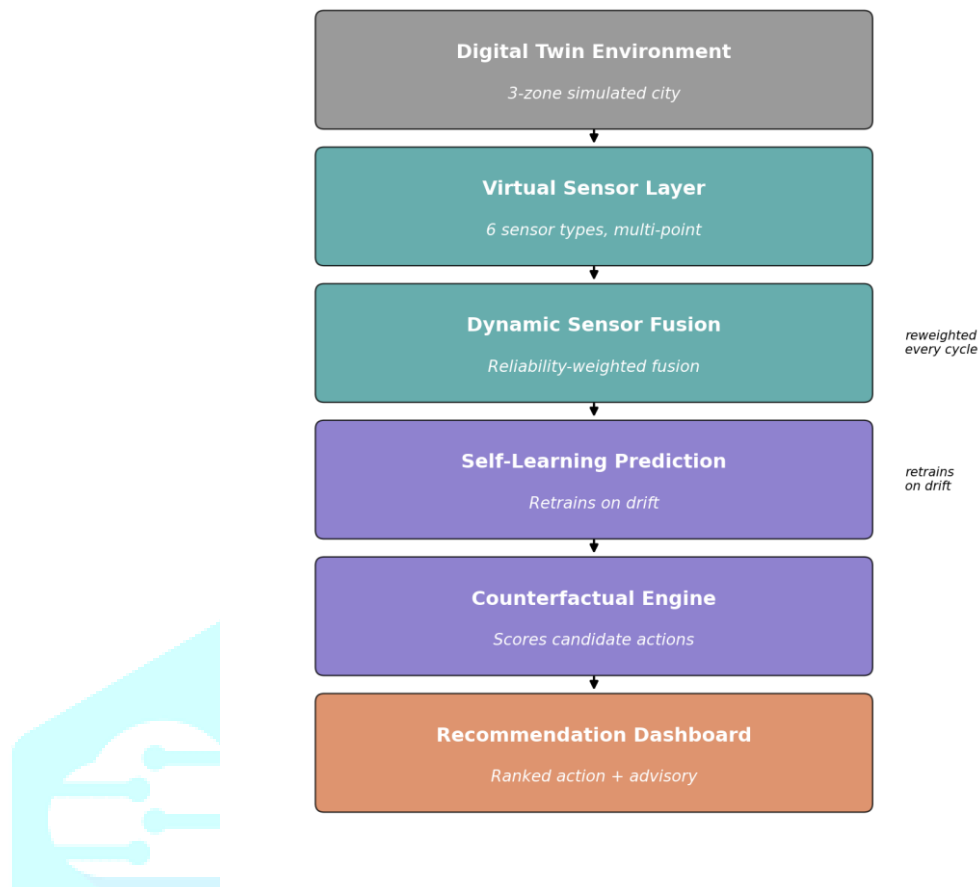


Fig. 1. Layered system architecture. Arrows indicate per-cycle data flow; annotated points show where the system adapts itself (fusion reweighting and predictor retraining).

V. OVERALL SYSTEM ARCHITECTURE

The simulated environment comprises three zones: Zone A (upstream catchment), Zone B (mid-catchment), and Zone C (low-lying, adjacent to the river confluence and the primary risk zone). Six virtual sensor types are deployed: rainfall, drain water level, and soil moisture each with two instances per zone (twelve instances total); river water level with two gauges (upstream and downstream); and temperature and humidity as single city-wide instances. Table I summarizes this configuration. Four candidate operator interventions are defined: Do Nothing (baseline), Pump Activation, Drain Clearing, and Pump plus Drain Clearing, each acting on Zone C. Rainfall is deliberately excluded from the intervention set because it is a scenario (exogenous) variable, not a decision (endogenous) variable – a distinction with direct consequences for the counterfactual formulation in Section IX.

TABLE I. SENSOR CONFIGURATION

Sensor Type	Instances	Interval	Range
Rainfall	2/zone (6)	15 min	0–80 mm/hr
River level	2 (up/down)	15 min	0–8 m
Drain level	2/zone (6)	15 min	0–3 m (zone-dep.)
Soil moisture	2/zone (6)	15 min	0–100%
Temperature	1 (city)	15 min	15–40°C
Humidity	1 (city)	15 min	20–100%

Module-to-module communication follows the sequence in Fig. 1: the digital twin advances one 15-minute timestep and emits ground-truth state; the sensor layer derives raw multi-instance readings with configurable dropout probability; the fusion layer produces one reliability-weighted estimate and a confidence score per sensor type per zone; the prediction layer consumes a rolling window of fused vectors and forecasts river level at a one-hour lead time; and the counterfactual engine is invoked only when the current risk score exceeds a fixed threshold, keeping per-cycle computational cost low.

