From Faulty Greenhouse Sensors to Usable Training Data: An End-to-End Sensor Health Management Pipeline for Fault Detection, Data Repair, and Downstream Validation

Mohamed Akram Lamhour[1], Mohamed Msalek[1], Mohammed Kasbouya[2], Soufiane Ardchir[1], and Mohamed Azzouazi[1]

[1] *LTIM (Laboratoire de Technologies de l'Information et Modélisation), Department of Mathematics and Computer Science, Faculty of Sciences Ben M'Sik, Hassan II University, Casablanca 20700, Morocco*
mohamed.lamhour-etu@etu.univh2c.ma (corresponding author)
mohamed.msalek-etu@etu.univh2c.ma
soufiane.ardchir@univh2c.ma
mohamed.azzouazi@univh2c.ma

[2] *Laboratory of Artificial Intelligence and Systems, Department of Mathematics and Computer Science, Faculty of Sciences Ben M'Sik, Hassan II University, Casablanca 20700, Morocco*
mohammed.kasbouya-etu@etu.univh2c.ma

ABSTRACT

Predictive models for smart greenhouses commonly assume that the low-cost sensor data they are trained on are correct. This paper treats the health of the sensing system as the object of study, using an uncurated 34-day greenhouse irrigation deployment (947,682 raw records, six streams, no fault-management layer). Every stream carries a naturally occurring fault signature: dropouts on 26.4% of the span, an eight-day out-of-range soil-moisture episode, a zero-variance channel, a CO_2 channel with a suspected calibration bias, stuck-at episodes, spikes, and, after the longest outage, a variance collapse that marks a change of operating regime. The pipeline characterizes these faults, benchmarks rule-based screening, Isolation Forest and an LSTM autoencoder on 162 injection scenarios and seven repair methods, and measures how training-data treatment changes a soil-moisture forecaster against persistence. Rules and the autoencoder are complementary, bias and drift stay hard for all detectors, and linear interpolation is the best repair or not significantly different from the best in nearly every condition up to 12 h. The original LSTM forecasters of the soil-moisture level do not beat persistence at 5 min to 6 h, mainly because they cannot extrapolate to the post-outage level; within the training range, repaired data cut their pooled 5-min error 3.9–4.9-fold against raw data when samples are grouped by the true target and 3.1–3.7-fold when grouped by the last observed value; at 5 min, with the long-gap fallback, rules-guided repair reaches the persistence level. In an exploratory analysis added after these results, level-free change models selected on training data only beat persistence by 14% at 5 min when trained on repaired data, against 5% for raw data; at 1 h validation selected persistence for raw data and for two of three repair variants, and at 6 h the selected repaired-data models are worse than persistence. A random-mask control does not show a benefit of targeted repair on the full test set.

Keywords: sensor health management; sensor fault detection; data repair; measurement reliability; anomaly detection; IoT sensing; smart greenhouse

1. INTRODUCTION

1.1. Background and motivation

Smart greenhouses instrument the growing environment with low-cost sensor nodes and increasingly delegate decisions to machine learning models trained on the resulting streams (Wolfert, Ge, Verdouw, & Bogaardt, 2017; Ojha, Misra, & Raghuwanshi, 2015; Tzounis, Katsoulas, Bartzanas, & Kittas, 2017; Kamilaris & Prenafeta-Boldú, 2018; Liakos, Busato, Moshou, Pearson, & Bochtis, 2018; Contreras-Castillo, Guerrero-Ibañez, Santana-Mancilla, & Anido-Rifón, 2023; Simo, Dzitac, Badea, & Meianu, 2022; Săcăleanu et al., 2024), from short-term prediction of irrigation-relevant variables such as soil moisture (Ghazouani, Azzouazi, & Lamhour,

Mohamed Akram Lamhour et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
https://doi.org/10.36001/IJPHM.2026.v17i2.5171

2023; Koné, Grati, Bouaziz, & Boukadi, 2023; Teshome, Bayabil, Schaffer, Ampatzidis, & Hoogenboom, 2024; Wang et al., 2024; Kandamali et al., 2025; Adeyemi, Grove, Peets, Domun, & Norton, 2018) to explainable models that justify their outputs to growers (Lamhour, Msalek, Ardchir, Ghazouani, & Azzouazi, 2026). Across this literature, one assumption is so pervasive that it is rarely stated: the sensor data feeding these models are correct.

Field experience says otherwise. Agricultural sensors operate unattended in hot, humid, dusty, and electrically noisy environments; nodes lose power and connectivity; probes corrode, drift, and die (Zou et al., 2023; Ojha et al., 2015). The wireless-sensor-network community documented this long ago: landmark studies of real deployments found transient and persistent faults to be routine rather than exceptional (Ni et al., 2009; Sharma, Golubchik, & Govindan, 2010). A recent review of fault diagnosis in agricultural IoT explicitly identifies the scarcity of *real-deployment* fault data as a barrier: nearly all published evaluations rely on synthetically injected faults layered onto curated datasets (Zou et al., 2023). As a consequence, the community knows fairly precisely how well detectors find the faults researchers simulate, and much less about the faults deployments actually produce, or about what those faults cost the downstream models that are the deployments' reason for existing.

1.2. An unplanned natural experiment

This paper turns a real, uncurated deployment into that missing evidence. The dataset, comprising 947,682 samples logged over 34 days by the greenhouse irrigation IoT system introduced by Ghazouani et al. (2023), was collected with no fault-management layer whatsoever. It consequently contains, without any injection, the full catalog of classic sensor pathologies: a soil-moisture probe that reported out-of-range values, far outside the nominal range of the index, for its first eight days, at the constant value -85 from 20 to 26 April, before abruptly recovering; a soil-temperature channel that reported a single constant value (zero variance) for the entire deployment; a CO_2 channel whose long-run median of 101.6 ppm (raw-sample basis; 113.3 ppm on the 1-min time basis), with no night-time rise of more than 3 ppm in any operating regime, indicates a suspected systematic calibration bias, punctuated by raw-sample spikes up to 23,553 ppm; fifteen transmission dropouts covering 26.4% of the deployment span; and, after the longest outage, an air-temperature channel whose daily range collapsed from about 16 °C to below 1 °C. Rather than treating these defects as an embarrassment to be silently cleaned away, we treat them as the object of study.

1.3. Research gap, questions, and contributions

From a health-management perspective, the sensing layer of an IoT deployment is itself a system whose condition must be assessed before its outputs can be trusted. The gap addressed here is that the three stages of such an assessment, diagnosing faulty measurements, restoring corrupted or missing streams, and verifying that the restored data actually support the operational task, are almost always studied in isolation and almost never on naturally faulty data. Detection studies stop at alarms, imputation studies assume the corrupted samples are already identified, and downstream studies assume clean inputs. In PHM terms, the monitored engineering system here is the greenhouse sensing subsystem; its health assessment asks whether the measurements it delivers are usable; diagnosis isolates the fault classes (dropout, out-of-range, stuck-at, spike, bias, drift, zero variance, variance collapse); the health-management actions are data repair within a validated horizon, recalibration, inspection, or replacement of the affected channel; and downstream validation quantifies how sensor health propagates into an operational decision-support output. This mirrors the concern, raised within the PHM community, that degraded sensors corrupt the measurements on which every subsequent diagnostic and decision step relies (Thai, Do, Iung, & Gardoni, 2025). Throughout, "usable" denotes a precise, limited property of training data: a model trained on them, evaluated on untouched screened-clean observations under a leakage-controlled chronological protocol, performs at the level of the operationally credible persistence reference within the range of the target covered by training. It does not imply formal uncertainty calibration, safe online operation, or forecasting skill beyond that reference. Because the target is unknown at prediction time, test samples are also assigned to that range by the last observed value: repaired data meet the definition at 5 min when samples are grouped by the true target, and only with rules-guided repair when they are grouped by the last observed value (Section 4.4). We therefore ask four questions:

- **RQ1, Characterization.** What fault classes occur in a real low-cost greenhouse deployment, at what rates, with what temporal structure, and in which operating regimes?
- **RQ2, Detection.** How well do practical unsupervised detectors (rule-based screening, Isolation Forest, and an LSTM autoencoder) identify each fault class, evaluated on labeled fault-injection scenarios built from screened-clean segments of the same deployment, and how sensitive are the conclusions to their thresholds?
- **RQ3, Repair.** Which imputation strategies best restore the signal across gap lengths and sensors, in both single-sensor and whole-node failure regimes?
- **RQ4, Downstream impact.** How does the detect-and-repair pipeline change the downstream forecaster relative to training on raw data or discarding flagged samples, measured against persistence and autoregressive references and separated by operating regime, and how does the answer depend on the forecasting model and horizon?

The paper carries one message: on a real low-cost deploy-

ment, the value of detecting and repairing faulty sensor data can only be judged downstream, against an operational reference, and it depends on the forecasting model and horizon as much as on the data treatment. It makes three contributions. (i) *Fault characterization and measurement reliability*: a time-basis taxonomy and prevalence of naturally occurring faults on an uncurated greenhouse deployment, including three operating regimes and a variance-collapse signature that point-wise rules cannot see (RQ1). (ii) *Detection and repair evaluation*: a leakage-controlled benchmark of three practical detectors with window-clustered uncertainty and threshold sensitivity, and a two-regime imputation benchmark reported per sensor and in normalized units, with the repair policy selected on training-period data only (RQ2, RQ3). (iii) *Downstream impact against baselines*: five training-data treatments, including deletion of faulty windows, evaluated against persistence and autoregressive references at three horizons with inference on temporally dependent errors, separating the effect of the training data from that of a regime change and from that of the forecasting model (RQ4). The forecasting models are standard (an LSTM and ridge regression); no new forecasting algorithm is proposed. The distinguishing contribution is the evaluation of the complete chain on one uncurated deployment rather than any single stage.

The pipeline diagnoses faulty measurements, quantifies the health of the recorded streams, repairs corrupted segments, and measures the consequences for an operational forecast; it does not estimate remaining useful life, model physical sensor degradation, or schedule maintenance, and the soil-moisture forecast is an agronomic prediction task, not a failure prognosis. This work completes an arc with our previous studies on the same deployment, which addressed prediction accuracy (Ghazouani et al., 2023) and interpretability (Lamhour et al., 2026); both presumed clean inputs, and the present paper supplies the upstream sensor-health layer that makes that presumption defensible.

2. RELATED WORK

2.1. Sensor fault detection in agricultural IoT

Shekarian, Aminian, Mohammad Fallah, and Akbary Moghaddam (2024) developed a cloud-based cross-sensor prediction framework for greenhouse fault detection and fault-tolerant operation, in which neural networks predict each sensor from the others and flag disagreements (93.2% detection accuracy), evaluated primarily with injected faults. Their focus is maintaining operation under sensor faults; reconstruction quality is not benchmarked across gap lengths, and the effect of repaired measurements on a downstream forecasting model is not quantified. A low-cost irrigation system by Benameur, Dahane, Kechar, and Benyamina (2024) detects anomalies in soil-moisture, temperature, and humidity streams with autoencoders and generative adversarial networks (90–97% accuracy), again on injected or curated anomalies and again without a repair stage or downstream evaluation. The review by Zou et al. (2023) surveys the field and identifies the absence of real-deployment fault datasets as a central limitation. Recent work on climate control integrates fault detection into actuation loops (Li, Abdullah, & Ishak, 2025), and a recent benchmark (Zhabko, Laktionov, Diachenko, Vinyukov, & Moroz, 2026) measures forecasting robustness under synthetic degradation of agroclimatic data, closest in spirit to our RQ4, but without detection or repair stages and not on greenhouse data.

2.2. Time-series anomaly detection methods

Anomaly detection in sensor streams spans three method families that map directly onto the detectors we benchmark. *Statistical screening* applies physical-range checks and robust filters; the Hampel identifier, which flags samples deviating from a rolling median by more than a multiple of the scaled median absolute deviation, remains a standard low-cost outlier test (Pearson, Neuvo, Astola, & Gabbouj, 2016). Foundational taxonomies of sensor data faults (Ni et al., 2009) and detection method classes (rule-based, estimation-based, time-series-model-based, and learning-based) (Sharma et al., 2010) established that no single family covers all fault types, a conclusion our results reinforce in the greenhouse setting. *Isolation-based and density-based learning methods* detect anomalies without labels: Isolation Forest exploits the shorter isolation paths of anomalous points (Liu, Ting, & Zhou, 2008, 2012), and local outlier factors compare local densities (Breunig, Kriegel, Ng, & Sander, 2000); general surveys cover the broader landscape (Chandola, Banerjee, & Kumar, 2009; Blázquez-García, Conde, Mori, & Lozano, 2021). *Reconstruction-based deep detectors* train an autoencoder on normal data and flag high reconstruction error: LSTM encoder–decoder architectures (Hochreiter & Schmidhuber, 1997; Lindemann, Maschler, Sahlab, & Weyrich, 2021) and LSTM prediction models with non-parametric thresholding (Hundman, Constantinou, Laporte, Colwell, & Soderstrom, 2018) are the canonical designs, and multivariate deep detectors have been applied to environmental monitoring streams such as water quality (El-Shafeiy, Alsabaan, Ibrahem, & Elwahsh, 2023). Across all families, low-magnitude bias and drift are recognized as the hardest classes for single-channel point-wise detection (Zou et al., 2023; Sharma et al., 2010; Trapani & Longo, 2023), motivating cross-sensor prediction approaches such as that of Shekarian et al. (2024), a difficulty our benchmark quantifies precisely.

Two further strands bear directly on our design. First, learned detectors are usually trained on data assumed to be normal, whereas field data are contaminated by the very faults to be detected; Ulmer, Zgraggen, and Goren Huber (2024) propose a generic refinement of contaminated training data for residual-based detectors, and Kim, Yu, Kim, Oh, and Choi

(2024) iteratively re-weight samples by estimated normality so that training tolerates contamination. Our detectors are trained on screened-clean windows obtained by an explicit algorithmic screen, which is the simplest member of this family; Section 4.2 quantifies how that screen affects the comparison. Second, the comprehensive evaluation of Schmidl, Wenig, and Papenbrock (2022) shows that the ranking of time-series anomaly detectors depends strongly on parameter choices and on the anomaly types present; we therefore report threshold sensitivity for every detector rather than a single operating point.

2.3. Missing-data imputation and its downstream effects

Missing-data handling ranges from classical statistical treatments (Little & Rubin, 2019) through simple temporal interpolation (Moritz & Bartz-Beielstein, 2017), nearest-neighbor imputation (Troyanskaya et al., 2001), and iterative chained-equation methods (van Buuren & Groothuis-Oudshoorn, 2011), to deep learning imputers such as self-attention models (SAITS) (Du, Côté, & Liu, 2023); systematic benchmarks organize this rapidly growing space (Khayati, Lerner, Tymchenko, & Cudré-Mauroux, 2020). Two findings from this literature shape our design. First, method rankings depend strongly on gap length and on whether correlated covariates remain observed, a regime distinction (single-sensor gap vs. whole-node outage) that evaluations on randomly scattered missingness miss (Khayati et al., 2020; AlSalehy & Bailey, 2025). Second, imputation error propagates into downstream models, as demonstrated for air-quality prediction (Hua, Nguyen, Dao, Nguyen, & Nguyen, 2024). However, this downstream line of work assumes the corrupted samples are already identified. To our knowledge, few studies couple an imperfect fault *detector* to the imputation stage and measure end-to-end consequences on a real agricultural deployment against an operational reference forecast.

2.4. Positioning relative to prior work

Table 1 positions our study within the sensor data-quality literature. Unlike the surveyed greenhouse fault-detection studies, which evaluate on injected faults and report alarms or control responses, this work characterizes naturally occurring fault classes on an uncurated deployment (including whole-sensor pathologies that injection protocols do not produce), benchmarks repair including the whole-node outage regime that dominated its missing time, and evaluates the complete chain on the deployment's own forecasting task against persistence and autoregressive references, including a comparison with deletion.

