information

MDPI

Article

Physics-Attested Federated Learning: Securing Collaborative Anomaly Detection in Critical Water Infrastructure

Jeff Nijsse [1,*], Shu Su [2], Benjamin Oholeguy [3] and Sreenivas Sremath Tirumala [4]

[1] Department of Software Engineering, RMIT University, Hanoi 10000, Vietnam
[2] Department of Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand; shu.su@aut.ac.nz
[3] Independent Researcher, Auckland 1010, New Zealand
[4] School of Science, Engineering and Technology, RMIT University, Hanoi 10000, Vietnam; sreenivas.tirumala@rmit.edu.vn
* Correspondence: jeff.nijsse@rmit.edu.vn

Abstract

Federated learning enables industrial operators to train shared intrusion detection models without disclosing proprietary operational telemetry. However, existing defences operate strictly in update space, leaving aggregators blind to data poisoning; model updates derived from fabricated telemetry remain indistinguishable from honest contributions. We repurpose process invariants, physical relations that always hold in normal operation, such as the conservation of mass, from runtime detection heuristics into a verifiable admission requirement for federated updates, mined automatically from clean operational data. We evaluate this admission gate across two physical water testbeds (SWaT and WADI) and a distribution benchmark (BATADAL), testing seven aggregation rules against telemetry fabrication, exposure-only replay poisoning, and an invariant-aware adaptive adversary. Across three testbeds, the mined invariants reject none of 25 honest shards except every cross-channel fabrication on SWaT and BATADAL, including optimised perturbations that the similarity-based FoolsGold admits in full. On real telemetry, five mined invariants detect 12 of SWaT's 35 attacks, while nine invariants detect 20. With nine invariants, the admission gate removes 78% of the targeted-attack recall lost to replay poisoning (95% CI 57–97%, $p < 0.001$ over 20 seeds) across five standard aggregators and 82% under FedAvg. To reconcile physical admission control with federated data privacy, we show invariant compliance using zero-knowledge proofs (zk-SNARKs) to allow clients to prove batch adherence without revealing operational telemetry.

Keywords: federated learning; data poisoning; industrial control systems; process invariants; anomaly detection; zero-knowledge proofs

Check for updates

Academic Editor: Chingnung Yang

Received: 28 September 2026
Revised: 29 October 2026
Accepted: 2 November 2026
Published: 4 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Critical infrastructure networks across water, energy, and transport are increasingly targeted through operational-technology attack surfaces. In 2025, Dragos recorded 119 ransomware groups active against industrial organisations, up from 80 the previous year, having a mean dwell time of 42 days inside operational networks [1]. Water and wastewater facilities face acute operational exposure: In late 2023, the United States Cybersecurity and Infrastructure Security Agency issued an advisory confirming active exploitation of Unitronics PLCs across municipal water authorities [2], reflecting systemic vulnerabilities

documented across the sector [3]. Countering such intrusions requires detection mechanisms that improve as rapidly as adversaries adapt. For learning-based detectors, however, detection capability is fundamentally constrained by training data scarcity rather than model architecture [4,5].

Applying machine learning (ML) for process data analysis is standard practice, as ML-based methods can learn complex patterns and relationships within data and facilitate the automated detection of anomalies and intrusions. An anomaly detector is trained on continuous sensor streams, actuator states, and set points from supervisory control and data acquisition (SCADA) systems to learn a plant's normal operating envelope and flag physical departures from it [6–8]. A principal bottleneck of this paradigm is the scarcity of localised data.

An isolated plant operates within a narrow regime, records rare documented incidents and has virtually no labelled attack telemetry. Consequently, detectors trained at one facility generalise poorly to others. Federated learning (FL) offers an architectural remedy by training locally and transmitting only parameter updates to an aggregation server, where operators can collaboratively train a detector informed by multiple facilities while retaining proprietary telemetry on site [9,10]. This formulation aligns naturally with industrial requirements, where operational data are commercially sensitive and subject to critical-infrastructure mandates such as the European Union's NIS2 Directive [11], motivating the expanding literature on federated intrusion detection in operational technology [12].

Because federated aggregation conceals local data to protect the privacy of individual participants, the coordinator cannot inspect the telemetry from which an update was derived [13,14]. A single compromised operator can exploit this blind spot by training on corrupted telemetry, thereby introducing an exposure poison that teaches the shared detector to reconstruct specific attack profiles as normal operating behaviour [15]. When the target system protects physical infrastructure, a detector conditioned to ignore an attack signature constitutes an immediate safety failure that persists throughout the adversary's dwell time. The federation thus concentrates operational risk at the exact juncture where it promised collective defence.

Existing federated defences operate exclusively in the update space. Byzantine-robust aggregation rules inspect update geometry, discounting vectors that deviate from the consensus [16–19], while cryptographic input validation enforces norm bounds on hidden updates [20–22]. Neither of these mechanisms examines the data behind the update. An update trained on entirely fabricated telemetry is indistinguishable from that computed on honest plant records. Update-space defences are known to fail against adversaries who optimise against them [23–25], while norm bounds constrain only vector magnitude rather than directional alignment [20]. More fundamentally, robust aggregation relies on consensus: It assumes that honest updates cluster closely enough that any outlier is malicious. However, in industrial infrastructure, differing equipment, set points, and diurnal cycles invalidate that consensus [26]. Honest updates are inherently dispersed, allowing an adversary to hide entirely within legitimate operational variance.

Industrial telemetry has an intrinsic constraint that consumer FL data lacks: It is bounded by the laws of physics. For example, mass balances dictate that tank levels change according to net flow, while hydraulic couplings require that active pumps move fluid and inactive pumps do not. These process invariants are well-established heuristics for real-time intrusion detection [27–30]. Although an adversary can construct fabricated telemetry that preserves every marginal statistic per-channel, it cannot easily satisfy multivariate physical constraints. In this paper, we convert process invariants from a detection heuristic into an *admission requirement*: a mathematical condition that a client's local training batch must satisfy before its update is accepted into the round (Figure 1).

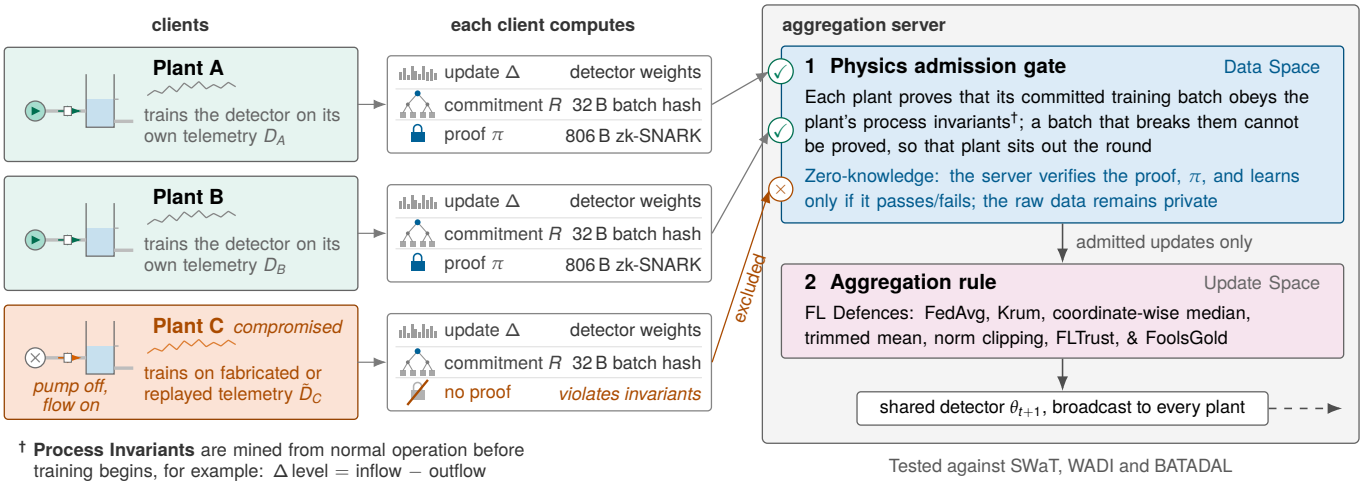


Figure 1. Physics-attested federated learning. Plants train a shared intrusion detector by only exchanging model updates. We introduce a data-space admission requirement: Each plant proves, in zero-knowledge, that its training batch satisfies mined process invariants (e.g., flow rates, mass balances). Batches violating these invariants cannot produce a proof and are excluded before aggregation.

Additionally, because these invariants are affine across measured channels, verification reduces to a set of bounded range checks that a client can prove in zero-knowledge over a committed training batch without disclosing raw telemetry. The resulting admission gate acts directly on the training data, operating before aggregation rules.

This paper investigates four primary research questions:

RQ1: Do process invariants mined from normal operation separate honest operational telemetry from fabricated telemetry, without rejecting honest clients?

RQ2: Can an adaptive adversary train on physics-violating telemetry to craft model updates that evade robust federated aggregation?

RQ3: How much poisoning capability is neutralised when an adaptive adversary is forced to satisfy the invariants, and does this hold across aggregation rules?

RQ4: Can a participant prove batch compliance with process invariants in zero-knowledge at a cost compatible with federated rounds?

The principal contributions of this work are fourfold:

1. A data-space admission gatebased on automatically mined affine process invariants that rejects every synthetic fabrication breaking cross-channel relations (every cross-channel fabrication on SWaT and BATADAL) while achieving zero false rejections across 25 distinct honest client shards (10 SWaT, 10 WADI, and 5 BATADAL).
2. An Orthogonality Result for Data- and Update-Space Defences. We demonstrate across seven aggregation rules and eight attacks that four of the rules admit physics-violating fabrications at 73–100% (including an optimised perturbation that FoolsGold admits in full), while the data gate rejects every fabrication and admits every update-space attack.
3. A Removal-and-Coverage Analysis Under Adaptive Attack. Against an adaptive adversary that projects corrupted batches back onto the invariants, we show that the gate removes 78% of targeted recall degradation pooled across five aggregation rules (95% CI 57–97%), reaching 82% under FedAvg.
4. A Zero-Knowledge Feasibility Demonstration (PA-FL Lite). We construct and benchmark a succinct zero-knowledge attestation protocol enabling clients to prove batch invariant compliance over sampled rows without disclosing operational telemetry.

The remainder of this article is organised as follows. Section 2 reviews related work in robust FL, cryptographic validation, and process invariants. Section 3 formalises the system

model, threat framework, and admission protocol. Section 4 details the experimental setup and benchmark datasets. Section 5 presents the empirical results across the three operational plants. Section 6 discusses the theoretical and practical implications, and Section 7 concludes this paper.

2. Background and Related Work

The existing literature is divided into three disconnected regimes: (1)Robust aggregation modifies server-side vector combination heuristics, (2) physics-based monitoring evaluates streaming SCADA telemetry at runtime, and (3) cryptographic federated learning inspects encrypted gradients or proves coordinator fidelity. However, none of these paradigms verify the physical validity of client training data.

2.1. Byzantine-Robust Aggregation and Evasion in Federated Learning

Byzantine-robust aggregation is the standard defence against poisoning. This regime addresses adversarial threats at the aggregation server. In traditional FL, distributed participants train local models on private data and transmit parameter updates to a central coordinator. Federated averaging (FedAvg) [9] combines these updates via a weighted arithmetic mean. Because linear averaging grants equal or proportional trust to all submissions, a single compromised participant can arbitrarily alter the trajectory of the shared model. Byzantine-robust aggregation presents myriad rules to attempt to neutralise this vulnerability by evaluating the geometric configuration of received update vectors. Table 1 summarises these representative approaches, their underlying mechanisms, and the key assumption on which their robustness depends.

Table 1. Representative Byzantine-robust aggregation mechanisms.

Category	Method	Mechanism	Key Assumption
Coordinate-wise statistics	Coordinate-wise median [17]	Evaluates updates coordinate-wise, selecting the median value for each parameter dimension.	Adversarial perturbations occupy an empirical minority in every coordinate.
	Trimmed mean [17]	Discards the upper and lower β-fractions along each coordinate before averaging.	Adversarial perturbations occupy an empirical minority in every coordinate.
Geometric proximity	Krum [16]	Identifies the single update that minimises the sum of squared Euclidean distances to nearest neighbours.	Benign updates form a dense spatial cluster in parameter space.
	Multi-Krum [16]	Averages multiple low-scoring updates selected via Krum's distance metric.	Honest updates maintain geometric coherence relative to Byzantine updates.
Magnitude control	Norm clipping [31]	Rescales update vectors whose Euclidean norm exceeds a threshold τ.	Bounding update norms limits the gradient influence of arbitrary perturbations.
Trusted reference	FLTrust [18]	Assigns client aggregation weights based on cosine similarity between each client update and a trusted reference gradient evaluated.	Server possesses a clean, representative validation dataset.
Temporal consistency	FoolsGold [19]	Penalises client groups exhibiting sustained multi-round collinearity.	Sybils controlled by a common adversary exhibit higher mutual similarity than honest clients.

Targeted poisoning strategies systematically bypass these update-space heuristics. Adversaries can inject persistent backdoors by scaling gradients to dominate the aggregate [13] or formulate data poisoning by matching surrogate gradients [32]. When adversaries possess knowledge of the aggregation rule and benign data distribution, they can optimise

perturbations to lie within the empirical coordinate variance of honest clients, evading distance-based and dimension-wise filters [14,23–25].

ML security surveys demonstrate that defences evaluated solely against static or generic attacks frequently suffer from an illusion of security; when evaluated against adaptive adversaries who possess full knowledge of the defence mechanism and optimise against it, empirical protections routinely collapse [33–35].

However, in these formulations, existing robust aggregation rules only evaluate the parameter vectors submitted Δ_i; none inspect or constrain the underlying training telemetry D_i (in Figure 1) from which those parameters were calculated.

2.2. Physics-Based Anomaly Detection in Industrial Control Systems

Industrial control security relies extensively on verifying sensor-actuator consistency against the governing hydraulic, thermodynamic, and mechanical relationships [30,36]. Process invariants formalise these relationships as algebraic constraints across monitored channels, including flow correlations across active pumps and mass balances across storage vessels. Whereas early methods manually derived invariants from process design schematics [27], subsequent techniques automated the extraction of invariants from uncorrupted SCADA historical records using association rule mining, linear regression, and first-order predicate logic [28,29]. Complementary physics-based formulations monitor sensor noise profiles [37] or pair neural system identification with Bayesian state estimation [38], while physics-informed ML embeds the governing equations in the learned model itself [39].

