

Adversarial Robustness of Anomaly Detectors in Industrial Digital Twins : A White-box Attack Study

Maya Melissa Mellaz

*Université de Lorraine, CNRS, CRAN,
F-54000 Nancy, France
mellaz.maya@univ-lorraine.fr*

Lemia Louail

*Université de Lorraine, CNRS, CRAN,
F-54000 Nancy, France
lemia.louail@univ-lorraine.fr*

Moufida Maimour

*Université de Lorraine, CNRS, CRAN,
F-54000 Nancy, France
moufida.maimour@univ-lorraine.fr*

Meroua Daoudi

*Laboratoire LITAN, École Supérieure en Sciences
et Technologies de l'Informatique et du Numérique
Bejaia, Algeria
daoudi@estin.dz*

Eric Rondeau

*Université de Lorraine, CNRS, CRAN,
F-54000 Nancy, France
eric.rondeau@univ-lorraine.fr*

Karl Andersson

*Luleå University of Technology
Skellefteå, Sweden
karl.andersson@ltu.se*

Abstract—Industrial Digital Twins continuously collect multi-variate sensor data streams from cyber-physical systems and embed machine learning (ML) models to perform real-time anomaly detection on this data. Despite their growing role in industrial safety, the robustness of these models against adversarial perturbations applied directly to sensor measurements remains largely unexplored. This paper systematically transposes three white-box attacks (FGSM, PGD, and Carlini & Wagner) to industrial tabular sensor data and evaluates them across three anomaly detectors of increasing complexity (Logistic Regression, MLP, and XGBoost). We further propose and evaluate architecture-specific adversarial training strategies that substantially improve robustness, although residual vulnerability remains model- and attack-dependent.

Index Terms—adversarial attacks, adversarial training, AI robustness, anomaly detection, CW, cybersecurity, Digital Twin, FGSM, PGD, white-box attacks

I. INTRODUCTION

Industrial systems increasingly rely on Digital Twins (DTs) to monitor, control, and optimise physical assets in real time [1]. A DT is a virtual representation of a physical system and its associated environment, continuously updated through the bidirectional exchange of information between physical and virtual counterparts [2]. In industrial contexts, DTs integrate heterogeneous sensor streams to maintain an up-to-date representation of the underlying process and embed Machine Learning (ML) models tasked with anomaly detection, predictive maintenance, and process optimization [3]. The correctness of these ML components is therefore directly tied to the safety and integrity of the entire Cyber-Physical System (CPS) : a missed anomaly can result in equipment failure, production loss, or physical damage.

This dependency on continuous sensor data streams makes industrial DT monitoring pipelines attractive targets for adversarial attacks [14]. In this context, the attack surface lies directly in the sensor data consumed by the ML models. Adversarial attacks consist of deliberately introducing small, carefully crafted perturbations into the inputs of a machine learning model. Although these perturbations may remain

bounded and numerically admissible, they can still mislead the model into producing incorrect outputs [4]. Applied to an industrial DT monitoring pipeline, such perturbations may cause genuine process anomalies to be misclassified as normal behavior while remaining plausible enough to avoid immediate operator detection. In wireless Industrial Internet of Things (IIoT) environments [5], sensor data streams may transit over radio channels, potentially enabling remote manipulation without direct physical access to the monitored system.

Adversarial attacks were first studied systematically in the image domain. Goodfellow et al. [6] introduced Fast Gradient Sign Method (FGSM), Madry et al. [7] extended it to the iterative Projected Gradient Descent (PGD) framework, and Carlini and Wagner [8] proposed an optimization-based attack that minimizes perturbation magnitude. These representative white-box methods are used in this work as stress tests for assessing the worst-case adversarial vulnerability of ML-based anomaly detectors embedded in industrial DT monitoring pipelines. However, adapting these attacks to tabular industrial sensor data remains non-trivial. Unlike images, CPS measurements lack spatial structure and involve heterogeneous feature scales and domain-specific physical constraints. Existing work on adversarial attacks in Industrial Control System (ICS) contexts either focuses on network-level intrusion detection [9] or investigates white-box attacks in a single model DT setting [10]. While the latter provides early evidence of white-box vulnerability in vehicular DTs, it does not systematically analyze how different ML architectures behave under advanced adversarial attacks on tabular industrial sensor data. Consequently, systematic multi-architecture white-box evaluation, together with architecture-specific adversarial training (AT) defenses, remains lacking.