3. MATERIALS AND METHODS

Let $x_t^{(s)}$ denote the reading of sensor s at grid slot t, σ the standard deviation of the segment under consideration, and $\mathcal{F}_t^{(s)} \in \{0, 1\}$ the fault flag. Table 2 summarizes the complete experimental workflow as a five-phase protocol; the following subsections detail each phase. In Table 2, m_t and MAD_t are the rolling median and median absolute deviation (61-slot centered window), $P_{\tau(t)}$ the smoothed clean-training diurnal profile at minute-of-day $\tau(t)$, ℓ_t the current constant-run length, and $h(\cdot)$ the isolation-tree path length with normalizer $c(n)$ (Liu et al., 2008).

3.1. Deployment, dataset, and operating regimes

The present study returns to the underlying high-frequency records of the greenhouse monitoring system used for the forecasting experiments of Ghazouani et al. (2023). Whereas that study reported forecasting experiments on a temporally processed representation of the system's data, the present work analyzes the raw high-frequency deployment segment logged from 18 April to 22 May 2022 (a 34-day span with 30 days containing data): 947,682 raw records across six sensor streams, with sampling density varying substantially over the deployment. No fault-management layer was active during logging, which is precisely what makes this segment valuable for sensor-health research.

The system instruments a tomato greenhouse in Casablanca, Morocco. Table 3 lists the sensing and communication hardware as documented in the project records (Ghazouani et al., 2023; Lamhour et al., 2026); specifications not present in those records are marked n/r (not recorded). The soil-moisture channel deserves a specific remark: the capacitive probe delivers an analog voltage that the node's analog-to-digital converter (ADC) samples, and the acquisition firmware maps the ADC count linearly onto a 0–100 scale between two fixed reference counts, the probe type's nominal dry-air and water-immersion readings; the two constants were not preserved in the project records, no gravimetric or volumetric calibration was performed, and readings beyond either reference count produce values below 0 or above 100 (which is how the -85 episode and the 122 maximum arise). The quantity is therefore a *relative moisture index* in integer steps (reported as soil moisture in percent by Ghazouani et al. (2023)), called here the soil-moisture index, in index units; the bounds 0 and 100 are the nominal range of the index, set by the mapping rather than by soil physics: a reading outside them is possible, but it means the probe output has left the interval between the two reference counts, which is why such values are flagged as out of range rather than treated as soil-moisture values. Figure 1 summarizes the acquisition chain and where the present pipeline attaches to it.

All analyses operate on a 1-minute median-resampled grid unless stated otherwise; we report fault prevalence on a *time basis* (fraction of 1-min grid slots affected) because the irregular raw sampling makes sample fractions misleading; for example, out-of-range soil-moisture readings constitute 1.16%

Table 1. Comparison of sensor data-quality approaches in agricultural and environmental IoT (2023–2026), as reported in the cited studies.

Study	Application domain	Fault evidence basis	Detection method(s)	Repair stage	Node outage regime	Downstream impact evaluation
Agri-IoT fault review (Zou et al., 2023)	Agricultural IoT (survey)	Survey; explicitly notes scarcity of real-deployment fault data	Taxonomy of rule-based, ML, and deep methods (conceptual)	Not addressed	Not addressed	Not addressed
Cross-sensor greenhouse FD (Shekarian et al., 2024)	Smart greenhouse	Evaluated primarily with injected faults	Cross-sensor CNN/ANN prediction (93.2% accuracy)	Fault-tolerant control response; reconstruction quality not benchmarked	Not addressed	Not addressed
Low-cost irrigation system (Benameur et al., 2024)	Irrigation (soil moisture, temp., humidity)	Injected/curated anomalies	Autoencoder + GAN (90–97% accuracy)	No repair stage reported	Not addressed	Not addressed
Climate-control FDD (Li et al., 2025)	Greenhouse climate control	Injected faults	Real-time detection wired to actuation	No (control response, not data repair)	Not addressed	Actuation only, no data-quality metric
Agroclimatic robustness benchmark (Zhabko et al., 2026)	Weather / agroclimatic forecasting	Synthetic degradation of clean data	No detection stage	No (robustness to degradation measured)	Not addressed	Yes, but degradation synthetic and undetected
Air-quality imputation (Hua et al., 2024)	Urban air quality	Missingness only (no fault classes)	No detection stage (missingness given)	Imputation benchmark	Not addressed	Yes, imputation $\rightarrow$ prediction only
This work (2026)	**Greenhouse irrigation IoT**	**Real deployment: 7 naturally occurring fault signatures over 34 days, time-basis prevalence, 3 operating regimes**	**Rules + Isolation Forest + LSTM-AE, leakage-controlled, per-class P/R/F1 on 162 labeled scenarios, threshold sensitivity**	**Yes: 7 methods $\times$ 5 gap lengths, per sensor and normalized; policy selected on training-period windows**	**Yes: all-sensor outage benchmarked**	**Yes: five training-data variants (incl. deletion), 3 horizons, persistence/AR references, identical clean test set**

of raw samples but 22.3% of deployment time.

Operating regimes. Inspection of sampling density and daily variability (Figure 2, Table 4) shows that the deployment did not operate in a single state. In regime R1 (18–27 April) the node logged about one sample per minute while the soil-moisture probe reported out-of-range values, at the constant value -85 from 20 to 26 April. In regime R2 (28 April to 11 May) the node logged about one sample per second and all dynamic channels varied with the expected diurnal amplitude (median daily air-temperature range 16.5 °C). After the 130.8-h outage, regime R3 (16–22 May) resumed at one sample per minute, the soil-moisture index returned at a different level (67–88 before settling at 48–52), and the air-temperature channel reported 8–13 distinct values per day with a median daily range of 0.8 °C, a variance collapse that no in-range, spike, or exact-constancy rule flags. These regimes matter for every subsequent phase: the clean windows available for RQ2 and RQ3 come from R1/R2 (humidity) and R3 (temperature, soil moisture), and the chronological RQ4 test period falls entirely in R3.

3.2. Fault taxonomy, thresholds, and flag definitions (RQ1)

We classify faults into point-level and whole-sensor classes (Table 2, Phase 1). Point-level: out-of-range (value outside the sensor's physically possible range), spike (Hampel criterion, 61-slot window), stuck-at (value exactly constant for longer than 1 h while sampling continues), and dropout (no samples received for more than 5 min). Whole-sensor: zero-variance channel (a single constant value over the entire deployment), suspected calibration bias (long-run median outside the physically plausible band for the quantity), and variance collapse

Table 2. End-to-end sensor fault detection and repair protocol. All thresholds in the input line were fixed before any held-out result was examined (Section 3.4).

Algorithm: End-to-End Fault Detection and Repair Pipeline	
Input:	Raw multivariate stream $\mathcal{X} = \{x_t^{(s)}\}$, $s = 1, \ldots, S$; physical bounds $[x_{\min}^{(s)}, x_{\max}^{(s)}]$ and plausible-median bands $[\mu_{\text{lo}}^{(s)}, \mu_{\text{hi}}^{(s)}]$ (Table 5); thresholds: stuck $\tau_s = 60$ min, dropout $\tau_d = 5$ min, Hampel $k_H = 6$ (61-slot window); seeds for injection and training
Output:	Fault taxonomy with time-basis prevalence and operating regimes; per-class detection P/R/F1 with window-clustered uncertainty; repaired series $\hat{\mathcal{X}}$; downstream skill relative to persistence
Phase 1: Fault characterization (RQ1)	
1.	Resample $\mathcal{X}$ to the 1-min median grid; mark every grid slot that received no raw sample as missing, and mark as dropout every missing slot inside a raw inter-sample interval $\Delta t > \tau_d$
2.	**for** each sensor s **do** flag out-of-range $\mathcal{F}_t^{\text{OOR}} = \mathbf{1}\{x_t \notin [x_{\min}, x_{\max}]\}$ (values exactly on a bound are in range); spikes $\mathcal{F}_t^{\text{spk}} = \mathbf{1}\{\lvert x_t - m_t\rvert > k_H \cdot 1.4826\,\text{MAD}_t\}$; stuck runs on the *raw samples*: a run of exactly equal consecutive values lasting longer than τ_s, split at dropout boundaries, is labelled stuck over its *entire* duration (retrospective labelling) and mapped to every grid slot it covers
3.	Test whole-sensor pathologies: zero-variance if $\text{Var}(x_{1:T}) = 0$; suspected calibration bias if $\text{med}(x_{1:T}) \notin [\mu_{\text{lo}}, \mu_{\text{hi}}]$; variance collapse if the daily range falls below 10% of its median over the dense-sampling regime for three or more consecutive days
4.	Report prevalence per class on the time basis, keeping point classes disjoint with priority OOR $>$ stuck $>$ spike; identify operating regimes from sampling density and daily variability
Phase 2: Clean-window selection and controlled injection (RQ2 ground truth)	
5.	Build the clean mask on the 1-min grid (present $\wedge$ in-range $\wedge$ not in a $>$1-h constant run of 1-min medians $\wedge \geq$30 min from any Hampel spike); cut contiguous clean stretches into non-overlapping 24-h windows (at most four per sensor)
6.	**for** each window, fault type $\in$ {spike, stuck, drift, bias, noise, dropout}, severity $\in$ {low, med, high} **do** inject with the temporal patterns of Table 6; values are not clipped to the physical bounds and keep their injected class label
7.	Record labels $y_t = \mathbf{1}\{\text{sample injected}\} \rightarrow$ 162 scenarios, 233,280 labeled samples; add near-threshold stuck (30/60/90 min) and quasi-stuck (constant $\pm$ one quantization step, 240/480 min) scenarios on the held-out windows
Phase 3: Detection benchmark (RQ2)	
8.	Split windows per sensor: earliest for detector training, remainder held out; evaluate all detectors on the same 72 held-out scenarios and on the non-injected held-out windows (false-positive rate)
9.	**Rules:** $\mathcal{F}_t = \mathcal{F}_t^{\text{OOR}} \vee \mathcal{F}_t^{\text{stk}} \vee \mathcal{F}_t^{\text{spk}} \vee \mathbf{1}\{x_t = \varnothing\}$ (1-min values in the injection benchmark)
10.	**Isolation Forest** (Liu et al., 2008): fit 200 trees on clean features $\mathbf{z}_t$ (z-scored value, first difference Δx_t, rolling deviation $\text{sd}_{30}(x_t)$, profile deviation $x_t - P_{\tau(t)}$, run length ℓ_t); flag if score $s(\mathbf{z}_t) = 2^{-\mathbb{E}[h(\mathbf{z}_t)]/c(n)}$ exceeds the contamination threshold (1% default; 0.5–5% sensitivity)
11.	**LSTM-AE** (Lindemann et al., 2021): train seq-60 autoencoder on clean windows; per-slot error $e_t = \frac{1}{\lvert W_t\rvert}\sum_{w \in W_t}(\hat{x}_t^{(w)} - x_t)^2$ over covering windows W_t; flag if $e_t > q^{\text{clean}}$ (99.5th percentile default; 99–99.9 sensitivity)
12.	Score point-wise precision, recall, F1 per fault class and severity, event-level recall per episode; report dispersion across scenarios and across held-out windows (the clustering unit)
Phase 4: Repair by imputation (RQ3)	
13.	Mask flagged samples; **for** gap length $L \in \{5, 15, 60, 240, 720\}$ min, regime $\in$ {single-sensor, node-outage} **do** benchmark 7 imputers (parameters in Section 3.5)
14.	Score RMSE/MAE on masked samples only, per sensor and normalized by the clean-signal standard deviation; select the lowest-error method by regime and gap length using only windows that precede the RQ4 training/test boundary (final policy: Table 10)
Phase 5: Downstream validation (RQ4)	
15.	Build five training variants of the 5-min stream: raw (forward-filled); rules-, LSTM-AE-, and (rules $\cup$ LSTM-AE)-flagged variants repaired with the Phase 4 policy, all repair components fitted on the training period only; drop-faulty (rule-flagged windows deleted)
16.	**for** each variant, horizon $H \in \{5\text{ min}, 1\text{ h}, 6\text{ h}\}$, seed $\in \{s_1, \ldots, s_{10}\}$ **do** train forecaster $\hat{y}_{t+H} = f_\theta(\mathbf{X}_{t-11:t})$, f_θ: LSTM(32)$\rightarrow$Dense(1), early stopping
17.	Evaluate RMSE, MAE, R^2, skill $\text{SS} = 1 - \text{RMSE}/\text{RMSE}_{\text{pers}}$ and relative MAE on the identical untouched test set (last 30%, inputs and target unflagged), against persistence, seasonal-naive and AR(1) references; decompose errors inside/outside the training support; test differences with moving-block bootstrap and Diebold–Mariano statistics on paired per-sample losses; ablate the $>$12-h gap treatment; repeat on three rolling-origin blocks with screened and operational test sets
18.	**Exploratory, added after steps 15–17:** random-mask control of the rules mask; level-free change models (persistence, ridge, level-free LSTM) selected on a training-only validation slice and evaluated on the same test set

(daily range below 10% of its dense-regime median for three or more consecutive days). Table 5 lists the bounds and bands used, their sources, and the observed extremes; all were fixed from datasheets, the climatic envelope of the site, and the firmware mapping before the deployment was inspected, except the CO_2 median band, whose lower edge (350 ppm) was

Table 3. Deployment hardware. All entries except the CO_2 row are documented in the project records (Ghazouani et al., 2023; Lamhour et al., 2026); the CO_2 stream is present in the raw analyzed database, but its sensor hardware is not recorded there. n/r = not recorded; datasheet values are nominal manufacturer specifications.

Quantity (unit)	Sensor / device	Interface	Nominal range / accuracy
Air humidity (%), air temperature (°C)	DHT22	digital, single-wire	0–100% RH (±2–5%); −40–80 °C (±0.5 °C)
Soil temperature (°C)	DS18B20 probe	1-Wire digital	−55–125 °C (±0.5 °C)
Soil-moisture index (relative, 0–100)	soil-moisture sensor v1.2 (capacitive probe (Lamhour et al., 2026))	analog (ADC)	uncalibrated relative output mapped linearly to 0–100, integer steps
Light (binary)	digital LDR module	digital	threshold day/night output
CO_2 (ppm)	model n/r	n/r	n/r
Node	ESP32 microcontroller	Wi-Fi, MQTT client	n/a
Gateway / logger	Raspberry Pi 4	MQTT broker, Python	n/a

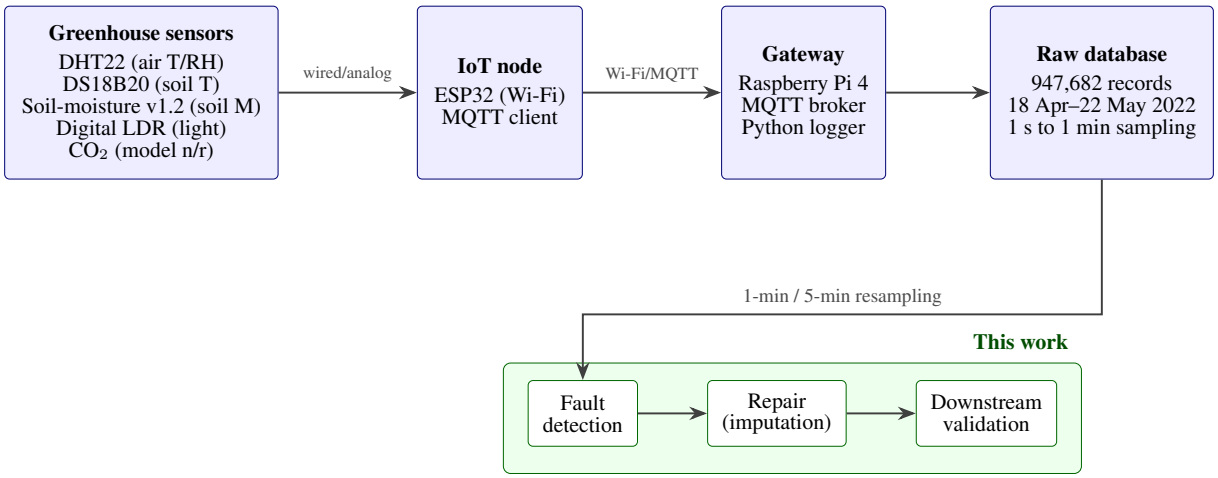


Figure 1. Acquisition chain of the deployment (hardware as documented by Ghazouani et al. (2023); the CO_2 stream is present in the raw database but its sensor model is not recorded) and the attachment point of the present fault detection, repair, and downstream validation pipeline.