A central operational reality of process invariants is that their defensive capability is strictly bounded by sensor instrumentation. An audit by Cho et al. [40] across the entire SWaT incident record revealed that mined invariants flagged only 13–16 of 36 physical attacks; intrusions targeting unmetered stages or shifting states within normal operational noise escape invariant violations entirely [36].

2.3. Cryptographic Verification in Federated Learning

Secure aggregation protocols rely on threshold cryptography to compute parameter sums over masked updates, preventing the central coordinator from inspecting individual client models [41]. Because homomorphic masking conceals individual client vectors, it precludes geometric anomaly detection by the server, inadvertently shielding poisoned updates. Cryptographic input validation addresses this limitation by verifying structural properties of hidden updates in ZK. RoFL [20] and ACORN [21] employ ZK arguments to verify that encrypted updates adhere to pre-committed ℓ_2 and ℓ_∞ norm bounds without exposing vector coordinates. Armadillo [22] formalises privacy-utility trade-offs for such validation under malicious server threat models. In parallel, verifiable FL frameworks confirm the integrity of the aggregation, proving that the coordinator executed the aggregation arithmetic faithfully or that global models maintain specific architectural constraints, using homomorphic hashing [42] or zero-knowledge proofs [43,44].

Succinct zero-knowledge arguments (zk-SNARKs), notably Groth16 [45] combined with arithmetisation-oriented hash functions such as Poseidon [46,47], have demonstrated practical efficiency for ML inference verification and sensor telemetry commitments [48,49]. However, proving end-to-end model training remains computationally prohibitive because rolling backpropagation and stochastic gradient descent generate billions of arithmetic constraints per iteration [50,51]; proof of learning is cheaper, re-executing only sampled training steps, but exposes their data to the verifier [52]. Available cryptographic schemes consequently verify operations on model updates or server-side arithmetic; none verify whether a participant's local training dataset represents physically valid telemetry.

2.4. Operational Testbeds and Baseline Detectors

Empirical evaluation of industrial control security relies on hardware-in-the-loop physical testbeds and high-fidelity hydraulic simulations that capture real-world cyber-physical dynamics. To this end we employ the following:

SWaT (Secure Water Treatment): A scaled operational water purification facility producing five gallons of treated water per minute in six sequential physical stages (raw water intake, chemical dosing, ultrafiltration, dechlorination, reverse osmosis, and backwash), fully instrumented with 51 physical sensors and actuators of the PLC [53,54].

WADI (Water Distribution): An operational water supply and distribution network testbed that captures municipal consumer distribution loops, booster pumps, and elevated storage reservoirs in 127 monitored sensor and actuator channels [55].

BATADAL: A city-scale benchmark hydraulic simulation of the C-Town distribution network (comprising seven storage tanks, 11 pumps, and 43 pipes) modelled in EPANET under realistic diurnal consumer demand [56].

HAI (Hardware-in-the-Loop Augmented ICS): A multi-process testbed integrating thermal-power generation, a boiler steam turbine loop, and industrial water treatment with hardware-in-the-loop simulation [57].

Unsupervised deep learning architectures constitute the standard detection baseline across these benchmarks. In particular, 1D convolutional and recurrent reconstruction autoencoders operating on sliding windows of normalised sensor telemetry learn the plant's normal operating envelope and flag anomalies when reconstruction errors exceed a calibrated threshold [7,8]. Although these testbeds serve as established benchmarks for centralised intrusion detection and have been used for federated anomaly detection [58], they have not been studied with a physics-based admission check on client training data nor under coordinated data-poisoning attacks that exploit physical process bounds.

3. Methodology: Physics-Attested Admission

3.1. System Model and Threat Framework

We consider a federation of n industrial facilities—in this context, regional water treatment plants or distribution networks—collaboratively training a shared anomaly detection model under the orchestration of a central aggregation server (Figure 2). Honest participants operate on distinct, non-overlapping temporal shards of historical plant telemetry D_i, retaining proprietary operational data on site. Training progresses across synchronous communication rounds $t = 1, \ldots, T$: At each round, the coordinator broadcasts the global parameter vector θ_t, and each client computes local updates $\Delta_i = \theta_{t,i} - \theta_t$. Crucially, unlike conventional federated learning where client updates pass directly to server-side aggregation, our framework introduces a data-space admission gate on the local training batch prior to parameter combination.

We assume an adversary who compromises and coordinates m of the n participating facilities ($m = 3$ of $n = 10$ in our headline evaluations, representing a 30% adversarial fraction, and $m = 2$ of $n = 5$ in smaller topologies). In accordance with Kerckhoffs' principle, the adversary possesses complete white-box knowledge of the federated protocol, the global parameter state θ_t, its own local telemetry records, and the full set of physical invariants $\mathcal{I}$ and tolerances $\{\varepsilon_j\}$ enforced by the coordinator.

Rather than mounting an untargeted denial-of-service attack that disrupts global convergence and is readily detected via validation loss monitoring, the adversary executes targeted exposure poisoning. By injecting anomalous process states and labelling them as normal operating conditions, the adversary forces the shared reconstruction autoencoder to reconstruct these attack signatures with low error, blinding the deployed detector to subsequent real-world intrusions. We evaluate two sophistication levels: a *naive attacker*,

who injects raw corrupted or replayed attack telemetry directly into local batches, and a *physics-aware attacker*, who explicitly projects corrupted batches onto the invariant-feasible subspace to evade the admission check while attempting to preserve poisoning efficacy. The gate inspects a client's training data, not its training. Its guarantee therefore assumes that each client computes its update from, and only from, the batch it committed. A client that commits honest data but trains on poisoned data falls outside this guarantee. We discuss this binding in Section 6.5 and Appendix D.3.

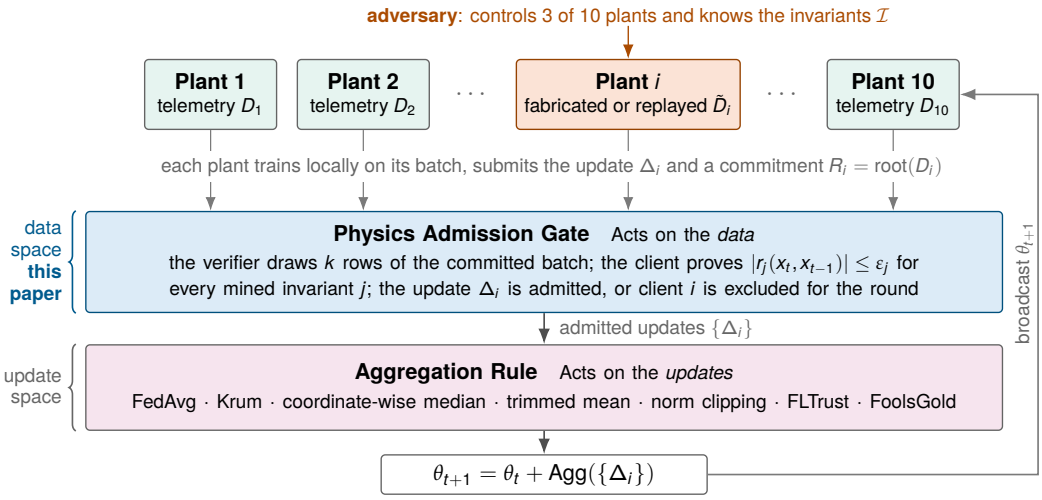


Figure 2. System model and threat architecture. Ten clients collaboratively train a shared intrusion detector under synchronous federated learning with three malicious participants. The admission gate acts directly on the committed training data batch D_i before the aggregation rule, providing an orthogonal defence prior to traditional update-space filtering.

3.2. Process Invariants and the Admission Gate

Water infrastructure obeys two physical relations that an attacker who fabricates telemetry must reproduce. The first is conservation of mass in each tank: for incompressible water and a tank of constant cross-section A_k, the level L_k changes with the net inflow:

$$A_k \frac{\mathrm{d}L_k}{\mathrm{d}t} = \sum_{\text{in}} Q - \sum_{\text{out}} Q,$$

so over a sampling interval Δt short enough for flows to be roughly constant (5 s here), ΔL_k is an affine function of the metered flows, with coefficients $\pm \Delta t / A_k$. The second is the operating logic of actuators: At a steady operating point, a running pump or open valve passes approximately its nominal flow, while a stopped pump passes none. These relations hold only where the plant meters them (at the sensor), and transients during switching can violate them briefly. Following established physics-based monitoring [27,30], we write them as two families of affine invariants (Figure 3), for which their parameters are mined from operational telemetry rather than taken from design drawings [28].

First, **actuator-to-flow couplings** relate the discrete operational state of an actuator $S_j \in \{0, 1\}$ (such as a raw water intake pump or motorised chemical valve) to the continuous downstream flow measurement F_j:

$$r_c(x_t) = F_j(t) - \mathbb{I}[S_j(t) = \text{on}] \cdot \bar{f}_j, \tag{1}$$

where $\bar{f}_j$ denotes the nominal steady-state volumetric flow rate, and $\mathbb{I}[\cdot]$ is the indicator function. Coupling constraints are evaluated exclusively when the actuator resides in a steady operating state; telemetry rows recorded during switching transitions are marked inapplicable and excluded from residual evaluation.

Second, **tank mass balances** relate differential liquid level changes across a storage or process tank L_k to the net volumetric flux across pipes instrumented with flow meters:

$$r_{\mathrm{b}}(x_t, x_{t-1}) = \Delta L_k(t) - \left(\sum_{j \in \mathcal{F}_k} a_{kj} F_j(t) + c_k \right), \tag{2}$$

where $\Delta L_k(t) = L_k(t) - L_k(t - \Delta t)$ is the level change over the sampling interval, $\mathcal{F}_k$ indexes the flow meters retained by the miner, coefficients a_{kj} approximate $\pm \Delta t / A_k$ for direct inflow and outflow, and offset c_k absorbs unmetered losses and sensor bias. The miner regresses ΔL_k on all plant flow meters under a sparsity penalty, so the fitted balances also include flows from other process stages. These are negligible; restricting the model to the tank's inflow and outflow lowers R^2 by at most 0.005, and for LIT401, for which its outflow is nearly constant, the inflow alone suffices.

Data: two recorded attacks from SWaT's labelled attack record; the rules and their tolerances $\pm\varepsilon$ (light blue band) are the ones mined from the attack-free recording.

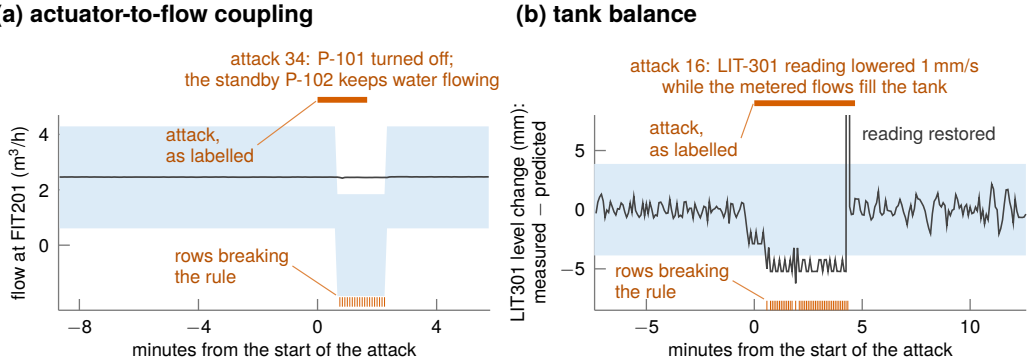


Figure 3. Process invariants detecting recorded SWaT attacks: (**a**) actuator-to-flow coupling during Attack 34 (P-101 off while flow persists) and (**b**) tank mass balance breach during Attack 16 (sensor spoofing while filling). Shaded blue bands denote tolerances $\pm\varepsilon$; orange ticks mark rows violating the invariant ($\leq$ 1% admission threshold).

3.2.1. Automated Invariant Mining and Calibration

Rather than relying on manual derivation from piping schematics, invariants are extracted automatically from an uncorrupted calibration record of normal plant operations [28]. Actuator couplings are identified by evaluating whether discrete actuator states partition continuous flow distributions into distinct, low-variance clusters, subject to minimum support and off-state ratio thresholds. Tank mass balances are identified by performing sparse linear regression of differential level changes against all metered flow rates, retaining candidate equations for which their coefficient of determination satisfies $R^2 \geq 0.60$.

Because industrial instrumentation exhibits natural measurement noise, hydraulic turbulence, and minor sensor drift, invariant residuals rarely evaluate to zero under normal operation conditions. For each mined invariant $j \in \mathcal{I}$, an operational tolerance ε_j is calibrated on clean holdout telemetry:

$$\varepsilon_j = 1.5 \times Q_{0.999}(|r_j|), \tag{3}$$

where $Q_{0.999}(\cdot)$ denotes the empirical 99.9th percentile of absolute residual magnitudes observed during honest operation.

3.2.2. Batch Admission Criterion

During federated training, a telemetry observation x_t violates invariant j if its absolute residual exceeds the calibrated tolerance: $|r_j(x_t)| > \varepsilon_j$. A row is classified as violating if it breaches at least one applicable invariant in $\mathcal{I}_{\text{app}}(x_t)$. A candidate training batch B submitted by a participant is admitted to the federated round if and only if the proportion of violating rows does not exceed the admission threshold $\alpha = 1\%$:

$$\frac{1}{|B|} \sum_{x \in B} \mathbb{I}\left(\bigvee_{j \in \mathcal{I}_{\text{app}}(x)} |r_j(x)| > \varepsilon_j \right) \leq \alpha. \tag{4}$$

The 1% accommodates minor transient disturbances while rejecting systematically corrupted telemetry. We evaluate two operational invariant configurations: a conservative *narrow set* (5 invariants on SWaT) and an expanded *wide set* (9 invariants capturing multi-stage dynamics). An attack pattern is defined as *covered* by an invariant set when its constituent telemetry produces a batch violation rate exceeding α, ensuring deterministic exclusion prior to parameter aggregation.

3.3. Attack Formulations and Threat Vectors

We classify adversary behaviours across two operational representation spaces: *data-space attacks* that corrupt local telemetry, D_i, and *update-space attacks* that alter parameter vectors Δ_i directly (Table 2).