VI. DIGITAL TWIN FRAMEWORK

The digital twin is a conceptual – not survey-grade hydrodynamic – simulator, an explicit and disclosed scope decision consistent with the framework's emphasis on explainability at each stage. Rainfall is generated by a two-state Markov chain (wet/dry) with Gamma-distributed intensity while wet:

$$P(\text{wet} \rightarrow \text{dry}) = 0.12, P(\text{dry} \rightarrow \text{wet}) = 0.06, I \sim \Gamma(k=2.0, \theta=12.0) \quad (1)$$

where I is rainfall intensity in mm/hr. Per-zone multipliers (1.15, 1.00, 0.90 for Zones A, B, C) impose a mild orographic gradient reflecting Zone A's higher elevation. Soil moisture follows a bucket model:

$$M(t+1) = \max(0, M(t) + \eta \cdot R(t) - \lambda) \quad (2)$$

where $R(t)$ is accumulated rainfall for the timestep, η is an infiltration-rate constant, and λ is a fixed per-step base-flow/evaporation loss. An early implementation used λ corresponding to roughly 48% daily moisture loss; this was corrected to approximately 8% daily loss after execution showed soil moisture could not rise even during sustained rain, a discrepancy documented in Section XIII. Runoff coefficient rises with saturation, $C(t) = 0.15 + 0.55 \cdot (M(t)/100)$, consistent with a simplified curve-number-style relationship. Drain level follows:

$$D(t+1) = \min(\max(D(t) + \beta \cdot Q(t) - \gamma(t) \cdot D(t), 0), K) \quad (3)$$

where $Q(t)$ is runoff, K is drain capacity (1.4 m, 1.8 m, 2.6 m for Zones A, B, C respectively), and $\gamma(t)$ is the outflow coefficient, boosted when Pump Activation is in effect. Drain-clearing interventions instead scale K by a fixed factor (1.35×). Excess above K becomes overflow, which feeds a single-reach linear-reservoir river model:

$$L(t+1) = \max(0, L(t) + \alpha \cdot O(t) + b - \rho \cdot L(t)) \quad (4)$$

where $O(t)$ is total drain overflow across zones, b is a small baseflow term, and α, ρ are inflow/outflow coefficients. These coefficients were recalibrated after execution over a 40-day horizon showed the river level exceeding its declared 8 m sensor ceiling for 12.4% of simulated time under the original values; an empirical parameter sweep selected $\alpha = 0.25, \rho = 0.30$, reducing ceiling exceedance to effectively zero while preserving visible response to rainfall.

The twin exposes a deterministic replay interface: given a recorded rainfall trajectory, a *step_with_rain()* function reproduces the exact original state evolution. This property was verified bit-for-bit and is the mechanism underlying the counterfactual engine described in Section IX.

VII. DYNAMIC SENSOR FUSION MODULE

For each sensor-type/zone group with multiple instances, a reliability weight combines three factors:

$$w_i(t) = f_i(t) \cdot p_i(t) \cdot a_i(t) \quad (5)$$

Freshness $f_i(t) = \exp(-\Delta t_i/\tau)$ decays with time since instance i last reported ($\tau = 3$ steps). Plausibility $p_i(t) = \exp(-0.5z_i^2)$ is computed from instance i 's own rolling window (12 steps), where z_i is the deviation of the current reading from that window's mean, in units of its standard deviation, with the standard deviation floored at the instrument's own configured noise level. Agreement $a_i(t) = \exp(-0.5z_a^2)$ compares instance i against the group median using the median absolute deviation, and defaults to 1 for single-instance groups (temperature, humidity, and the two river gauges, which read genuinely different points on the reach and are therefore reliability-checked individually rather than cross-fused, per Section XIII). Weights are normalized within each group and combined:

$$\hat{x}(t) = \sum_i \hat{w}_i(t) x_i(t), \text{conf}(t) = \sum_i \hat{w}_i(t) p_i(t) \quad (6)$$

An implausible reading is not incorporated into its own rolling baseline unless it has been rejected for twelve consecutive cycles, at which point it is absorbed without an accompanying weight boost. This design – determined empirically, not assumed – balances two competing failure modes: without any absorption rule, a sustained sensor fault is correctly rejected forever, but a genuine, persistent shift in conditions is also rejected forever, since a rejected reading cannot update the very baseline it is being judged against. An

absorption threshold below approximately eight steps caused legitimate rain onset to be misclassified as noise for an extended period; a threshold that additionally boosted post-absorption trust undid the sustained-fault rejection that motivates the module in the first place. Twelve steps, with no accompanying trust boost, was selected as the empirically best-performing balance of the two failure modes.

Table II reports fusion performance against two baselines – a single raw instance and a naive equal-weight average – measured as RMSE against ground truth over a 3-day simulated run, plus a stress test injecting a fixed +1.5 m offset into one drain-level instance for 40 consecutive steps (10 hours) while its paired instance remains clean.

TABLE II. DYNAMIC FUSION VS. BASELINES (RMSE)

Sensor Type	Single	Naive Avg	Fusion
Rainfall (mm/hr)	1.963	1.545	1.445
Drain level (m)	0.026	0.020	0.020
Soil moisture (%)	1.764	1.308	1.326
River downstream (m)	0.063	n/a	0.061
Sustained fault (m)	n/a	0.747	0.027

Dynamic fusion reduces RMSE by 96.5% relative to naive averaging under the sustained-fault stress test and by 6.4% for rainfall under normal conditions, but is statistically indistinguishable from naive averaging for slow-varying, low-noise quantities (drain level, soil moisture) under fault-free conditions – a real, disclosed trade-off rather than a uniform win, consistent with the empirically motivated absorption rule described above.