Table 4. Operating regimes of the deployment, derived from the raw records (daily statistics computed on 1-min medians over days with at least 1,000 min of data). Δt = median raw sampling interval; range and distinct values refer to the air-temperature channel.

Regime	Dates (2022)	Days	Δt	Daily range	Distinct/day	Soil-moisture index	Salient state
R1	18 Apr – 27 Apr	9.5	60 s	16.1 °C	102	OOR; −85 (constant) 20–26 Apr	probe out of range
R2	28 Apr – 11 May	12.6	1 s	16.5 °C	111	40–50	dense, dynamic
R3	16 May – 22 May	6.3	60 s	0.8 °C	9	67–88, then 48–52	variance collapse, level shift

set from outdoor ambient and is discussed in Section 4.1.

Flag definitions on the resampled grid are as follows. A 1-min slot is *missing* when no raw sample fell into it; every missing slot that lies inside a raw inter-sample interval longer than 5 min is a *dropout* slot, so the whole gap is flagged, not only its boundary. On the 1-min grid, 26.36% of slots are missing and 26.34% of the span lies inside the fifteen gaps longer than 5 min; the two bases differ only in isolated missing minutes shorter than 5 min, and we report 26.4% throughout. A stuck-at run is identified on the raw samples (exact equality of consecutive values for more than 60 min, split at dropout boundaries) and labelled retrospectively over its entire duration; this choice yields the fault prevalence of a curation pass, whereas an online rule can only raise the flag after minute 60 and therefore recovers at most the remainder of each episode (for the in-range stuck-at episodes of this deployment's soil-moisture channel, 53.8% of stuck time lies beyond the first hour). Defining the stuck rule on the raw samples rather than on the resampled medians is essential for slow, integer-quantized channels: the 1-s soil-moisture samples alternate between adjacent integers, so exact constancy over an hour is rare in the raw stream (2.9% of all slots,

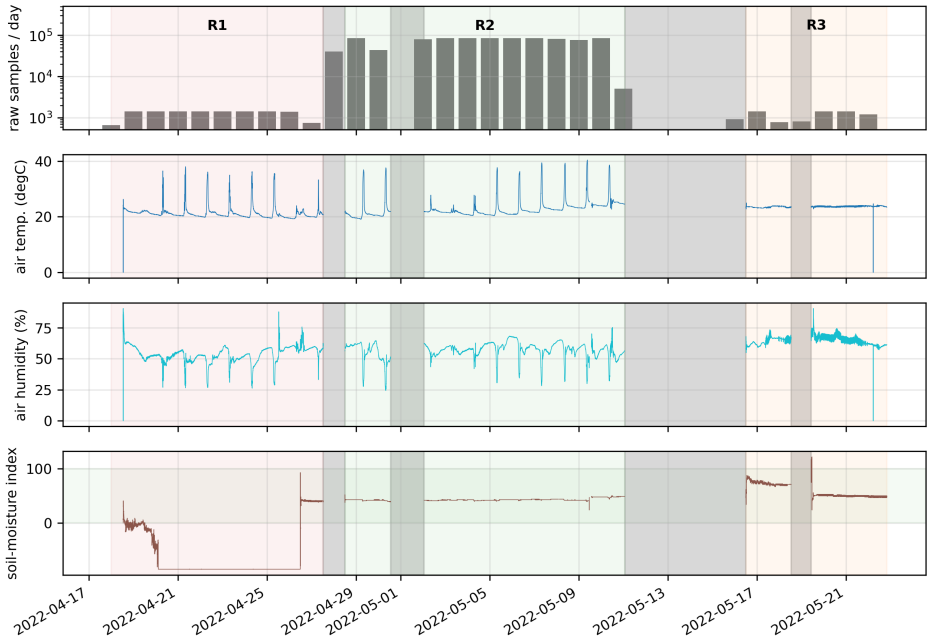


Figure 2. Operating regimes of the deployment: raw samples per day (log scale), air temperature, air humidity and soil-moisture index on the 1-min grid. Gray bands mark outages longer than 12 h; shaded backgrounds mark regimes R1 (red), R2 (green) and R3 (orange). The air-temperature channel loses its diurnal amplitude after the 130.8-h outage.

Table 5. Physical bounds $[x_{\min}, x_{\max}]$ and plausible long-run median bands $[\mu_{\text{lo}}, \mu_{\text{hi}}]$ used by the rule screen, their sources, and the observed extremes and median of the raw samples. Values exactly on a bound count as in range; missing observations are never out of range.

Sensor (unit)	$x_{\min}$	$x_{\max}$	μ_{lo}	μ_{hi}	**Source of bounds**	**Observed min / median / max**
Air humidity (% RH)	0	100	20	100	DHT22 datasheet; median band from greenhouse agronomic plausibility	0.0 / 57.7 / 90.5
Air temperature (°C)	−5	55	5	45	DHT22 datasheet narrowed to the spring climatic envelope of Casablanca	0.0 / 21.9 / 40.4
Soil-moisture index (index units)	0	100	0	100	nominal range of the firmware mapping of the ADC count; a value outside it means the probe output left the interval between the two reference counts, and is flagged	−85 / 43 / 122
Soil temperature (°C)	−5	55	5	45	DS18B20 datasheet narrowed to the climatic envelope	10.0 / 10.0 / 10.0
CO_2 (ppm)	0	5000	350	2000	non-negative by physics; 5000 ppm upper edge of enrichment practice; median band from outdoor ambient to enrichment set points	44 / 101.6 / 23,553
Light (binary)	0	1	0	1	digital output	0 / 1 / 1

not counting the out-of-range episode), whereas the 1-min medians of the same healthy signal are exactly constant for hours (26.7% of all slots, not counting that episode; 45.3% against 21.5% for the raw-sample rule when the constant out-of-range run is included) and the 5-min medians even more so (40.6% of all training-period rows, not counting the out-of-range episode, against 3.3%). Section 4.2 quantifies this effect on the detectors and Section 4.4 on the downstream experiment. Point-level classes are kept disjoint with priority OOR > stuck-at > spike, and dropout minutes are excluded from the point-level classes; the binary light channel is excluded from stuck and spike analysis. All real-fault labels are algorithmic: they are inferred from physical plausibility and temporal behavior of the recorded values. No co-located reference instrumentation, calibration record, or maintenance log is available to confirm the underlying hardware causes, so we refer to the whole-sensor anomalies as a zero-variance channel, a *suspected* calibration bias, and a variance collapse

rather than as confirmed hardware faults.

3.3. Controlled fault injection (RQ2 ground truth)

Real faults lack sample-level labels, so detector evaluation uses controlled injection into *screened-clean* segments: a 1-min slot passes the clean screen for a sensor if it is present, in physical range, not part of a >1 h constant run of 1-min medians, and at least 30 min away from any Hampel-flagged spike. The term emphasizes that cleanliness is established by this algorithmic screen, not by external verification; the injected labels themselves, by contrast, constitute exact ground truth. Contiguous clean stretches are cut into non-overlapping 24-h windows, at most four per sensor: 9 windows total (4 air-humidity, 2 air-temperature, 3 soil-moisture), itself a finding, as only 24.4% of soil-moisture minutes qualify as clean (48.1% if constancy is evaluated on the raw samples, which would give nine soil-moisture windows instead of three). The screen keeps the median-based constancy criterion as a conservative choice: on the slow, integer-quantized index it rejects many healthy minutes, but none of the selected windows contains a minute that is stuck on the raw samples. Soil temperature, CO_2, and light are not covered by the detection benchmark (zero variance, suspected bias, and binary output respectively). The clean-screen definition shares its fault family with the rule detector; Section 4.2 reports the sensitivity of the window set and of the detector comparison to the Hampel constant k_H and the detectors' false-positive rates on the non-injected clean windows, so that the magnitude of this favorable bias is visible.

Into each window we inject six fault types at three severities each, parameterized in units of the window's own standard deviation and seeded for reproducibility; Table 6 gives the complete temporal pattern of every injector and its amplitude in native units. This produces 162 scenarios (233,280 labeled samples, 29.1% faulty). Two properties of the design are worth noting because they shape the results. First, the clean temperature windows, both from regime R3, have a standard deviation of only 0.18–0.20 °C, so σ-scaled temperature faults are small in native units (a high-severity spike is 1.5 °C); the benchmark therefore probes the detectors on a nearly flat channel for temperature. Second, the high-severity spike injections cross the physical bounds in 28.6% (humidity) and 26.2% (soil moisture) of injected samples and are then trivially caught by the range check, as are 0.2% of the high-severity humidity noise samples; they keep their injected label. Both facts are reflected in the interpretation of Section 4.2. Because all injected stuck durations (90–480 min) exceed the 60-min rule threshold, we add near-threshold stuck episodes of 30, 60, and 90 min and quasi-stuck episodes (a frozen level perturbed by $\pm$ one quantization step, 240 and 480 min) on the held-out windows, three seeded repetitions each. Supplementary Figure S1 shows one example of each injected fault type.

3.4. Detectors (RQ2)

Three detectors representative of practice are compared; all share a trivial missingness flag for dropout, so the informative comparison concerns the value-corrupting classes and we also report macro-F1 over those five classes only.

1. **Rule-based screening** (no training): OOR bounds, stuck-run detection (>1 h), Hampel filter (Pearson et al., 2016) ($k_H = 6$, 61-slot window), missingness. In the injection benchmark, whose windows are 1-min series without raw samples, the stuck rule is applied to the 1-min values; the screened windows contain no 1-min constant run longer than 1 h by construction of the clean screen.
2. **Isolation Forest** (Liu et al., 2008) (semi-supervised): 200 trees, contamination 1%, trained per sensor on clean training windows over five features: z-scored value, first difference, 30-min rolling standard deviation, deviation from the training-window diurnal profile, and current constant-run length.
3. **LSTM autoencoder** (Hochreiter & Schmidhuber, 1997; Lindemann et al., 2021) (semi-supervised): 60-step sequence autoencoder (LSTM(32) encoder/decoder) trained per sensor on z-scored clean windows; a time step is flagged when its reconstruction error (squared error averaged over all sliding windows covering the step) exceeds the 99.5th percentile of clean-training errors, following the clean-quantile thresholding practice of reconstruction-based detectors (Lindemann et al., 2021; Hundman et al., 2018).

Threshold selection and sensitivity. The three operating points (Hampel $k_H = 6$, contamination 1%, clean-error quantile 99.5%) were fixed a priori from common practice before any held-out scenario was scored, and no parameter was tuned on held-out results; the 60-step sequence length matches one hour on the 1-min grid. The autoencoder threshold is the 99.5th percentile of the reconstruction errors on the same clean training sequences the autoencoder was fitted on, not on a separate calibration split: with one or two clean training windows per sensor, a calibration split would remove half or all of a sensor's training data. Its sensitivity is reported instead: moving the quantile from 99% to 99.9% changes the autoencoder's macro-F1 by at most 0.015 (Supplementary Table S1). Because a single operating point can misrepresent a detector (Schmidl et al., 2022), we report the full benchmark for $k_H \in \{4, 6, 8\}$, contamination $\in \{0.5, 1, 2, 5\}\%$ and quantile $\in \{99, 99.5, 99.9\}\%$, together with each configuration's false-positive rate on the non-injected held-out clean windows.

Leakage control and uncertainty. Per sensor, the earliest clean windows (two, or all but one when only two exist) train the learned detectors; evaluation uses only the 72 scenarios injected into the remaining four held-out windows (two humidity, one temperature, one soil moisture), identically for

Table 6. Injection protocol. σ is the standard deviation of the clean 24-h window; the native-unit columns give the resulting amplitude range over the windows of each sensor (humidity σ = 3.9–7.5% RH, temperature σ = 0.18–0.20 °C, soil-moisture index σ = 1.3–3.8). Positions are drawn with a seeded generator (seed 42); injected values are not clipped.

Fault	Temporal pattern	Severity (low / med / high)	Humidity (% RH)	Temperature (°C)	Soil index
Spike	$n = \max(1, f \cdot 1440)$ isolated samples at uniformly random positions, each shifted by $\pm k\sigma$ with random sign	k = 3 / 5 / 8; f = 0.2 / 0.5 / 1.0%	12–23 / 20–38 / 31–60	0.55 / 0.9 / 1.5	3.8–12 / 6.3–19 / 10–31
Stuck-at	one episode, random start, value frozen at the start sample	L = 90 / 240 / 480 min	n/a	n/a	n/a
Drift	linear ramp from 0 to $k\sigma$ starting at a random position in the second quarter of the window and lasting to its end	k = 1 / 3 / 5	3.9–7.5 / 12–23 / 20–38	0.19 / 0.55 / 0.9	1.3–3.8 / 3.8–12 / 6.3–19
Bias	step of $\pm k\sigma$ (random sign) from a random position in the second quarter of the window to its end	k = 1 / 2 / 3	3.9–7.5 / 7.8–15 / 12–23	0.19 / 0.37 / 0.55	1.3–3.8 / 2.5–7.7 / 3.8–12
Noise	Gaussian $\mathcal{N}(0, (k\sigma)^2)$ added over one 360-min episode starting in the first half of the window	k = 0.5 / 1 / 2	2–3.8 / 3.9–7.5 / 7.8–15	0.09 / 0.19 / 0.37	0.6–1.9 / 1.3–3.8 / 2.5–7.7
Dropout	one episode set to missing, random start	L = 15 / 60 / 240 min	n/a	n/a	n/a
Near-threshold stuck	one episode, random start, three repetitions per window	L = 30 / 60 / 90 min	n/a	n/a	n/a
Quasi-stuck	frozen level plus jitter of $\{-q, 0, +q\}$ per sample, q = quantization step (0.1% RH, 0.1 °C, 1 index unit)	L = 240 / 480 min	n/a	n/a	n/a

all three detectors. The injected scenario is the scoring unit, but scenarios injected into the same 24-h window share that window's signal and are not independent; we therefore report dispersion both across scenarios and across held-out windows (the clustering unit, $n = 4$), and paired detector differences with bootstrap intervals over windows. Metrics: point-wise precision, recall, F1 per fault class and severity, plus event-level recall (an injected episode counts as detected if at least one of its samples is flagged). All detectors are implemented with the scikit-learn and TensorFlow libraries.