Table 2. Classification of evaluated adversarial threat vectors across data and update spaces.

Attack Class/Variant	Locus	Mechanism/Preserved Property	Physical Relation Broken
Historical Replay	Data (D_i)	Spliced uncorrupted physical attack segments	Only where recorded attack deviates
Statistical Fabrication			
Channel Roll	Data (D_i)	Cyclic actuator shift (60 rows); preserves marginals	Actuator-to-flow couplings
Regime Permutation	Data (D_i)	Intra-regime actuator shuffle; preserves regime stats	Actuator-to-flow couplings
Conservation Scaling	Data (D_i)	Flow scaling ($\times 1.3$); leaves tank levels untouched	Tank mass balances
Regime Splicing †	Data (D_i)	Reordered honest segments (12 blocks); rows nominal	Segment boundary joins
Optimised Perturbation	Data (D_i)	Continuous channel shifts within $\pm 2.5\sigma$ (surrogate loss)	Both couplings and mass balances
Update Baselines	Update (Δ_i)	Sign-flip, gradient scaling, free-riding, min-max	None (data space untouched)

† Evaluated in the Separation Protocol (Section 4.6) as a boundary benchmark for row-level checking.

Data-space attacks coerce the autoencoder into learning attack patterns as nominal operations. As classified in Table 2, these attacks differ systematically in which physical invariant they breach. Within statistical fabrications, channel roll and regime permutation decouple actuator states from flow readings while leaving mass balances intact; conversely, conservation scaling alters flow-to-level rates and breaches mass balances while satisfying actuator couplings. Optimised perturbation violates both invariant families simultaneously, whereas historical replay breaches physical relations only during phases where the recorded attack itself contradicts instrumented physics. Regime splicing serves as a boundary case in our separation protocol, isolating single-row transition discontinuities. Conversely, update-space baselines train on uncorrupted data and modify only Δ_i, establishing the reference benchmark where defence relies exclusively on server-side aggregation heuristics.

Adaptive Optimisation (Physics-Aware Adversary)

To assess the gate against an adaptive adversary [33,34], we model a physics-aware attacker who possesses the complete invariant set $\mathcal{I}$ and tolerances $\{\varepsilon_j\}$. Given a target poisoned batch $\tilde{X}$, the adversary seeks the minimum-norm correction that renders the batch admissible:

$$\min_{X} \|X - \tilde{X}\|_{\mathrm{F}}^2 \quad \text{subject to} \quad |r_j(x_t)| \leq 0.95\,\varepsilon_j \quad \forall t, j \in \mathcal{I}_{\mathrm{app}}(x_t). \tag{5}$$

Because discrete actuator states cannot be continuously modified without immediately revealing fabrication, the adversary holds discrete actuator channels fixed and perturbs only continuous sensor readings. Rather than solving a global quadratic programme to optimality, the adversary executes an iterative active-set correction. In each pass (up to 30), it identifies active constraint violations and computes a minimum-norm step via conjugate gradients to shift continuous readings within a $0.95\,\varepsilon_j$ margin. The loop terminates as soon as the batch violation rate falls below 0.5%, providing a twofold safety buffer beneath the 1% admission threshold. Batches that cannot be repaired within the 30-pass budget are submitted unamended and face gate rejection.

3.4. Zero-Knowledge Verification Architecture (PA-FL Lite)

If clients sent raw training batches to the coordinator for inspection, they would breach data confidentiality. PA-FL Lite instead lets a client prove, without revealing telemetry, that a batch it has committed to satisfies the invariants, using Groth16 zk-SNARKs [45] and the Poseidon hash [46] (protocol in Appendix D). The client commits to its batch ($N = 1024$ rows) as a Poseidon Merkle root; an unpredictable challenge from the verifier selects $k = 32$ rows; the proof shows that these rows open to the root and that at most $v_{\max}$ of them violate an invariant. A client whose proof fails is excluded from the round. Each round the client uploads only the 32-byte root and an 806-byte proof, about 0.2% of a model update, and no telemetry. At $k = 32$, the circuit has 297,736 constraints and proves in 12.0 s on commodity hardware. PA-FL Lite is a feasibility demonstration: it checks sampled rows of one committed batch and does not bind the update to that batch (Section 6.5, Appendix D.3).

4. Experimental Setup

We outline operational testbeds, federation partitioning methodology, training parameters, and evaluation metrics. The experiment evaluates 3169 runs shown in Table 3.

Table 3. Experimental grid across industrial benchmark records, detailing participant counts, aggregation rules, attack modes, federations (modes), seeds, invariant sets, and total trained models (cells).

Record	Clients (Malicious)	Rules	Attacks	Modes	Seeds	Invariant Sets	Cells
SWaT	10 (3)	7	8	5	20	narrow (5), wide (9)	3007
WADI	10 (3)	3	2	5	3	default (7)	90
BATADAL	5 (2)	3	2	4	3	mined (10)	72

4.1. Benchmark Records and Preprocessing

Having established the physical testbed architectures in Section 2.4, we detail the concrete telemetry records, sampling strides, and preprocessing pipelines: (1) SWaT:We evaluate the December 2015 release (seven clean days; four days containing 36 cyber-physical attacks), downsampled to a 5-second stride with the initial 6-hour start-up transient removed. Chemical analyser (AIT) channels are excluded from detector training due to severe

non-stationary electrochemical drift (18σ baseline shifts across records; Appendix A.1), and discrete actuators are normalised to unit range. (2) WADI: We evaluate the October 2017 release (14 clean days; 2 attack days under 15 intrusion scenarios) downsampled to a 5-second stride, with four uninformative zero-variance alarm channels dropped and transient communication dropouts filled via linear interpolation. (3) BATADAL: We utilise the full 365-day hourly SCADA recording across 14 cyber-physical attack scenarios under realistic diurnal demand patterns. (4) HAI: We evaluate release version 21.03 as an explicit negative control, testing non-hydraulic thermal loops where linear mass balances do not apply (Section 5). Grouping contiguous labelled attack observations at the operational stride yields 35 distinct attack segments on SWaT (including a sustained 10-hour filtration disruption) and 14 on WADI.

4.2. Federation Partitioning and Threat Synthesis

Following strict temporal separation to eliminate data leakage [35], a 10-client federation is constructed from the normal operational record of one plant, modelling a utility that operates several sites of the same design (Figure 4). The chronological sequence is partitioned into an invariant discovery slice (30%: half for fitting, half for tolerance calibration), a validation slice (15%) for anomaly threshold selection, a server root slice (5%) for FLTrust reference updates, and client training telemetry (50%). The client partition is divided into ten disjoint temporal shards (4734 rows each on SWaT at 5 s stride). The entire attack record is held out as the global evaluation set.

In compromised federations ($m = 3$ of 10 clients on SWaT and WADI; 2 of 5 on BATADAL), malicious clients splice attack segments across 25% of shard rows (10% on WADI), coordinating against a single target attack set per seed. Attackers either inject statistical fabrications, optimised perturbations, invariant-projected data, or replayed historical attacks, oversampling attack windows to 50% of local batches to ensure gradient impact.

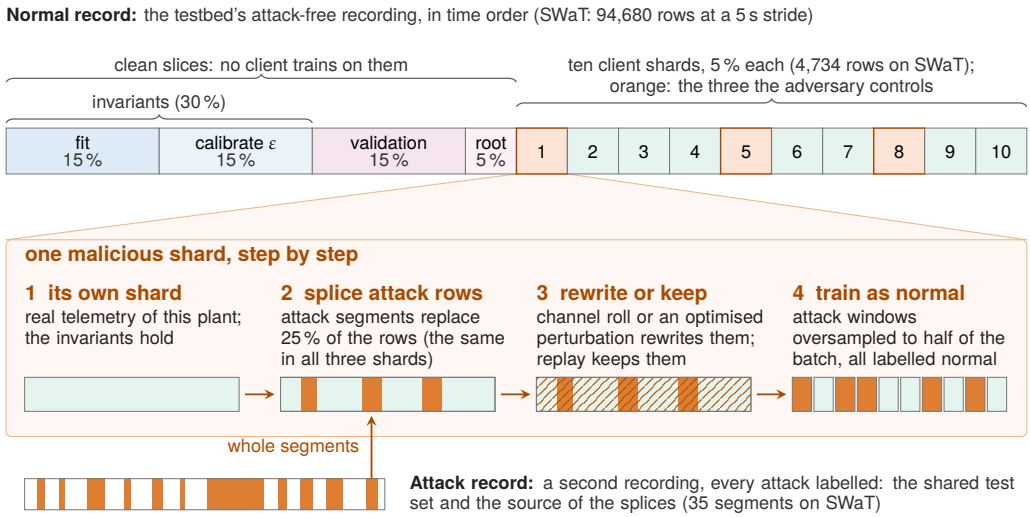


Figure 4. Federation partitioning and malicious shard synthesis from a single plant record. Chronological normal telemetry is segmented into invariant mining (30%), validation (15%), server root (5%), and ten disjoint client training shards (50%). Compromised clients ($m = 3$) splice attack segments across 25% of rows and oversample attack windows to 50% of the batch.

4.3. Local Detector Architecture and Training Setup

Following standard ICS anomaly detection benchmarks [7,8], each facility trains an unsupervised deep reconstruction autoencoder (we use a standard architecture without tuning and evaluate the gate by admission and the detector by the damage that admitted poison causes) over sliding temporal windows of normalised telemetry ($W = 10$). The

network compresses each flattened window through a 128-unit dense layer with ReLU activations into a 24-dimensional latent bottleneck, which a symmetric decoder reconstructs under mean squared error. Observations for which their reconstruction error exceeds the 99.5th percentile of clean validation loss (τ_{det}) trigger intrusion alarms.

Federated training executes for 25 rounds (2 local epochs per round, Adam with $\eta = 10^{-3}$, mini-batches of $B = 128$ windows). On commodity hardware, a full 25-round run completes in 17 seconds, enabling exhaustive evaluation across the 3169-run experimental grid (Table 3). Each seeded run redraws the malicious clients, the target attack set, the fabrication, and the weight initialisation keeping the temporal shards fixed. Every removal ratio is measured on SWaT across 20 seeds and WADI and BATADAL across three. (WADI and BATADAL use three seeds each: On WADI the naive attack never costs a point of targeted recall, and on BATADAL the gate excludes no attacker, so more seeds cannot add removal evidence. Both records serve separation and admission.)

4.4. Baseline Aggregation Rules

Because the admission gate operates in data space prior to local gradient computation, it is designed to complement server-side aggregation. To test orthogonality across the defensive paradigms introduced in Section 2.1, we benchmark the gate alongside seven representative aggregation rules: (1) unfiltered baseline: FedAvg; (2–3) coordinate-wise estimators: median and trimmed mean (trimming fraction $\beta = 20\%$); (4–5) geometric filters: norm clipping (threshold τ) and Krum (configured with Byzantine allowance $f = m = 3$); and (6–7) directional and historical scoring: FLTrust (evaluated against the 5% server root partition, see Section 4.2) and FoolsGold (tracked across all 25 rounds).

4.5. Federated Execution Modes and Evaluation Metrics

To isolate defensive mechanics, experiments evaluate five federated operational modes: (1) **clean**, where all $n = 10$ participants train honestly; (2) **honest-only**, where the $n - m = 7$ honest clients train in isolation, establishing the reference baseline that isolates poisoning damage from changes in collective participant capacity; (3) **naive attack**, where 7 honest and 3 malicious clients aggregate without data-space admission checks; (4) **physics-aware attack**, where projected malicious updates are aggregated unconditionally; and (5) **gated**, the deployed system where batches breaching Equation (4) are excluded prior to aggregation ($w_i = 0$).

The threat model is strictly targeted: the adversary trains the shared autoencoder to reconstruct specific attack patterns as nominal operations, leaving benign and untargeted operations unperturbed. Because normal operations constitute 88% of the test record (attacks $< 12\%$), aggregate metrics averaged throughout the test record barely register severe disruptions confined to brief attack segments—a well-documented pitfall in security evaluation [35,40]. Blinding a detector to three targeted attacks alters global F1 by < 0.01 points [13], and under FedAvg, the four data-space attacks lower targeted recall by 6.0 to 8.6 points (11.7 points in our 20-seed replay experiment) while global F1 does not fall at all (Table A2). A defender relying on F1 would judge the deployed model uncompromised.

We therefore define *targeted-attack recall* (R_{targ}) as the primary metric, measuring detection within the coordinated target attack set $\mathcal{A}_{\text{targ}}$:

$$R_{\text{targ}} = \frac{\sum_{t \in \mathcal{A}_{\text{targ}}} y_t \cdot \hat{y}_t}{\sum_{t \in \mathcal{A}_{\text{targ}}} y_t}, \tag{6}$$

where $y_t \in \{0, 1\}$ is the ground-truth label and $\hat{y}_t \in \{0, 1\}$ indicates whether reconstruction error exceeds threshold τ_{det}.

The *damage-weighted removal ratio* quantifies the proportion of naive poisoning damage removed by the admission gate relative to the honest-only reference baseline:

$$\text{Removal} = 1 - \frac{R_{\text{ref}} - R_{\text{gated}}}{R_{\text{ref}} - R_{\text{naive}}}, \tag{7}$$

where R_{ref}, R_{naive}, and R_{gated} denote targeted recall under the honest reference baseline, naive attack, and gated admission, respectively. Removal is evaluated over informative seeds where naive attack damage is measurable ($R_{\text{ref}} - R_{\text{naive}} \geq 1.0\%$). Evaluation metrics thus decouple into two distinct families: gate-level metrics evaluate data admission (honest shards rejected and malicious batches excluded; while detector-level metrics measure the operational damage that admitted poison causes to the federated detector (targeted and untargeted recall, global F1, and AUC-PR; Appendix Table A2).

4.6. Verification Protocols and Reproducibility

Invariant performance is verified prior to federated training via two protocols (detailed in Appendix A): (1) a *separation protocol* evaluating 200 random 1000-row batches per record across clean and fabricated telemetry to verify zero false rejections; and (2) a *coverage audit* scoring all 35 SWaT and 14 WADI attacks individually against narrow and wide invariant sets.

The federated simulation, invariant miner, and analysis scripts are implemented in Python 3.13 with PyTorch 2.14 and the circuit in Circom 2.1.9 with SnarkJS 0.7.4; all runs use an Apple M3 Pro (36 GB RAM, CPU only), with hyperparameters fixed a priori and no search. For more information, refer to the Data Availability Statement.