To address this limitation, this paper studies the adversarial robustness of ML-based anomaly detectors that can be embedded within industrial DT monitoring pipelines. More specifically, we adapt FGSM, PGD, and Carlini & Wagner (CW) to tabular industrial sensor data and evaluate them across three anomaly detectors of increasing complexity with

different architectural properties trained on two industrial CPS anomaly-detection benchmarks

The contributions are threefold. First, we formalize the adaptations required to apply white-box attacks to non-spatial, heterogeneous CPS inputs. Second, we provide a multi-architecture comparison of adversarial vulnerability under identical attack conditions, showing that architectural choices strongly influence robustness and that non-differentiability alone does not constitute a reliable defense. Third, we propose and evaluate architecture-specific AT strategies, and assess their effectiveness across models and attack configurations.

The remainder of the paper is organized as follows : Sections III-A, III-B, and III-C present the three attacks and their adaptation to industrial sensor data. Section IV details the experimental setup, with attack results reported in Section V-B and defense results in Section V-D. Section VI concludes the paper.

II. THREAT MODEL AND PROBLEM FORMULATION

In an industrial DT, the anomaly detector continuously processes the multivariate sensor stream generated by the physical process. We consider an adversary who modifies these measurements before they are processed by the anomaly detector.

a) Attacker Knowledge: The attacks considered in this work operate under the white-box threat model. The adversary is assumed to have complete access to the model architecture, parameters, and gradients [10]. While remote sensor-stream manipulation may not always grant full white-box access, this setting establishes a worst-case upper bound on vulnerability.

b) Attack Objective: The adversary aims to modify the sensor measurements fed to the anomaly detector so that a genuine process attack is misclassified as normal behavior, while the altered readings remain sufficiently stealthy to avoid immediate detection by operators or monitoring mechanisms.

c) Perturbation Constraints: The adversarial perturbation is bounded in each input dimension. For FGSM and PGD, this requirement is expressed through an (ℓ_∞) constraint : $\|x^{\text{adv}} - x\|_\infty \leq \epsilon$. This constraint guarantees that the perturbation is bounded in each dimension, making it difficult to detect.

d) Normalization and domain-admissibility constraint: The input features are constrained to the normalized domain $[0, 1]^n$. Adversarial examples are required to remain within this admissible range. The mechanism used to enforce this constraint depends on the attack. In particular, the change of variable used by CW is described in Section III-C5.

III. WHITE-BOX ATTACKS ON INDUSTRIAL SENSOR DATA

A. Adversarial Attacks using the Fast Gradient Sign Method

1) Principle: A trained neural network learns a decision boundary in the feature space. The core idea behind adversarial examples [6] is the following : if one knows in which direction to modify an input in order to maximize the loss, it is possible to construct an imperceptible perturbation that pushes the model toward a wrong prediction.

FGSM [6] exploits exactly this idea in an efficient single step : rather than using the magnitude of the gradient, which varies across features, only its sign is retained. The perturbation is therefore uniform across all directions that increase the loss, and its overall intensity is controlled by a single hyperparameter ϵ [11].

2) Mathematical Formulation: Let f_θ be the model with parameters θ , $\mathcal{L}(f_\theta(x), y)$ the loss function, and $\epsilon > 0$ the perturbation intensity. The adversarial example x^{adv} is constructed as :

$$x^{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(f_\theta(x), y)), \quad (1)$$

The terms are :

- $\nabla_x \mathcal{L}$: gradient of the loss with respect to the inputs x , not with respect to the weights (θ). This is the same backpropagation as during training, but the weights are frozen, and the gradient is allowed to flow back to the data.
- $\text{sign}(\cdot) \in \{-1, +1\}^d$: only the direction of the gradient is retained, not its magnitude, so that each feature is perturbed by exactly ϵ , bounding the perturbation in ℓ_∞ norm : $\|x^{\text{adv}} - x\|_\infty \leq \epsilon$.
- ϵ : a scalar controlling the overall intensity of the attack. A small value ($\epsilon \rightarrow 0$) leaves the model unaffected ; a larger value produces coarser but more effective perturbations.