3.5. Repair by imputation (RQ3)

Repair masks flagged samples and imputes them. Seven methods are benchmarked: forward-fill (the naive baseline: last observation carried forward, leading gaps filled backward), linear interpolation, cubic spline (Moritz & Bartz-Beielstein, 2017), multivariate KNN (Troyanskaya et al., 2001) and iterative (MICE-style) imputation (van Buuren & Groothuis-Oudshoorn, 2011) using the co-recorded channels (air humidity, air temperature, soil-moisture index, light), diurnal-profile filling (training-window minute-of-day profile, 31-min smoothed, anchored to the observed window level by the mean offset), and a hybrid combining linear interpolation for gaps $\leq$1 h with the diurnal profile beyond, included as a candidate. The KNN imputer uses $k = 10$ neighbors with the nan-Euclidean metric and uniform weights; the iterative imputer uses a Bayesian ridge estimator, 10 iterations, tolerance 10^{-3}, mean initialization, ascending imputation order and a fixed random state. Both are fitted on the 24-h window being repaired and may therefore use rows on both sides of the gap: all evaluated procedures are retrospective data-repair methods for dataset curation and model training, not real-time substitutes during an active outage. The multivariate imputers operate on the native units of the channels; because KNN distances depend on scaling, we also evaluate variants with per-window standardization of all channels. Deep imputers (Du et al., 2023) are excluded because the nine clean 24-h windows provide too little training data for them to be competitive, a constraint typical of single-deployment settings (Khayati et al., 2020).

Protocol and metrics. Into each held-out clean window we carve seeded contiguous gaps of 5, 15, 60, 240, and 720 min (five repeats each) under two regimes, single-sensor gap (covariates intact) and node outage (all channels missing simultaneously, as when a node loses power), and score RMSE/MAE on the masked samples only. Because the three channels have different units and variances, errors are reported per sensor and as a dimensionless normalized RMSE, $\mathrm{nRMSE} = \mathrm{RMSE}/\mathrm{sd}_{\mathrm{clean}}$, where $\mathrm{sd}_{\mathrm{clean}}$ is the standard deviation of the sensor's held-out clean signal (6.1% RH, 0.18 °C, 1.27 index units); aggregates across sensors average nRMSE first within each sensor (over its held-out windows and repeats) and then with equal weight over the three sensors, so that humidity, which has two held-out windows against one for each other sensor, does not count twice. Paired method differences are summarized with bootstrap 95% intervals obtained by re-

sampling (window, repeat) units within each sensor. The final repair policy is not fixed a priori: it is selected from the benchmark separately by outage regime and gap length, preferring the simplest method among those whose paired difference to the numerically best method has a bootstrap interval spanning zero (simplicity order: linear, forward-fill, spline, profile, hybrid, KNN, iterative). To keep the downstream experiment free of test-period influence, the selection is made on the windows that precede the RQ4 training/test boundary (the two held-out humidity windows, 23–25 April) and then checked against the full window set (Table 10).

3.6. Downstream validation protocol (RQ4)

The downstream task mirrors that of Ghazouani et al. (2023): forecasting the soil-moisture index H steps ahead on a 5-min grid from 1 h (12 steps) of air-humidity, air-temperature, and soil-moisture history, with a compact LSTM (LSTM(32) $\rightarrow$ Dense(1), early stopping). We evaluate three horizons, $H \in \{1, 12, 72\}$ steps (5 min, 1 h, 6 h). Five training variants of the identical architecture are compared: **raw** (as recorded, missing values forward-filled, the naive pipeline); three repaired variants in which the samples flagged by **rules**, by the **LSTM autoencoder**, or by their **union** are masked and repaired; and **drop-faulty** (training windows containing any rule-flagged row excluded). In the repaired variants, flagged segments up to 4 h (48 rows) are imputed with linear interpolation, which Table 10 selects in this range; longer flagged segments are filled with the anchored diurnal profile, an *unvalidated operational fallback* used only to construct a complete numerical training stream, whose influence is quantified by an ablation (Section 4.4). The RQ4 repair therefore departs from Table 10 for flagged runs of 4 to 12 h, for which that table selects linear interpolation (KNN for 12-h single-sensor gaps), and KNN is not used in RQ4; with the rules mask such runs contain 139 soil-moisture training target rows (2.0% of the training rows), against 3,170 (45.8%) in flagged runs longer than 12 h, where no method is validated. No imputation method is validated beyond 12 h and no reconstruction-fidelity claim is made for those segments.

Chronological split and leakage control. The split is strictly chronological: the first 70% of the 5-min grid (18 April to 12 May 13:00) is the training period, the last 30% (12 May 13:00 to 22 May) the test period. Every data-derived component is fitted on the training period only: each variant's input/target normalization statistics, the diurnal profiles and anchoring offsets used for repair, and the autoencoder weights and its 99.5th-percentile threshold (retrained at the 5-min scale on screened-clean training sequences, with the threshold estimated from those clean sequences only). These components are refitted for RQ4 and are separate from the RQ2 detectors, whose training windows partly lie in the test period; no RQ2 model or threshold is used in RQ4. The rule screen uses the fixed a-priori thresholds of Table 5 with the stuck rule evaluated on the raw samples (Section 3.2); its masks, including the raw-sample stuck runs, which are evaluated only on the raw samples of each partition, and the autoencoder inference used to flag training samples for repair, are computed per chronological partition so that no rolling, constancy or reconstruction window crosses the train/test boundary; the same holds for every rolling-origin block. This RQ4 chronological protocol is distinct from the RQ2 evaluation, which splits clean 24-h *windows* between detector training and held-out injection scenarios; five of the nine clean windows (all three soil-moisture and both temperature windows: three detector-training and two held-out windows) lie inside the RQ4 test period, which is why the repair policy is selected on the humidity windows only (Section 3.5).

Common untouched test set and reference forecasts. The five models differ only in their training data; all are evaluated on one common test set built once from the original observed 5-min stream and never repaired, filled, or reconstructed by any variant. A test sample is retained only if all 12 input steps and the target at $t + H$ are unflagged by the rule screen in *all three* input channels (missing slots count as flagged), so inputs and targets are untouched real observations. Three reference forecasts are computed on exactly the same samples: *persistence* ($\hat{y}_{t+H} = y_t$), *seasonal naive* (the observation 24 h before the target, falling back to persistence when it is missing), and an *AR(1)* regression of y_{t+H} on y_t fitted on clean training pairs. Beyond RMSE, MAE and R^2, we report the skill score $\mathrm{SS} = 1 - \mathrm{RMSE}/\mathrm{RMSE}_{\mathrm{pers}}$ and the relative mean absolute error $\mathrm{rMAE} = \mathrm{MAE}/\mathrm{MAE}_{\mathrm{pers}}$, both relative to persistence on the same test samples; these are relative accuracy measures in the sense of Hyndman and Koehler (2006), not the in-sample-scaled MASE. Because the test period lies in regime R3, where the soil-moisture index partly exceeds the range seen in training, every error is additionally decomposed into test samples whose target lies inside the training support (the range of screened-clean training targets) and samples outside it.

Inference. Training stochasticity is quantified by ten seeds (42–51) per variant, reported as mean $\pm$ standard deviation; paired comparisons across seeds are kept as a robustness statement only, because seeds do not sample the greenhouse process. Inference about forecast accuracy uses the paired per-sample squared losses on the common test set, averaged over seeds for each variant: differences in RMSE between variants, and between each variant and persistence, are given 95% intervals from a moving-block bootstrap (Künsch, 1989) (block length 24 samples, 2,000 replicates, identical block draws for both members of a pair) and Diebold–Mariano statistics (Diebold & Mariano, 1995) with a Bartlett heteroskedasticity-and-autocorrelation-consistent variance (lag $H + 11$: the horizon plus the 12-step input window minus one, because overlapping input windows share observations) and the small-sample correction of Harvey, Leybourne, and Newbold (1997). Re-

tained test samples are not contiguous in time, so the blocks approximate rather than reproduce the serial dependence; the interval widths should be read with this caveat. Two RMSE aggregates are therefore reported: Table 11 and Supplementary Table S5 give the mean ± standard deviation of the per-seed RMSE, whereas Table 12 gives the pooled RMSE of the seed-averaged squared errors, which by Jensen's inequality is never smaller than the mean per-seed RMSE and exceeds it most for variants whose error varies strongly across seeds; this is why autoencoder-guided repair minus raw training is $+0.07$ in Table 11 (11.01 vs. 10.94) but -0.37 in Table 12 (pooled 11.05 vs. 11.42). As a second, independent line of evidence, the whole experiment is repeated at the 5-min horizon in a *rolling-origin* design with three chronological test blocks (5–8 May, 8–11 May, and the original 12–22 May block), every fitted component being re-estimated on the data preceding each block, and differences are summarized across blocks. For each block we also report a *broader operational test set* that keeps every observed target of the block without the rule screen, with inputs taken as recorded, missing input slots forward-filled inside the block, and flagged inputs left in place; it measures what a deployed model would face, at the price of including faulty inputs and targets. TensorFlow operation determinism is enabled and all random generators are seeded (Python 3.11, NumPy 2.3, TensorFlow 2.20, CPU).

Original and exploratory analyses. The design described so far, with the LSTM forecaster of the soil-moisture level, is the original downstream analysis. Two further analyses were added after its results were known, each under a protocol written before its models were trained, and are reported as exploratory: a random-mask control and level-free change models. Their intervals describe the temporal sampling of the test errors given each analysis; they do not account for the sequence of analysis choices that led to them.

Random-mask control. To test whether the effect of rules-guided repair depends on *which* observations are masked, beyond masking and repairing a comparable amount of training data, the observed training values flagged by the rules are replaced, per channel, by randomly placed masks with the same count and the same run lengths (five mask seeds crossed with the ten training seeds, 5-min horizon); genuine missing values and every other step are those of rules→repair. Because the longest soil-moisture run (the out-of-range episode) fits only in the observed stretch that contains it, the primary control re-masks most of that episode; a split control, also fixed before training, places it as two halves and re-masks part of it. Design and diagnostics are in Supplementary Section S3.

Level-free change models. Because the level forecasters failed mainly through extrapolation to the post-outage level (Section 4.4), we evaluated forecasters that predict the change $y(t+H)-y(t)$ from inputs without the absolute soil-moisture level; the prediction is the last observed value plus the predicted change. For every training variant and horizon three standard candidates are compared: persistence (zero change), ridge regression on level-free features of the 12-step window (soil-moisture changes over the last 1, 2, 3, 6 and 11 steps, last air humidity and temperature and their changes over the window, and the hour of day as sine and cosine), and the LSTM of the original design with the soil-moisture input replaced by its difference to the last value of the window. The candidate and the ridge regularization are selected on a chronological validation slice (the last 20% of the training windows, rule-clean windows only). Detection masks, autoencoder, repair profiles, offsets and normalization used to fit the candidates for this selection are computed on the earlier part of the training period only, so the validation slice does not influence its own preprocessing. The selected model is refitted on the full training partition and evaluated on the common test set; all candidates are reported, and the ridge result is given for every variant so that training data can be compared within one model family. For these exploratory models Table 14 reports the more conservative bootstrap over contiguous runs of retained test samples; with the moving-block bootstrap of Table 12 the 6-h drop-faulty difference to persistence would exclude zero. A residual LSTM that predicts the change but keeps the absolute level among its inputs was evaluated first and is reported in Supplementary Table S7.

Table 7. Inferred real-fault prevalence per sensor stream, on a 1-min time basis (all values in % of deployment time). Point classes are disjoint. Convention: when a whole-sensor flag applies (zero variance, suspected calibration bias), the channel is unusable throughout and the total is set to 100% regardless of the point-class sum; Figure 5 uses the same convention. "Var. collapse" marks the variance collapse of regime R3, which affects part of the span and does not change the total. "Susp. bias" denotes a suspected calibration bias inferred from physical implausibility; no reference calibration record was available.

Sensor	Dropout	OOR	Stuck	Spike	Whole-sensor	Total
Air humidity	26.4	0.0	0.0	0.6	n/a	27.0
Air temperature	26.4	0.0	13.7	0.2	var. collapse	40.3
Soil moisture	26.4	22.3	2.9	0.0	n/a	51.6
Soil temperature	26.4	0.0	73.4	0.0	zero-variance	100
CO_2	26.4	0.0	0.1	1.2	susp. bias	100
Light	26.4	n/a	n/a	n/a	n/a	26.4

4. RESULTS

4.1. Every sensor stream exhibits at least one fault signature, and the deployment has three regimes (RQ1)

Figure 3, Figure 2, and Table 7 summarize the characterization. Six points stand out. (i) Fifteen dropouts >5 min total 216.8 h (26.4% of the span), with two events exceeding one day (longest 130.8 h; Figure 4). (ii) The soil-moisture probe reported the constant value -85 from 20 to 26 April inside an out-of-range episode spanning 18 to 26 April (22.3% of deployment time) before abruptly recovering; this 153.1-h

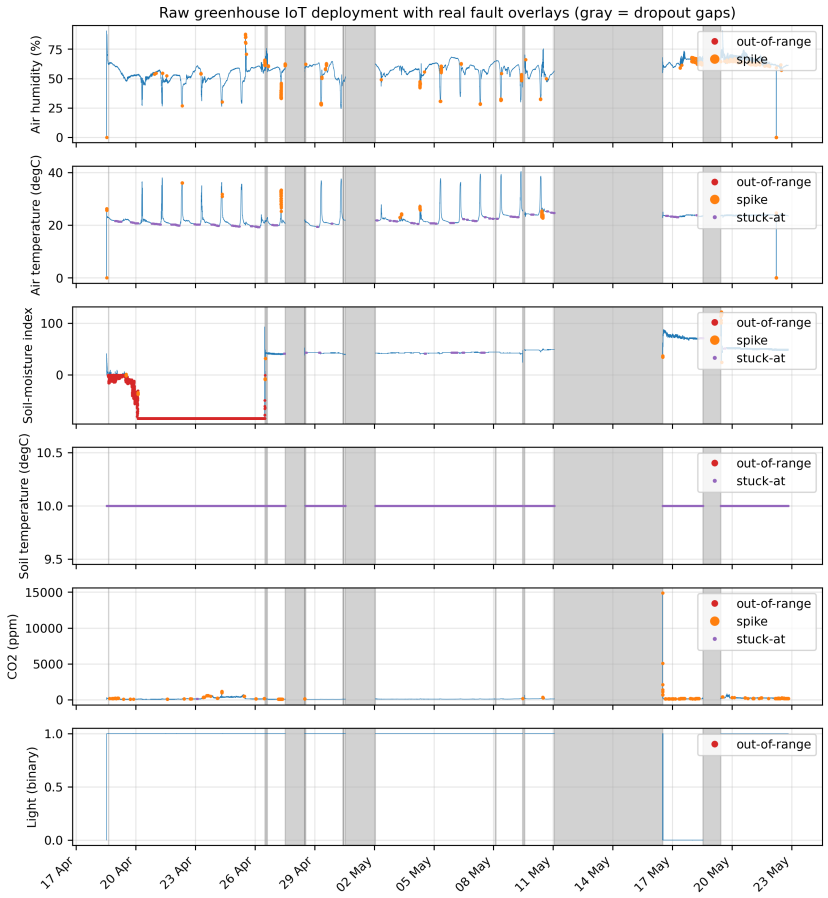


Figure 3. The raw 34-day deployment resampled to 1-min medians, with naturally occurring (real) fault overlays: out-of-range (red), spikes (orange), stuck-at (purple), transmission dropouts (gray bands). Note the eight-day out-of-range soil-moisture episode, the zero-variance soil-temperature channel, and CO_2 spikes. The raw-sample CO_2 maximum of 23,553 ppm is not visible at this resolution because 1-min medians attenuate single-sample spikes.

constant run is the longest constant run of the deployment, but it is counted as out-of-range under the class priority, and the longest in-range stuck-at episode of the soil-moisture channel is 5.5 h (5 May). Complete fault–recovery trajectories of this kind are rarely available in published agricultural datasets. (iii) The sample-basis vs. time-basis discrepancy (1.16% vs. 22.3% for soil-moisture OOR) shows time-basis reporting is essential under irregular sampling. (iv) The CO_2 channel's long-run median of 101.6 ppm (raw-sample basis) is a strong long-run plausibility anomaly rather than proof of a hardware fault. Outdoor ambient (∼420 ppm) is not a hard floor for a greenhouse, since photosynthetic drawdown can depress daytime concentrations well below ambient in closed canopies (Poorter et al., 2012). The 350-ppm lower edge of the median band (Table 5) only triggers the screen; the classification rests on the features below. Three features nevertheless make a calibration or measurement bias the most plausible explanation: 94.7% of all raw samples lie below 150 ppm, the daily median is below 150 ppm on 22 of 30 days, and night-time readings (22:00–05:00, median 104.5 ppm) are only about 6 ppm higher than daytime readings (10:00–16:00, median 98.8 ppm), whereas without photosynthetic uptake night-time concentrations in a closed canopy would be expected to rise. These three figures are raw-sample weighted, and 97.8% of the raw samples come from the 1-Hz regime R2; on the 1-min time basis used elsewhere in this paper the median is 113.3 ppm and 70.4% of the minutes lie below 150 ppm, the regime medians are 122.6 (R1), 101.6 (R2) and 223.8 ppm (R3), and in no regime does the night-time median exceed the daytime median by more than 2.8 ppm (night vs. day: R1 122.6 vs. 125.9, R2 101.6 vs. 98.8, R3 209.2 vs. 223.8 ppm). This absence of a substantial night-time rise is consistent with a measurement artifact but does not establish its cause; because the sensor model, its calibration record and any reference instrument are unavailable, we label the channel as carrying a *suspected* calibration bias. (v) After the 130.8-h outage the air-temperature channel reports a daily range of 0.8 °C over four complete days (16.5 °C in R2) with 8–13 distinct values per day; this variance collapse covers five consecutive post-outage days (17–21 May, 14.6% of the deployment span), is invisible to

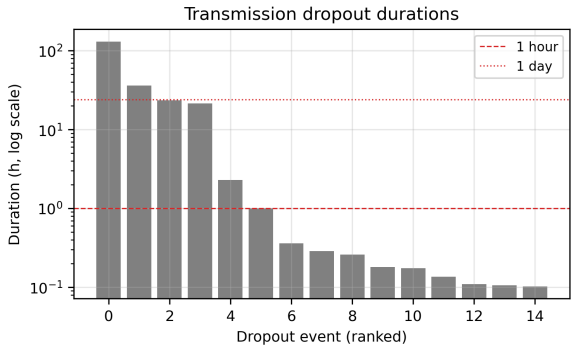


Figure 4. Durations of the fifteen transmission dropout events (log scale).