5. Results

5.1. RQ1: Separating Honest from Fabricated Telemetry

We first ask whether process invariants mined from normal operation can separate honest operational telemetry from fabricated batches without rejecting honest participants. Following our Separation Protocol (Section 4.6), we evaluate 200 randomly placed batches of $N = 1000$ consecutive observations per benchmark under the 1% admission threshold $\alpha = 0.01$ (Figure 5).

Across the three benchmarks (SWaT, WADI, and BATADAL), honest data is not rejected (FRR $= 0.000$). Across 25 distinct honest client shards (10 SWaT, 10 WADI, 5 BATADAL), the highest observed single-shard row violation rate is 0.51% on SWaT (under wide invariants), which sits safely below the 1% admission gate. In contrast, when data statistics are fabricated, they violate physical constraints: (1) *channel roll* desynchronises control commands from hydraulic responses, violating invariants in 28.1% to 96.8% of batch rows; (2) *within-regime permutation* reassigns actuator states, producing violation rates of 22.2% to 75.6%; and (3) *conservation scaling* disrupts mass and actuator-flow balances, generating violation rates between 26.9% to 99.0%. These are batch means; per batch, WADI, which has no mass balances, is the exception: its check admits 3% of channel-roll and 23.5% of conservation-scaling batches, against none on SWaT and BATADAL.

Two boundaries emerge from Figure 5. First, *regime splicing* produces an empirical violation rate of only 0.12% to 0.36%. Because spliced sequences consist of genuine historical telemetry segments, physical laws are satisfied internally throughout each contiguous segment, triggering violations exclusively across splice boundaries. Spliced fabrications thus pass row-level percentage checks. Second, *HAI*, a boiler and steam-turbine testbed, has almost no actuator switching and no tank-style mass balance in its training record, so the miner finds no couplings and only three weak balances. HAI marks our work's bound-

ary and confirms that the framework applies strictly to hydraulic and mass-conserving physical processes.

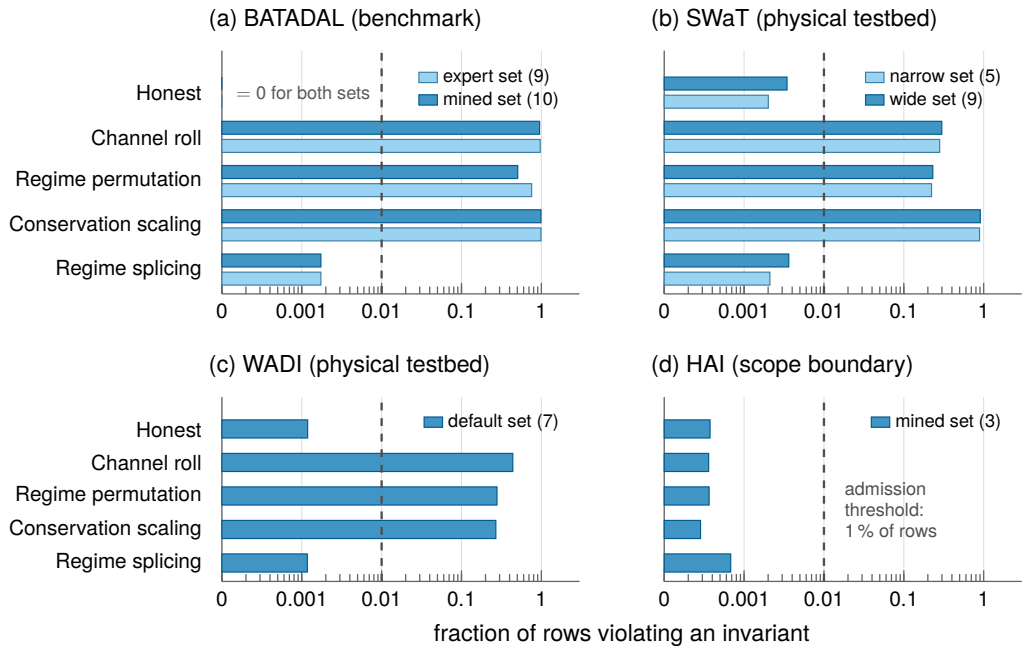


Figure 5. Separation of honest from fabricated telemetry by process invariants. Mean fraction of rows violating at least one invariant over 200 batches of 1000 rows (log scale; zero plotted at 10^{-4}). The vertical dashed line marks the 1% admission threshold α. (**a**) BATADAL under expert (9) and mined (10) invariant sets. (**b**) SWaT under narrow (5) and wide (9) mined invariant sets. (**c**) WADI under the default mined set (7 couplings). (**d**) HAI is a scope boundary since the miner finds no couplings and only three weak balances, so every batch is admitted.

In sum, yes, process invariants mined from normal operation can separate honest from fabricated telemetry within the coverage of the invariants.

5.2. RQ2: Evasion of Robust Aggregation and Data-Space Admission

We next ask whether updates trained on fabricated or replayed telemetry evade update-space robust aggregation and whether data-space admission closes that blind spot. The damage is measured based on targeted-attack recall as in Section 4.5. Figure 6 reports the targeted recall loss and client acceptance rates across seven aggregation rules under naive data-space attacks (left block) and update-space attacks (right block) in the absence and presence of the admission gate.

Across all four data-space attacks (replay, channel roll, permutation, and optimised perturbation), update-space aggregation rules fail to prevent model poisoning.

Coordinate estimators and geometric filters:FedAvg, coordinate-wise median, and norm clipping admit 100% of the poisoned updates and lose 6.0 to 8.6, 4.2 to 6.2, and 2.7 to 6.5 points, respectively, across the data-space attacks. Norm clipping rescales the update but keeps its direction, so it behaves like FedAvg. Trimmed mean discards the extreme 20% of each coordinate and admits only 34% of the malicious updates yet still loses 5.2 to 5.3 points under replay, channel roll and regime permutation (1.4 under optimised perturbation). Krum admits none of them and still shows a uniform 9.2-point drop on every attack: that drop is selection variance, since Krum forwards a single client's update and not poisoning.

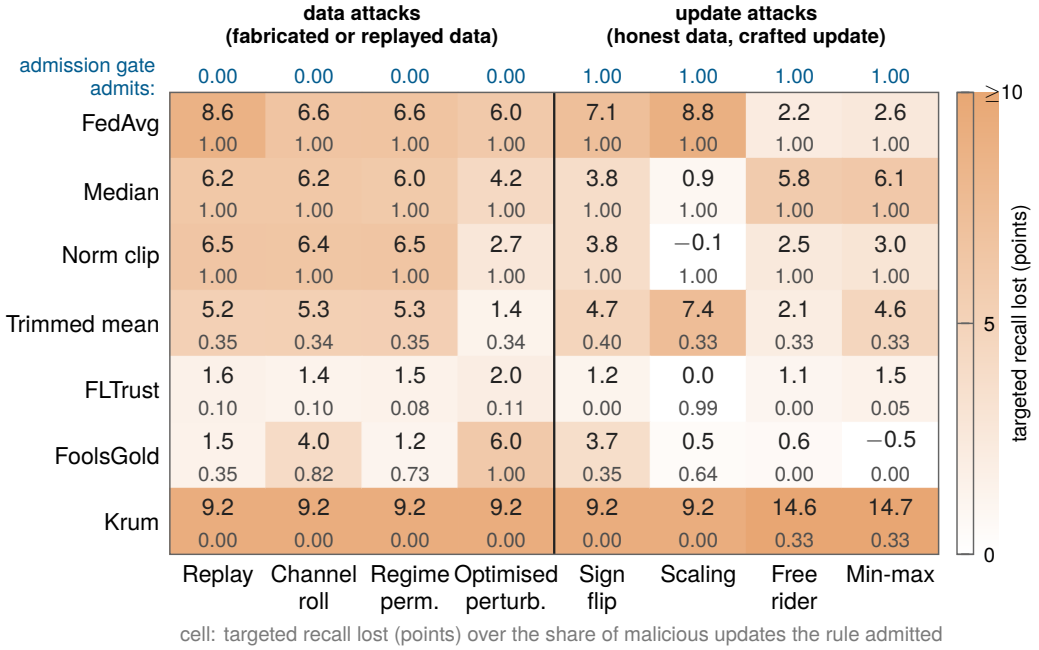


Figure 6. Blind spots of the two defences on SWaT (wide invariant set, no gate 3 seeds). Rows are the aggregation rules; columns are the four data-space attacks on the left and the four update-space attacks on the right. Each cell gives the targeted-attack recall lost in points (upper number and colour) and, below it, the share of malicious updates the rule admitted. The top row is the share of malicious batches the admission gate admits: 0.00 for every data attack, for which its batches break the invariants, and 1.00 for every update attack, for which its clients train on honest data and tamper only with the update. Each defence is blind to one half of the grid.

Directional and Similarity-Based Defences: FLTrust admits 8% to 11% of the poisoned updates by comparing them with a clean server root and holds the loss to 1.4 to 2.0 points. FoolsGold, which penalises updates that stay collinear across rounds, admits 100% of the optimised perturbation updates and loses 6.0 points on them, and admits 82% under channel roll (4.0 points).

Structural Division: Figure 6 illustrates a structural division between data-space admission and update-space aggregation rules. When malicious participants poison training data (the four data-space attacks on the left), update-space defences are ineffective, while the admission gate rejects 100% of these batches ($w_i = 0$), preventing corrupted gradients from entering aggregation. Conversely, when clients train on honest telemetry and manipulate only the parameter update (the four update-space attacks on the right: sign flip, scaling, free rider, min–max), the admission check admits every batch ($w_i = 1.0$), and the aggregation rule is the sole defence. Here, performance is uneven: FLTrust holds every update attack to ≤ 1.5 points and FoolsGold resists all but sign flip, whereas FedAvg loses 7.1 and 8.8 points to sign flip and scaling, median and trimmed mean lose 4 to 7 points, and Krum is degraded heavily by free rider and min-max (14.6 and 14.7 points). The choice of aggregation rules thus governs update-space resilience, while data-space admission neutralises data poisoning.

In summary, we answer RQ2: Do updates trained on fabricated telemetry evade update-space robust aggregation? Four of the seven aggregation rules admit 73–100% of the updates trained on poisoned telemetry, and FoolsGold admits the optimised perturbation in full. The gate rejects every data-space attack and admits every update-space attack, so the two defences cover complementary halves of the threat and are needed together.

5.3. RQ3: Damage Removal Under an Adaptive Adversary

Next, we investigate how much targeted damage admission removes when an adaptive adversary knows the invariants and projects its poisoned data onto them, and how this depends on invariant coverage.

Narrow vs. wide attack coverage. Because an admission gate operates by evaluating algebraic invariant residuals, it can exclude only those poisoning attacks for which their underlying telemetry contradicts instrumented physical relations. We quantify this relationship across the 35 labelled cyber-physical attacks on SWaT and 14 on WADI (Figure 7 and Table 4).

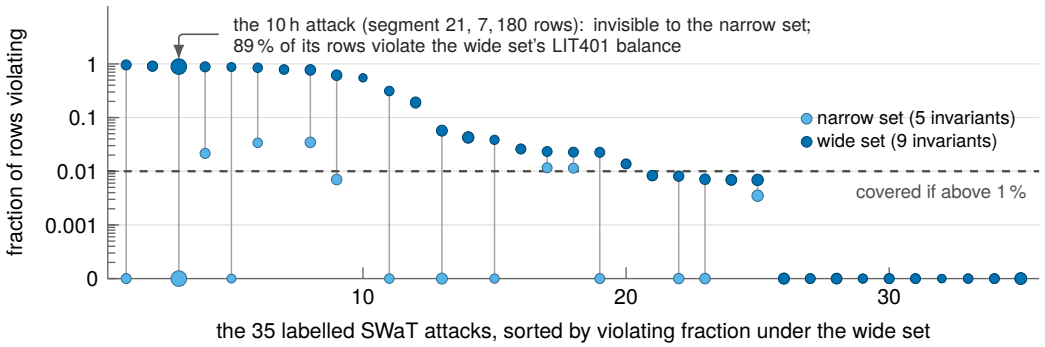


Figure 7. SWaT attacks identified by invariant set. The fraction of rows violating at least one invariant for each of the 35 labelled SWaT attacks under narrow (5 invariants) and wide (9 invariants) sets, sorted by the wide-set value (log scale; zero plotted at 10^{-4}). Stems join each attack's narrow and wide values; larger markers denote longer attacks; the dashed line is the 1% admission threshold: an attack to its right is covered.

Table 4. Coverage of labelled attacks across invariant sets on SWaT and WADI. An attack segment is covered when its data violation rate strictly exceeds the 1% threshold. Honest row violations are audited over the first 60,000 observations following the discovery slice.

Record, Setting	Rules	Couplings/Balances	Honest Rows Violating	Attacks Covered	Attack Rows Covered	Share
SWaT, narrow	5	3/2	0.21%	12/35	1137/10,931	10%
SWaT, wide	9	6/3	0.36%	20/35	8991/10,931	82%
WADI, default	7	7/0	0.14%	3/14	395/1996	20%

As shown in Figure 7 and Table 4, invariant coverage functions as an operational dial. Under conservative miner defaults ($R^2 \geq 0.60$, support 0.020, coupling off-ratio 0.05), the narrow set keeps only actuator couplings and single-tank balances and cannot see attacks that move water between stages, such as the 10-hour ultrafiltration backwash drain (*Segment 21*), which violates none of its invariants and passes the check unseen. Relaxing the thresholds ($R^2 \geq 0.40$, support 0.005, coupling off-ratio 0.10), a setting we introduced after the narrow set missed this attack, incorporates multi-stage mass balances P1 through P4: The same attack now violates the LIT401 balance in 89% of its rows while increasing the honest violation rate to only 0.36%, safely below the 1% threshold. On the WADI distribution network, the miner discovers seven actuator couplings but zero reliable mass balances due to unmetered consumer demand loops, covering 3 of 14 attacks (19.8% of rows) and demonstrating that physical coverage is fundamentally bounded by metering capability.

Poison Damage Removal Across Aggregation Rules. Having established that robust aggregation cannot prevent data-space poisoning, we examine whether process physics can

neutralise this threat across all seven aggregation rules (Figure 8 and Appendix Table A1). We evaluate two distinct defensive barriers relative to the unpoisoned reference: first, whether forcing an adaptive adversary to project poisoned telemetry onto physical invariants diminishes attack potency, and second, whether filtering non-compliant batches at the admission gate restores detection performance.