Its main limitation is its single-step nature : the perturbation is optimized only once and does not seek to approach a local optimum. Iterative attacks such as PGD exploit this weakness by repeating the FGSM step multiple times with projection.

B. Adversarial Attacks using the Projected Gradient Descent Method

a) Principle: PGD [7] extends FGSM by iteratively applying smaller gradient-sign steps instead of a single perturbation step. At each iteration, the perturbed point is projected back onto the admissible ℓ_∞ ball $\mathcal{B}_\infty(x, \epsilon)$, ensuring that the perturbation remains within the prescribed budget. This iterative refinement allows PGD to search for more deceptive perturbations than FGSM, making it a standard reference attack for evaluating adversarial robustness.

PGD repeats this update T times with a step size $\alpha \leq \epsilon$, followed by a projection onto $\mathcal{B}_\infty(x, \epsilon)$.

b) Random start: A key difference from FGSM is the random initialization : instead of starting from x itself, PGD initializes the perturbation with uniform noise in $[-\epsilon, \epsilon]^d$. The surface $\phi(\mathbf{x}') = \mathcal{L}(f_\theta(\mathbf{x}'), y)$ restricted to $\mathcal{B}_\infty$ is non-convex due to the network's nonlinearities : it exhibits several local maxima of different heights.

c) Mathematical Formulation: Let $\Pi_{\mathcal{B}}$ be the projection operator onto the ball $\mathcal{B}_\infty(x, \epsilon) = \{x' : \|x' - x\|_\infty \leq \epsilon\}$. PGD iterates the following rule :

$$x^{(0)} = x + \delta_0, \quad \delta_0 \sim \mathcal{U}(-\epsilon, \epsilon)^d, \quad (2)$$

$$x^{(t+1)} = \Pi_{\mathcal{B}_\infty(x, \varepsilon)} \left(x^{(t)} + \alpha \cdot \text{sign} \left(\nabla_{x^{(t)}} \mathcal{L} \left(f_\theta(x^{(t)}), y \right) \right) \right), \quad (3)$$

where :

- α is the step size at each iteration, with $\alpha \leq \varepsilon$. Too large an α risks jumping outside the ball before projection ; too small slows convergence.
- $\Pi_{\mathcal{B}_\infty(x, \varepsilon)}$ is the projection onto the ℓ_∞ ball, computed by a simple clamp :

$$\Pi_{\mathcal{B}_\infty(x, \varepsilon)}(x') = x + \text{clamp}(x' - x, -\varepsilon, \varepsilon)$$

- T is the total number of iterations. After T steps, $x^{(T)}$ is the returned adversarial example.

The projection guarantees that the constraint $\|x^{(T)} - x\|_\infty \leq \varepsilon$ is satisfied at every iteration. The random start mitigates non-convexity :by restarting PGD from several points $x^{(0)}$ drawn uniformly from $\mathcal{B}_\infty$, the probability of reaching a high-loss perturbation increases.

Figure 1 illustrates the geometry of iterative attacks in the sensor space of the digital twin. PGD takes T small steps α along the loss gradient, each followed by a projection back onto the ball to enforce the ℓ_∞ perturbation budget. FGSM, by contrast, takes a single large step directly toward the boundary.

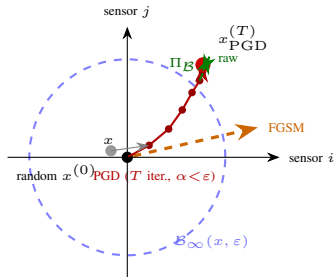


Fig. 1: Comparison of FGSM and PGD in the perturbation space of the industrial sensor measurements.

C. Carlini & Wagner Attack

1) *Principle*: FGSM and PGD require the attacker to fix the perturbation budget ε before launching the attack. In an industrial DT, this is a critical weakness : a budget set too large produces sensor readings that are immediately flagged by engineering alarm thresholds, while a budget set too small may leave the embedded anomaly detector unaffected. Neither outcome meets the attacker's objective.

The CW method [8] removes this pre-specification requirement by seeking the smallest possible perturbation that achieves evasion. Applied to the anomaly detectors studied here, this translates into sensor perturbations that are simultaneously effective against the ML model and maximally covert against process-level monitoring.