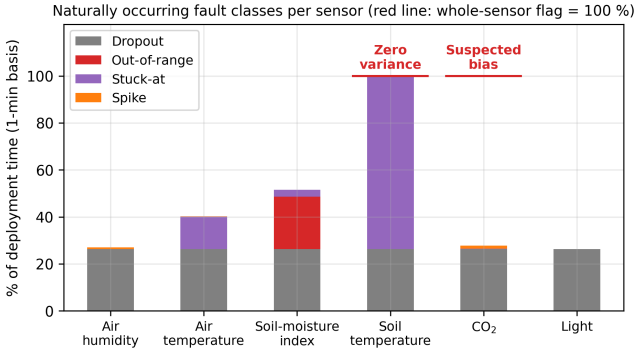


Figure 5. Prevalence of naturally occurring point fault classes per sensor (real data, 1-min time basis). The soil-temperature channel shows zero variance throughout and the CO_2 channel carries a suspected whole-sensor calibration bias; for these two channels the whole-sensor total of 100% is drawn as a red line, matching the convention of Table 7.

range, spike, and exact-constancy rules, and is detectable only from the loss of diurnal amplitude. (vi) The deployment comprises three operating regimes (Table 4); the clean windows and the chronological test period are not drawn from a stationary process, a fact that conditions the interpretation of RQ2–RQ4. The same caveat applies to all real-fault labels in this section, which are inferred from physical plausibility and temporal behavior rather than from independent reference-grade instrumentation. Figure 5 aggregates prevalence per class.

4.2. Detection: rules and learned detectors are complementary, and the conclusions survive threshold changes (RQ2)

Table 8 reports the benchmark; Supplementary Figure S2 gives F1 by severity, and Supplementary Tables S1–S3 the threshold sensitivity, the window-length sensitivity and the near-threshold scenarios. Findings: (i) rules dominate the deployment's most prevalent classes, stuck-at (F1 0.99) and dropout (1.00), and are precise on spikes (precision 0.75); (ii) the LSTM autoencoder (LSTM-AE) is the best learned detector, leading on spikes (F1 0.62, event recall 0.75) and the only method with meaningful traction on bias and drift; (iii) Isolation Forest's run-length feature gives it moderate stuck-at capability (F1 0.47) that the autoencoder lacks; (iv) **no detector handles low-magnitude bias and drift well** (all F1 $\leq$ 0.17), consistent with the literature (Sharma et al., 2010; Trapani & Longo, 2023) and motivating cross-sensor redundancy methods (Shekarian et al., 2024) as complementary future work. Overall macro-F1 (mean over scenarios) is 0.44 for rules, 0.36 for LSTM-AE and 0.33 for Isolation Forest; restricted to the five value-corrupting classes, which is the informative comparison since the dropout flag is shared, it is 0.33, 0.23 and 0.20.

Uncertainty. Scenarios injected into the same window are not independent, and the window-level ranges in Table 8 show that several large scenario-level dispersions are between-window rather than between-severity effects: Isolation Forest's stuck-at F1 varies only from 0.44 to 0.49 across severities (Supplementary Figure S2) but from 0.00 to 0.95 across windows, because its run-length feature was fitted on the dense humidity windows and does not transfer to the one held-out soil-moisture window. These dispersions therefore cannot be attributed to severity dependence alone. With four held-out windows, paired detector differences are decisive only for the largest effects: over windows, rules beat LSTM-AE on stuck-at faults by 0.97 (bootstrap 95% interval 0.94–1.00) and Isolation Forest on spikes by 0.31 (0.15–0.46), while LSTM-AE beats rules on noise by 0.19 (0.05–0.32); the LSTM-AE-versus-rules difference on bias (0.16, 0.002 to 0.32), the rules-versus-LSTM-AE difference on spikes (-0.03, -0.33 to 0.28) and the LSTM-AE-versus-Isolation-Forest difference on stuck-at (-0.45, -0.93 to 0.04) are not resolved by this benchmark. A bootstrap over four windows has only 35 distinct resamples, so the signs of the per-window differences (released with the code) are a more direct summary: rules beat LSTM-AE on stuck-at faults in all four windows and Isolation Forest on spikes in all four; LSTM-AE beats rules on bias and on noise in three windows, the fourth being tied; on spikes rules and LSTM-AE each win in two windows, and on stuck-at LSTM-AE beats Isolation Forest in one window, loses in two and ties in one. Event-level recall follows the same complementary pattern: 1.00 for dropout (all detectors) and for stuck-at under rules, 0.75 for spikes under both rules and the autoencoder, and 0.67 for bias and noise under the autoencoder.

Thresholds and the clean screen. The detector ranking depends on the operating point through a trade-off between detection and false alarms (Supplementary Table S1). Rules (k_H = 4–8) and the autoencoder (quantile 99–99.9%) change macro-F1 by at most 0.02 with false-positive rates on the non-injected clean windows below 0.5% on average at every setting (rules $\leq$0.09%, LSTM-AE $\leq$0.33%; worst window 1.11%). Isolation Forest rises to 0.41 at 5% contamination, above the

Table 8. Point-wise F1 per detector and injected fault class on the 72 held-out scenarios at the default operating points. Each cell gives the mean over scenarios, the standard deviation over scenarios, and in brackets the range of the four per-window means (the clustering unit). Bold marks the best detector per class.

F1	bias	drift	dropout	noise	spike	stuck
Rules	0.00 ± 0.00 [0.00–0.00]	0.07 ± 0.12 [0.01–0.26]	$\mathbf{1.00 \pm 0.01}$ [0.99–1.00]	0.01 ± 0.01 [0.00–0.02]	0.59 ± 0.40 [0.24–0.97]	$\mathbf{0.99 \pm 0.02}$ [0.98–1.00]
Isolation Forest	0.04 ± 0.07 [0.00–0.11]	0.01 ± 0.01 [0.00–0.03]	0.98 ± 0.04 [0.95–1.00]	$\mathbf{0.23 \pm 0.32}$ [0.00–0.45]	0.28 ± 0.29 [0.00–0.48]	0.47 ± 0.49 [0.00–0.95]
LSTM-AE	$\mathbf{0.16 \pm 0.28}$ [0.00–0.38]	$\mathbf{0.17 \pm 0.27}$ [0.00–0.40]	0.97 ± 0.05 [0.92–1.00]	0.19 ± 0.24 [0.00–0.34]	$\mathbf{0.62 \pm 0.41}$ [0.00–1.00]	0.02 ± 0.06 [0.00–0.08]

default rules over the five value-corrupting classes (0.40 vs. 0.33), but its clean-window false-positive rate then reaches 11.6% on average and 18.8% on one window, so the ordering of the default operating points holds only at comparable false-alarm rates (below 0.5% on average). Window lengths matter less than thresholds, and the rules remain the best detector at every window length (Supplementary Table S2). The false-positive rate of the rules on the clean windows is exactly zero at $k_H = 6$ because the clean screen excludes the Hampel-flagged neighbourhoods of the same rule family; this is the favorable bias acknowledged in Section 3.3, and the rate at the detector's $k_H = 4$ (0.09%) does not bound it, because the bias arises from which windows the screen admits. Changing the clean screen's own k_H changes the number of humidity 24-h windows (none at $k_H = 4$, five at the default $k_H = 6$, of which four are used, and twelve at $k_H = 8$) but not the temperature and soil-moisture sets, and varying the spike exclusion margin (15–60 min) changes no window count; because no humidity window survives the $k_H = 4$ screen, the benchmark cannot be repeated on that screen, and the settings of Supplementary Table S1 vary the detector's k_H on the fixed default windows. On 24-h windows selected by the autoencoder alone, without any rule criterion, the rules flag 0.70% of humidity slots and 6.5% of soil-moisture slots, the latter mostly the real in-range stuck-at episodes of 5 and 7 May that the autoencoder does not see (Supplementary Section S1).

Near-threshold and quasi-stuck faults. The main injection design uses stuck durations of 90–480 min, all beyond the 60-min rule, which builds the rule detector's stuck-at result into the scenario design. On the added near-threshold scenarios (Supplementary Table S3) the rule detector behaves as its definition dictates: F1 0.00 at 30 min, 0.25 at 60 min and 0.99 at 90 min, whereas Isolation Forest, whose run-length feature is continuous, reaches F1 0.23, 0.37 and 0.42 with event recall 0.5 at all three durations, and the autoencoder detects none. Quasi-stuck episodes, in which the value jitters by one quantization step around a frozen level for 4–8 h, defeat all three detectors (F1 $\leq$ 0.03, event recall $\leq$ 0.17). This is the injected counterpart of the natural variance collapse of Section 4.1, and it confirms that practical stuck-at ambiguity, not only exact constancy, must be addressed by amplitude-based plausibility checks over longer windows.

Application to the real stream. Applied to the full real soil-moisture stream (Figure 6), rules flag 51.6% of deployment time, the same one-minute slots as the Table 7 total for this channel (26.4% dropout, 22.3% out-of-range, 2.9% stuck-at and 0.02% spikes under the class priority of Table 7; the Hampel test itself flags 0.07% of slots, 0.05 percentage points of which lie inside the out-of-range episode), Isolation Forest 71.6%, and LSTM-AE 49.7%. Evaluating the stuck rule on constant runs of 1-min medians instead would flag 75.3%, 23.7 percentage points more, because that rule marks 26.7% of all slots as stuck (not counting the out-of-range episode) instead of 2.9% where the healthy, integer-quantized index is merely slow (Section 3.2). For detection coverage the two families are thus complementary; Section 4.4 examines how the choice of detector flags propagates to the downstream task.

4.3. Repair: simple methods win at practical gap lengths, per sensor and in normalized units (RQ3)

Figure 7 and Table 9 report the normalized imputation error against gap length in both regimes, with equal weight per sensor, and Supplementary Table S4 the per-sensor results; Supplementary Figure S3 shows a qualitative 4-h repair; MAE rankings are consistent with RMSE and are provided with the released tables. Findings: (i) for gaps up to 1 h, linear interpolation (to which the hybrid reduces in this range) is the numerically best method at 5 min in both regimes (nRMSE 0.50 and 0.47) and at 1 h in the node-outage regime; elsewhere in this range KNN (single-sensor, 15 min and 1 h) or the cubic spline (node outage, 15 min) is numerically lower, but by 0.04–0.06 with paired intervals spanning zero; (ii) in the node-outage regime, which dominated this deployment's missing time, linear interpolation is never significantly worse than the best method: at 4 h it is within 0.01 of the multivariate fallback (0.93 vs. 0.92, interval spanning zero) and at 12 h it is the best method (0.87 vs. 1.00 for KNN, interval excluding zero), while the multivariate methods degrade to the column-mean fallback as soon as their covariates vanish (nRMSE 0.79–1.00) and cubic splines diverge on long gaps (nRMSE up to 4.3); (iii) in the single-sensor regime, multivariate KNN is numerically best from 15 min to 12 h, not significantly different from linear interpolation from 15 min to 4 h (paired intervals spanning zero, which with few windows does not establish equivalence)

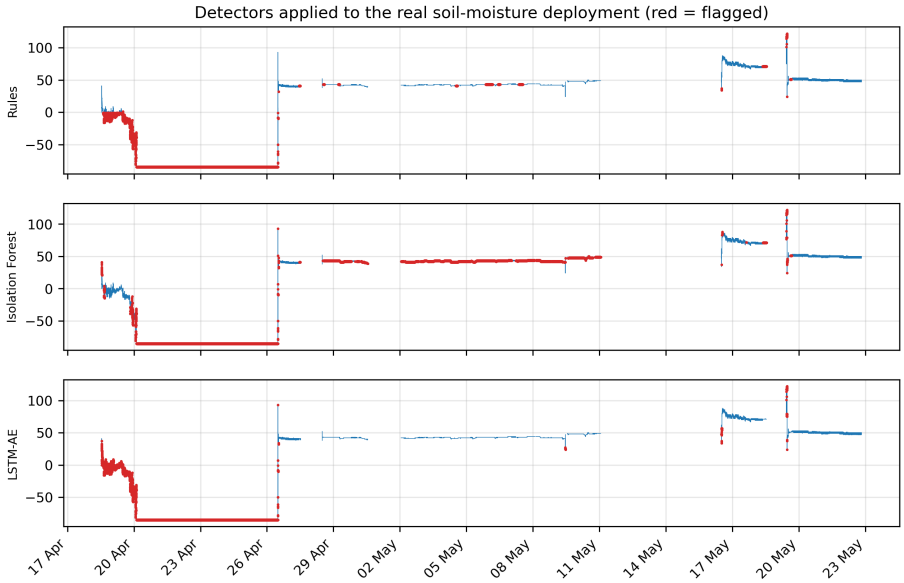


Figure 6. Detectors applied to the full real soil-moisture stream, 1-min medians (red = flagged). The rule detector uses the raw-sample stuck definition of Section 3.2.

and clearly better at 12 h (0.76 vs. 0.91, interval 0.03–0.30), where its covariates remain observed; (iv) forward-fill, the default in naive pipelines, is 20–46% worse than the best method across the evaluated gap lengths. Linear interpolation is therefore not the winner in every condition, but it is the simplest method that is the best or not significantly different from the best in every condition except 12-h single-sensor gaps.