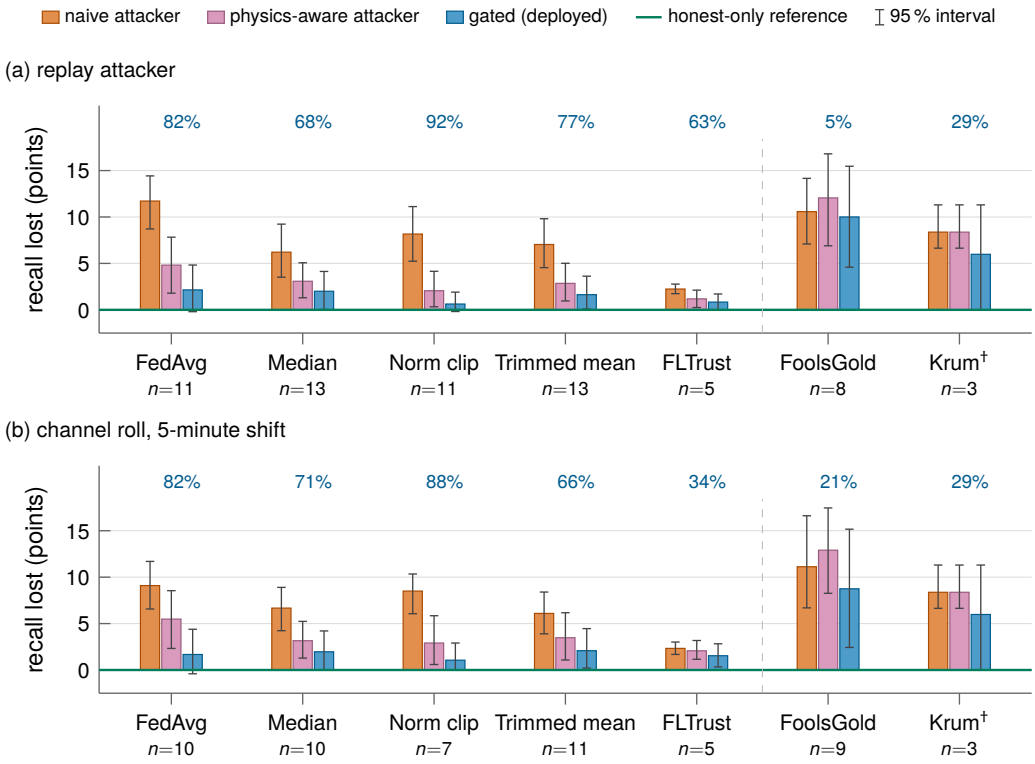


Figure 8. Poison damage removal across seven aggregation rules on SWaT under the wide invariant set (10 clients, 3 malicious, 20 seeds). Targeted-attack recall lost is evaluated across four federated execution modes: *honest-only* (unpoisoned reference at 0), *naive* (unfiltered poisoned training), *physics-aware* (adversary projects telemetry onto invariants), and *gated* (deployed server rejects non-compliant batches). Subplots show (**a**) historical attack replay and (**b**) channel roll (5 min shift). Bars are means over the n informative seeds (naive damage ≥ 1 point) with 95% bootstrap intervals; percentages give the share of naive damage removed by the gate. † Krum admits no malicious update, so its damage reflects which honest update it selects.

Forcing an adaptive adversary to satisfy the physical invariants already costs it a substantial share of its poisoning damage. Projection alone removes 50% to 75% of replay damage under FedAvg, median, trimmed mean, and norm clipping yet leaves 63% of projected batches admissible to the gate (Table 5). The threshold $\alpha = 1\%$ is the tightest setting tested in the sweep that rejects no honest shard (Appendix C). Because discrete actuator states remain fixed (e.g., on/off), satisfying the invariants forces sensor readings into unnatural configurations (in one projected batch, flow meter FIT401 reads -62 m^3/h). This distortion weakens the poison but does not prevent the model from learning the attack.

The admission gate removes most of the remaining damage. Under FedAvg, it removes 82% of naive replay damage (95% CI 61–102%) from an 11.7-point drop in targeted recall and 82% of channel-roll damage (57–105%). Median, trimmed mean and norm clipping retain 66% to 92% of the benefit across both attackers, each significant at $p \leq 0.016$ (exact one-sided Wilcoxon test; Table A1). Pooled over the five rules with the seed as the unit, the gate removes 78% of the damage (95% CI 57–97%, $p < 0.001$), significantly more than projection alone ($p = 0.008$). The variance of removal across seeds is structural, following the number of attackers the gate excludes. Removal is 19–40% in seeds where no attacker

is excluded, 75–86% where one is, and 100% where all three are. FLTrust, which already admits few malicious updates, gains less: 63% of replay damage and 34% of channel-roll damage, the latter not significant. FoolsGold shows no reliable removal (intervals span zero), and Krum admits no malicious update, so its residual damage reflects which honest update it selects rather than the poison.

Table 5. Batch admission and client exclusion rates across industrial benchmark records before and after adaptive projection under the 1% gate. *Malicious excluded* is the mean per run. Honest shards are counted once each.

	Malicious Batches Admitted				Honest Shards	
Record (Rules)	**Fabricated**	**Replay, Before**	**Replay, After Proj.**	**Malicious Excluded**	**Rejected**	**Worst Violating**
BATADAL (10)	0/6	0/6	6/6	0.0 of 2	0/5	0.00%
SWaT narrow (5)	0/60	24/60	51/60	0.5 of 3	0/10	0.32%
SWaT wide (9)	0/60	3/60	38/60	1.1 of 3	0/10	0.51%
WADI (7)	0/9	1/9	3/9	2.0 of 3	0/10	0.13%

Effect of Invariant Coverage on Damage Removal. To confirm that damage removal is governed by physical invariant coverage rather than extraneous artefacts, Figure 9 contrasts gate removal under the narrow set (12/35 attacks, 10% rows) against the wide set (20/35 attacks, 82% rows) across identical seeds. Under the narrow set, the admission gate admits 85% of projected batches, excluding an average of only 0.5 of three malicious clients per run (attackers are excluded in 3 of 20 seeds); its removal of 53% to 70% of replay damage is almost entirely the effect of projection, and the gate adds no significant removal beyond it (pooled 48% to 58%, $p = 0.50$). Under the wide set, the additional multi-stage mass balances reduce batch admission to 63%, excluding an average of 1.1 of three malicious clients per run, and the gate raises removal from 55% under projection to 78% (pooled, $p = 0.008$; 66% to 92% per rule across both attackers). Wider invariant coverage leaves more projected batches unable to comply, and it is these exclusions that let the gate remove damage beyond what projection alone achieves.

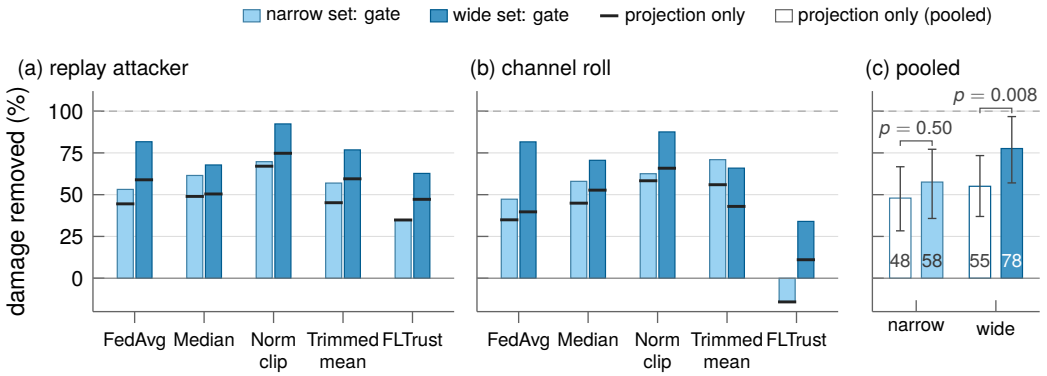


Figure 9. (**a**,**b**) The percentage of targeted damage removed by the deployed gate under narrow versus wide invariant sets on SWaT. The solid line marks removal by projection alone, so the gap above is the gate's contribution. (**c**) All five rules combined: removal by projection alone (open bars) and by the gate (filled bars), with 95% bootstrap intervals.

In summary, projection alone already costs the adversary about half its damage (55% pooled). Under the wide invariant set, the gate removes most of the rest by excluding the batches that projection cannot bring under α: 78% pooled (57–97%), significantly more than projection ($p = 0.008$) and 66–92% for FedAvg, median, trimmed mean, and norm clipping.

Under the narrow set, it adds no significant removal ($p = 0.50$). Invariant coverage decides how much the gate adds.

5.4. RQ4: Zero-Knowledge Verification Feasibility

Finally, we evaluate whether a participant can prove batch compliance with process invariants in zero-knowledge at a cost compatible with federated rounds, without disclosing proprietary operational telemetry.

Cryptographic Benchmarks and Circuit Overhead. We evaluate the PA-FL Lite protocol instantiated with Groth16 zk-SNARKs over the BN128 elliptic curve on an Apple M3 Pro workstation (Table 6). PA-FL Lite achieves practical operational efficiency at the $k = 32$ operating point (with complete $k = 16$ benchmarks detailed in Appendix Table A3). The Circom circuit compiles to 297,736 R1CS constraints (9304 constraints per sample, dominated by chunked Poseidon hashing and Merkle opening paths, compared to 148,888 constraints at $k = 16$). On the local plant client, generating the attestation proof π_i requires 12.0 s and 3.8 GB peak RAM (6.55 s and 2.4 GB at $k = 16$).

Table 6. PA-FL Lite at the operating point $k = 32$ on wide SWaT with $N = 1024$ rows per batch (Groth16 over BN128, Apple M3 Pro): the cost of one attestation and the sampling guarantee it buys. The full benchmark, including $k = 16$, is shown in Appendix Table A3.

PA-FL Lite at $k = 32$ Sampled Rows	
Circuit size (R1CS constraints)	297,736
Proof generation on the plant (peak RAM)	12.0 s (3.8 GB)
Proof verification at the server	290 ms
Proof size π_i	806 B
Honest batch accepted, $P(v \leq 1)$, 19,992 draws	98.5%
Replayed attack batch detected, $P(v > 1)$	99.8%
Channel-roll batch detected, $P(v > 1)$	99.8%

On the coordinator side, proof verification executes in just 290 ms, while the submitted Groth16 proof π_i consumes only 806 bytes of communication bandwidth. Because proof verification requires only three elliptic curve pairings regardless of circuit size, the verifier workload remains constant and shows good promise to scale to large federations.

Sampling soundness and detection guarantees. To verify that sampling $k = 32$ rows ensures statistical security against poisoning, we evaluate attestation across 19,992 Monte Carlo draws ($v_{\max} = 1$, $u_{\max} = 8$; Table A3). Honest batches pass with 98.51% probability ($P(v \leq 1)$), rarely triggering false rejections. Conversely, unconstrained attacks—such as channel roll (21.29% violations) and historical replay (23.34%)—are rejected with 99.8% certainty ($P(v > 1) = 0.998$), triggering Circom witness-generation aborts that prevent valid proof construction. While adaptive projection onto the invariant subspace lowers violations to 0.68% to pass verification, this projection alone removes only part of the attack's poisoning damage. (Section 5.3).

In summary, we answer RQ4: A participant can verify batch compliance in zero knowledge with practical overhead. By pairing Poseidon Merkle commitments with Groth16 row-sampling proofs ($k = 32$ of 1024 rows), PA-FL Lite generates an 806-byte proof in 12.0 s that verifies in 290 ms. Honest batches pass with 98.5% probability while unprojected attacks are rejected with 99.8% certainty. Binding the model update to the committed batch remains future work.

6. Discussion

6.1. Can Process Invariants Reliably Separate Honest from Fabricated Telemetry?

In short, yes, honest batches pass the admission check and fabricated ones fail it. This *physical separation* of batches succeeds because the fabrications preserve individual channel plausibility while breaking the algebraic relationships between coupled channels (Section 5.1). For instance, channel roll preserves every marginal distribution identically, and conservation scaling maintains each variable strictly within its nominal operational range; standard per-channel outlier filters or univariate distributional tests would admit both of these. Invariant admission rejects them because coupled states no longer agree—a running pump exhibits zero flow, or a fluctuating tank level contradicts the measured net flux. Conversely, honest telemetry, recorded under natural operational sensor noise, produces an empirical violation rate that peaks at 0.51% (roughly half the 1% gate), demonstrating that α must be calibrated against clean plant baselines: tightening α to 0.5% triggers false rejections on honest client shards (Appendix C).

This also establishes the benefit of data-space admission: internal physical consistency. Spliced sequences composed of authentic operational telemetry satisfy physical laws internally and pass the row-level check by construction, as do historical attack segments for which their manipulated variables fall outside the instrumented invariant set (Section 5.3). Detecting such attacks requires cryptographic provenance rather than physical constraints. Furthermore, physical separation presupposes an instrumented, coupled industrial process. Under partial sensor coverage, some uncoordinated fabrications pass (such as conservation scaling on WADI, which lacks metered mass balances); in the absence of coupled actuator–hydraulic dynamics (HAI), the gate admits all batches and defence is in the hands of update-space aggregation. Crucially, facility operators can audit this boundary prior to deployment directly from the support and fit of invariants mined on their clean historical records.

6.2. Why Are Update-Space Defences Blind to Poisoned Data?

Intrusion detection for FL typically begins by checking the model updates produced by each individual plant for statistical anomalies. Updates that pass are admitted to the subsequent aggregation round. The primary issue here is that the update could have been produced by training on dishonest data; the update itself cannot discern either way. In other words, poisoning that enters through the training data cannot be picked up by rules that evaluate the update. In the next round, the aggregation server will evaluate the update for potential anomalies and have none to find (Section 5.2).

It is well known that Byzantine-robust aggregation rules fail against clever adversaries [23–25], and our results deepen the failure. An adversary manipulating the data space bypasses most robust aggregation rules without needing to model or optimise against the server's defence. FLTrust is the exception, at the cost of a clean root dataset held by the server. The invariants play that role for the admission gate as a compact physical reference rather than a dataset.