2) *The CW Reformulation :a Continuous Optimization Problem*: Classical adversarial attacks minimize the perturbation magnitude $\|\delta\|_p$ subject to the hard constraint $f(x + \delta) = t$, which is non-differentiable and thus incompatible with

gradient-based optimization. CW addresses this by recasting the constraint as a continuous surrogate loss. CW reformulates the problem by replacing the binary constraint with an additive term in the objective function :

$$\min_{\delta} \|\delta\|_p + c \cdot \mathcal{L}(f(x + \delta), t), \quad (4)$$

where the first term penalizes perturbation magnitude (stealthiness) and the second term penalizes misclassification failure (attack effectiveness). These two terms are in constant tension : if δ grows, the stealthiness term penalizes ; if δ remains too small, the attack term stays high and the attack fails. The algorithm naturally finds the right balance via gradient descent. The constant $c > 0$ governs this trade-off ; its optimal value is determined by binary search, retaining the smallest c for which the attack succeeds.

3) *Measuring the Perturbation : the Three Norms*: The stealthiness term Eq. (4) involves a norm $\|\delta\|_p$ that summarizes the magnitude of the perturbation. CW proposes three variants : ℓ_0 counts the number of affected sensors, ℓ_2 measures the total energy spread across all sensors, and ℓ_∞ controls the worst-case perturbation on any single sensor. We retain the ℓ_2 norm in this work.

4) *Measuring Attack Success :the Objective Function*: The attack-effectiveness term relies on a continuous surrogate $\mathcal{L}$ of the hard classification constraint, enabling gradient computation. Given the model logits $Z(x')$, it is defined as :

$$\mathcal{L}(x') = \max \left(Z(x')_{\text{attack}} - Z(x')_{\text{no attack}}, -\kappa \right), \quad (5)$$

The function is negative when the attack succeeds, zero at the decision boundary, and positive when it fails. The margin $\kappa \geq 0$ forces a minimum confidence gap before declaring success, preventing fragile adversarial examples that small sensor noise could reverse (Fig. 2).

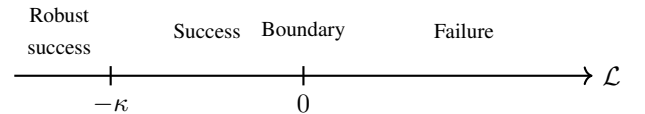


Fig. 2: CW confidence margin κ : robust success, success, and failure regions.

5) *Constraining the Perturbation : the tanh Change of Variable*: To keep $x + \delta$ within admissible bounds $[0, 1]^n$, CW optimizes over an unconstrained variable $w \in \mathbb{R}^n$ and constructs :

$$x + \delta = \frac{1}{2} (\tanh(w) + 1), \quad (6)$$

6) *Final Formula*: Substituting equations (6) and (5) into problem (4), the complete CW ℓ_2 optimization is:

$$\min_{w \in \mathbb{R}^n} \|\tilde{x}(w) - x\|_2^2 + c \cdot \max(Z_{\text{attack}}(\tilde{x}) - Z_{\text{no attack}}(\tilde{x}), -\kappa). \quad (7)$$

where $\tilde{x}(w) = \frac{1}{2} (\tanh(w) + 1)$.

D. Adaptation to Tabular CPS Data

Throughout this section, the three attacks were adapted to the constraints of tabular CPS data, which lacks the spatial structure of images and involves heterogeneous feature scales and domain-specific physical constraints. FGSM and PGD enforce these constraints through the ℓ_∞ ball and clamp projection, while CW enforces them through the tanh change of variable (Section III-C5), keeping all adversarial examples within the admissible range $[0, 1]^n$.

IV. EXPERIMENTAL SETUP

A. Dataset

We evaluate our framework on two widely used benchmarks for anomaly detection in industrial cyber-physical systems. Secure Water Treatment (SWaT) [9] reproduces a real water-treatment plant instrumented with sensors and actuators ; the model must classify each sample as normal or attack. Battle of the Attack Detection Algorithms (BATADAL) [12] reproduces a water distribution network under the same normal/attack classification setting, and is used in Section V-B to assess the generalizability of the reported vulnerability profiles across datasets. Table I summarizes the two benchmarks. Both datasets contain a mix of continuous sensor readings and discrete actuator states, fit on the training split only; no feature selection, temporal windowing, or aggregation is applied, each row being treated as an independent sample. Missing values, present only in SWaT, are imputed with the training-split per-feature mean.