Per-sensor consistency. Native-unit errors are not comparable across sensors (the humidity windows have a standard deviation of 6.1% RH, 30 times that of the temperature windows), and averaging over windows would count humidity twice; both effects are removed by the equal-sensor nRMSE above, and the per-sensor table shows where the aggregate hides disagreement. For humidity, which has the largest diurnal amplitude, linear interpolation is best or not significantly different from the best at every gap length except 12-h single-sensor gaps. For the soil-moisture index, KNN is the best single-sensor method at four of five gap lengths (RMSE 1.2–1.3 index units vs. 1.3–1.6 for linear), because the slow index is well predicted by the co-recorded channels, while linear interpolation is best in the node-outage regime at 12 h and within 0.25 index units of the best method elsewhere. For temperature, the two clean windows come from the variance-collapse regime R3 and the window standard deviation is only 0.18 °C and the seven methods differ by at most 0.17 °C; these windows discriminate poorly between methods, so we draw no ranking from them. Standardizing the channels before KNN or iterative imputation changes nRMSE by less than 0.05 at every gap length and does not alter any ranking. Because the channels are quantized, KNN distance ties are frequent; breaking them at random changed the equal-sensor KNN nRMSE by at most 0.025 and no ranking in our software environment (Supplementary Section S2). The common policy of Table 10 is therefore a robust compromise rather than a uniformly optimal choice.

Table 9. Normalized imputation RMSE (nRMSE = RMSE / sd of the clean held-out signal), averaged within each sensor and then with equal weight over the three sensors, by regime and gap length; bold = numerically lowest among the methods shown (the hybrid equals linear up to 1 h; the iterative imputer equals KNN in the node-outage regime). Full table and paired intervals in the released results.

Regime	Method	5 min	15 min	1 h	4 h	12 h
Single-sensor	forward-fill	0.68	0.84	0.92	0.91	1.06
	linear	**0.50**	0.72	0.73	0.79	0.91
	KNN	0.56	**0.68**	**0.69**	**0.73**	**0.76**
	profile	0.83	0.88	0.98	0.93	0.99
	spline	0.57	0.72	0.73	1.67	2.54
Node outage	forward-fill	0.69	0.90	0.87	1.21	1.14
	linear	**0.47**	0.76	**0.73**	0.93	**0.87**
	KNN	0.98	1.00	0.79	**0.92**	1.00
	profile	0.96	0.99	0.87	1.02	1.00
	spline	0.56	**0.70**	0.87	1.96	4.27

Repairability horizon. The benchmark validates repair for gaps of 5 min to 12 h only. Of the fifteen real dropouts, nine are shorter than 1 h and two fall between 1 and 4 h, so eleven of fifteen events lie within the validated horizon; however, the four long outages (21.5, 23.4, 36.2, and 130.8 h) account for 211.7 of the 216.8 missing hours (98%). For gaps beyond 12 h, reliable reconstruction cannot be established from the present benchmark, and we do not extrapolate the reported

Table 10. Repair policy. "Training-period windows" = simplest method not significantly different from the numerically best one on the two held-out humidity windows of 23–25 April, which precede the RQ4 training/test boundary (paired bootstrap over 10 window-repeat units); "all windows" = numerically best method on all four held-out windows (equal-sensor nRMSE), with methods not significantly different from it in brackets ("n.s."). The policy used downstream is the training-period column; it is the best or not significantly different from the best on all windows in every row.

Failure regime	Gap duration	Selected (training-period windows)	Selected (all windows)
Single-sensor	$\leq$1 h	linear	linear at 5 min; KNN at 15 min and 1 h (linear n.s.)
Single-sensor	4 h	linear	KNN (linear n.s.)
Single-sensor	12 h	KNN (covariates observed)	KNN
Node outage	$\leq$1 h	linear	linear; spline at 15 min (linear n.s.)
Node outage	4 h	linear	KNN/iterative (linear n.s.)
Node outage	12 h	linear	linear
Any	$>$12 h	none validated	none validated

errors to them. In the downstream experiment such gaps are filled with the diurnal profile as an unvalidated operational fallback whose sole purpose is to yield a complete numerical training stream; the ablation of Section 4.4 measures what this fallback contributes.

4.4. Downstream: training-data treatment helps the forecaster, but how much depends on the model and horizon (RQ4)

Scope of the test set. The common untouched test set at the 5-min horizon contains 1,162 samples, 39.2% of the 2,964 rows of the chronological test period, all between 16 and 22 May (regime R3); the first four days of the test period fall inside the 130.8-h outage and contribute nothing. The retained targets range from 45 to 85 index units with mean 55.3 (sd 9.7), whereas the 403 observed test-period rows excluded by the rule screen have mean 70.0 (sd 11.3), so the test set describes the drier, quieter part of the regime. Because screened-clean training targets range from 0 to 50, only 43.9% of the retained test targets (510 samples) lie inside the training support; the remaining 56.1%, concentrated on 16–18 May when the index read 68–85, require extrapolation from every learned model. The sets at 1 h and 6 h are defined identically and contain 1,117 and 1,040 samples.

Reference forecasts. Persistence attains RMSE 1.29, 1.53 and 1.97 index units at 5 min, 1 h and 6 h (R^2 = 0.98, 0.97 and 0.95 against the test mean); the AR(1) regression fitted on clean training pairs has slope 1.00 at 5 min and coincides with persistence at 5 min and 1 h (1.29 and 1.53), but at 6 h its mean reversion makes it slightly better than persistence (1.81 vs. 1.97, skill 0.08), so persistence is the strongest reference only up to 1 h; the seasonal-naive forecast is far worse (5.6–5.7). The target is slow and integer-quantized: at the 5-min horizon it does not change at all in 51% of the retained steps and never by more than 4 units. Persistence is therefore the operationally credible reference, and R^2 against the test mean overstates every model.

Original level-predicting LSTM. Table 11 reports the five training-data variants against the references at the three horizons. *No original level-predicting LSTM surpasses persistence on the full test set*: skill scores are between -4.8 and -9.4, and the moving-block bootstrap intervals of the RMSE difference to persistence exclude zero for every variant (Table 12). Deletion keeps too little data for this model: with 61.3% of all 5-min rows (64.2% of training-period rows) carrying at least one rule flag, drop-faulty keeps 1,741 of 6,902 training windows (351 at the 6-h horizon), and on the full set it is worse than raw training by 2.10 at 5 min, although the bootstrap interval (-0.11 to 3.35) includes zero (DM $p = 0.007$).

Failure diagnosis: inside and outside the training support. The gap to persistence is almost entirely a regime effect. The split used here is post hoc: it is defined by the true value of the target, unknown at prediction time, so it explains where the error comes from but is not an operational rule. Inside the support the repaired-data models at 5 min reach RMSE 1.05 (rules), 1.17 (autoencoder) and 1.30 (union), at or below the persistence level of 1.34 on the same samples, whereas the raw-data model is at 4.91, 3.7 times worse, with a seed-to-seed standard deviation of 1.8 that reflects how strongly the eight-day out-of-range training episode dominates what it learns. Outside the support every learned model fails (RMSE 13–18) while persistence is unaffected (1.25), because the index returned from the outage at a level that no training variant had seen (Supplementary Figure S4). On the inside-support samples the temporal tests support this reading at 5 min: the RMSE difference between rules-guided repair and persistence is -0.28 (95% interval -0.43 to -0.14, DM $p < 0.001$), between autoencoder-guided repair and persistence -0.16 (-0.31 to -0.01, p = 0.021), and between each repaired variant and the raw-data model -4.13 and -4.01 (both $p < 0.001$). Drop-faulty is about twice as bad as the repaired variants inside the support (2.62 vs. 1.05–1.30). At 1 h and 6 h the repaired variants remain far better than raw training inside the support (by 3.7–4.4 units, intervals excluding zero), but at 1 h only autoencoder-guided repair remains at the persistence level (difference $+0.15$, interval -0.07 to 0.31; rules-guided $+0.35$, 0.07 to 0.57; union-guided $+0.26$, 0.05 to 0.46), and at 6 h none does (2.9–3.2 vs. 1.6). Grouping the samples instead by the last observed value, which is known

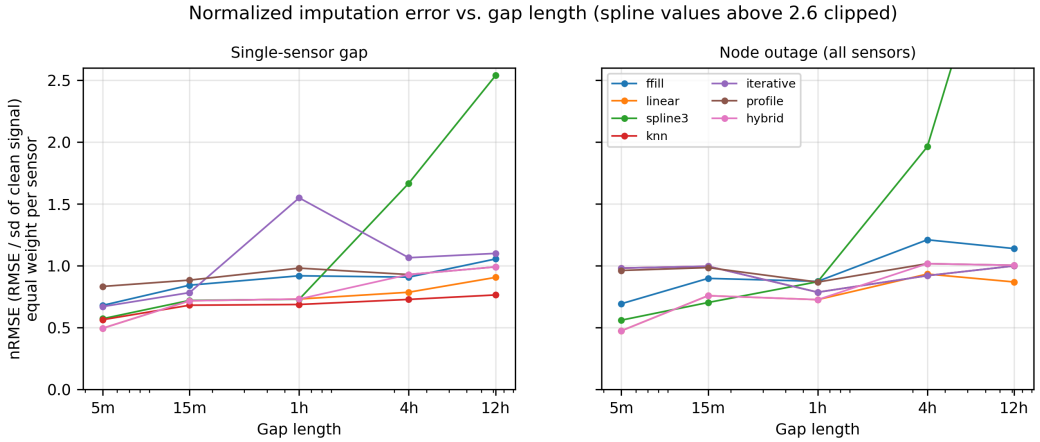


Figure 7. Normalized imputation RMSE (equal weight per sensor) vs. length of controlled gaps carved into held-out clean windows, under the single-sensor-gap and node-outage regimes; spline values above 2.6 are clipped. Per-sensor native-unit errors are in Supplementary Table S4.

Table 11. Original level-predicting LSTM forecasts of the soil-moisture index on the common untouched test set (original 5-min observations, inputs and targets unflagged), by training-data variant and horizon. Learned models: mean ± sd over 10 seeds; SS = skill relative to persistence on the same samples ($1 - \mathrm{RMSE}/\mathrm{RMSE}_{\mathrm{pers}}$); inside/outside = RMSE on test targets inside/outside the training support (0–50). Windows = training windows available to the variant. Stuck-at flags follow the raw-sample definition of Section 3.2. Bold row label: rules→repair, the primary repaired variant; bold does not mark the best value.

Horizon	Training data	n_{test}	RMSE	MAE	R^2	SS	RMSE inside	RMSE outside	Windows
5 min	Persistence (reference)	1162	1.29	0.87	0.98	0.00	1.34	1.25	n/a
	AR(1) (reference)	1162	1.29	0.88	0.98	0.00	1.34	1.25	n/a
	Raw (forward-filled)	1162	10.94 ± 3.45	8.98	-0.39	-7.48	4.91	13.93	6,902
	Rules → repair	1162	9.90 ± 0.88	5.85	-0.06	-6.68	1.05	13.19	6,902
	LSTM-AE → repair	1162	11.01 ± 1.00	6.50	-0.31	-7.53	1.17	14.66	6,902
	Union → repair	1162	11.19 ± 1.11	6.73	-0.35	-7.67	1.30	14.89	6,902
	Drop-faulty	1162	13.48 ± 0.97	8.99	-0.95	-9.45	2.62	17.85	1,741
1 h	Persistence (reference)	1117	1.53	1.11	0.97	0.00	1.52	1.55	n/a
	AR(1) (reference)	1117	1.53	1.13	0.97	0.01	1.50	1.55	n/a
	Raw (forward-filled)	1117	10.71 ± 3.40	8.85	-0.43	-5.98	5.29	13.69	6,891
	Rules → repair	1117	10.03 ± 0.76	6.34	-0.16	-5.54	1.80	13.50	6,891
	LSTM-AE → repair	1117	10.47 ± 1.04	6.42	-0.26	-5.83	1.64	14.13	6,891
	Union → repair	1117	10.51 ± 1.09	6.53	-0.27	-5.85	1.74	14.16	6,891
	Drop-faulty	1117	12.20 ± 0.81	7.68	-0.71	-6.95	2.06	16.44	1,300
6 h	Persistence (reference)	1040	1.97	1.33	0.95	0.00	1.58	2.26	n/a
	AR(1) (reference)	1040	1.81	1.30	0.96	0.08	1.48	2.06	n/a
	Raw (forward-filled)	1040	12.83 ± 2.93	11.06	-1.29	-5.51	7.08	16.28	6,831
	Rules → repair	1040	11.39 ± 1.21	8.08	-0.74	-4.77	3.17	15.32	6,831
	LSTM-AE → repair	1040	11.46 ± 0.98	7.93	-0.75	-4.81	2.88	15.47	6,831
	Union → repair	1040	11.69 ± 0.85	8.27	-0.82	-4.93	3.19	15.75	6,831
	Drop-faulty	1040	14.46 ± 0.42	11.55	-1.78	-6.33	6.22	18.94	351

when the forecast is made (the two groupings agree on 82.1%, 77.8% and 72.7% of the test samples at 5 min, 1 h and 6 h; Supplementary Table S9), gives the same repair-versus-raw differences (-4.16 and -4.01 for rules- and autoencoder-guided repair at 5 min, -3.6 to -4.4 at 1 h and 6 h, intervals excluding zero), but it does not confirm that the repaired models reach persistence: at 5 min only rules-guided repair stays at the persistence level ($+0.21$, interval -0.02 to 0.43), autoencoder- and union-guided repair are worse ($+0.35$ and $+0.51$, intervals excluding zero), and at 1 h and 6 h every repaired variant is worse (rules-guided $+0.97$ and $+2.51$). In R3 about 70% of the retained test targets lie at 48–52, around the support edge of 50 where the level models saturate, and wherever the two groupings disagree the target lies at 48–53; grouping by the true target therefore selects the samples on which saturation happens to be right. In the predictions (Figure 8) the raw-trained model carries a persistent negative bias inherited from the out-of-range episode, which repair removes, and above the support every level model saturates near the top of its training range.

Table 12. Temporal inference on paired per-sample squared losses (seed-averaged) of the common test set: difference in RMSE ($a - b$), its 95% moving-block bootstrap interval (block 24 samples, 2,000 replicates) and the Diebold–Mariano statistic with Harvey correction. "Inside" restricts both members of the pair to test targets inside the training support. Negative differences favor a. Samples: 1,162, 1,117 and 1,040 (full) and 510, 510 and 487 (inside) at 5 min, 1 h and 6 h. The union variant and intervals with block lengths 12 and 48 and with a bootstrap over contiguous runs of retained samples are in Supplementary Table S8.