We have two main defences and argue that both are required because they operate on distinct objects: The admission gate evaluates the training batch before local computation, while the aggregation rule evaluates the parameter vectors afterward. Each defence is blind to the threat space governed by the other (Figure 6). In this sense, the admission gate acts as a pre-admission filter, similar to how norm bounding is framed in secure aggregation [22]. On the aggregation side, the federated learning literature typically defers poisoning to robust aggregation [12,59]. Our findings draw a line here: robust aggregation guards update-space manipulations but provides no barrier against data-space corruption.

6.3. How Much Protection Does Process Physics Actually Provide?

Process invariants began as hand-derived physical balances for testbed security to sound real-time alarms [27] and have since been refined through automated mining [28,29]. The invariant check can be classified as rule-based by design as affine relations are deterministic, interpretable, and provable in zero knowledge. Admission is also an easier job than runtime detection: Run offline once per round, it averages out transient row-level noise, a violation excludes an update (rather than shutting down the plant), and attackers must still fake coupled multivariate physics across actuators and flows [29]. Runtime monitoring and the gate are complementary: one protects the plant, the other the training.

Projection of the invariants removes about half of the damage (55% pooled), while batch exclusion adds the rest on the wide set ($p = 0.008$; Section 5.3). As a result, the coverage ceiling that has traditionally constrained invariant-based detectors can be used as a tunable design parameter (Section 5.3, Figure 9). Operators can therefore audit and set the coverage boundary before training begins. When a batch is rejected, the cause is interpretable: the failed invariant is named, for example, a pump reporting on while its flow meter reads zero. By the same token, protection extends only as far as the plant's metering.

So, how much protection does process physics actually provide? As much as plant instrumentation permits: the gate removes 66%–92% of targeted poisoning damage across the four main aggregation rules on coupled, metered balances but offers no defence over unmonitored dynamics.

6.4. Can We Verify the Data Without Seeing It?

A fundamental challenge in designing a data-space admission gate is establishing trust: If a compromised client trains on poisoned telemetry, what mechanisms prevent it from falsely claiming that its data has passed the validation process? The intuitive workaround, uploading telemetry to the coordinator so that the server can verify the invariants directly, is a privacy leak. Centralising operational SCADA logs violates participant privacy and, at least in the EU, violates critical-infrastructure cybersecurity directives and data-governance mandates [11]. On the other hand, relying on an industrial client's honest updates is an open invitation to poisoning. Zero-knowledge attestation bridges the gap (Section 5.4): by evaluating process invariants inside an arithmetic circuit, a client can mathematically prove that its training batch respects the plant's governing physics without disclosing a sensor reading.

Other data-side defences do not remove this need, for example, a trusted execution environment attests a computation rather than the validity of its input data [60], and differential-privacy clipping acts on updates, which is norm clipping in its weak form, and among the rules we evaluate.

Early secure aggregation protocols explicitly left the validation of well-formed client inputs as an unresolved challenge [41]. Subsequent cryptographic input validation addressed this for update vectors by proving Euclidean norm bounds [20–22], while verifiable FL proved the integrity of server aggregation [43,44]. PA-FL Lite advances this work by moving the proved statement directly to the local training data, establishing batch telemetry as a verifiable object in federated systems. While proving full backpropagation training remains computationally prohibitive [51], proving batch compliance with physical predicates represents a compact, well-defined statement along that path, offering a practical intermediary without waiting for full proof of training to become tractable.

There is a trade-off to using ZK for physical admission: The proof that hides the data also hides the explanation trace. Invariant rejections on their own allow straightforward tracing to their root cause, but after the zk-SNARK, the operator only knows whether it is valid or invalid. Revealing the underlying data would negate the ZK aspect and data privacy.

6.5. Limitations

The limitations of this study are multifaceted, reflecting the challenges of addressing both theoretical and practical considerations. Although we have sought to address existing gaps and maintain a positive mass flow, the experiments presented herein necessarily simplify the underlying complex and multidimensional systems. With that said, a few words are due regarding adaptive adversaries, prototyping, and generalisability.

As in any setting with intelligence on both sides advancing offensive and defensive capabilities, adversaries can exploit blind spots: replayed sequences that obey the invariant set, regime-spliced data, and successfully projected batches all evade defences. The damage removal reported in Section 5.3 must therefore be read regarding covered attacks rather than universal protection. An adversary that directly optimises the poisoning objective within the feasible invariants and tailors updates to evade downstream aggregation rules, represents a stronger threat. It remains open how much poisoning capability survives when an adversary simultaneously optimises across both the physical invariants and the server's filters.

PA-FL Lite attests batch compliance over sampled rows but does not cryptographically bind the submitted update to the committed batch. A client committing honest telemetry while training on poisoned data would evade the gate, reducing to a naive attacker whose updates bypass most robust aggregation rules (Figure 6). Proving the training step itself remains future work. Furthermore, the circuit checks a fixed-point representation of the invariants for which its strict equivalence to floating-point constraints is argued and not proven [48]. And while ZK conceals the raw telemetry batch, the public verification circuit embeds the invariant coefficients, disclosing aspects of plant topology to the coordinator. This leaves open whether the invariant equations themselves can be evaluated privately.

Lastly, regarding data (and their public availability), the empirical evaluations rely on testbed facilities and simulated distribution networks governed by mass conservation and their publicly available datasets. Our federations have ten clients; because the gate is a per-client check its cost grows linearly with their number, and larger, heterogeneous federations are to be explored. Extending data-space admission to thermodynamic cycles, electrical grids with uncoupled dynamics (HAI), and live facilities remains future work.

7. Conclusions

This work establishes that process physics can serve as an effective, privacy-preserving admission gate for federated intrusion detection in critical water infrastructure. Across physical SWaT and WADI records, automatically mined affine invariants admit all honest client shards while removing most of the targeted damage of historical attack replay, even when adversaries possess complete knowledge of the invariant set. By adjusting between narrow and wide invariant sets, operators explicitly control their defensive coverage boundaries prior to training. Finally, we show that a zk-SNARK data-space admission check proves batch compliance in seconds on local clients and verifies in milliseconds at the coordinator's end, showing that data verification and privacy can be reconciled for sampled batches.

Author Contributions: Conceptualisation, J.N., S.S. and B.O.; methodology, J.N.; software, J.N. and B.O.; validation, J.N., S.S., B.O. and S.S.T.; writing—original draft preparation, J.N. and S.S.; writing—review and editing, J.N., S.S. and S.S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The SWaT and WADI datasets analysed during this study are subject to third-party licensing restrictions and cannot be redistributed or republished. They are available for academic research upon formal request to the iTrust Centre for Research in Cyber Security at the Singapore University of Technology and Design (SUTD) at https://itrust.sutd.edu.sg/itrust-labs_datasets/, accessed on 1 November 2026. The BATADAL benchmark dataset is publicly accessible at http://www.batadal.net/data.html, accessed on 1 November 2026, and the HAI dataset is publicly available at https://github.com/icsdataset/hai, accessed on 1 November 2026. All source code, invariant mining scripts, zero-knowledge verification circuits, unit tests, and raw experimental evaluation results (JSON) supporting this study are publicly available in our GitHub repository at https://github.com/RMIT-BDSL/pafl, accessed on 1 November 2026.

Acknowledgments: During the preparation of this manuscript, the authors used Claude Code (Fable 5.1) and Gemini (Flash 3.8) for assistance with coding and debugging, and figure generation. Writeful (Overleaf) was used for language editing and figure editing. The authors reviewed and edited the output and take full responsibility for the content of the published article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Data Preparation and Verification Protocols

This appendix records the channel handling applied to the two physical testbeds and specifies the two protocols by which the invariant sets are verified before any federated training (Section 4.6).

Appendix A.1. Testbed Channel Handling

SWaT Analyser Channels. The Secure Water Treatment facility monitors its chemical dosing loops with Analyser Indicator Transmitters (AITs). Between the seven-day normal record and the attack record, these channels drift by up to 18σ of their normal-record spread, a shift that reflects reagent depletion and probe fouling rather than any manipulation of the plant. Left in, the drift dominates an unsupervised detector's reconstruction error on the attack record: together with a numerical fault in the scaler (six constant actuators divided by a 10^{-8} spread), it held the clean detector at F1 0.27 with a precision of 0.16; excluding the AIT channels and flooring, the scaler raised the clean F1 to 0.756. The AIT channels are therefore excluded from detector training and from invariant mining.

WADI Channel Handling. Four channels of the WADI release are entirely empty in the historian export (`2_LS_001_AL`, `2_LS_002_AL`, `2_P_001_STATUS`, `2_P_002_STATUS`) and are dropped; a few analyser channels have short gaps, which are filled from the neighbouring rows.

Appendix A.2. Separation Protocol

The separation protocol asks whether the mined invariants distinguish honest telemetry from fabricated telemetry (without rejecting honest batches):

1. **Batch Sampling.** For each record (SWaT, WADI, BATADAL and HAI), 200 batches of $N = 1000$ consecutive rows are drawn at random start positions from the attack-free record.
2. **Fabrications.** Each batch is fabricated in the four ways defined in Section 3.3: channel roll by 60 rows (5 min at the 5 s stride; 7 rows on BATADAL and HAI), regime permutation over four k-means regimes, conservation scaling of the flows by 1.3, and regime splicing into twelve reordered segments.
3. **Outcome.** Every honest and fabricated batch is scored against the calibrated invariant set under the 1% admission rule. The protocol records the fraction of violating rows

per batch, the honest false-rejection rate and the rejection rate of each fabrication (Figure 5).

Appendix A.3. Attack Coverage Audit

The coverage audit determines which of the plant's own labelled attacks an invariant set can see:

1. **Segmentation.** Each contiguous run of labelled attack rows at the 5 s stride is one attack: 35 on SWaT and 14 on WADI (Section 4.6 gives the counting convention).
2. **Scoring.** Every attack is scored under each invariant set. An attack counts as *covered* when more than 1% of its rows violate at least one invariant, the same rule the gate applies to a batch.
3. **Honest Rate.** The honest violation rate of each set is measured on the first 60,000 rows after the invariant slice of the normal record; it is the cost side of the coverage dial (Table 4).

Appendix B. Targeted-Recall Damage and Removal and Detector Performance per Rule

Table A1 shows the targeted-recall damage and removal across aggregation rules. For each aggregation rule, attacker, and invariant set, the damage of the naive attack in the targeted recall points, the share of that damage removed by projection alone and by the deployed gate, and the share of malicious updates the rule admitted in each federation are shown. The removal ratios are damage-weighted over the informative seeds.

Table A1. Targeted-recall damage and its removal per aggregation rule on SWaT (20 seeds). n: informative seeds (naive damage $\geq$ 1 point); damage is their mean, in points. Rule accepts: share of the attacker's updates the rule aggregates. Removal: share of naive damage removed, with 95% bootstrap intervals; * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$ (exact one-sided Wilcoxon test, gated vs naive). Values above 100% lie within seed variance. † Krum accepts no malicious update, so its damage is not poisoning.

				Damage Removed (%)				
Rule	**Rule Accepts**	***n***	**Naive Damage**	**Proj.**	**Gate [95% CI]**			
Replay, wide set								
FedAvg	1.00	11	11.7	59	82	61	102	***
Median	1.00	13	6.2	50	68	41	99	***
Norm clip	1.00	11	8.2	75	92	74	103	***
Trimmed mean	0.34	13	7.0	59	77	56	98	***
FLTrust	0.22	5	2.2	47	63	34	100	*
FoolsGold	0.66	8	10.6	−14	5	−53	53	
Krum †	0.00	3	8.4	0	29	0	100	
Channel roll, wide set								
FedAvg	1.00	10	9.1	40	82	57	105	**
Median	1.00	10	6.7	53	71	39	100	***
Norm clip	1.00	7	8.5	66	88	61	100	*
Trimmed mean	0.34	11	6.1	43	66	34	96	**
FLTrust	0.15	5	2.3	11	34	−2	81	
FoolsGold	0.90	9	11.1	−16	21	−43	73	
Krum †	0.00	3	8.4	0	29	0	100	

Table A1. *Cont.*

			Damage Removed (%)					
Rule	**Rule Accepts**	***n***	**Naive Damage**	**Proj.**	**Gate [95% CI]**			
Replay, narrow set								
FedAvg	1.00	11	11.7	44	53	30	77	**
Median	1.00	13	6.2	49	61	29	86	***
Norm clip	1.00	11	8.2	67	70	37	97	***
Trimmed mean	0.34	13	7.0	45	57	33	81	***
FLTrust	0.22	5	2.2	35	35	5	77	
FoolsGold	0.66	8	10.6	−29	−16	−82	46	
Krum †	0.00	3	8.4	0	29	0	100	
Channel roll, narrow set								
FedAvg	1.00	10	9.1	35	47	21	78	**
Median	1.00	10	6.7	45	58	27	85	**
Norm clip	1.00	7	8.5	58	63	27	91	*
Trimmed mean	0.34	11	6.1	56	71	42	92	**
FLTrust	0.15	5	2.3	−14	−14	−44	19	
FoolsGold	0.90	9	11.1	−23	−5	−76	36	
Krum †	0.00	3	8.4	0	29	0	100	

Table A2 reports the global detector's own metrics in every federation (FedAvg). Poisoning barely moves F1 or AUC-PR but lowers recall on the targeted attacks, which is why removal is measured on targeted recall; gated federations match the honest-only reference, so the gate costs no detection utility.

Table A2. Detector metrics of the global model per federation (FedAvg; mean ± standard deviation over seeds). The alarm threshold is calibrated on clean data; AUC-PR is threshold-free. The clean and honest-only federations do not involve the attacker; BATADAL has no honest-only federation and no untargeted attack windows.