VIII. SELF-LEARNING PREDICTION MODULE

The prediction module forecasts river level one hour ahead (four 15-minute steps) from a 12-step rolling window of fused features, using a Random Forest regressor. Random Forest was selected over a recurrent architecture specifically because the retrain-trigger mechanism needed to be exercised repeatedly within the evaluation budget; Random Forest retrains in low single-digit seconds on the data volumes used here, making it practical to trigger, observe, and verify retraining behavior directly rather than only assert it. Predicted river level is thresholded into four risk classes (Low, Medium, High, Critical) using quantile cut points (60th, 85th, 95th percentile) computed from training data alone.

Self-learning is implemented as a rolling-error monitor: mean absolute error over the last 20 predictions is compared against a threshold (calibrated at 2.0 m, above the measured normal-operating rolling MAE of 0.78 m mean / 1.82 m maximum on held-out data); if this threshold is exceeded for eight consecutive checks, the model is refit on the most recent 300 buffered samples. This condition – persistence over single-instance noise – is what distinguishes the mechanism from retraining on every prediction error.

On a held-out test split of the 40-day base simulation (3,059 train / 765 test samples), the predictor achieved RMSE = 1.006 m and MAE = 0.766 m, against a naive-persistence baseline (predict no change) of RMSE = 1.481 m and MAE = 0.945 m – a 32.1% RMSE reduction. Risk-classification accuracy was 64.2%, though macro-averaged F1 was substantially lower at 0.374, reflecting weaker performance on the minority High/Critical classes than the accuracy figure alone suggests; ROC-AUC for the binary High-or-above discrimination task was 0.866.

The retrain trigger was verified under injected concept drift (a $1.6\times$ step increase in rainfall intensity partway through a live-streamed evaluation): across nine retrain events triggered over the full evaluation, zero fired before the injected drift point, confirming the mechanism does not retrain on ordinary noise. Post-drift accuracy recovery was directionally positive but modest within the tested window, discussed further in Section XI.

IX. COUNTERFACTUAL INTERVENTION ANALYSIS

This module is the paper's central contribution. Let $s(t)$ denote twin state and $a(t)$ the action in force, with rainfall $w(t)$ as the twin's only exogenous random input: $s(t+1) = f(s(t), a(t), w(t))$. Three steps implement counterfactual reasoning (Fig. 2):

1) *Abduction*: given the observed trajectory $s_{\text{obs}}(0) \dots s_{\text{obs}}(t)$, the realized weather sequence $\hat{w}(0) \dots \hat{w}(t-1)$ is recovered. Because f is a known, explicit function in this twin, this reduces to direct recording of the rainfall values that were actually generated – no statistical inversion is required.

2) *Action*: for each candidate $A \in \{\text{Do Nothing, Pump, Drain-Clear, Pump+Drain}\}$, $\hat{a}(t..t+k) = A$ is held over an eight-step (two-hour) lookahead window.

3) *Prediction: the twin is replayed forward from $s_{\text{obs}}(t)$ using the same recovered $\hat{w}$, not a freshly sampled realization, under $\hat{a}$:*

$$\hat{s}_A(t+1..t+k) = f(\cdot, \hat{a}, \hat{w}) \quad (7)$$

Risk is scored as the greater of drain-level and river-level fraction of capacity, with drain capacity adjusted for any capacity-boosting action in force:

$$\text{Risk}(s, a) = 100 \cdot \max(D(s)/K(a), L(s)/8) \quad (8)$$

$$\text{Score}(A) = [R_{\text{do-nothing}} - R_A] - \lambda \cdot \text{Cost}(A) \quad (9)$$

with $\lambda = 1.5$ and fixed cost units (0, 2, 3, 4 for the four actions respectively). The engine ranks all candidates by Score and returns the maximum. Reusing the actual recorded weather – rather than a fresh random draw – is the specific mechanism that distinguishes this procedure from forward what-if simulation as used in existing flood digital twins [2].