		Full test set				**Inside training support**			
Horizon	**Pair (a vs. b)**	ΔRMSE	95% CI	DM	p	ΔRMSE	95% CI	DM	p
5 min	rules→repair vs. persistence	+8.65	[+5.56, +10.49]	+4.88	<0.001	-0.28	[-0.43, -0.14]	-4.22	<0.001
	LSTM-AE→repair vs. persistence	+9.76	[+6.25, +11.95]	+4.92	<0.001	-0.16	[-0.31, -0.01]	-2.31	0.021
	raw vs. persistence	+10.13	[+8.17, +11.78]	+8.10	<0.001	+3.85	[+2.49, +4.92]	+5.11	<0.001
	rules→repair vs. raw	-1.48	[-2.88, -0.75]	-3.63	<0.001	-4.13	[-5.12, -2.85]	-5.32	<0.001
	LSTM-AE→repair vs. raw	-0.37	[-2.12, +0.54]	-0.68	0.494	-4.01	[-5.04, -2.71]	-5.28	<0.001
	drop-faulty vs. raw	+2.10	[-0.11, +3.35]	+2.69	0.007	-2.55	[-3.26, -1.66]	-4.82	<0.001
	LSTM-AE→repair vs. rules→repair	+1.11	[+0.75, +1.32]	+5.03	<0.001	+0.12	[+0.10, +0.15]	+11.28	<0.001
1 h	rules→repair vs. persistence	+8.52	[+5.53, +10.46]	+3.63	<0.001	+0.35	[+0.07, +0.57]	+2.33	0.020
	LSTM-AE→repair vs. persistence	+8.99	[+5.62, +11.05]	+3.58	<0.001	+0.15	[-0.07, +0.31]	+1.35	0.177
	raw vs. persistence	+9.66	[+7.80, +11.22]	+5.93	<0.001	+4.06	[+2.61, +5.24]	+3.78	<0.001
	rules→repair vs. raw	-1.14	[-2.66, -0.38]	-1.98	0.048	-3.71	[-4.61, -2.55]	-3.86	<0.001
	LSTM-AE→repair vs. raw	-0.67	[-2.51, +0.17]	-1.00	0.316	-3.91	[-4.88, -2.69]	-3.87	<0.001
	drop-faulty vs. raw	+1.04	[-0.99, +2.17]	+1.09	0.275	-3.47	[-4.24, -2.37]	-3.88	<0.001
	LSTM-AE→repair vs. rules→repair	+0.46	[+0.24, +0.60]	+3.09	0.002	-0.20	[-0.25, -0.13]	-4.00	<0.001
6 h	rules→repair vs. persistence	+9.47	[+6.40, +11.73]	+2.08	0.038	+1.70	[+1.41, +2.08]	+4.55	<0.001
	LSTM-AE→repair vs. persistence	+9.53	[+6.43, +11.93]	+2.02	0.043	+1.38	[+1.10, +1.73]	+4.19	<0.001
	raw vs. persistence	+11.16	[+9.30, +13.30]	+3.35	<0.001	+5.76	[+4.22, +7.17]	+2.30	0.022
	rules→repair vs. raw	-1.68	[-3.54, -0.69]	-1.49	0.136	-4.05	[-5.18, -2.77]	-2.09	0.037
	LSTM-AE→repair vs. raw	-1.63	[-3.56, -0.46]	-1.37	0.171	-4.38	[-5.58, -3.09]	-2.15	0.032
	drop-faulty vs. raw	+1.34	[-0.68, +2.64]	+0.79	0.432	-1.09	[-2.38, +0.45]	-0.70	0.483
	LSTM-AE→repair vs. rules→repair	+0.05	[-0.06, +0.13]	+0.52	0.600	-0.33	[-0.36, -0.30]	-5.98	<0.001

Which detector should feed the repair. With the stuck rule evaluated on constant runs of 5-min medians, the rules flag 89.6% of the training-period soil-moisture rows, the "rules-repaired" training stream consists largely of profile-filled values, and autoencoder-guided repair appears clearly better (RMSE 6.28 vs. 7.11); that difference is an artifact of the rule. With the raw-sample stuck rule the rules flag 52.4% of these rows, close to the autoencoder's 50.0%, and the two guided repairs differ by small amounts whose sign changes with the horizon: inside the support rules-guided repair is better at 5 min (difference 0.12, interval 0.10–0.15) and the autoencoder at 1 h and 6 h (by 0.20 and 0.33, intervals excluding zero), small against the 3.7–4.4-unit gain of either repair over raw training; the union adds nothing.

Treatment of the long outages. In the training period, 16.5% of rows lie inside outages longer than 12 h and 47.9% (rules) or 47.4% (autoencoder) of the soil-moisture target rows are filled with the unvalidated diurnal profile because their flagged run exceeds the 4-h linear-interpolation limit; the test set contains no filled value. Table 13 separates what the fallback does and does not drive. On the full test set, leaving every flagged run longer than 4 h unfilled and dropping the training windows that touch one (3,471 windows remain for rules, 3,579 for the autoencoder) changes the 5-min RMSE from 9.90 to 10.70 (rules) and from 11.01 to 10.89 (autoencoder), within one seed standard deviation; additionally excluding every window that intersects a real outage longer than 12 h (5,737 windows remain) degrades both variants to about 12.3–12.5, because the windows removed include the last days before the split, whose index values (44–50) are the closest the training data come to the test regime. Inside the training support (samples grouped by the true target), every treatment keeps the repaired models well ahead of raw training (1.5–3.0 vs. 4.9), but only the default fallback reaches the persistence level (1.05 and 1.17 vs. 1.34); without it the inside-support error is 1.5–1.6, and with the stricter exclusion 2.4–3.0. The benefit of repair over raw training is therefore robust to the long-gap treatment, while reaching persistence under the target grouping depends on it.

Additional level-free analysis (exploratory). Table 14 reports the change models. Removing the absolute level from the inputs removes the extrapolation failure: errors inside and outside the training support become similar. At 5 min the model selected for rules-repaired training data (ridge) attains RMSE 1.11 against 1.29 for persistence (skill 0.14; difference -0.18, run-based interval -0.22 to -0.12), and the model selected for raw data (the level-free LSTM) 1.22 (skill 0.05); rules-guided repair is better than raw training by 0.11 with the selected models and by 0.17 within the ridge family (both intervals excluding zero). The other repaired variants reach 1.16. Drop-faulty, which deletes rather than repairs, reaches 1.11 with a wider interval (-0.25 to -0.01). At 1 h validation selected

Table 13. Original level-predicting LSTM: ablation of the treatment of flagged runs longer than the validated 4-h linear-interpolation limit (5-min horizon, 10 seeds, RMSE mean $\pm$ sd on the full test set and on test targets inside the training support; persistence: 1.29 and 1.34). (i) as (ii), minus every training window that intersects a real outage longer than 12 h; (ii) default used in Table 11; (iii) long flagged runs left unfilled and the training windows that touch them dropped. Raw training: 10.94 and 4.91.

Variant	Long-gap treatment	Windows	RMSE, full	RMSE, inside
Rules $\rightarrow$ repair	(ii) diurnal fallback	6,902	9.90 ± 0.88	1.05 ± 0.12
	(i) minus >12-h outage windows	5,737	12.53 ± 1.32	2.97 ± 0.81
	(iii) no fill beyond 4 h	3,471	10.70 ± 1.04	1.55 ± 0.38
LSTM-AE $\rightarrow$ repair	(ii) diurnal fallback	6,902	11.01 ± 1.00	1.17 ± 0.18
	(i) minus >12-h outage windows	5,737	12.28 ± 0.78	2.39 ± 0.39
	(iii) no fill beyond 4 h	3,579	10.89 ± 0.58	1.54 ± 0.19

persistence for raw data and for rules- and autoencoder-guided repair, with very small validation margins; union-guided repair beats persistence (1.34 vs. 1.53, interval excluding zero), as does drop-faulty (1.32), and within the ridge family every repaired variant beats persistence (1.34–1.36) while raw training does not (1.73). At 6 h the selected models of the three repaired variants (ridge) are worse than persistence (2.25–2.30 vs. 1.97, intervals excluding zero), and drop-faulty is the only selected model below persistence (1.88), with a run-based interval that includes zero (the moving-block interval, -0.177 to -0.035, excludes it). The short-horizon gain is therefore produced jointly by the training-data treatment, deletion included, and by the level-free representation, not by imputation alone; deletion is competitive here because a linear model needs far fewer windows than an LSTM. The residual LSTM, which predicts the change but still sees the absolute level, reduces the 5-min error of the level models to 1.3–1.8 but does not beat persistence (Supplementary Table S7).

Robustness of the level-model results. Three further checks concern the original level-predicting LSTM (details in the Supplementary Material). (a) *Random-mask control* (Supplementary Table S6). On the prespecified primary outcome, the full test set, the random control has lower error than rules-guided repair for all five mask seeds (pooled difference -2.34, interval -5.11 to -0.62), and so does the split control. The advantage comes from outside the support, where models trained with random masks, which leave part of the out-of-range episode in the training data, follow their lagged input more closely; this is consistent with the full-set ranking of level models reflecting extrapolation rather than data quality. Inside the support rules-guided repair has lower error than the split control for all five mask seeds and all ten training seeds when averaged over the other factor (44 of 50 individual fits; pooled $+1.20$, interval 0.96 to 1.75), whereas against the primary control the result depends on the mask realization (three of five masks favor rules). The control therefore does not show a benefit of targeting on the full test set. (b) *Rolling-origin blocks and operational test sets* (Supplementary Table S5). In all three chronological blocks, with both the screened and the operational test set, persistence is the best forecast on the full test set and rules-guided repair beats raw training inside the support. The other comparisons are not uniform: on the full operational set of the post-outage block raw training is marginally better than rules-guided repair (13.37 vs. 13.55), autoencoder-guided repair is worse than raw training in that block, and in the first block, which follows a mostly healthy training period, deletion is as good as repair. (c) *Interval and seed sensitivity* (Supplementary Table S8). The inside-support conclusions hold with block lengths of 12 to 48 samples and with a bootstrap over contiguous runs of retained samples, whereas on the full test set that bootstrap rests on 12–14 runs and its intervals for repaired-versus-raw differences include zero; full-set comparisons between level-model variants are therefore indicative. Paired comparisons over the ten training seeds are kept as a statement about training stochasticity only.

5. DISCUSSION

Training-data treatment helps, but how much depends on the model and the horizon. The downstream results separate three effects that a single headline number would conflate. For the original level-predicting LSTM, repairing the training data removes the bias inherited from the out-of-range episode and cuts the error within the training range by about 4 index units relative to raw training at every horizon; it brings the forecaster to the persistence level there at 5 min when samples are grouped by the true target (at 1 h only for autoencoder-guided repair) and, when they are grouped by the last observed value, only for rules-guided repair. No treatment helps outside the training range, where the index returned from the outage at a level never seen in training. That is a regime change, and it has to be recognized as a sensor-health event (the node re-entered service in a different state, with one channel in variance collapse) rather than repaired as a data defect; because the support split uses the true target, a deployed system would need an input-side indicator of the regime, such as the sampling-density and daily-amplitude indicators of Section 3.1. When the forecaster receives no absolute level, treated training data support a forecaster that beats persistence on the whole test set at the shortest horizon, and the benefit over raw training appears there; at 1 h only some treatments beat persistence, at 6 h the selected repaired-data models are worse than it, and the ceiling stays low for a target that does not change in most 5-min steps. The value of a data treatment is therefore a property of the treatment, the model and the horizon together, and it should be measured

Table 14. Level-free change models (prediction = last observed value + predicted change; no absolute soil-moisture level among the inputs). For each training variant and horizon the candidate (persistence, ridge regression, level-free LSTM) with the lowest error on a chronological validation slice at the end of the training period is selected (detection, normalization and repair for that slice use only the earlier part of the training period) and then evaluated on the common test set. Ridge = RMSE of the ridge candidate for every variant (same model family, no selection). RMSE on the full test set and inside/outside the training support; SS = skill relative to persistence; Δ = pooled RMSE difference to persistence with the 95% bootstrap interval over contiguous runs of test samples (n/a when persistence is selected). For these exploratory models we report this more conservative bootstrap rather than the moving-block bootstrap of Table 12; with the moving-block bootstrap the 6-h drop-faulty difference to persistence would exclude zero (-0.177 to -0.035). Bold row label: rules→repair, the primary repaired variant; bold does not mark the best value. The selection among candidates is part of an exploratory analysis added after the main results; the intervals do not account for it. Persistence: 1.29, 1.53 and 1.97 at 5 min, 1 h and 6 h.

Horizon	Training data	Selected	RMSE full	SS	RMSE inside	RMSE outside	Ridge	Δ vs. persistence [95% CI]
5 min	Raw	LSTM	1.22	0.05	1.24	1.21	1.28	-0.07 [-0.09, -0.05]
	Rules→repair	ridge	1.11	0.14	1.13	1.10	1.11	-0.18 [-0.22, -0.12]
	LSTM-AE→repair	ridge	1.16	0.10	1.19	1.13	1.16	-0.13 [-0.16, -0.10]
	Union→repair	ridge	1.16	0.10	1.19	1.13	1.16	-0.13 [-0.16, -0.10]
	Drop-faulty	ridge	1.11	0.14	1.11	1.12	1.11	-0.18 [-0.25, -0.01]
1 h	Raw	persistence	1.53	0.00	1.52	1.55	1.73	n/a
	Rules→repair	persistence	1.53	0.00	1.52	1.55	1.36	n/a
	LSTM-AE→repair	persistence	1.53	0.00	1.52	1.55	1.34	n/a
	Union→repair	ridge	1.34	0.13	1.35	1.33	1.34	-0.20 [-0.23, -0.09]
	Drop-faulty	ridge	1.32	0.14	1.24	1.39	1.32	-0.21 [-0.25, -0.08]
6 h	Raw	persistence	1.97	0.00	1.58	2.26	5.02	n/a
	Rules→repair	ridge	2.30	-0.17	1.64	2.76	2.30	+0.33 [+0.23, +0.53]
	LSTM-AE→repair	ridge	2.27	-0.15	2.00	2.48	2.27	+0.29 [+0.13, +0.56]
	Union→repair	ridge	2.25	-0.14	1.98	2.47	2.25	+0.28 [+0.13, +0.56]
	Drop-faulty	ridge	1.88	0.05	1.32	2.25	1.88	-0.10 [-0.15, +0.02]

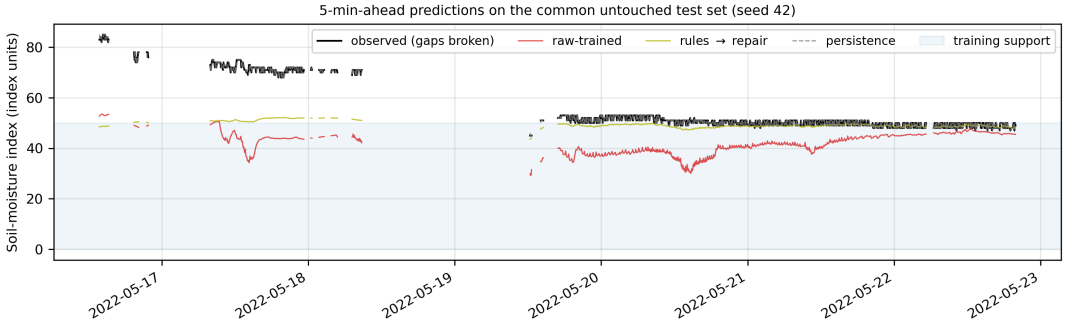


Figure 8. 5-min-ahead predictions of the original level-predicting LSTMs on the common untouched test set (seed 42). The observed series is broken at every gap in the retained samples; the shaded band is the training support (0–50 index units). The raw-trained model carries a persistent negative bias; the model trained on rules-repaired data, the primary repaired variant, tracks the observations within the support and saturates above it, as does every learned level model.

against persistence for the model that will actually consume the data. The detector that feeds the repair matters less than how it is defined: once the stuck rule is evaluated on the raw samples, rules- and autoencoder-guided repair differ only slightly, whereas a rule evaluated on resampled medians flags a healthy, slow channel as stuck and changes the downstream conclusion. Detection benchmarks and downstream utility are related but not identical objectives.

Repair is not always better than deletion, and it does not fix every fault. Deleting faulty windows was the worst option for the data-hungry level LSTM, but for a linear change model it was as good as repair, and at 1 h it was one of the two treatments that beat persistence; repair is preferable when the model needs the windows that deletion removes, not universally. Repair also has hard limits. Its fidelity was validated only up to 12 h, whereas almost all missing time lay in longer outages, which were filled for training with an unvalidated diurnal profile on which reaching the persistence level (samples grouped by the true target) depends; the zero-variance and suspected-bias channels could not be repaired at all; and faults that are not detected cannot be repaired. Low-magnitude bias and drift evade all three single-sensor detectors and quasi-stuck episodes evade them entirely. The deployment shows both ends of this gap: the CO_2 level error is detectable only from its implausible long-run median and the absence of a nighttime rise, and the air-temperature variance collapse only from the loss of diurnal amplitude. Rules and the autoencoder are

complementary for coverage (the rules for the dominant overt classes, the autoencoder for transients, Isolation Forest near the stuck threshold), but cross-sensor prediction (Shekarian et al., 2024), amplitude-based plausibility checks over daily windows, periodic reference calibration and hardware redundancy remain necessary complements. Training learned detectors on contaminated data is a related open problem (Ulmer et al., 2024; Kim et al., 2024); our screened-clean windows are the simplest answer, and the zero false-positive rate of the rules on those windows shows its cost.