Federation	**F1**	**AUC-PR**	**Targeted Recall**	**Untargeted Recall**
SWaT, wide set (20 seeds)				
clean	0.771 ± 0.010	0.797 ± 0.005	0.206 ± 0.112	0.719 ± 0.014
honest-only	0.771 ± 0.016	0.797 ± 0.005	0.198 ± 0.119	0.715 ± 0.018
replay, naive	0.773 ± 0.008	0.780 ± 0.004	0.133 ± 0.094	0.711 ± 0.015
replay, physics-aware	0.770 ± 0.010	0.790 ± 0.012	0.171 ± 0.108	0.705 ± 0.014
replay, gated	0.770 ± 0.014	0.792 ± 0.011	0.186 ± 0.118	0.711 ± 0.017
roll, naive	0.774 ± 0.007	0.782 ± 0.005	0.152 ± 0.102	0.710 ± 0.015
roll, physics-aware	0.771 ± 0.011	0.797 ± 0.006	0.173 ± 0.105	0.707 ± 0.014
roll, gated	0.771 ± 0.014	0.797 ± 0.006	0.192 ± 0.118	0.712 ± 0.016
WADI (3 seeds)				
clean	0.260 ± 0.002	0.352 ± 0.008	0.197 ± 0.091	0.096 ± 0.142
honest-only	0.264 ± 0.006	0.348 ± 0.011	0.198 ± 0.091	0.104 ± 0.135
replay, naive	0.256 ± 0.002	0.305 ± 0.014	0.192 ± 0.093	0.097 ± 0.144
replay, physics-aware	0.256 ± 0.001	0.304 ± 0.012	0.192 ± 0.093	0.097 ± 0.143
replay, gated	0.262 ± 0.008	0.332 ± 0.032	0.196 ± 0.089	0.104 ± 0.135
roll, naive	0.258 ± 0.003	0.309 ± 0.016	0.194 ± 0.092	0.098 ± 0.141
roll, physics-aware	0.257 ± 0.002	0.312 ± 0.019	0.194 ± 0.093	0.097 ± 0.142
roll, gated	0.262 ± 0.007	0.333 ± 0.031	0.197 ± 0.090	0.103 ± 0.136
BATADAL (3 seeds)				
clean	0.470 ± 0.000	0.653 ± 0.005	0.337 ± 0.000	–
replay, naive	0.458 ± 0.019	0.619 ± 0.011	0.323 ± 0.024	–
replay, physics-aware	0.458 ± 0.019	0.619 ± 0.010	0.323 ± 0.024	–
replay, gated	0.458 ± 0.019	0.619 ± 0.010	0.323 ± 0.024	–
roll, naive	0.433 ± 0.036	0.593 ± 0.014	0.306 ± 0.032	–
roll, physics-aware	0.443 ± 0.014	0.595 ± 0.008	0.314 ± 0.011	–
roll, gated	0.443 ± 0.014	0.595 ± 0.008	0.314 ± 0.011	–

Appendix C. Sensitivity to the Admission Threshold

An alpha sweep is shown in Figure A1 repeating the gated federations at $\alpha = 0.5$, 2 and 5% (SWaT wide set, 20 seeds). Tightening α to 0.5% rejects one honest shard while loosening it to 2% or more admits nearly every projected batch. The paper's $\alpha = 1\%$ is the tightest setting tested that rejects no honest shard.

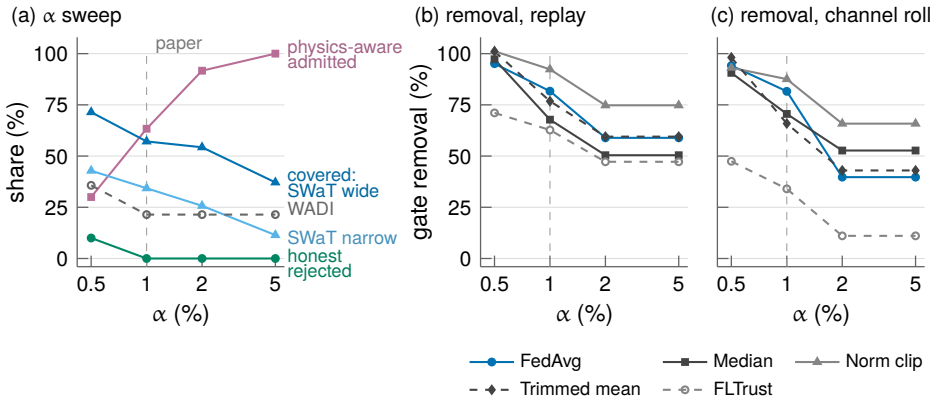


Figure A1. Sensitivity to the admission threshold α (SWaT wide set unless stated; 20 seeds). (**a**) Distinct honest shards rejected, physics-aware batches admitted, and labelled attacks covered. No honest shard is rejected on SWaT narrow, WADI or BATADAL at any α; at 0.5% one SWaT wide-set shard (0.51% violating rows) is rejected. (**b**,**c**) Share of naive damage removed by the gate, per rule; at $\alpha \geq 2\%$ the gate excludes no attacker in an informative seed, so removal equals the projection-only value. The dashed line marks our paper's $\alpha = 1\%$.

Appendix D. Zero-Knowledge Circuit Arithmetisation (PA-FL Lite)

This appendix gives the protocol sequence and the arithmetisation of the PA-FL Lite Circom circuit, including fixed-point encoding, the biased representation that prevents field wrap-around, the Poseidon leaf hashing, and the constraint breakdown and prover cost in Table A3.

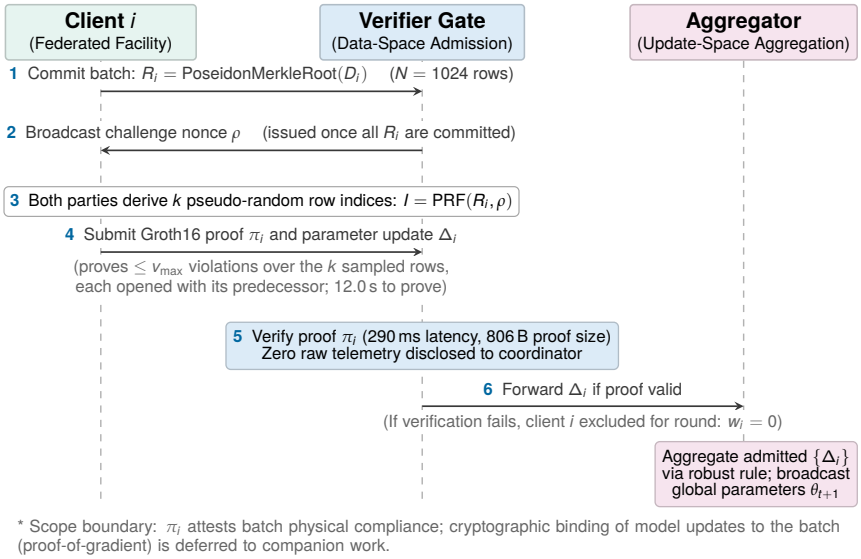


Figure A2. PA-FL Lite protocol round. The client commits to the training batch ($N = 1024$ rows) as a Poseidon Merkle root; once every commitment is in, the verifier broadcasts a challenge nonce, and both parties derive k pseudo-random row indices. A Groth16 proof is returned that at most $v_{\max}$ sampled rows violate an invariant and that at most $u_{\max}$ checks are inapplicable. The verifier checks the proof, and the aggregator combines the admitted updates with a robust rule; a client whose proof fails is excluded for the round ($w_i = 0$). Timings are for $k = 32$ (Table 6).

Appendix D.1. Circuit Pipeline and R1CS Optimisations

To validate telemetry inside a rank-1 constraint system within an edge device's budget, the circuit uses four arithmetisation choices:

1. Constants Baked into the Template.The invariant parameters, the coefficient matrices $(J_{\text{prev}}, J_{\text{cur}})$, offsets c_j and tolerances ε_j are compiled into the Circom template as numerical constants. The affine residuals $r_j = J_{\text{prev},j}\, x[i-1] + J_{\text{cur},j}\, x[i] + c_j$ are then linear combinations of witness values and cost no multiplication gates. The public inputs are only $(R_i, I, v_{\max}, u_{\max})$, $k+3$ field elements (35 at $k = 32$).
2. Fixed-Point Scaling with a Biased Encoding. Sensor readings and tolerances are scaled to integers by $S = 2^{16}$. Field elements of $\mathbb{F}_p$ ($p \approx 2^{254}$ on the BN128 scalar field) have no sign, so a negative value would wrap to near p; projected batches do carry negative values, since the adversary's projection drives unconstrained flow channels below zero (in one projected SWaT batch, FIT401 reads $-62\,\text{m}^3/\text{h}$). Every value therefore enters the circuit as $\hat{x} = \lfloor x \cdot 2^{16} \rceil + 2^{31}$ and is range-checked to 32 bits, and every residual stays below $2^{53} \ll p$, so field arithmetic reproduces signed integer arithmetic exactly.
3. Chunked Poseidon Leaf Hashing. A Poseidon permutation takes at most 16 inputs, so the 18 constrained channels of a row are hashed in two chunks whose digests are combined with a precomputed digest of the row's unconstrained channels. A leaf costs 1095 constraints, against about 1500 for a single wide sponge.
4. Soft Violation Accumulation. Instead of asserting $|r_j| \le \varepsilon_j$ row by row, which would make the circuit unsatisfiable at the first violation, each check is a 56-bit comparison on the biased residual that emits a Boolean flag. The flags, and the flags of inapplicable actuator checks, are summed over the k samples and compared with the public budgets $v_{\max}$ and $u_{\max}$ once at the circuit boundary.

Appendix D.2. Constraint Breakdown and Prover Cost

At $k = 32$ samples over a batch of $N = 1024$ rows (Merkle depth 10), the circuit has 297,736 R1CS constraints, 9304 per sample. The two Merkle path openings (2×2420) and the two leaf hashes (2×1095) take 7030 of these, about 76%; the 36 range checks on the two sampled rows take 1152 (12%), the nine tolerance checks 1035 (11%), and the applicability and index-bit check the remaining 80. Benchmarked on an Apple M3 Pro (Circom 2.1.9, SnarkJS 0.7.4, Groth16 over BN128, Powers of Tau 2^{20}), proving takes 12.0 s at 3.8 GB peak memory; verification is 290 ms independent of k; the Groth16 proof consists of three group elements ($A \in G_1$, $B \in G_2$, $C \in G_1$), consuming 256 bytes in binary and 806 bytes in SnarkJS's JSON encoding. At $k = 16$, the circuit has 148,888 constraints and proves in 6.55 s with the same verification cost (Table A3).

Table A3. PA-FL Lite full benchmark across sample sizes k. Attestation outcomes are for the seed-0 SWaT batches under a budget of $v_{\max} = 1$ violations and $u_{\max} = 8$ inapplicable checks; sampling soundness is estimated from 19,992 random index draws over 56 honest windows.

Protocol Metric/Circuit Property	$k = 16$ Samples	$k = 32$ Samples
R1CS arithmetic constraints (total)	148,888	297,736
Constraints per sample (2 paths, 2 leaves, physics)	9306	9304
Public/private circuit inputs	19/928	35/1856
Circuit compilation time (Circom)	7.47 s	15.0 s
Proving key size (`.zkey`)	108.7 MB	217.4 MB

Table A3. *Cont.*

Protocol Metric/Circuit Property	$k = 16$ Samples	$k = 32$ Samples
Verification key size (`.vk`)	6.1 kB	8.9 kB
Witness generation (honest batch)	0.41 s	0.56 s
Proof generation time π_i (peak RAM)	6.55 s (2.3 GB)	12.0 s (3.8 GB)
Proof verification time (coordinator)	300 ms	290 ms
Proof size (Groth16 π_i)	808 B	806 B
Empirical Attestation Outcome ($v_{\max} = 1, u_{\max} = 8$ *budget)*		
Honest batch (0.39% violating rows)	Verified (0/0)	Verified (0/0)
Channel roll (21.29% violating rows)	Hard-abort (6/1)	Hard-abort (18/1)
Historical replay (23.34% violating rows)	Hard-abort (5/0)	Hard-abort (6/0)
Physics-aware projected (0.68% violating rows)	Verified (0/0)	Verified (0/0)
Monte Carlo Sampling Soundness (19,992 *draws)*		
Honest batch acceptance ($P(v \leq 1)$)	99.41%	98.51%
Detection of channel roll ($P(v > 1)$)	93.5%	99.8%
Detection of historical replay ($P(v > 1)$)	91.7%	99.8%

Appendix D.3. Binding the Update to the Committed Batch

PA-FL Lite proves a statement about a committed batch, while the aggregator receives an update, and nothing in the protocol ties the two: A client can commit an honest batch, pass verification, and submit an update trained on other data. Such a client is equivalent to the naive attacker (Section 3.3), and the gate does not stop it. Narrowing this gap is an important direction for future work. Two promising pathways stand out: zero-knowledge proofs of training [51], which cryptographically bind the update to the committed batch but currently incur substantial circuit overheads for backpropagation, and probabilistic spot-auditing, where selected rounds are verified by re-executing training on revealed batches. In the interim, robust aggregation remains the active backstop: in our empirical sweep, FLTrust limited data poisoning damage to 1.4–2.0 percentage points of targeted recall, whereas FedAvg, coordinate median, and norm clipping admitted every poisoned update (Figure 6).

