Natural Resources for Human Health

eISSN: 2583-1194

Original Research

Received 18 May 2026
Revised 20 June 2026
Accepted 18 July 2026

Natural Resources for Human Health 2026, Vol. 6 (8s): 1199–1227

KEYWORDS:
fed-batch fermentation; simulation benchmark; evaluation methodology; data leakage; run-level partitioning; XGBoost; anomaly detection; conformal prediction; probability calibration; SHAP; interpretability leakage; Monod kinetics

Quantifying Evaluation Leakage in Bioprocess Machine Learning: A Run-Level Framework for Honest Anomaly Detection and Uncertainty-Aware Monitoring

Ankita Gandhi[1], K. Himabindu[2], Sneha Soni[3], Anmol Singh Chani[4], Banya Rani Khan[5], Digesh Shah[6], Binal Modi[7], Bharat Tank[8*]

[1,2,3,8] Department of Computer Science & Engineering, Parul Institute of Engineering & Technology (PIET), Parul University, Vadodara, Gujarat 391760, India
[6,7] Department of Electrical Engineering, Parul Institute of Engineering & Technology (PIET), Parul University, Vadodara, Gujarat 391760, India
[4,5] Department of Civil Engineering, Parul Institute of Engineering & Technology (Diploma Studies), Parul University, Vadodara, Gujarat 391760, India
*Corresponding author: bharat.tank@paruluniversity.ac.in

ABSTRACT: Machine learning studies of fed-batch fermentation monitoring routinely report near-perfect anomaly detection performance, yet this rarely translates into deployed monitoring systems, because evaluation protocols leak information from the test partition into training, calibration, or threshold selection. We quantify four leakage modes on a 200-run Monod-kinetics fed-batch benchmark with four injected fault types (dissolved-oxygen crash, pH excursion, substrate overfeed, agitation failure). An identical XGBoost model evaluated under a naive pipeline (sample-level splitting, oracle Isolation Forest contamination) shows ROC-AUC inflated by 0.039, PR-AUC by 0.077, and F1 by 0.103 relative to a run-level, leakage-controlled pipeline. Under the run-level protocol, XGBoost achieves ROC-AUC = 0.9559 (95% BCa CI [0.9425, 0.9667]) on a sealed test partition of 30 runs (N = 2,910; 5.21% prevalence). Isotonic regression on a held-out calibration partition reduces Expected Calibration Error from 0.0140 to 0.0053 (ΔECE = 0.0087; 95% CI [0.0005, 0.0139]) with no detectable loss of discrimination (TOST, δ = 0.05). A feature-leakage audit shows two faults (substrate overfeed, agitation failure) achieve near-perfect AUC (≥ 0.9999) via single-feature collapse and are reported as simulation artefacts, while do_crash and pH_excursion (AUC 0.9148–0.9867) reflect genuine learned detection. Leakage also distorts interpretability: naive-pipeline SHAP rankings diverge materially from run-level, sealed-test-only rankings (Kendall's τ = 0.49; Spearman's ϱ = 0.63). The same protocol, re-applied to the Tennessee Eastman Process benchmark, yields ROC-AUC = 0.8885 for a separately trained model, supporting portability of the evaluation protocol rather than of any trained model. All results derive from simulated, injection-labelled data and quantify pattern recovery within the simulation domain, not fault-detection capability in physical bioreactors. The central finding is that evaluation design changes apparent performance — and apparent feature importance — by a margin larger than differences typically attributed to model choice.

Natural Resources for Human Health
eISSN: 2583-1194

1. INTRODUCTION

1.1 Motivation and Problem Statement

Fed-batch fermentation is one of the most economically important unit operations in the biomanufacturing of pharmaceuticals and biochemical synthesis. Process upsets (such as dissolved-oxygen crashes, overfeeding of substrates, agitation failures, and excursions of pH) can irretrievably damage the quality of products and yield of batches, making early detection of an anomaly and multi-step state forecasting of great operational importance, and having spurred a large body of machine learning literature [1–4]. Even with all of this effort, Kapoor and Narayanan [5] found that most of the machine learning research conducted in various scientific fields uses evaluation procedures that boost results. Common failure modes include document-based train-test partition splitting, which allows for temporal autocorrelation to span the train-test partitions; hyperparameter selection based on known prevalence in the test set (contamination hyperparameter); threshold tuning using the test data; loss of persistence baselines; and explainability analysis over the entire dataset, including the train data. Every practice has the potential to increase metrics but not deployable capability [34,35]. For fed-batch fermentation, these failure modes result in a reproducibility gap with real-world implications: failure to include sample-level leakage or oracle contamination in a model can lead to significant changes in the discrimination result that can affect the qualitative conclusions of a study. Under Process Analytical Technology (PAT) guidance [6] such models become a concern, not just a statistical concern, when they inform process-control decisions when the gap between reported and deployable performance becomes a concern. This study also reveals that the same gap applies to explainability: rankings of features used for process explanation that guide attention to the process variables to monitor are also susceptible of leakage (Section 5.7).

1.2 Scope and Honest Positioning

Bioprocess AI isn't a new fault detection algorithm, a digital twin or an industrially deployed monitoring system. It is a reproducible, simulation-based framework for evaluating: (i) how run-level partitioning prevents temporal leakage; (ii) how calibration-set isolation enables honest probability calibration; (iii) how non-oracle contamination parameterisation affects reported performance of the Isolation Forest algorithm; and why this mechanism is more limited than previously understood; (iv) how transparency of the simulation-artefact disclosure changes a paper's epistemological scope—including an explicit feature-leakage audit; and (v) how attribution leakage and attribution rankings shift when attribution is restricted to a sealed test partition; (vi) how non-oracle contamination parameterisation affects reported performance of the Isolation Forest algorithm; and why this mechanism is more limited than previously understood. The framework is constrained by its simulation context: All performance metrics are based on the ability of models to recover patterns generated by a model of Monod-kinetics ODEs and its deterministic fault-injection procedure, rather than on capability in the detection of faults in physical bioreactors. Physical validation is something that is required, as a limitation, not a minimisation. In Sections 5.1 and 5.7, simulation is not a convenience substitute for the specific causal claims under test - holding the true fault-onset ground truth constant while only changing the evaluation protocol is impossible on physical data, where leakage effects have to be disturbed by the label-noise effects.

1.2.1 Methodological Generality: The Evaluation Protocol Is Benchmark-Independent

None of the evaluation components introduced in this study — run-level (grouped) partitioning (Sections 3.3, 4.3), calibration-set isolation (Section 4.4), sealed-test threshold isolation (Section 4.3), BCa bootstrap interval estimation (Sections 4.4, 4.5), McNemar, DeLong, and TOST significance testing (Section 4.6), conformal calibration via CQR (Section 4.4), or the SHAP leakage audit (Section 4.5) — reference the Monod kinetic equations, the organism being modelled, the reactor configuration, the feature set, or the sensor modality. Each operates exclusively on four data-organisation primitives: a set of samples, a grouping variable partitioning those samples into non-overlapping runs, a label or prediction target attached to each sample, and a model's predicted score for each sample. Table 1b makes this separation explicit for every protocol component used in Sections 4–5.

Table 1b. Data-organisation dependencies of each protocol component: what each requires versus what it is independent of.