TABLE I: Dataset summary.

	SWaT	BATADAL
Features (sensors / actuators)	25 / 26	31 / 12
Samples (total)	1,441,719	8,980
Train / Val / Test split	64% / 16% / 20%, stratified	
Attack ratio (%)	3.79	2.44

B. Models and Attack Configuration

Models are trained on 64% of the data, with 16% held out for validation and 20% reserved for testing, using a stratified split by class. We evaluate three anomaly detectors of increasing complexity: Logistic Regression, Multilayer Perceptron (MLP), and XGBoost. The MLP uses four layers, ReLU activations, dropout $p=0.3$, Adam ($lr=0.001$), and threshold $\tau=0.5$; the network outputs a raw logit, with the sigmoid applied downstream to obtain a probability against which τ is compared. Logistic Regression uses the SAGA solver with $C = 1.0$ and balanced class weights. XGBoost uses 500 estimators, $\max_depth=6$, ($lr = 0.1$), and early stopping on log-loss. Both Logistic Regression and XGBoost use their standard `predict_proba` output, thresholded at $\tau=0.5$.

C. Metrics

Three complementary metrics are reported on the attack class : Attack Success Rate (ASR), F1, and Recall. ASR measures the fraction of attack samples misclassified as normal. F1 is the harmonic mean of precision and recall on the full test

set. Recall measures the fraction of true attacks still detected. In an ICS context, low recall is the primary safety risk, as it means genuine physical attacks go undetected.

D. Attack Parameters

We consider three white-box strategies: FGSM, PGD, and CW. All attacks are evaluated under $\varepsilon = 0.1$, targeting attack-class samples only. For PGD, we use $T = 200$, $R = 10$, and $\alpha = \varepsilon/4$. Since XGBoost is non-differentiable, we use full model access together with centred finite-difference gradient estimation and majority-vote sign aggregation across four step sizes. The resulting attacks should therefore be interpreted as numerical-gradient adaptations of FGSM, PGD, and CW rather than exact analytical-gradient attacks. For CW, we use $\kappa = 0$. CW is warm-started from the best PGD solution for MLP and XGBoost. For each seed, attacks target a random subset of true positives (correctly classified attack samples on the clean data) capped at 500 for SWaT and 200 for BATADAL, which is why results are aggregated over $N = 10$ independent seeds.

V. RESULTS

A. Clean Baseline

Before any adversarial perturbation, all three models achieve strong anomaly-detection performance on the clean sensor data (Table II). These results confirm that all three classifiers are capable of reliably distinguishing normal sensor behaviour from attack patterns, providing a solid and realistic starting point for adversarial evaluation.

TABLE II: Baseline performance on clean test data.

Model	Acc. (%)	F1	Prec.	Recall
MLP	99.59	0.9478	0.9070	0.9924
LogReg	92.71	0.4776	0.3277	0.8798
XGBoost	99.99	0.9992	0.9984	0.9915

B. Attack Results

Results are reported as median ASR over $N = 10$ independent runs with $\varepsilon = 0.1$; standard deviation remains below 1.8 percentage points across all models and attacks.

Figure 3 summarises the ASR obtained for each attack across the three models.

1) *MLP*: The MLP shows moderate but consistent vulnerability to gradient-based attacks. FGSM reaches an ASR of 16.6% (Figure 3a), reflecting the inherent limitation of a single gradient step : the non-linear decision boundary of the MLP is not well approximated by a single linear move in the gradient direction, so a fraction of samples requires multiple steps to cross it.

PGD and CW, which leverage iterative optimisation, raise the ASR to 20.6% for both, confirming that additional iterations provide a modest but consistent gain by following the curvature of the loss surface more closely.