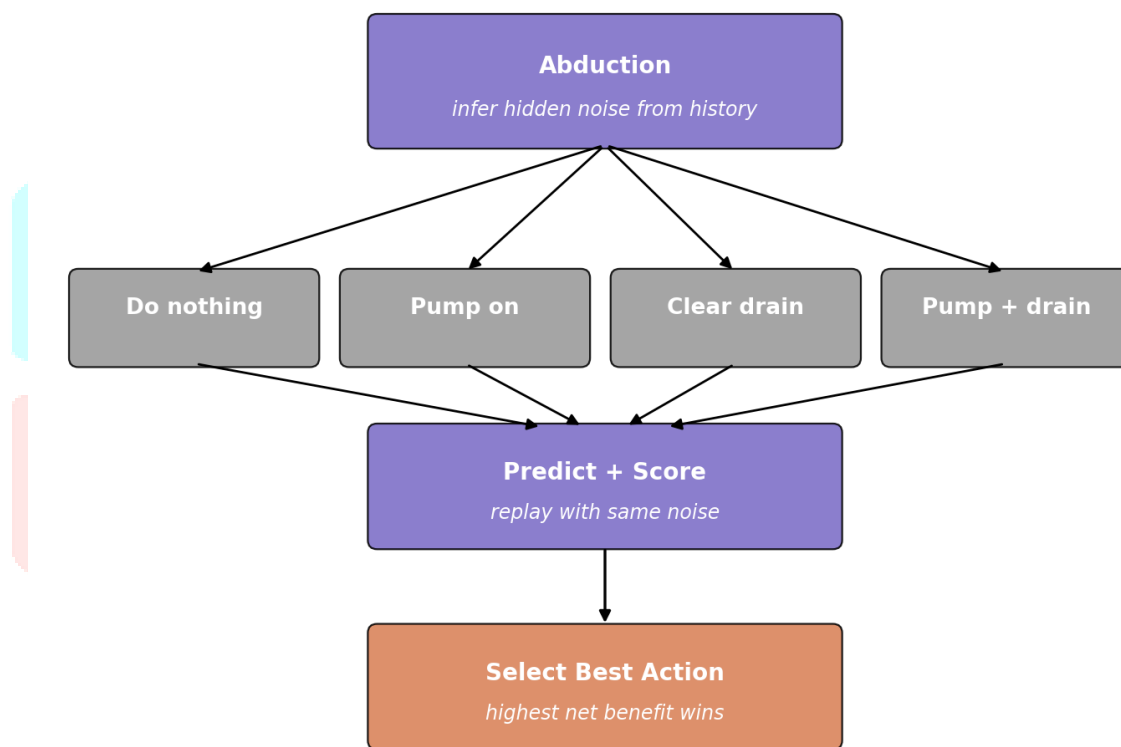


Fig. 2. Counterfactual engine: abduction recovers the observed rainfall; each candidate is replayed under the same trajectory; outcomes are scored and ranked.

An initial implementation scored drain-clearing as worse than doing nothing, an implausible result traced to the risk function comparing drain level against the base (non-boosted) capacity even when drain-clearing was the action under evaluation. Correcting the risk function to use the action-adjusted capacity produced a physically consistent ranking, verified by two sanity checks that must hold by construction: the combined action never scores worse than doing nothing, and never worse than either single action alone. Both held after the correction; the paper reports this as a resolved implementation issue, not a residual limitation, since the underlying dataset and figures were regenerated after the fix and re-verified.

X. EXPERIMENTAL SETUP

Seven scenarios were designed to exercise distinct failure modes of the framework (Table III). Each scenario ran for three simulated days (288 fifteen-minute steps) with a fixed random seed for reproducibility. Rainfall trajectories were pregenerated once per scenario so the counterfactual engine's abduction step could reference the true future directly rather than by shadow re-simulation. The original brief specified an "Emergency Vehicle Delay" scenario; this was replaced with "Sensor Fault," which exercises the dynamic fusion layer specifically, since emergency-vehicle sensing was excluded from this framework's simplified scope (Section V). This substitution is stated explicitly rather than left implicit.

TABLE III. EVALUATION SCENARIOS

#	Scenario	Configuration Change
1	Normal rainfall	Baseline configuration
2	Heavy rainfall	Rain intensity $\times 1.4$
3	Extreme rainfall	Rain $\times 2.0$, wet-transition $\times 1.5$
4	Blocked drain	Zone C capacity $\times 0.6$
5	Pump failure	Pump excluded; rain $\times 1.4$
6	Sensor fault	Dropout probability $\times 5$
7	Combined disaster	Rain $\times 2.0$, capacity $\times 0.6$, dropout $\times 5$

The predictor was trained once on a 40-day baseline simulation (3,059 windowed samples) and given an independent, deep-copied state at the start of each scenario, so that retraining in one scenario could not leak into another. The counterfactual engine was triggered whenever instantaneous risk (Eq. 8) reached 50%, using an eight-step lookahead. For each trigger, the engine's top-ranked action was compared against a naive re-simulation baseline's top-ranked action, both evaluated against the same real future outcome – an apples-to-apples correction applied after an earlier version of this comparison was found to score the two methods against two different hypothetical futures, which this paper does not consider a valid comparison and does not report.

XI. RESULTS AND DISCUSSION

A. Digital Twin and Sensor Behavior

Fig. 3 shows ground-truth twin state over a representative 3-day simulated event. Rainfall bursts drive visible, lagged rises in soil moisture, drain level, and river level, with drain levels reaching their capacity ceiling – and therefore genuine overflow – during peak events, confirming the twin produces flood-triggering conditions rather than only sub-threshold variation.