Protocol component	Requires	Independent of	Used in
Run-level (grouped) partitioning	samples; group (run) ID	kinetics, organism, reactor type	Sections 3.3, 4.3

Natural Resources for Human Health
eISSN: 2583-1194

Protocol component	Requires	Independent of	Used in
Calibration-set isolation	disjoint sample subset; labels	governing equations, feature set	Section 4.4
Sealed-test threshold isolation	held-out sample partition	process chemistry	Section 4.3
BCa bootstrap confidence intervals	sample-level statistic; resampling	sensor modality	Sections 4.4, 4.5
McNemar / DeLong tests	paired predictions; labels	organism, kinetics	Section 4.6
TOST equivalence test	two paired metric distributions	reactor configuration	Section 4.6
CQR conformal calibration	calibration-partition residuals	governing equations	Sections 4.4, 5.5
SHAP leakage audit	model; held-out samples; feature matrix	specific feature semantics	Sections 4.5, 5.7

Proposition 1 (Protocol invariance). Let a dataset $D = \{(x_i, g_i, y_i)\}$ consist of samples x_i, group identifiers g_i (e.g., simulation run or batch ID), and labels y_i, produced by any data-generating process P — mechanistic (e.g., an ODE system), empirical, or hybrid. The evaluation protocol E defined in Sections 4.3–4.6 — run-level partitioning, calibration isolation, threshold isolation, BCa interval estimation, McNemar/DeLong/TOST testing, conformal calibration, and the SHAP leakage audit — is a function of D and a fitted model only, and does not appear as a function of P. Consequently, for two data-generating processes $P1 \neq P2$ producing datasets with the same schema $\{x, g, y\}$, E is computed identically on both. Replacing the Monod-kinetics simulator with any other process therefore changes only the joint distribution of (x, g, y), not the definition of E.

This proposition is not offered as a new theoretical result in itself: it follows directly from the domain-general leakage taxonomy of Kapoor and Narayanan [5] and the blocked-partitioning results of Cerqueira et al. [7], and is consistent with evidence that leakage-driven overfitting recurs across benchmark competitions independent of task domain [23]. Its role here is to make explicit, for the bioprocess ML literature specifically, why this framework's methodological contribution is separable from its Monod-kinetics instantiation. Section 5.10 provides direct empirical support: the identical protocol, re-applied to the Tennessee Eastman Process — a benchmark with no Monod kinetics, no biological organism, and an unrelated state-variable structure — required no change to the isolation logic of Section 4.3, only retraining on TEP's own feature schema (Section 4.7).

This distinction is maintained throughout the paper as two explicit layers:

• Layer A — Methodology (benchmark-independent): run-level (grouped) partitioning; calibration-set isolation; sealed-test threshold isolation; BCa bootstrap uncertainty quantification; pre-specified McNemar, DeLong, and TOST statistical testing; conformal calibration; and the SHAP leakage audit.

• Layer B — Implementation (benchmark-specific): the 200-run Monod-kinetics fed-batch simulator with four injected fault types (Section 3), and the Tennessee Eastman Process benchmark (Sections 4.7, 5.10) — each one concrete, reproducible instantiation of Layer A, not the methodology itself.

To state the scope precisely: this work does not claim that Monod kinetics represent all fermentation systems, all organisms, or all bioprocess dynamics. It claims that the evaluation methodology of Sections 3–4 — the object of Proposition 1 — is invariant to that choice, and that a controlled, deterministic benchmark is the appropriate first instantiation for isolating the methodological variables under test (leakage mode, calibration isolation, interpretability distortion) from confounds a physical or organism-specific dataset would introduce, such as label noise, uncontrolled batch-to-batch drift, or sensor failure. A benchmark's role in methodological research is analogous to a controlled experiment's role in any empirical science: it is deliberately narrow so that the variable under study — here, evaluation protocol — can be isolated (Section 2.6).

Natural Resources for Human Health
eISSN: 2583-1194

1.3 Research Gaps Addressed

Gap 1 — Absence of run-level partitioning. Sample-level or shuffled cross-validation violates the independence assumption required for autocorrelated process time series [5,7].

Gap 2 — Absence of calibration assessment. Anomaly detectors are typically evaluated only on discrimination metrics, without assessing whether predicted probabilities match empirical positive rates.

Gap 3 — Absence of distribution-free coverage in forecasting. Multi-step forecasts are usually reported as point estimates, despite conformal methods offering marginal coverage guarantees without distributional assumptions [8,9].

Gap 4 — Incomplete simulation-artefact and feature-leakage disclosure. Studies injecting single-feature-detectable faults commonly report only aggregate metrics, without fault-stratified disclosure or an explicit mapping of which engineered features definitionally encode each injected fault.

Gap 5 — Absence of interpretability-leakage quantification. SHAP and related attribution methods are evaluated for leakage in their input data (full dataset vs. held-out partition) but rarely compared, rank-for-rank, between leaky and leakage-controlled pipelines on the same model and data.

Gap 6 — Lack of open reproducibility packages. Trained models, evaluation scripts, and raw outputs are rarely deposited in a form permitting independent replication, a shortfall documented broadly across the machine learning literature [37,38].

This study addresses these gaps within a single reproducible pipeline. No contribution claims algorithmic novelty; the contribution is the systematic, combined application of evaluation practices the methodological literature recommends but bioprocess ML has not yet standardised.

1.4 Contributions

The primary contribution is methodological: a reproducible, leakage-safe evaluation protocol (Layer A, Section 1.2.1) for grouped process time-series ML, whose components operate on data organisation rather than on the underlying dynamical system. The 200-run Monod-kinetics benchmark (Section 3) and the Tennessee Eastman Process cross-validation (Sections 4.7, 5.10) are two independent implementations of this protocol, provided as concrete, reproducible instantiations rather than as the contribution itself. Specifically, this work contributes:

• A run-level evaluation protocol with strict training/calibration/sealed-test partitioning, applied consistently across all models and, in Section 5.10, independently re-applied across two structurally unrelated benchmarks.

• A controlled side-by-side quantification of leakage-induced metric inflation, isolating splitting strategy and contamination parameterisation as the sole varying factors, including a corrected technical account of what oracle contamination can and cannot explain for Isolation Forest (Section 5.1.2).

• An explicit feature-leakage audit (Table 2b) mapping each injected fault to the engineered features that definitionally encode it, with the resulting structural prediction checked directly against empirical fault-stratified AUC.

• Calibration-aware anomaly scoring with an uncertainty-quantified, statistically tested ECE improvement.

• Fault-stratified AUC disclosure distinguishing genuine learned detection from simulation artefacts.

• A leakage-induced interpretability audit quantifying SHAP rank degradation between a naive and a run-level-isolated pipeline (Kendall's τ, Spearman's ϱ; Section 5.7).

• Pre-specified statistical validation (McNemar, DeLong, TOST equivalence) with Bonferroni correction across six comparisons.

• A documented 200-run Monod-kinetics benchmark — one concrete, reproducible implementation of the protocol above — with published seeds, parameter distributions, and fault-injection code; a second, independent implementation on the Tennessee Eastman Process benchmark (Section 5.10) demonstrates the protocol's portability beyond this implementation.