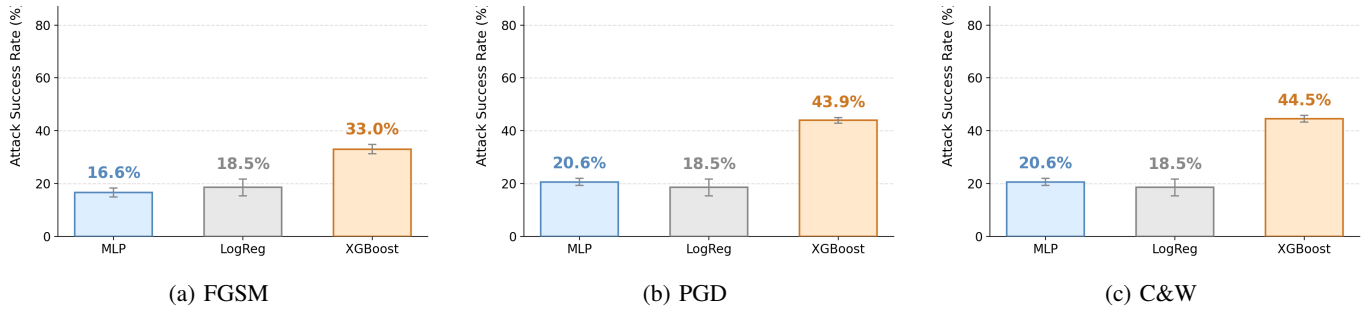


Fig. 3: Median ASR per attack, for each whitebox attack type, Results are averaged over $N=10$ independent runs

2) *Logistic Regression*: All three attacks achieve an identical ASR of 18.5% on LogReg (Figure 3b), with a standard deviation of 3.11 percentage points. The fact that FGSM, PGD, and CW yield the same result immediately rules out any algorithmic explanation: for a linear model, the gradient $\nabla_x \mathcal{L} = (\sigma(z) - y) \mathbf{w}$ always points in the same direction regardless of the current x , so iterating PGD or running C&W brings no additional information beyond what FGSM already exploits in one step.

The 18.5% ceiling is therefore a hard geometric limit imposed by the budget $\varepsilon = 0.1$. For a linear model, the maximum logit reduction achievable in ℓ_∞ norm is exactly $\varepsilon \cdot \|\mathbf{w}\|_1$. For the coefficients learned on SWaT, this quantity is sufficient to cross the decision boundary for approximately 18.5% of attack samples, but the majority sit far enough from the hyperplane that a perturbation of magnitude 0.1 cannot reach them. Increasing ε would raise this ceiling proportionally, but at the cost of range validity. This result highlights a fundamental property of linear models under ℓ_∞ attacks: their vulnerability is entirely determined by the geometry of the learned hyperplane and the perturbation budget, not by the sophistication of the attack.

3) *XGBoost*: XGBoost exhibits a mixed vulnerability profile driven by its non-differentiable structure. FGSM achieves 33.0% ASR (Figure 3c). Using the finite-difference gradient estimator described in Section IV, a single step along the approximate gradient direction already crosses a meaningful share of tree decision boundaries, though less effectively than iterative refinement. PGD and CW, operating iteratively via numerical gradients, reach 43.9% and 44.5% respectively. The gap between FGSM and iterative attacks (33.0% vs. $\sim 44.2\%$) confirms that non-differentiability provides partial but not reliable protection: tree discontinuities effectively blunt single-step attacks, yet repeated numerical gradient steps can progressively navigate the piecewise-constant loss surface and locate perturbations that cross decision boundaries.

C. Dataset Comparison: SWaT and BATADAL

To further assess the generalizability of the reported vulnerability profiles, we replicated the multi-run evaluation protocol on the BATADAL dataset, a benchmark for anomaly detection in water distribution networks. Under identical attack conditions ($\varepsilon = 0.1$, 10 independent seeds), the MLP and

LogReg exhibit substantially higher ASRs on BATADAL than on SWaT: the MLP reaches a median ASR of 37.5% across all three attacks (vs. 16.6–20.6% on SWaT), and LogReg 47.06% across all three attacks (vs. 18.5% on SWaT), both with zero observed standard deviation. XGBoost shows the most pronounced shift, reaching 90.48% under FGSM, 91.67% under PGD, and 79.76% under CW (std up to 5.38 pp). These results suggest that the vulnerability orderings observed on SWaT are dataset-dependent, and that models trained on BATADAL’s sparser attack distribution are exposed to greater adversarial risk under the same perturbation budget.