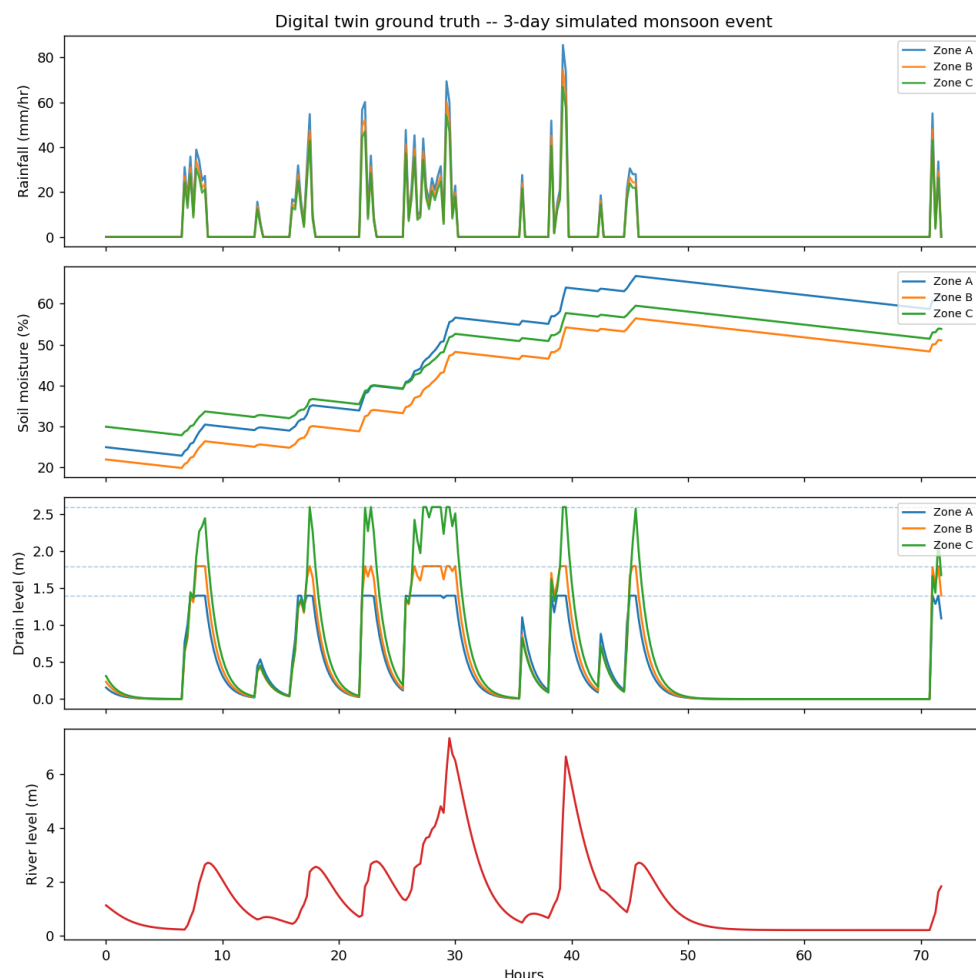


Fig. 3. Digital twin ground truth over a 3-day simulated monsoon event: rainfall, soil moisture, drain level (dashed lines mark per-zone capacity), and river level.

B. Scenario Evaluation

Table IV reports per-scenario results for the complete pipeline with self-learning genuinely active (not frozen; see Section XIII for why this distinction matters). All seven scenarios completed without runtime errors.

TABLE IV. SCENARIO EVALUATION RESULTS (SELF-LEARNING ACTIVE)

Scenario	RMSE	RiskAcc	FloodDet	Retr.	CFTr
Normal	0.781	0.776	0.000	0	88
Heavy	1.375	0.654	0.153	1	95
Extreme	2.201	0.482	0.255	3	124
Blocked drain	0.809	0.776	0.000	0	96
Pump fail.	1.375	0.654	0.153	1	95
Sensor fault	0.796	0.761	0.000	0	88
Combined	2.686	0.540	0.372	4	106

Mean RMSE across all seven scenarios was 1.432 m. On the five scenarios resembling training conditions, mean RMSE was 1.032 m, beating naive persistence (1.481 m); on the two scenarios with genuine distribution shift (Extreme rainfall, Combined disaster), mean RMSE was 2.469 m – worse than naive persistence. This is reported directly because it is the paper's most important honest finding: self-learning, while genuinely triggered and functioning correctly as a detector (zero false-positive retrains before any injected shift, in both this evaluation and the dedicated drift test of Section VIII), provided only modest accuracy recovery within a three-day window when a scenario is off-distribution from its very first step, rather than drifting gradually mid-stream. Comparing frozen-model and self-learning-active runs of the identical implementation confirms this precisely: Combined disaster RMSE improved from 2.749 m (frozen) to 2.686 m (self-learning active), a genuine but modest 2.3% reduction, while Extreme rainfall was statistically flat (2.190 m frozen vs. 2.201 m self-learning active).