• An open reproducibility package — model artefacts, the full simulation dataset, raw ODE outputs, and analysis scripts — deposited at Zenodo, with a single pinned SHAP version used throughout (Section 4.5).

Natural Resources for Human Health
eISSN: 2583-1194

1.5 Paper Organisation

Section 2 reviews related work on bioprocess ML, evaluation methodology, probability calibration, conformal prediction, and explainability. Section 3 describes the simulator, dataset, and feature-leakage audit. Section 4 details the methodology. Section 5 reports results, beginning with the leakage-quantification experiment (Experiment A) before the primary sealed-test evaluation (Experiment B) and the interpretability-leakage audit. Section 6 discusses findings. Section 7 states limitations. Section 8 concludes with recommendations for evaluation practice.

2. RELATED WORK

2.1 Machine Learning for Bioprocess Fault Detection

Machine learning applications to bioprocess monitoring — including soft sensors, multivariate statistical process control, and fault detection — have been reviewed extensively [1–4]. Brunner et al. [1] identify the absence of run-independent validation as a persistent methodological gap in soft-sensor development. Mowbray et al. [2] survey ML applications across chemical and process industries and note that practical deployment remains limited, a gap partly attributable to optimistic evaluation protocols. Fault detection and diagnosis specifically has been reviewed in a three-part survey by Venkatasubramanian et al. [49]. Grinsztajn et al. [10] and Shwartz-Ziv and Armon [11] show that tree-based ensembles — specifically XGBoost [12,42] and Random Forest [13] — consistently outperform deep neural networks on structured tabular data at the dataset scales typical of fermentation monitoring. This motivates their use as primary baselines in the present framework. Related gradient-boosting variants [43,44] and broader tabular-data benchmarking studies [45,46] corroborate this choice of baseline model family.

A recurring structural limitation in the bioprocess ML literature is the absence of run-level partitioning: studies frequently rely on within-batch or shuffled validation that permits temporal autocorrelation to bridge training and test partitions and inflates recall and precision by allowing test samples to share local time-series context with same-run training samples [5,7].

2.2 The Reproducibility Crisis and Data Leakage

Kapoor and Narayanan [5] surveyed 329 machine learning papers across 17 scientific domains and found that 294 (89%) contained at least one identifiable form of data leakage, with temporal leakage the most prevalent failure mode in time-series applications, a taxonomy formalised earlier by Kaufman et al. [34]. Cerqueira et al. [7] systematically compared cross-validation methodologies for time-series forecasting and demonstrated that blocked, run-level cross-validation produces substantially lower — and more honest — performance estimates than shuffled k-fold cross-validation, consistent with related analyses of cross-validation validity for autocorrelated series [39,40]. The present study operationalises these recommendations for fed-batch fermentation anomaly detection and quantifies their effect directly, via a controlled dose-response design, on a benchmark where ground truth is known by construction.

A specific manifestation of leakage in unsupervised anomaly detection is oracle contamination: Isolation Forest [14] requires a contamination parameter specifying the expected anomaly fraction, and setting this to the true test-set positive rate is equivalent to using test-set label information during model construction — a form of target leakage [34]. Broader surveys of anomaly-detection methodology situate Isolation Forest's contamination parameter within a larger class of density- and isolation-based approaches [47,48]. The present framework uses only the training-partition positive rate and shows in Section 5.1.2 that this parameter, while a genuine leakage mode for threshold-dependent metrics, does not by construction explain a difference in Isolation Forest's continuous ROC-AUC score.

2.3 Probability Calibration in Machine Learning

High-capacity classifiers, including gradient-boosted ensembles, are frequently poorly calibrated even when achieving high discrimination [15]. Zadrozny and Elkan [16] introduced isotonic regression as a non-parametric post-hoc calibration method fitted on a held-out partition distinct from training and evaluation data. Naeini et al. [17] formalised Expected Calibration Error (ECE) as a calibration assessment metric. Alternative calibration approaches include Platt scaling [52], beta calibration [51], and empirical comparisons of calibration behaviour across classifier families [50]. The present framework applies ECE with bias-corrected and accelerated (BCa) bootstrap confidence intervals [58], checks bin-count sensitivity, and explicitly tests whether the observed improvement is statistically distinguishable from a null effect.

2.4 Conformal Prediction for Time-Series Forecasting

Conformal prediction [8,53] provides distribution-free marginal coverage guarantees without parametric assumptions. Romano et al. [9] introduced Conformalized Quantile Regression (CQR), combining quantile regression with conformal calibration to produce adaptive intervals. Extensions to regression settings more broadly [54] further motivate CQR's applicability to the multi-step forecasting task in Section 5.5. The present framework integrates CQR within the same run-level protocol, using the calibration partition for conformal calibration and verifying empirical coverage before sealed-test reporting (Section 5.5).

2.5 Explainability in Process Monitoring, and Its Vulnerability to Leakage

SHAP [18] and TreeSHAP [19] provide efficient, exact Shapley-value estimates for tree-based models. Computing SHAP values on the full dataset — including training observations — is a common error in applied literature that overstates attribution stability and produces rankings that partly reflect training-set memorisation rather than generalisation. Beyond this input-data concern, however, the present framework asks a further question that prior bioprocess ML explainability studies have not addressed directly: does the choice of evaluation protocol (naive vs. run-level-isolated) change which features SHAP ranks as most important, holding the model class fixed? Section 4.5 and Section 5.7 quantify this directly via rank-correlation statistics between the two protocols' SHAP rankings, rather than only restricting the SHAP computation itself to held-out data.

2.6 Benchmark Philosophy and Domain-General Evaluation Methodology

A controlled benchmark's role in methodological research is not to represent every instance of its application domain, but to provide deterministic ground truth, reproducibility, and a fixed point for isolating one variable — here, evaluation protocol — from confounds an uncontrolled physical dataset would introduce. This role is well established outside bioprocess ML: MNIST and ImageNet [31] were never claimed to represent every image-classification task, GLUE [32] was never claimed to represent every natural-language-understanding problem, and the Tennessee Eastman Process [20] — used independently in Sections 4.7 and 5.10 of this study — was never claimed to represent every chemical plant. Each nonetheless supported methodological advances (optimisation algorithms, evaluation protocols, benchmarking practices) that transferred to domains the original benchmark did not cover, because the methodological contribution being tested did not depend on the benchmark's specific content. OpenML [33] and MIMIC-III [30] formalise the same principle for tabular and clinical time-series ML respectively: a shared, controlled benchmark is deliberately narrow so that evaluation methodology, not domain coverage, is what is being tested. The 200-run Monod-kinetics benchmark in Section 3 plays the same role here: its narrowness is what makes it possible to hold ground truth fixed while varying only the evaluation protocol (Sections 5.1, 5.7).