References

1. Dragos, Inc. *OT Cybersecurity Year in Review 2026*; Technical Report; Dragos, Inc.: Hanover, MD, USA, 2026. Available online: https://www.dragos.com/ot-cybersecurity-year-in-review/ (accessed on 1 November 2026).
2. Cybersecurity and Infrastructure Security Agency. Exploitation of Unitronics PLCs Used in Water and Wastewater Systems. Cybersecurity Alert, 2023. Available online: https://www.cisa.gov/news-events/alerts/2023/11/28/exploitation-unitronics-plcs-used-water-and-wastewater-systems (accessed on 1 November 2026).
3. Tuptuk, N.; Hazell, P.; Watson, J.; Hailes, S. A Systematic Review of the State of Cyber-Security in Water Systems. *Water* **2021**, *13*, 81. [CrossRef]
4. Raman, G.M.R.; Ahmed, C.M.; Mathur, A. Machine Learning for Intrusion Detection in Industrial Control Systems: Challenges and Lessons from Experimental Evaluation. *Cybersecurity* **2021**, *4*, 27. [CrossRef]
5. Koay, A.M.Y.; Ko, R.K.L.; Hettema, H.; Radke, K. Machine Learning in Industrial Control System (ICS) Security: Current Landscape, Opportunities and Challenges. *J. Intell. Inf. Syst.* **2023**, *60*, 377–405. [CrossRef]
6. Goh, J.; Adepu, S.; Tan, M.; Lee, Z.S. Anomaly Detection in Cyber Physical Systems Using Recurrent Neural Networks. In Proceedings of the IEEE International Symposium on High Assurance Systems Engineering (HASE), Singapore, 12–14 January 2017; pp. 140–145. [CrossRef]
7. Kravchik, M.; Shabtai, A. Detecting Cyber Attacks in Industrial Control Systems Using Convolutional Neural Networks. In Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy (CPS-SPC), Toronto, ON, Canada, 19 October 2018; pp. 72–83. [CrossRef]
8. Kravchik, M.; Shabtai, A. Efficient Cyber Attack Detection in Industrial Control Systems Using Lightweight Neural Networks and PCA. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 2179–2197. [CrossRef]

9. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; Agüera y Arcas, B. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), Lauderdale, FL, USA, 20–22 April 2017*; Proceedings of Machine Learning Research: Fort Lauderdale, FL, USA, 2017; Volume 54, pp. 1273–1282.
10. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
11. European Parliament and Council of the European Union. Directive (EU) 2022/2555 on Measures for a High Common Level of Cybersecurity Across the Union (NIS2 Directive), Article 23. *Off. J. Eur. Union* **2022**, 80–152. Available online: https://eur-lex.europa.eu/eli/dir/2022/2555/oj (accessed on 1 November 2026).
12. Hernandez-Ramos, J.L.; Karopoulos, G.; Chatzoglou, E.; Kouliaridis, V.; Marmol, E.; Gonzalez-Vidal, A.; Kambourakis, G. Intrusion Detection Based on Federated Learning: A Systematic Review. *ACM Comput. Surv.* **2025**, *57*, 1–65. [CrossRef]
13. Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; Shmatikov, V. How to Backdoor Federated Learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics (AISTATS), Sicily, Italy, 26–28 August 2020*; Proceedings of Machine Learning Research: Fort Lauderdale, FL, USA, 2020; Volume 108, pp. 2938–2948.
14. Xia, Q.; Chen, W.; Yu, Z.; Ma, J. Poisoning Attacks in Federated Learning: A Survey. *IEEE Access* **2023**, *11*, 10708–10722. [CrossRef]
15. Nguyen, T.D.; Rieger, P.; Miettinen, M.; Sadeghi, A.R. Poisoning Attacks on Federated Learning-based IoT Intrusion Detection System. In *Proceedings of the Workshop on Decentralized IoT Systems and Security (DISS), Co-Located with NDSS, San Diego, CA, USA, 23 February 2020*; Internet Society: Reston, VA, USA, 2020. [CrossRef]
16. Blanchard, P.; El Mhamdi, E.M.; Guerraoui, R.; Stainer, J. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 119–129.
17. Yin, D.; Chen, Y.; Ramchandran, K.; Bartlett, P. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In *Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018*; Proceedings of Machine Learning Research: Fort Lauderdale, FL, USA, 2018; Volume 80, pp. 5650–5659.
18. Cao, X.; Fang, M.; Liu, J.; Gong, N.Z. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 21–25 February 2021. [CrossRef]
19. Fung, C.; Yoon, C.J.M.; Beschastnikh, I. The Limitations of Federated Learning in Sybil Settings. In Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID), San Sebastian, Spain, 14–16 October 2020; pp. 301–316.
20. Lycklama, H.; Burkhalter, L.; Viand, A.; Küchler, N.; Hithnawi, A. RoFL: Robustness of Secure Federated Learning. In Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 21–25 May 2023; pp. 453–476. [CrossRef]
21. Bell, J.; Gascón, A.; Lepoint, T.; Li, B.; Meiklejohn, S.; Raykova, M.; Yun, C. ACORN: Input Validation for Secure Aggregation. In Proceedings of the 32nd USENIX Security Symposium (USENIX Security), Anaheim, CA, USA, 9–11 August 2023; pp. 4805–4822.
22. Ma, Y.; Guo, Y.; Karthikeyan, H.; Polychroniadou, A. Armadillo: Robust Single-Server Secure Aggregation for Federated Learning with Input Validation. In Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS), Taipei, Taiwan, 13–17 October 2025. [CrossRef]
23. Fang, M.; Cao, X.; Jia, J.; Gong, N.Z. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In Proceedings of the 29th USENIX Security Symposium (USENIX Security), Boston, MA, USA, 12–14 August 2020; pp. 1605–1622.
24. Baruch, G.; Baruch, M.; Goldberg, Y. A Little Is Enough: Circumventing Defenses for Distributed Learning. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019; Volume 32, pp. 8635–8645.
25. Shejwalkar, V.; Houmansadr, A. Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 21–25 February 2021. [CrossRef]
26. Zhou, X.; Cheng, Z.; Wang, C.; Wang, S.; Tao, C.; Zhou, Z.; Chen, X.; Luo, J.; Wang, D.; Zhou, H. A Dataset Collected in Real-World Industrial Control Systems for Network Attack Detection. *Sci. Data* **2026**, *13*, 112. [CrossRef] [PubMed]
27. Adepu, S.; Mathur, A. Using Process Invariants to Detect Cyber Attacks on a Water Treatment System. In Proceedings of the 31st IFIP TC 11 International Conference on ICT Systems Security and Privacy Protection (IFIP SEC), Ghent, Belgium, 30 May–1 June 2016; pp. 91–104. [CrossRef]
28. Feng, C.; Palleti, V.R.; Mathur, A.; Chana, D. A Systematic Framework to Generate Invariants for Anomaly Detection in Industrial Control Systems. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 24–27 February 2019. [CrossRef]
29. Zhu, Q.; Ding, Y.; Jiang, J.; Yang, S.H. Anomaly Detection Using Invariant Rules in Industrial Control Systems. *Control Eng. Pract.* **2025**, *154*, 106164. [CrossRef]

30. Giraldo, J.; Urbina, D.; Cardenas, A.; Valente, J.; Faisal, M.; Ruths, J.; Tippenhauer, N.O.; Sandberg, H.; Candell, R. A Survey of Physics-Based Attack Detection in Cyber-Physical Systems. *ACM Comput. Surv.* **2018**, *51*, 1–36. [CrossRef] [PubMed]
31. Sun, Z.; Kairouz, P.; Suresh, A.T.; McMahan, H.B. Can You Really Backdoor Federated Learning? *arXiv* **2019**, arXiv:1911.07963. [CrossRef]
32. Geiping, J.; Fowl, L.; Huang, W.R.; Czaja, W.; Taylor, G.; Moeller, M.; Goldstein, T. Witches' Brew: Industrial Scale Data Poisoning via Gradient Matching. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 4 May 2021. [CrossRef]
33. Tramèr, F.; Carlini, N.; Brendel, W.; Madry, A. On Adaptive Attacks to Adversarial Example Defenses. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtually, 6–12 December 2020; Volume 33, pp. 1633–1645.
34. Carlini, N.; Athalye, A.; Papernot, N.; Brendel, W.; Rauber, J.; Tsipras, D.; Goodfellow, I.; Madry, A.; Kurakin, A. On Evaluating Adversarial Robustness. *arXiv* **2019**, arXiv:1902.06705. [CrossRef]
35. Arp, D.; Quiring, E.; Pendlebury, F.; Warnecke, A.; Pierazzi, F.; Wressnegger, C.; Cavallaro, L.; Rieck, K. Dos and Don'ts of Machine Learning in Computer Security. In Proceedings of the 31st USENIX Security Symposium (USENIX Security), Boston, MA, USA, 10–12 August 2022; pp. 3971–3988.
36. Urbina, D.I.; Giraldo, J.A.; Cardenas, A.A.; Tippenhauer, N.O.; Valente, J.; Faisal, M.; Ruths, J.; Candell, R.; Sandberg, H. Limiting the Impact of Stealthy Attacks on Industrial Control Systems. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS), Vienna, Austria, 24–28 October 2016; pp. 1092–1105. [CrossRef]
37. Ahmed, C.M.; Ochoa, M.; Zhou, J.; Mathur, A.P.; Qadeer, R.; Murguia, C.; Ruths, J. NoisePrint: Attack Detection Using Sensor and Process Noise Fingerprint in Cyber Physical Systems. In Proceedings of the 2018 ACM on Asia Conference on Computer and Communications Security (AsiaCCS), Incheon, Republic of Korea, 4 June 2018; pp. 483–497. [CrossRef]
38. Feng, C.; Tian, P. Time Series Anomaly Detection for Cyber-Physical Systems via Neural System Identification and Bayesian Filtering. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD), Singapore, 14–18 August 2021; pp. 2858–2867. [CrossRef]
39. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-Informed Machine Learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. [CrossRef]
40. Cho, G. Attack-Level Failure Analysis of Invariant-Rule-Based Anomaly Detection in Industrial Control Systems. *Mathematics* **2026**, *14*, 2016. [CrossRef]
41. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), Dallas, TX, USA, 30 October–3 November 2017; pp. 1175–1191. [CrossRef]
42. Xu, G.; Li, H.; Liu, S.; Yang, K.; Lin, X. VerifyNet: Secure and Verifiable Federated Learning. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 911–926. [CrossRef]
43. Wang, Z.; Dong, N.; Sun, J.; Knottenbelt, W.; Guo, Y. zkFL: Zero-Knowledge Proof-Based Gradient Aggregation for Federated Learning. *IEEE Trans. Big Data* **2024**, *10*, 447–460. [CrossRef]
44. Wang, T.; Jin, Y.; Yang, Q.; Xia, Y.; Shi, L.; Zhang, S. Zero-Knowledge Federated Learning: A New Trustworthy and Privacy-Preserving Distributed Learning Paradigm. *arXiv* **2025**, arXiv:2503.15550. [CrossRef]
45. Groth, J. On the Size of Pairing-Based Non-interactive Arguments. In *Proceedings of the Advances in Cryptology—EUROCRYPT 2016, Vienna, Austria, 8–12 May 2016*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9666, pp. 305–326. [CrossRef] [PubMed]
46. Grassi, L.; Khovratovich, D.; Rechberger, C.; Roy, A.; Schofnegger, M. Poseidon: A New Hash Function for Zero-Knowledge Proof Systems. In Proceedings of the 30th USENIX Security Symposium (USENIX Security), Vancouver, BC, Canada, 11–13 August 2021; pp. 519–535.
47. Guo, H.; Feng, Y.; Wu, C.; Li, Z.; Xu, J. Benchmarking ZK-Friendly Hash Functions and SNARK Proving Systems for EVM-Compatible Blockchains. *arXiv* **2024**, arXiv:2409.01976. [CrossRef]
48. Chen, B.J.; Waiwitlikhit, S.; Stoica, I.; Kang, D. ZKML: An Optimizing System for ML Inference in Zero-Knowledge Proofs. In Proceedings of the Nineteenth European Conference on Computer Systems (EuroSys), Athens, Greece, 22–25 April 2024; pp. 560–574. [CrossRef]
49. Kuznetsov, O.; Kuznetsova, Y.; Ziyatbekova, G.; Kovalenko, Y.; Palahusynets, R. Performance Evaluation of zk-SNARK Protocols for Privacy-Preserving Sensor Data Verification. *Sensors* **2026**, *26*, 2486. [CrossRef] [PubMed]
50. Xing, Z.; Zhang, Z.; Zhang, Z.; Li, Z.; Li, M.; Liu, J.; Zhang, Z.; Zhao, Y.; Sun, Q.; Zhu, L.; et al. Zero-Knowledge Proof-Based Verifiable Decentralized Machine Learning in Communication Network: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2025**, *28*, 985–1024. [CrossRef]
51. Peng, Z.; Zhao, C.; Wang, T.; Liao, G.; Lin, Z.; Liu, Y.; Cao, B.; Shi, L.; Yang, Q.; Zhang, S. A Survey of Zero-Knowledge Proof Based Verifiable Machine Learning. *Artif. Intell. Rev.* **2026**, *59*, 57. [CrossRef]

52. Jia, H.; Yaghini, M.; Choquette-Choo, C.A.; Dullerud, N.; Thudi, A.; Chandrasekaran, V.; Papernot, N. Proof-of-Learning: Definitions and Practice. In Proceedings of the 42nd IEEE Symposium on Security and Privacy (S&P), Virtual, 24–27 May 2021; pp. 1039–1056. [CrossRef]
53. Mathur, A.P.; Tippenhauer, N.O. SWaT: A Water Treatment Testbed for Research and Training on ICS Security. In Proceedings of the 2016 International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater), Vienna, Austria, 11 April 2016; pp. 31–36. [CrossRef]
54. Goh, J.; Adepu, S.; Junejo, K.N.; Mathur, A. A Dataset to Support Research in the Design of Secure Water Treatment Systems. In *Proceedings of the Critical Information Infrastructures Security (CRITIS 2016), Paris, France, 10–12 October 2017*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2016; Volume 10242, pp. 88–99. [CrossRef]
55. Ahmed, C.M.; Palleti, V.R.; Mathur, A.P. WADI: A Water Distribution Testbed for Research in the Design of Secure Cyber Physical Systems. In Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater), Pittsburgh, PA, USA, 18–21 April 2017; pp. 25–28. [CrossRef]
56. Taormina, R.; Galelli, S.; Tippenhauer, N.O.; Salomons, E.; Ostfeld, A.; Eliades, D.G.; Aghashahi, M.; Sundararajan, R.; Pourahmadi, M.; Banks, M.K.; et al. Battle of the Attack Detection Algorithms: Disclosing Cyber Attacks on Water Distribution Networks. *J. Water Resour. Plan. Manag.* **2018**, *144*, 04018048. [CrossRef]
57. Shin, H.K.; Lee, W.; Yun, J.H.; Min, B.G. Two ICS Security Datasets and Anomaly Detection Contest on the HIL-Based Augmented ICS Testbed. In Proceedings of the 14th Cyber Security Experimentation and Test Workshop (CSET), Virtual Event, 9 August 2021; pp. 36–40. [CrossRef]
58. Huong, T.T.; Bac, T.P.; Ha, K.N.; Hoang, N.V.; Hoang, N.X.; Hung, N.T.; Tran, K.P. Federated Learning-Based Explainable Anomaly Detection for Industrial Control Systems. *IEEE Access* **2022**, *10*, 53854–53872. [CrossRef]
59. Pecherle, G.D.; Győrödi, R.Ş.; Győrödi, C.A. Federated Learning-Based Intrusion Detection in Industrial IoT Networks. *Future Internet* **2026**, *18*, 2. [CrossRef]
60. Chen, Y.; Luo, F.; Li, T.; Xiang, T.; Liu, Z.; Li, J. A Training-Integrity Privacy-Preserving Federated Learning Scheme with Trusted Execution Environment. *Inf. Sci.* **2020**, *522*, 69–79. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.