D. Defense Results

Having characterised the vulnerability of each model, we apply adversarial training as a defense strategy [13], [15]. Defense models are trained with $\varepsilon = 0.3$ to expose the model to robust perturbations during hardening, following a robust-optimisation margin principle [7]: the ℓ_∞ ball of radius 0.3 strictly contains that of radius 0.1, so a model hardened up to 0.3 is, by construction, at least as robust at the operational budget of 0.1. Model selection (early stopping, hyperparameters) is however anchored at the deployment budget $\varepsilon = 0.1$, via the combined score $F1_{\text{val}} - \text{ASR}_{\text{val}}(\text{FGSM}, \varepsilon = 0.1)$, so the larger training radius only shapes the robustness margin while retention decisions track the true operational budget. Results are reported as median ASR over $N = 10$ independent seeds.

1) Training Procedure:

a) *AT-FGSM / AT-PGD for MLP*: At each mini-batch, adversarial examples are generated on-the-fly using FGSM or PGD, and mixed 50/50 with clean examples before the gradient update. Early stopping on validation F1 is retained.

b) *Aug-FGSM for LogReg*: Logistic Regression has no iterative training loop, so adversarial augmentation is applied before fitting. FGSM is run on attack-class samples only, and the resulting adversarial examples are concatenated with the original training set. The model is then refit on this augmented dataset.

c) *Aug-FGSM-Iter for XGBoost*: The same centred finite-difference gradient estimator used during attack evaluation is applied to generate AT examples. The procedure runs for three rounds: at each round, adversarial examples are generated on the current XGBoost model, accumulated into

TABLE III: Post-defense results at $\varepsilon = 0.1$ ($\varepsilon_{\text{train}} = 0.3$), median ASR(%) over $N = 10$ seeds (SWaT).

Model	Defense	Attack	ASR(%)	F1
MLP	AT-FGSM	FGSM	1.8	0.9889
MLP	AT-FGSM	PGD	2.3	0.9874
MLP	AT-FGSM	C&W	2.2	0.9874
MLP	AT-PGD	FGSM	1.1	0.9910
MLP	AT-PGD	PGD	1.2	0.9905
MLP	AT-PGD	C&W	1.2	0.9905
LogReg	Aug-FGSM	FGSM/PGD/C&W	10.6	0.9309
XGBoost	Aug-FGSM-Iter	FGSM	5.7	0.9707
XGBoost	Aug-FGSM-Iter	PGD	17.5	0.9041
XGBoost	Aug-FGSM-Iter	C&W	18.2	0.8999

the training set, and the model is refitted. This iterative self-hardening ensures that perturbations target XGBoost’s current vulnerabilities at each stage.

2) *Baseline and Post-defense Results*: Defense results are evaluated against the undefended baseline reported in Section V-B; post-defense results are reported in Table III.

a) *MLP*: Both AT variants substantially reduce vulnerability. AT-PGD achieves the best overall results : ASR drops to 1.1–1.2% across all attacks (from a 16.6–20.6% undefended baseline), with F1 recovering to 0.990–0.991. AT-FGSM is close behind, with ASR at 1.8–2.3% and F1 at 0.987–0.989. Both defenses generalise well beyond their training attack, with comparable gains against FGSM, PGD, and CW.

b) *LogReg*: Adversarial augmentation reduces ASR from 18.5% to 10.6% ($\Delta = -7.9$ pp), with F1 recovering to 0.9309. As on the undefended model, all three attacks share the same post-defense ASR, consistent with the fact that all attacks exploit the same fixed gradient direction for a linear model.

c) *XGBoost*: The iterative self-hardening strategy yields asymmetric results across attacks, mirroring the undefended vulnerability profile. Against FGSM, ASR drops from 33.0% to 5.7% ($\Delta = -27.3$ pp). Against CW, the reduction is comparable, from 44.5% to 18.2% ($\Delta = -26.3$ pp). Against PGD, the reduction is similar in magnitude, from 43.9% to 17.5% ($\Delta = -26.4$ pp), though the post-defense ASR remains the highest of the three attacks, consistent with PGD’s ability to iteratively navigate the piecewise-constant loss surface of tree ensembles even after self-hardening.