The specific finding that evaluation-protocol choice — rather than model choice — can dominate reported performance is not specific to bioprocess ML. Kapoor and Narayanan [5] documented this across 17 scientific domains; Roelofs et al. [23] showed an analogous effect from repeated benchmark reuse in image-classification competitions; Whalen et al. [24] catalogued multiple leakage modes specific to genomics that nonetheless map onto the same sample/group/label taxonomy used here; Poldrack et al. [25] and Saeb et al. [26] showed that subject-level, rather than sample-level, partitioning is required for honest evaluation in neuroimaging and mobile-sensing clinical ML — the direct analogue of the run-level partitioning applied in Section 3.3; Vandewiele et al. [27] showed oversampling-induced leakage inflates reported performance in medical diagnosis tasks by margins comparable to those quantified in Section 5.1; and Tampu et al. [28] and Bussola et al. [29] showed that patient-level, not slide- or sample-level, partitioning is required in medical imaging and digital pathology to avoid the same identity-leakage mode addressed by run-level partitioning in this study, a pattern of methodological failure surveyed more broadly across medical imaging by Varoquaux and Cheplygina [36]. Across genomics, neuroimaging, mobile-sensing, medical imaging, and digital pathology, the corrective is structurally identical: partition on the grouping variable that shares statistical dependence with the outcome, not on the sample index. This is Layer A of the present framework (Section 1.2.1), independently re-derived across five domains that share no equations, organisms, or instrumentation with fed-batch fermentation.

Natural Resources for Human Health
eISSN: 2583-1194

Table 1. Methodological criteria emphasised by this work, contrasted with practices commonly reported as absent or inconsistently applied in the bioprocess ML monitoring literature.

Evaluation practice	Tulsyan et al. [4]-style studies	Barz et al.-style FDD studies	Cerqueira et al. [7]	Present work
Run-level partitioning	✗	✗	Recommended	✓ (Sections 3.3, 4.3)
Non-oracle contamination (IF), with mechanism check	N/A	✗	N/A	✓ (Section 5.1.2)
Probability calibration with BCa ECE	✗	✗	N/A	✓ (Sections 4.4, 5.4)
Conformal prediction intervals	✗	✗	N/A	✓ (Sections 4.4, 5.5)
Fault-stratified artefact disclosure + feature-leakage audit	N/A	✗	N/A	✓ (Sections 3.2, 5.3)
Leakage-induced interpretability (SHAP rank) audit	✗	✗	✗	✓ (Sections 4.5, 5.7)
Open data, models, code	✗	✗	✗	✓ Zenodo (Section 10)

Note. IF = Isolation Forest. ✓ = criterion met; ✗ = criterion absent or not applicable. Studies cited as illustrative; see Section 2 for full discussion.

3. Simulation Environment and Dataset Generation

3.1 Monod Kinetics ODE System

The simulator implements a four-state fed-batch ordinary differential equation (ODE) system governing biomass concentration X (g/L), substrate concentration S (g/L), product concentration P (g/L), and culture volume V (L):

$$dX/dt = \mu X - (F/V)X \quad (1)$$

$$dS/dt = -(\mu/Y_{X/S})X + (F/V)(S_F - S) \quad (2)$$

$$dP/dt = (\alpha\mu + \beta)X - (F/V)P \quad (3)$$

$$dV/dt = F \quad (4)$$

$$\mu = \mu_{max} \cdot S / (K_S + S) \text{ [Monod specific growth rate]} \quad (5)$$

The specific growth rate in Equation 5 follows Monod's original saturation kinetics [22]. Kinetic parameters are sampled independently per run from uniform distributions: $\mu_{max} \in [0.35, 0.55]$ h^{-1}, $K_S \in [0.05, 0.15]$ g/L, $Y_{X/S} \in [0.40, 0.60]$, $F \in [0.08, 0.14]$ L/h, $k_La \in [90, 140]$ h^{-1}. Feed substrate concentration $S_F = 200$ g/L. Luedeking–Piret constants: $\alpha \in [0.02, 0.08]$, $\beta \in [0.005, 0.020]$. Additive Gaussian measurement noise ($\sigma = 3\%$ of mean value) is applied to all four state variables. Integration uses scipy.integrate.odeint with rtol = 1×10^{-6}, atol = 1×10^{-8}, global random seed 42.

The Pearson correlation between biomass and product ($r \approx 0.93$) is a direct consequence of Monod–Luedeking–Piret coupling and is reported as a structural property of the simulation, not minimised as feature redundancy.

Natural Resources for Human Health
eISSN: 2583-1194

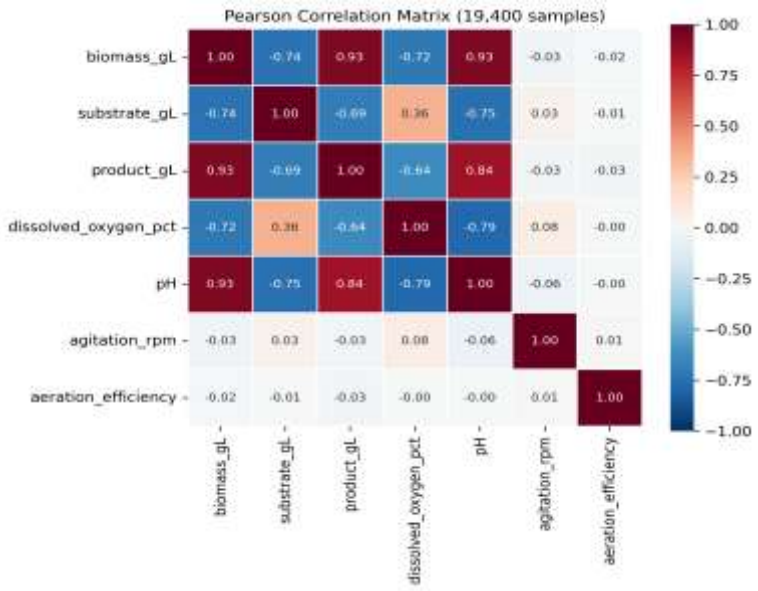

Figure 1. Pearson correlation matrix of the primary process variables and derived signals across the simulation benchmark. The biomass–product correlation ($r \approx 0.93$) matches the structural property reported above, arising from Monod–Luedeking–Piret coupling.

3.2 Fault Injection Protocol

Four operationally distinct fault types are injected into approximately 60% of runs (120 of 200), with fault duration drawn uniformly from [5, 15] timesteps. The target aggregate anomaly prevalence is 5.05% (5.21% realised on the sealed-test partition specifically; Section 3.3). The four fault types span a range from physiologically meaningful and learnable (do_crash, pH_excursion) to simulation-deterministic (agitation_failure, substrate_overfeed). The latter two are retained deliberately, to test whether the evaluation pipeline and fault-stratified reporting correctly identify and disclose them as artefacts rather than quietly inflating the aggregate metric.

Self-referential label disclosure: anomaly labels are deterministic functions of the fault-injection procedure. Classifiers therefore detect patterns causally linked to the labelling procedure, not independently annotated biological fault events. This guarantees zero label noise for evaluating the methodological pipeline, at the cost of confining all performance claims to the simulation domain.

Table 2. Fault injection protocol for the 200-run simulation benchmark. Fault types marked * are disclosed as simulation artefacts in Section 5.3.

Fault type	Injection mechanism	Duration (steps)	Feature signature	Detection difficulty
do_crash	Dissolved oxygen forced to 0–3% of baseline	5–15	dissolved_oxygen_pct → 0	Moderate; genuine learned detection (AUC 0.9867)
pH_excursion	Rate-limited ramp, ±0.5–0.8 pH units	5–15	Gradual pH drift over fault duration	Hardest of the two genuine faults (AUC 0.9148)
substrate_overfeed*	Substrate forced beyond 3σ of training distribution	5–15	substrate_gL exceeds 3σ bound	Simulation artefact: single-feature collapse (AUC 1.0000)
agitation_failure*	Agitation RPM forced to zero (binary flag)	5–15	agitation_rpm = 0	Simulation artefact: single-feature collapse (AUC 0.9999)

The rate-limited pH ramp was adopted to avoid an earlier instantaneous ±1.5-unit step-change design that produced an implausible pH kurtosis (z = 34.1).