Scope of the evidence. The study examines one uncurated deployment in depth, a single greenhouse over 34 days, so that every stage of the chain is evaluated on the same faults; the fault classes are canonical (Ni et al., 2009; Sharma et al., 2010), but their prevalences describe this deployment rather than a population. Only four clean windows are held out, so detector comparisons are decisive only for large effects; the temperature windows come from the variance-collapse regime; soil temperature, CO_2 and light are not covered by the detection benchmark; real-fault labels are inferred from physical plausibility and temporal behavior without reference instrumentation; and the clean-window screen shares its family with the rules detector, which favors rules by an amount that cannot be quantified without independently labeled data. The repair policy was selected on humidity windows only, because no clean soil-moisture or temperature window precedes the downstream training/test boundary under the median-based clean screen; with the raw-sample constancy criterion six soil-moisture windows would precede it, and selecting the policy on them gives linear interpolation in every condition (Supplementary Section S2). The downstream test set keeps only screened-clean samples, which cover less than half of the test period and its drier, quieter part, and it cannot measure accuracy during faulty periods, where forecasts may matter most; temporal inference rests on a block bootstrap over non-contiguous samples, and the rolling-origin design adds only three blocks. The level-free change models and the random-mask control were added after the main results were known: their protocols were fixed before training and their selection used training data only, but their intervals do not account for the sequence of analysis choices, so they are exploratory and need confirmation on other deployments. Finally, the pipeline is a diagnostic and data-quality assessment of the sensing layer; it does not model physical sensor degradation or predict when a sensor will fail, and generalization to other farms, sensor models, seasons or climates is not demonstrated.

6. CONCLUSION

We turned an uncurated 34-day greenhouse IoT deployment into a study of sensor data health, from fault diagnosis through data repair to the consequences for a downstream forecaster. Every one of its six streams carried at least one fault signature, including a change of operating regime after the longest outage that point-wise rules cannot see. Rule-based screening and an LSTM autoencoder were complementary (macro-F1 0.44 and 0.36 overall, 0.33 and 0.23 on the value-corrupting classes), low-magnitude bias, drift and quasi-stuck faults remained difficult for all detectors, and linear interpolation was the best repair, or not significantly different from the best, in every evaluated condition except 12-h single-sensor gaps, with no fidelity claim beyond 12 h. Downstream, under a leakage-controlled chronological protocol, the original LSTM forecasters of the soil-moisture level did not beat persistence at any horizon from 5 min to 6 h. Within the training range, repaired data reduced their error by about 4 index units relative to raw data at every horizon; at 5 min they reached the persistence level when test samples were grouped by the true target (mean RMSE over seeds 1.05–1.30 vs. 1.34, Table 11) and, when grouped by the last observed value, which is known at prediction time, only with rules-guided repair (pooled RMSE 1.56 vs. 1.35, Supplementary Table S9). Under the grouping available at prediction time, the training data therefore meet our definition of usable only with rules-guided repair and only at 5 min. A random-mask control did not show a benefit of targeted repair on the full test set. In an exploratory analysis added after these results, level-free change models selected on training data beat persistence on the whole test set at 5 min when trained on repaired data (skill 0.14, against 0.05 for raw data); at 1 h only union-guided repair and, separately, deletion of faulty windows did, and at 6 h the selected repaired-data models were worse than persistence. The practical lessons are to define screening rules on the raw time base, report data health on a time basis, attach a validated horizon to every repair, compare deletion with repair for the model that will consume the data, evaluate every forecaster against persistence, and treat a change of operating regime as a sensor-health event rather than a data defect. The conclusions are deployment-specific, and the pipeline addresses diagnosis and data restoration rather than prognosis of sensor failure. Together with our earlier studies of prediction (Ghazouani et al., 2023) and explanation (Lamhour et al., 2026) on the same system, this work adds the upstream sensor-health layer those studies presumed.

ACKNOWLEDGMENT

The authors acknowledge the use of artificial intelligence language models (ChatGPT, OpenAI; Claude, Anthropic) and the writing assistant Grammarly to assist with the drafting, linguistic refinement, and formatting of parts of this manuscript. All technical content, experimental design, analyses, and conclusions were conceived, implemented, and verified by the authors. The authors reviewed and edited all AI-assisted text and bear full responsibility for the final content of the publication.

DATA AND CODE AVAILABILITY

The raw deployment dataset and the complete analysis code supporting this study, including every analysis of the paper and of the Supplementary Material with its protocols, per-fit predictions and the numerical tables underlying every figure, are publicly available at `https://github.com/Lamhour-Mohamed-Akram/greenhouse-iot-fault-detection-repair`.

Supplementary material. Supplementary Tables S1–S9 and Figures S1–S4 (injected fault examples, threshold and window-length sensitivity, near-threshold scenarios, per-sensor imputation errors, rolling-origin and operational test sets, the random-mask control, the residual LSTM, interval sensitivity, and the training-support split by the last observed value) accompany this article.

Funding. This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest. The authors declare no conflict of interest.

REFERENCES

Adeyemi, O., Grove, I., Peets, S., Domun, Y., & Norton, T. (2018). Dynamic neural network modelling of soil moisture content for predictive irrigation scheduling. *Sensors*, *18*(10), 3408. doi: 10.3390/s18103408

AlSalehy, A. S., & Bailey, M. (2025). Improving time series data quality: Identifying outliers and handling missing values in a multilocation gas and weather dataset. *Smart Cities*, *8*(3), 82. doi: 10.3390/smartcities8030082

Benameur, R., Dahane, A., Kechar, B., & Benyamina, A. E. H. (2024). An innovative smart and sustainable low-cost irrigation system for anomaly detection using deep learning. *Sensors*, *24*(4), 1162. doi: 10.3390/s24041162

Blázquez-García, A., Conde, A., Mori, U., & Lozano, J. A. (2021). A review on outlier/anomaly detection in time series data. *ACM Computing Surveys*, *54*(3), 1–33. doi: 10.1145/3444690

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. In *Proceedings of the 2000 acm sigmod international conference on management of data* (pp. 93–104). doi: 10.1145/342009.335388

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), 1–58. doi: 10.1145/1541880.1541882

Contreras-Castillo, J., Guerrero-Ibañez, J. A., Santana-Mancilla, P. C., & Anido-Rifón, L. (2023). SAgric-IoT: An IoT-Based platform and deep learning for greenhouse monitoring. *Applied Sciences*, *13*(3), 1961. doi: 10.3390/app13031961

Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *13*(3), 253–263. doi: 10.1080/07350015.1995.10524599

Du, W., Côté, D., & Liu, Y. (2023). SAITS: Self-attention-based imputation for time series. *Expert Systems with Applications*, *219*, 119619. doi: 10.1016/j.eswa.2023.119619

El-Shafeiy, E., Alsabaan, M., Ibrahem, M. I., & Elwahsh, H. (2023). Real-time anomaly detection for water quality sensor monitoring based on multivariate deep learning technique. *Sensors*, *23*(20), 8613. doi: 10.3390/s23208613

Ghazouani, M., Azzouazi, M., & Lamhour, M. A. (2023). A drip irrigation prediction system in a greenhouse based on long short-term memory and connected objects. *Mathematical Modeling and Computing*, *10*(2), 524–533. doi: 10.23939/mmc2023.02.524

Harvey, D., Leybourne, S., & Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, *13*(2), 281–291. doi: 10.1016/S0169-2070(96)00719-4

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Hua, V., Nguyen, T., Dao, M.-S., Nguyen, H. D., & Nguyen, B. T. (2024). The impact of data imputation on air quality prediction problem. *PLOS ONE*, *19*(9), e0306303. doi: 10.1371/journal.pone.0306303

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., & Soderstrom, T. (2018). Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining* (pp. 387–395). doi: 10.1145/3219819.3219845

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688. doi: 10.1016/j.ijforecast.2006.03.001

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, *147*, 70–90. doi: 10.1016/j.compag.2018.02.016

Kandamali, D. F., Porter, E., Porter, W. M., McLemore, A., Kiobia, D. O., Tavandashti, A. P., & Rains, G. C. (2025). Hybrid LSTM method for multistep soil moisture prediction using historical soil moisture and weather data. *AgriEngineering*, *7*(8), 260. doi: 10.3390/agriengineering7080260

Khayati, M., Lerner, A., Tymchenko, Z., & Cudré-Mauroux, P. (2020). Mind the gap: an experimental evaluation of imputation of missing values techniques in time series. *Proceedings of the VLDB Endowment*, *13*(5), 768–782. doi: 10.14778/3377369.3377383

Kim, M., Yu, J., Kim, J., Oh, T.-H., & Choi, J. K. (2024). An iterative method for unsupervised robust anomaly detection under data contamination. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(10), 13327–13339. doi: 10.1109/TNNLS.2023.3267028

Koné, B. A. T., Grati, R., Bouaziz, B., & Boukadi, K. (2023). A new long short-term memory based approach for soil moisture prediction. *Journal of Ambient Intelligence and Smart Environments*, *15*(3), 255–268. doi: 10.3233/AIS-230035

Künsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics*, *17*(3), 1217–1241. doi: 10.1214/aos/1176347265

Lamhour, M. A., Msalek, M., Ardchir, S., Ghazouani, M., & Azzouazi, M. (2026). Multi-seed XAI validation for reliable LSTM irrigation control. *IEEE Access*, *14*, 120366–120387. doi: 10.1109/ACCESS.2026.3720462

Li, Z., Abdullah, N., & Ishak, M. K. (2025). Smart greenhouse climate control with real-time fault detection and energy-aware automation. *Smart Agricultural Technology*, *13*, 101707. doi: 10.1016/j.atech.2025.101707

Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine learning in agriculture: A review. *Sensors*, *18*(8), 2674. doi: 10.3390/s18082674

Lindemann, B., Maschler, B., Sahlab, N., & Weyrich, M. (2021). A survey on anomaly detection for technical systems using LSTM networks. *Computers in Industry*, *131*, 103498. doi: 10.1016/j.compind.2021.103498

Little, R. J. A., & Rubin, D. B. (2019). *Statistical analysis with missing data* (3rd ed.). Wiley. doi: 10.1002/9781119482260

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. In *Proceedings of the 2008 eighth ieee international conference on data mining* (pp. 413–422). doi: 10.1109/ICDM.2008.17

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2012). Isolation-Based anomaly detection. *ACM Transactions on Knowledge Discovery from Data*, *6*(1), 3. doi: 10.1145/2133360.2133363

Moritz, S., & Bartz-Beielstein, T. (2017). imputeTS: Time series missing value imputation in R. *The R Journal*, *9*(1), 207–218. doi: 10.32614/RJ-2017-009

Ni, K., Ramanathan, N., Chehade, M. N. H., Balzano, L., Nair, S., Zahedi, S., … Srivastava, M. (2009). Sensor network data fault types. *ACM Transactions on Sensor Networks*, *5*(3), 25. doi: 10.1145/1525856.1525863

Ojha, T., Misra, S., & Raghuwanshi, N. S. (2015). Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. *Computers and Electronics in Agriculture*, *118*, 66–84. doi: 10.1016/j.compag.2015.08.011

Pearson, R. K., Neuvo, Y., Astola, J., & Gabbouj, M. (2016). Generalized Hampel filters. *EURASIP Journal on Advances in Signal Processing*, *2016*, 87. doi: 10.1186/s13634-016-0383-6

Poorter, H., Fiorani, F., Stitt, M., Schurr, U., Finck, A., Gibon, Y., … Pons, T. L. (2012). The art of growing plants for experimental purposes: A practical guide for the plant biologist. *Functional Plant Biology*, *39*(11), 821–838. doi: 10.1071/FP12028

Săcăleanu, D.-I., Matache, M.-G., Roşu, Ş.-G., Florea, B.-C., Manciu, I.-P., & Perişoară, L.-A. (2024). Iot-enhanced decision support system for real-time greenhouse microclimate monitoring and control. *Technologies*, *12*(11), 230. doi: 10.3390/technologies12110230

Schmidl, S., Wenig, P., & Papenbrock, T. (2022). Anomaly detection in time series: A comprehensive evaluation. *Proceedings of the VLDB Endowment*, *15*(9), 1779–1797. doi: 10.14778/3538598.3538602

Sharma, A. B., Golubchik, L., & Govindan, R. (2010). Sensor faults: Detection methods and prevalence in real-world datasets. *ACM Transactions on Sensor Networks*, *6*(3), 23. doi: 10.1145/1754414.1754419

Shekarian, S. M., Aminian, M., Mohammad Fallah, A., & Akbary Moghaddam, V. (2024). AI-powered sensor fault detection for cost-effective smart greenhouses. *Computers and Electronics in Agriculture*, *224*, 109198. doi: 10.1016/j.compag.2024.109198

Simo, A., Dzitac, S., Badea, G. E., & Meianu, D. (2022). Smart agriculture: IoT-based greenhouse monitoring system. *International Journal of Computers Communications & Control*, *17*(6), 5039. doi: 10.15837/ijccc.2022.6.5039

Teshome, F. T., Bayabil, H. K., Schaffer, B., Ampatzidis, Y., & Hoogenboom, G. (2024). Improving soil moisture prediction with deep learning and machine learning models. *Computers and Electronics in Agriculture*, *226*, 109414. doi: 10.1016/j.compag.2024.109414

Thai, T.-T. N., Do, P., Iung, B., & Gardoni, P. (2025). Impact of sensor degradation on measurement uncertainty in prognostics and maintenance decision-making: State of the art and challenges. *International Journal of Prognostics and Health Management*, *16*(4). doi: 10.36001/ijphm.2025.v16i4.4306

Trapani, N., & Longo, L. (2023). Fault detection and diagnosis methods for sensors systems: a scientific literature review. *IFAC-PapersOnLine*, *56*(2), 1253–1263. doi: 10.1016/j.ifacol.2023.10.1749

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., … Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, *17*(6), 520–525. doi: 10.1093/bioinformatics/17.6.520

Tzounis, A., Katsoulas, N., Bartzanas, T., & Kittas, C. (2017). Internet of Things in agriculture, recent advances and future challenges. *Biosystems Engineering*, *164*, 31–48. doi: 10.1016/j.biosystemseng.2017.09.007

Ulmer, M., Zgraggen, J., & Goren Huber, L. (2024). A generic

machine learning framework for fully-unsupervised anomaly detection with contaminated data. *International Journal of Prognostics and Health Management*, *15*(1), 1–12. doi: 10.36001/ijphm.2024.v15i1.3589

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, *45*(3), 1–67. doi: 10.18637/jss.v045.i03

Wang, Y., Shi, L., Hu, Y., Hu, X., Song, W., & Wang, L. (2024). A comprehensive study of deep learning for soil moisture prediction. *Hydrology and Earth System Sciences*, *28*(4), 917–943. doi: 10.5194/hess-28-917-2024

Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M.-J. (2017). Big data in smart farming – a review. *Agricultural Systems*, *153*, 69–80. doi: 10.1016/j.agsy.2017.01.023

Zhabko, O., Laktionov, I., Diachenko, G., Vinyukov, O., & Moroz, D. (2026). Agroclimatic forecasting under degraded sensor data: A robustness benchmark of machine-learning models. *Applied Sciences*, *16*(10), 5075. doi: 10.3390/app16105075

Zou, X., Liu, W., Huo, Z., Wang, S., Chen, Z., Xin, C., … Shu, L. (2023). Current status and prospects of research on sensor fault diagnosis of agricultural Internet of Things. *Sensors*, *23*(5), 2528. doi: 10.3390/s23052528