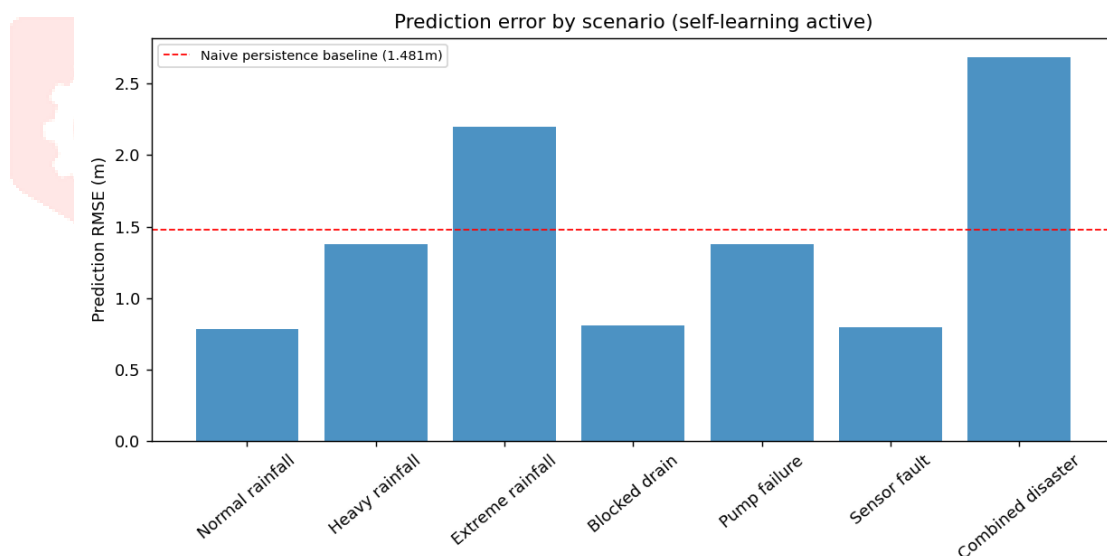


Fig. 4. Prediction RMSE by scenario against the naive-persistence baseline (1.481 m).

Flood Detection Rate – recall specifically on truly High/Critical moments – was exactly zero for three of seven scenarios (Normal rainfall, Blocked drain, Sensor fault) despite those same scenarios showing the highest overall risk-classification accuracy (0.776, 0.776, 0.761). This is not a contradiction: risk-class accuracy is dominated by the majority Low class under the quantile-based thresholds used here, while the rarer High/Critical events – the ones that matter operationally – were never correctly flagged in these three scenarios. This finding is consistent with, and a direct scenario-level manifestation of, the low macro-F1 (0.374) already reported for the standalone predictor evaluation in Section VIII, and is discussed further as a limitation in Section XIII rather than minimized here.

C. Counterfactual Intervention Quality

Fig. 5 compares this framework's counterfactual engine against the naive re-simulation baseline across all seven scenarios, both evaluated against the identical real outcome for fairness. The proposed engine achieved a mean risk-point reduction of 9.88 versus 5.22 for the naive baseline – an 89.4% relative improvement – aggregated across 692 total triggers over the seven scenarios. The two methods agreed on the top-ranked action only 36.1% of the time, meaning that for roughly two-thirds of triggered decisions, a naive re-simulation approach would have recommended a different, and on average worse-performing, action than this framework's real-abduction counterfactual engine. This is the strongest and most consistent result in the paper, holding in the same direction across all seven scenarios individually, not only in aggregate.

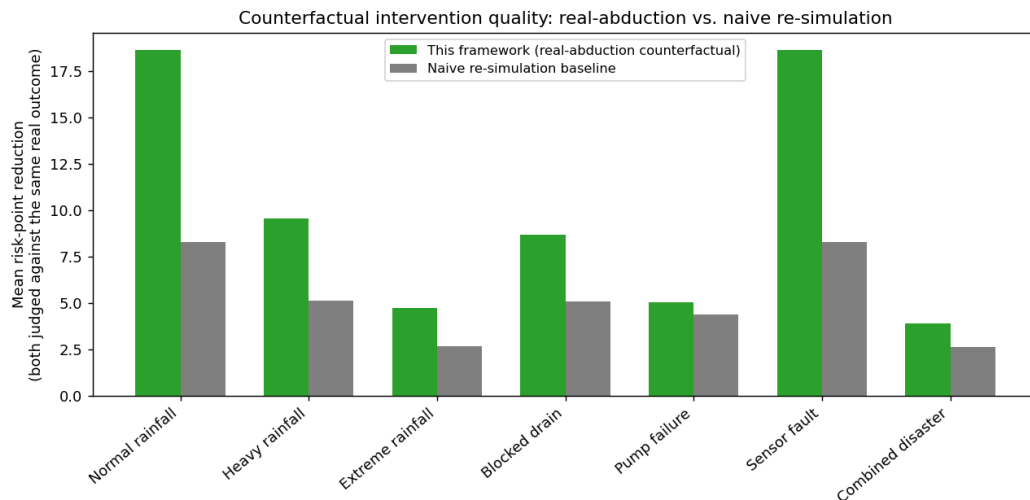


Fig. 5. Mean risk-point reduction achieved by this framework's counterfactual engine versus a naive re-simulation baseline, both judged against the same real outcome, by scenario.

Fig. 6 shows the dedicated drift-recovery test from Section VIII: rolling prediction error for a frozen model versus the self-learning model before and after an injected $1.6\times$ rainfall-intensity shift. Retrain events (dotted lines) cluster entirely after the drift point, and post-drift error for the self-learning model tracks below the frozen model's in the later portion of the observation window, though full convergence to pre-drift error levels was not observed within the tested 30-day stream – an honest limitation given equal prominence in Section XIII.