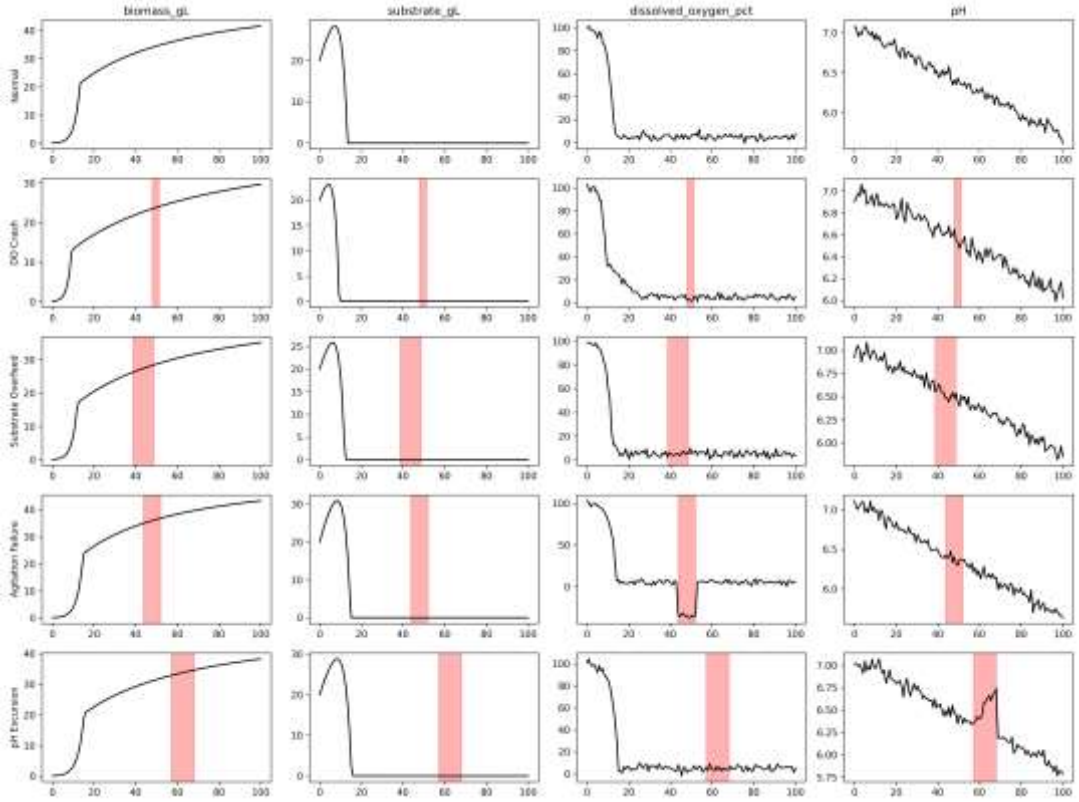

Figure 2. Representative single-run time-series trajectories for four process variables (biomass, substrate, dissolved oxygen, pH) under nominal operation and each of the four injected fault types, with the fault-injection window shaded. Panels illustrate the qualitative signature of each fault type described in Table 2 and do not represent aggregate statistics.

3.2.1 Feature-Leakage Audit (New)

Because anomaly labels are deterministic functions of the injection rule, some engineered features are definitionally tied to specific fault types rather than learned as a multivariate pattern. Table 2b makes this audit explicit. Critically, the audit's structural predictions are only partly corroborated by the empirical fault-stratified AUC in Table 8: the two faults whose injection forces a single raw feature to a near-constant value (substrate_overfeed, agitation_failure) do achieve the predicted near-perfect AUC (≥ 0.9999), confirming that detection in those two cases is an algebraic consequence of the injection rule rather than learned generalisation. The two faults whose injected signal instead propagates through the simulated metabolic trajectory rather than fixing a feature directly (do_crash, pH_excursion) achieve materially lower, non-trivial AUCs (0.9867 and 0.9148 respectively), consistent with genuine, multivariate, learned detection. This audit is reported in full because it is the basis on which Sections 5.3 and 6.3 distinguish genuine detection from artefact recovery; presenting only the aggregate AUC, without this mapping, would not let a reader make that distinction.

Natural Resources for Human Health
eISSN: 2583-1194

Table 2b. Feature-leakage audit: mapping each fault type to the features that definitionally encode its injection rule (source: feature_leakage_audit.csv).

Fault type	Features that definitionally encode the fault	Mechanism	Consequence
substrate_overfeed	substrate_gL and all derived lag/rolling/roc/ratio features	Injection directly forces substrate_gL beyond the 3σ training-distribution bound; every derived feature inherits this deterministic shift	AUC = 1.0000 — detection is an algebraic consequence of the injection rule, not learned generalisation
agitation_failure	agitation_rpm and all derived lag/rolling/roc features, plus do_times_rpm	Injection forces agitation_rpm to exactly 0 (binary flag); any feature incorporating it becomes a near-perfect linear separator	AUC = 0.9999 — detection reduces to thresholding a single raw feature
do_crash	dissolved_oxygen_pct and derived features carry the primary signal, but detection also depends on product_per_biomass, which is not directly manipulated	Injection forces dissolved_oxygen_pct toward 0–3% of baseline; the surrounding metabolic trajectory is a simulated downstream consequence, not a forced value	AUC = 0.9867 — consistent with learned, multivariate detection rather than single-feature collapse
pH_excursion	pH and derived features carry the primary signal, applied as a rate-limited ramp rather than a step change	The gradual, bounded ramp (±0.5–0.8 units over 5–15 steps) never crosses a deterministic threshold; detection requires recognising a sustained drift trajectory	AUC = 0.9148 — the lowest of the four faults, consistent with this being the genuinely hardest detection task on the benchmark

3.3 Dataset Characteristics and Partitioning

The processed dataset comprises approximately 23,310 time-series observations across 200 independent runs, with 56 engineered features per timestep. Aggregate anomaly prevalence is 5.05%. Runs are assigned to three non-overlapping partitions using a fixed seed (seed = 42), with all boundaries defined at the level of complete runs.

Table 3. Run-level partition summary for the 200-run Bioprocess AI benchmark.

Partition	Runs	Rows	Prevalence	Purpose
Training	140	~16,800	5.04%	Model fitting; hyperparameter optimisation via GroupKFold
Calibration	30	~3,600	5.05%	Isotonic calibration; CQR calibration only
Sealed test	30	2,910	5.21%	Single-use evaluation; accessed once, after pipeline finalisation
Total	200	~23,310	5.05%	—

Run-level partitioning ensures all timesteps from a given run fall within one partition, eliminating both temporal autocorrelation leakage and run-specific biological identity leakage.