The results reveal a clear vulnerability ordering grounded in the geometric properties of each model. The fixed gradient direction of Logistic Regression explains why FGSM, PGD, and CW achieve identical post-defense success rates. The MLP’s non-linear decision boundary responds best to adversarial training, reaching near-clean performance. XGBoost provides the weakest defense of the three, its non-differentiability continuing to leave it more exposed to PGD than to FGSM or CW even after self-hardening.

Overall, AT substantially reduces vulnerability across all three architectures, although its effectiveness remains model- and attack-dependent. Table IV summarises the ASR reduction achieved by each defense, grouping results by model and attack.

TABLE IV: ASR at $\varepsilon = 0.1$: undefended vs. defended ($\varepsilon_{\text{train}} = 0.3$) model (SWaT), per model and attack.

Model	Attack	Before (%)	After (%)	Δ
MLP (AT-PGD)	FGSM	16.6	1.1	-15.5
	PGD	20.6	1.2	-19.4
	C&W	20.6	1.2	-19.4
LogReg (Aug-FGSM)	all	18.5	10.6	-7.9
XGB (Aug-FGSM-Iter)	FGSM	33.0	5.7	-27.3
	PGD	43.9	17.5	-26.4
	C&W	44.5	18.2	-26.3

VI. CONCLUSION

This paper evaluated the adversarial robustness of three anomaly detectors (Logistic Regression (LogReg), MLP, and XGBoost) under FGSM, PGD, and CW white-box attacks, and investigated architecture-specific AT strategies as defenses. The results highlight that robustness is strongly influenced by the geometric structure of each model’s decision boundary, with architectural choices playing a central role in vulnerability to adversarial perturbations. Future work should extend this analysis to black-box and transfer attacks, incorporate physical plausibility constraints on perturbations, and explore certified defenses for non-differentiable models.

REFERENCES

- [1] M. Javaid et al. Digital Twin Applications toward Industry 4.0: A Review. *Cognitive Robotics*, 3:71–92, Jan 2023.
- [2] E. VanDerHorn et al. Digital Twin: Generalization, Characterization and Implementation. *Decision Support Systems*, 145:113524, Feb 2021.
- [3] A. Maradapu Vera Venkata Sai et al. Navigating the Digital Twin Network Landscape: A Survey on Architecture, Applications, Privacy and Security. *High-Confidence Computing*, 4(4):100269, Dec 2024.
- [4] E. Anthi et al. Adversarial Attacks on Machine Learning Cybersecurity Defences in Industrial Control Systems, Apr 2020.
- [5] M. Kherbache et al. Network Digital Twin for the Industrial Internet of Things. In *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 573–578, 2022.
- [6] I. Goodfellow et al. Explaining and Harnessing Adversarial Examples. *arXiv 1412.6572*, Dec 2014.
- [7] A. Madry et al. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*, Feb 2018.
- [8] N. Carlini et al. *Towards Evaluating the Robustness of Neural Networks*. May 2017.
- [9] J. Goh et al. A Dataset to Support Research in the Design of Secure Water Treatment Systems. In Grigore Havarneanu et al. *Critical Information Infrastructures Security*, pages 88–99, Cham, 2017. Springer International Publishing. <https://www.kaggle.com/datasets/vishala28/swat-dataset-secure-water-treatment-system>.
- [10] W. Patterson et al. A White-Box Adversarial Attack Against a Digital Twin, Oct 2022.
- [11] M.M. Mellaz et al. Digital Twins and Adversarial Machine Learning: Evaluating and Hardening AI Model Robustness. SAGIP 2026.
- [12] R. Taormina et al. The Battle of the Attack Detection Algorithms: Disclosing Cyber Attacks on Water Distribution Networks. *Journal of Water Resources Planning and Management*, Aug 2018.
- [13] M. Zhao et al. Adversarial Training: A Survey, Oct 2024.
- [14] Y. Liu et al. Adversarial Sample Attacks and Defenses Based on LSTM-ED in Industrial Control Systems. *Computers & Security*, 140:103750, 2024.
- [15] V. Pozdnyakov et al. Adversarial Attacks and Defenses in Fault Detection and Diagnosis: A Comprehensive Benchmark on the Tennessee Eastman Process. *IEEE Open Journal of the Industrial Electronics Society*, 5:428–440, 2024.