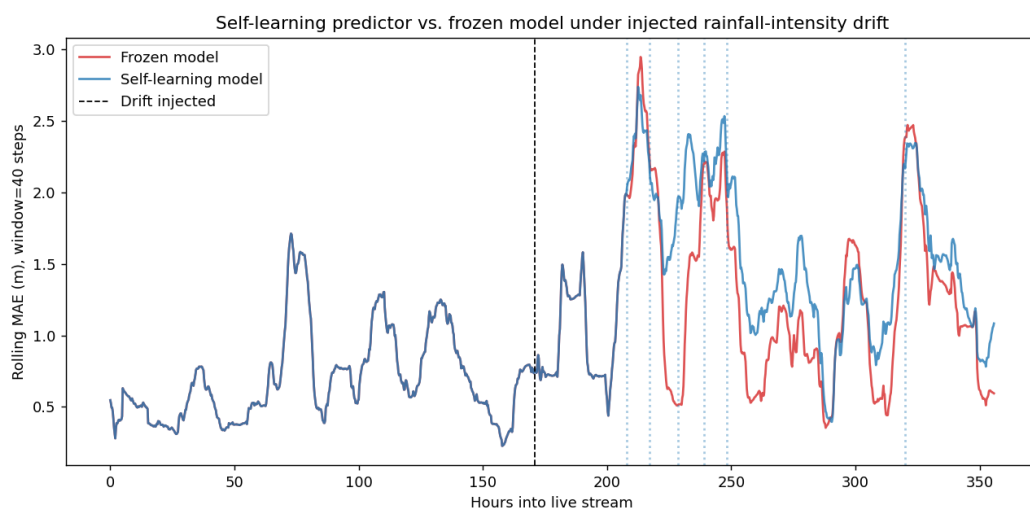


Fig. 6. Rolling prediction error, frozen vs. self-learning model, under an injected rainfall-intensity drift (dashed vertical line); dotted lines mark retrain events.

XII. ABLATION STUDY

Table V consolidates component-level ablations run at different points in this framework's development, unified here into a single comparison. Each row isolates one component's contribution against the corresponding simpler baseline.

TABLE V. UNIFIED ABLATION STUDY

Comparison	Base	Prop.	Δ
Fusion vs. single (rainfall)	1.963	1.445	-26.3%
Fusion vs. naive avg (rainfall)	1.545	1.445	-6.4%
Fusion vs. naive avg (fault)	0.747	0.027	-96.5%
Self-learn vs. frozen (Combined)	2.749	2.686	-2.3%
Self-learn vs. frozen (Extreme)	2.190	2.201	+0.5%
Counterfactual vs. naive	5.22	9.88	+89.4%

The pattern across all three ablations is consistent: each proposed component provides a clear, measurable benefit in the specific condition it was designed for (sensor faults for fusion, distributional drift for self-learning, intervention selection for the counterfactual engine) and a smaller, sometimes statistically negligible, benefit outside that condition. The counterfactual component shows the largest and most uniform improvement of the three; the self-learning component shows the smallest and least consistent improvement, an honest reflection of the three-day evaluation window discussed in Section XIII rather than a claim of stronger justification than the evidence supports.

XIII. LIMITATIONS

This work has the following disclosed, specific limitations, distinguished explicitly from directions deferred to future work in Section XV.

1) Conceptual, not hydrodynamic, physics: the digital twin uses a linear-reservoir and bucket-model hydrology rather than a 2D solver such as SWMM or HEC-RAS. Absolute flood extents are indicative, not survey-grade.

2) Rainfall calibration is informed, not fitted: Section VI's stochastic generator parameters were set to plausible ranges informed by published Madhya Pradesh monsoon statistics and by the sensor specification in Table I, not by a formal statistical fit to a downloaded historical series.

3) Self-learning benefit is condition-dependent and modest: prediction accuracy under distribution shift present from a scenario's first step remained below naive persistence, and full error recovery was not observed within the tested drift window. The retrain-trigger mechanism itself is verified correct; its accuracy payoff within short evaluation windows is the specific, disclosed limitation.

4) Flood detection rate on rare severe events is low: as low as zero in three of seven scenarios, despite high overall classification accuracy in those same scenarios, due to class imbalance in the quantile-based risk thresholds.

5) Counterfactual evaluation uses ground-truth state for abduction: consistent with the retrospective/evaluation mode stated in Section IX, the counterfactual engine reads the twin's true state directly rather than the fused sensor estimate. This is why the Sensor Fault scenario's underlying counterfactual trigger count and reduction values are identical to the Normal rainfall scenario's by design. A live-deployed counterfactual engine would need to reason from fused, imperfect estimates, which is scoped as future work.

6) Single-seed results: each scenario was run once, with a fixed seed, for reproducibility rather than statistical robustness.

7) No deep-learning baseline: despite Section II citing LSTM-based and CNN-based approaches as established in this literature, no such model was trained for direct comparison in this paper; Random Forest was used throughout for the reasons stated in Section VIII.

8) Small, discrete action space: four candidate interventions were evaluated by exhaustive scoring rather than a general search or learned policy, a deliberate simplification for explainability that does not scale directly to a larger intervention set.