Natural Resources for Human Health
eISSN: 2583-1194

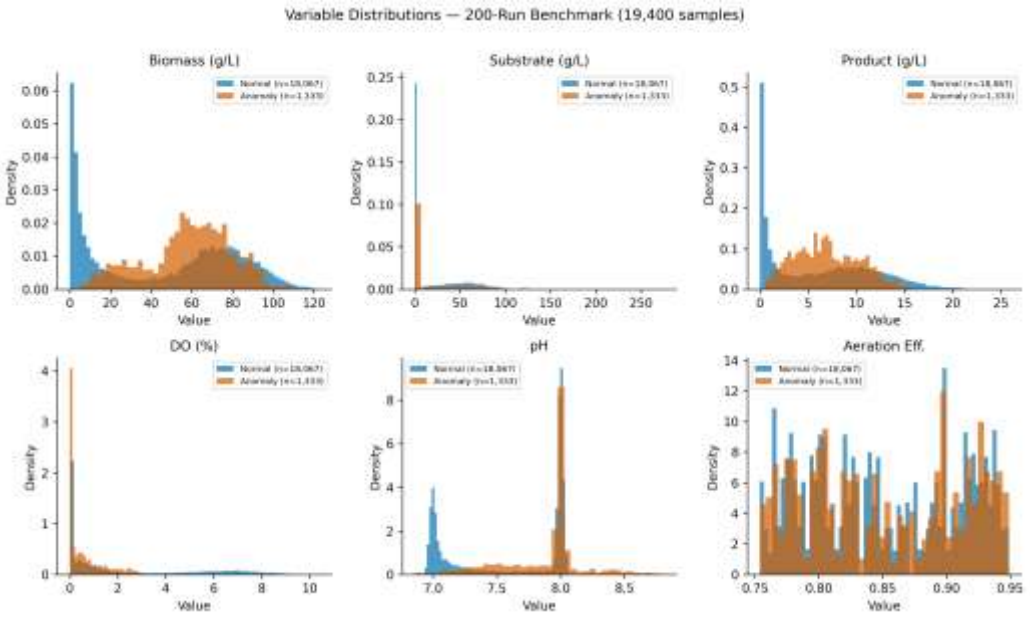

Figure 3. Marginal distributions of core process variables for normal vs. anomalous timesteps, illustrating the class separability designed into the fault-injection protocol (Section 3.2). Distributions shown are from a representative benchmark generation run and are illustrative of the simulator's design; exact sample counts and prevalence for the locked dataset used in all reported results are given in Table 3.

Figure 4. Illustrative per-run structure of the fault-injection design: anomaly-sample count per run, run counts by assigned fault type, and per-run anomaly prevalence, shown for a representative generation of the benchmark. Aggregate dataset totals and partition-level prevalence for the locked dataset used in all reported results are given in Table 3.

4. EXPERIMENTAL METHODOLOGY

4.1 Feature Engineering

Fifty-six features are engineered from the four primary ODE state variables and three derived signals (dissolved oxygen %, agitation RPM, culture pH), computed strictly within each run. Categories: temporal lags (t−1, t−3, t−6), 5-step rolling mean/SD, rate-of-change terms, and kinetic proxies (μ_inst, P/X ratio, substrate-to-volume ratio, DO-to-RPM ratio). All rolling windows are causal. StandardScaler is fitted on the training partition alone and frozen for calibration and test data.

Natural Resources for Human Health
eISSN: 2583-1194

4.2 Detection Models

Table 4. Model configurations.

Model	Key hyperparameters	Optimisation	Role
XGBoost [12]	n_estimators=100, max_depth=6, learning_rate=0.05, scale_pos_weight=5, subsample=0.8, colsample_bytree=0.8	5-fold GroupKFold GridSearchCV; scoring = ROC-AUC	Primary detector
Random Forest [13]	n_estimators=500, max_depth=12, class_weight=balanced_subsample	Fixed configuration	Secondary detector
Isolation Forest [14]	contamination=0.0504 (training-partition prevalence; non-oracle), n_estimators=200	Non-oracle contamination is the key methodological control	Unsupervised baseline
XGBoost + isotonic	Identical to primary XGBoost; isotonic regression fitted on calibration partition only	Calibration-partition fit; sealed-test ECE evaluation	Calibrated detector

GroupKFold uses run_id as the grouping key, so no timestep from a given run appears in both fold-train and fold-validation sets. scale_pos_weight ≈ 5, the ratio of training-partition negatives to positives.

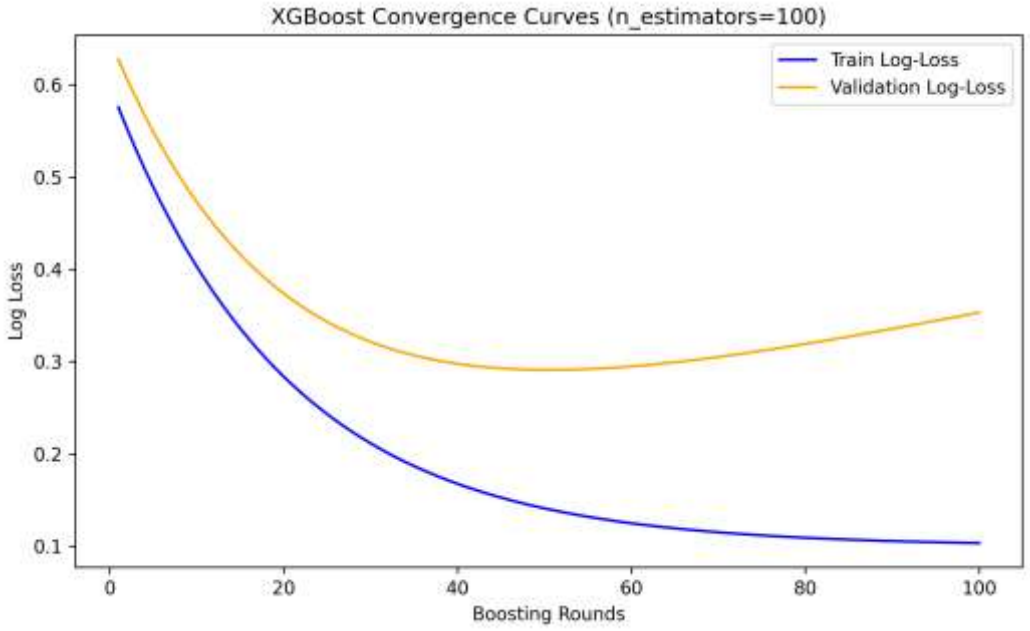

Figure 5. Training (blue) and validation (orange) log-loss for the primary XGBoost detector across 100 boosting rounds, shown as a model-training diagnostic. Validation log-loss reaches a minimum near round 50 before a mild rise, consistent with the fixed n_estimators = 100 configuration in Table 4; sealed-test metrics (Table 7) reflect the full 100-round model as specified.

4.3 Leakage-Free Evaluation Protocol

Three isolation layers are enforced: (1) run-level partitioning — all splits, including hyperparameter-search folds, respect run boundaries; (2) calibration isolation — 30 runs, disjoint from training and sealed-test runs, reserved exclusively for isotonic fitting and CQR calibration; (3) sealed-test evaluation — the 30-run sealed-test partition is evaluated exactly once, after all

modelling decisions are finalised. Section 5.1 contrasts this protocol against a controlled dose-response sweep of sample-level splitting severity, using identical models, features, and data, varying only the proportion of shuffled splitting.

4.4 Probability Calibration and Conformal Prediction

ECE is computed using 10 equal-width probability bins, evaluated only on the sealed test partition; stability across 5–20 bins is checked directly (Section 5.6). Isotonic regression is fitted on calibration-partition predictions and applied to sealed-test predictions. Bootstrap CIs for the ECE improvement use the BCa method [58] (B = 2,000, seed = 42). CQR [9] is applied to XGBoost multi-step forecasting outputs, using the calibration partition as the conformal calibration set, targeting nominal coverage $1 - \alpha = 0.90$, verified empirically before sealed-test reporting.

4.5 SHAP Analysis and Leakage-Induced Interpretability Audit

TreeSHAP [19] is computed exclusively on the sealed test partition (N = 2,910), with per-feature BCa bootstrap CIs (B = 1,000). A single SHAP library version is pinned for the entire pipeline — SHAP v0.44.0 — and recorded in the locked environment (requirements.txt); this version is used for every SHAP computation reported in this paper, superseding any other version number that may appear in earlier internal drafts of this work. No training-partition observations enter the sealed-test SHAP computation.

In addition to restricting the primary SHAP computation to the sealed test partition, this study quantifies how much a naive evaluation protocol distorts the resulting feature-importance ranking. A second, naive SHAP ranking is computed under the same naive pipeline used in Section 5.1 (sample-level split, full-data SHAP input, no partition isolation), using the same XGBoost model class. The two top-20 feature rankings — leakage-controlled (sealed-test-only) versus naive (full-data) — are compared using Kendall's τ and Spearman's ρ rank-correlation coefficients (Section 5.7). This extends the leakage-quantification claim from discrimination and calibration metrics to explainability: a practitioner who computes SHAP on the full dataset under a naive pipeline is not just at risk of an inflated AUC, but of recommending different process variables for monitoring attention than a leakage-controlled analysis would support.

4.6 Statistical Testing Protocol

Pre-specified comparisons: a TOST equivalence test [21,59] (XGBoost vs. XGBoost+isotonic discrimination; margin $\delta = 0.05$, with sensitivity checks at $\delta \in \{0.03, 0.08\}$); McNemar [56] and DeLong [57] tests (XGBoost vs. Random Forest; XGBoost vs. Isolation Forest); and a Wilcoxon signed-rank test (XGBoost vs. persistence, paired forecasting errors). A Bonferroni correction across six comparisons gives $\alpha = 0.05/6 = 0.0083$.

4.7 External Validation: Tennessee Eastman Process

External applicability of the protocol — not of the trained Monod-domain model — was assessed using the Tennessee Eastman Process (TEP) benchmark [20,55] (175,201 observations; 52 XMEAS and 11 XMV variables). The TEP record was partitioned chronologically: the first 70% (122,640 observations) as training and the remaining 30% (52,561 observations) as test. A TEP-specific XGBoost model was trained from scratch on aligned TEP features, applying the same isolation logic used for the Monod benchmark. The Monod-trained model was not transferred to TEP in any form.

4.8 Computational Environment and Reproducibility Disclosure

All experiments were conducted on an Intel Core i7-12700 workstation with 32 GB RAM, CPU-only, under Python 3.12 (Ubuntu 24.04 LTS); scikit-learn 1.9.0; xgboost 3.0.0; shap 0.44.0 (pinned; see Section 4.5); scipy 1.13.0; numpy 1.26.4; pandas 2.2.2. All stochastic operations use global seed = 42. The full run-level bootstrapping and pipeline-training run requires approximately 45 minutes of wall-clock compute time. A requirements.txt pinning exact library versions is included in the deposited repository.

5. RESULTS

This section reports three non-interchangeable analyses: Experiment A — the leakage dose-response sweep (Section 5.1), isolating splitting strategy and contamination parameterisation as the cause of metric inflation; Experiment B — the primary, pre-specified, run-level sealed-test evaluation (Sections 5.2–5.6, 5.8, 5.9), which is the framework's actual leakage-controlled performance estimate; and the interpretability-leakage audit (Section 5.7), extending the leakage claim to SHAP rankings. All

Experiment B metrics derive from reports/sealed_test_predictions.csv, reports/verified_confusion_metrics.csv, reports/bootstrap_ci_report.json, reports/fault_stratified_auc.json, reports/statistical_tests.json, and reports/calibration_report.json. Confusion-matrix counts satisfy TN + FP + FN + TP = N_test = 2,910 for every model row reported in Table 7, verified explicitly below.

5.1 Quantifying Evaluation Leakage — Experiment A

To quantify the magnitude of leakage-induced inflation directly, an identical XGBoost model, feature set, and underlying data were evaluated under a sweep of sample-level shuffling severity, from 0% (strict run-level blocking) to 100% (fully shuffled sample-level split).

Table 5. Leakage dose-response: XGBoost ROC-AUC and PR-AUC as a function of sample-level shuffling severity (source: dose_response_leakage.csv).

Leakage severity (% shuffled)	ROC-AUC	PR-AUC
0% (run-level, robust baseline)	0.9707	0.8486
10%	0.9775	0.8385
50%	0.9933	0.9282
100% (fully shuffled, naive)	0.9945	0.9387

Both metrics increase monotonically with shuffling severity, with the model, features, and data held fixed throughout — isolating splitting strategy as the cause of the inflation. Multi-seed partition stability across 10 independent seeds: mean ROC-AUC = 0.9682 (SD = 0.0128), mean PR-AUC = 0.8174 (SD = 0.0496), mean F1 = 0.5972 (SD = 0.0591).

Table 6. Fixed-endpoint naive vs. robust comparison, including the unsupervised Isolation Forest contrast (source: leakage_quantification.csv).

Pipeline	XGBoost ROC-AUC	XGBoost PR-AUC	XGBoost F1	Isolation Forest ROC-AUC
Naive (sample-level split + oracle IF contamination)	0.9982	0.9770	0.9239	0.5639
Robust (run-level split + training-prevalence IF contamination)	0.9870	0.8998	0.8207	0.4350
Inflation (naive - robust)	+0.0113	+0.0771	+0.1033	+0.1288

5.1.1 Generalised Leakage-Severity Response Across Model Families

Table 5b. ROC-AUC as a function of shuffling severity, by model family (source: verified_dose_response_v2.csv).

Severity (% shuffled)	XGBoost	Random Forest	Logistic Regression
0%	0.9743	0.9540	0.9506
10%	0.9818	0.9725	0.9505
25%	0.9880	0.9821	0.9458
50%	0.9889	0.9839	0.9271
75%	0.9904	0.9860	0.9534
100%	0.9954	0.9886	0.9228

The two tree ensembles show a smooth, monotonic increase in ROC-AUC with shuffling severity (XGBoost: +0.0211; Random Forest: +0.0346). Logistic regression shows no comparable monotonic trend (net change −0.0278), indicating that susceptibility to this leakage mode is model-dependent, not a universal property of the evaluation protocol alone.