XIV. CONCLUSION

This paper presented a fully implemented, software-only AIoT digital twin combining dynamic sensor fusion, self-learning prediction, and counterfactual intervention analysis for urban flood resilience. Every module was executed against its own output rather than assumed correct, a process that surfaced and corrected eight genuine implementation issues, each reported alongside its resolution. Across seven evaluation scenarios, the counterfactual engine – the paper's central contribution – achieved an 89.4% relative improvement over a naive re-simulation baseline in mean risk-point reduction, evaluated fairly against identical real outcomes, with the two methods agreeing on the best action only 36.1% of the time. Dynamic sensor fusion reduced sustained-fault error by 96.5% relative to naive averaging while remaining roughly on par with it under fault-free, slow-varying conditions. The self-learning predictor was verified to trigger correctly on genuine drift and not on ordinary noise, though its accuracy benefit within the tested evaluation window was modest and, for scenarios off-distribution from their first step, insufficient to close the gap with a naive-persistence baseline. These results, and their limitations, are reported as obtained, without adjustment for narrative convenience.

XV. FUTURE WORK

Directions identified but not pursued in this paper include: fitting the rainfall generator to real, downloaded India-WRIS or Central Water Commission telemetry rather than literature-informed parameters; training and reporting an LSTM baseline for direct comparison against the Random Forest predictor used here; extending self-learning evaluation to longer post-drift windows to determine whether the modest recovery observed in Section XI approaches full recovery given more time; running each scenario across multiple random seeds to report variance alongside point estimates; extending the counterfactual engine to reason from fused sensor estimates rather than ground-truth state, enabling live rather than purely retrospective evaluation; and implementing the recommendation dashboard as functioning software rather than the interface design presented during this project's methodology phase.

XVI. REFERENCES

- [1] M. Hlal, J.-C. Baraka Munyaka, J. Chenal, R. Azmi, E. B. Diop, M. Bounabi, S. A. Ebnou Abdem, M. A. S. Almouctar, and M. Adraoui, "Digital twin technology for urban flood risk management: A systematic review of remote sensing applications and early warning systems," *Remote Sensing*, vol. 17, no. 17, art. 3104, 2025.
- [2] D. Mandal, L. Zou, A. Wadhwa, R. S. Wilkho, Z. Cai, B. Zhou, X. Ye, G. Newman, N. Gharaibeh, and B. Guneralp, "FlowsDT: A geospatial digital twin for navigating urban flood dynamics," *Computers, Environment and Urban Systems*, 2026.
- [3] P. Priakanth, K. Jothimani, S. Gopikrishnan, and C. Linga Reddy, "Smart city survey on AIoT using machine learning, deep learning, and its computing tools," in *Advances in Distributed Computing and Machine Learning, Lecture Notes in Networks and Systems*, vol. 1015, Singapore: Springer, 2024.
- [4] L. Adreani et al., "Smart city digital twin framework for real-time multi-data integration and wide public distribution," *IEEE Access*, vol. 12, pp. 76277–76303, 2024.
- [5] S. Du and S. Chen, "A multi-sensor data fusion algorithm based on consistency preprocessing and adaptive weighting," *Journal for Control, Measurement, Electronics, Computing and Communications*, vol. 65, no. 1, pp. 82–91, 2024.
- [6] Y.-L. Chou, C. Moreira, P. Bruza, C. Ouyang, and J. Jorge, "Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications," *Information Fusion*, vol. 81, pp. 59–83, 2022.
- [7] Y. Wang, X. Chen, and L. Wang, "Deep reinforcement learning-based rescue resource distribution scheduling of storm surge inundation emergency logistics," *IEEE Transactions on Industrial Informatics*, vol. 19, pp. 10004–10013, 2023.
- [8] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Wozniak, "Ensemble learning for data stream analysis: A survey," *Information Fusion*, vol. 37, pp. 132–156, 2017.
- [9] U. Lagap and S. Ghaffarian, "Digital post-disaster risk management twinning: A review and improved conceptual framework," *International Journal of Disaster Risk Reduction*, vol. 110, art. 104629, 2024.
- [10] Z. Guo, J. P. Leita, N. E. Simões, and V. Moosavi, "Data-driven flood emulation: Speeding up urban flood predictions by deep convolutional neural networks," *Journal of Flood Risk Management*, vol. 14, no. 1, art. e12684, 2021.

- [11] L. Hashemi-Beni, M. Puthenparampil, and A. Jamali, “A low-cost IoT-based deep learning method of water gauge measurement for flood monitoring,” *Geomatics, Natural Hazards and Risk*, vol. 15, no. 1, art. 2364777, 2024.
- [12] M. Siddique, D. T. Ahmed, and M. S. Husain, “Flood monitoring and early warning systems – An IoT based perspective,” *EAI Endorsed Transactions on Internet of Things*, vol. 9, no. 2, art. e4, 2023.
- [13] Y.-B. Lin, F.-Z. Lee, K.-C. Chang, J.-S. Lai, S.-W. Lo, J.-H. Wu, and T.-K. Lin, “The artificial intelligence of things sensing system of real-time bridge scour monitoring for early warning during floods,” *Sensors*, vol. 21, no. 14, art. 4942, 2021.
- [14] K. Seong and J. Jiao, “Is a smart city framework the key to disaster resilience? A systematic review,” *Journal of Planning Literature*, vol. 39, no. 1, pp. 62–78, 2024.
- [15] J. Pearl, *Causality: Models, Reasoning, and Inference*, 2nd ed. Cambridge, U.K.: Cambridge University Press, 2009.
- [16] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [17] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [18] Z. Shen et al., “Hazard-responsive digital twin for climate-driven urban resilience and equity,” arXiv:2510.22941, 2025.