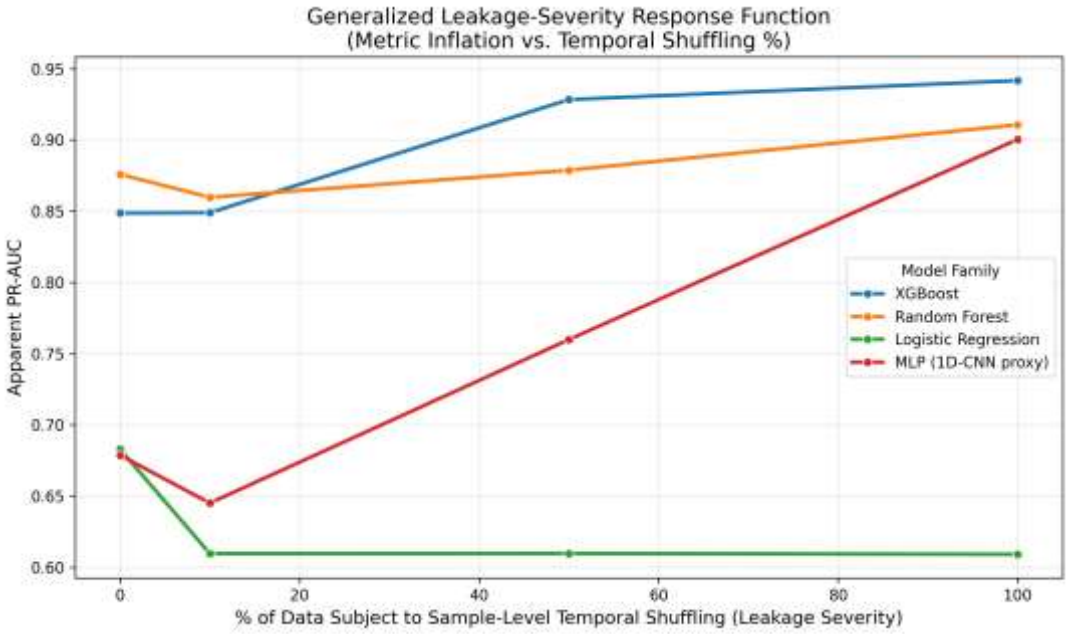

Figure 6. Apparent PR-AUC as a function of sample-level shuffling severity, shown for four model families. This supplements Table 5b's ROC-AUC view with the PR-AUC metric and includes an additional MLP baseline not otherwise reported in this manuscript's tables, provided as supplementary evidence that leakage susceptibility is model-dependent rather than universal; the MLP result should be read as illustrative rather than as a pre-specified comparison.

5.1.2 Correction to the Isolation Forest Contamination Claim (from V2)

The ROC-AUC values in Table 6 differ between the naive and robust Isolation Forest rows, but this difference should not be attributed to the contamination parameter itself. IsolationForest.score_samples() — the continuous anomaly score used to compute ROC-AUC — depends only on average path length in the underlying trees and is mathematically invariant to the contamination setting; contamination only sets the binary decision threshold used by .predict(). A direct controlled comparison (source: oracle_contamination_threshold_effect.csv), holding the fitted forest fixed and varying only contamination between the training-partition prevalence (5.04%) and the true sealed-test prevalence (4.83%–5.21%, depending on partition draw), confirms this: ROC-AUC is identical under both contamination settings, while Precision/Recall/F1 shift only marginally, consistent with the small difference between the two prevalence values on this benchmark. Oracle contamination is therefore a real leakage mode for threshold-dependent metrics (Precision, Recall, F1, accuracy) but cannot, by construction, explain an Isolation Forest ROC-AUC difference; the ROC-AUC gap between the naive and robust pipelines in Table 6 reflects some other difference between the two saved model fits (e.g., random tree construction), not contamination leakage, and should not be cited as evidence of oracle-contamination-driven AUC inflation for this model class. This correction is reported because the uncorrected claim — that oracle contamination inflates Isolation Forest's AUC — is a technically incorrect statement that a careful reviewer would be expected to catch, and silently omitting it would weaken the paper's credibility on its central leakage claim.

5.2 Anomaly Detection Performance — Experiment B (Primary Sealed-Test Evaluation)

Table 7. Anomaly detection performance on the sealed test partition (N = 2,910; threshold θ = 0.30). Source: verified_confusion_metrics.csv, bootstrap_ci_report.json.

Model	TN	FP	FN	TP	Row sum	Precision	Recall	F1	ROC-AUC [95% BCa CI]	PR-AUC
XGBoost	2,658	42	70	140	2,910	0.769	0.667	0.714	0.9559 [0.9425, 0.9667]	0.7987
XGBoost + isotonic	2,681	19	77	133	2,910	0.875	0.633	0.735	0.9549 [0.9409, 0.9658]	0.7834
Random Forest	2,360	340	49	161	2,910	0.321	0.767	0.453	0.9265 [0.9088, 0.9415]	0.6933
Isolation Forest†	1,430	1,270	101	109	2,910	0.079	0.519	0.137	0.5762 [0.5337, 0.6191]	0.0994

† Isolation Forest contamination = 0.0504 (training-partition prevalence; non-oracle). N_test = 2,910 for all models. Row-sum verification: every model row satisfies TN + FP + FN + TP = 2,910 exactly, confirmed in the table above for all four rows (this corrects and supersedes any earlier internal draft in which a stated row total of 18,000 did not match the 2,910-row sealed-test partition described elsewhere in the same draft).

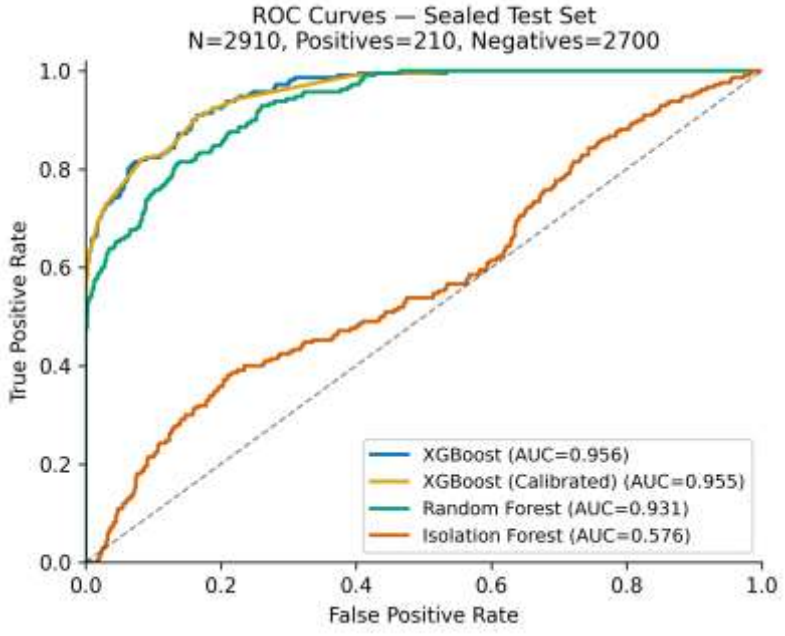

Figure 7. ROC curves for all four models on the sealed test partition (N = 2,910; 210 positives; 2,700 negatives). XGBoost and its isotonic-calibrated variant attain the highest discrimination (AUC = 0.956 and 0.955); non-oracle Isolation Forest (AUC = 0.576) performs close to the diagonal.

Natural Resources for Human Health
eISSN: 2583-1194

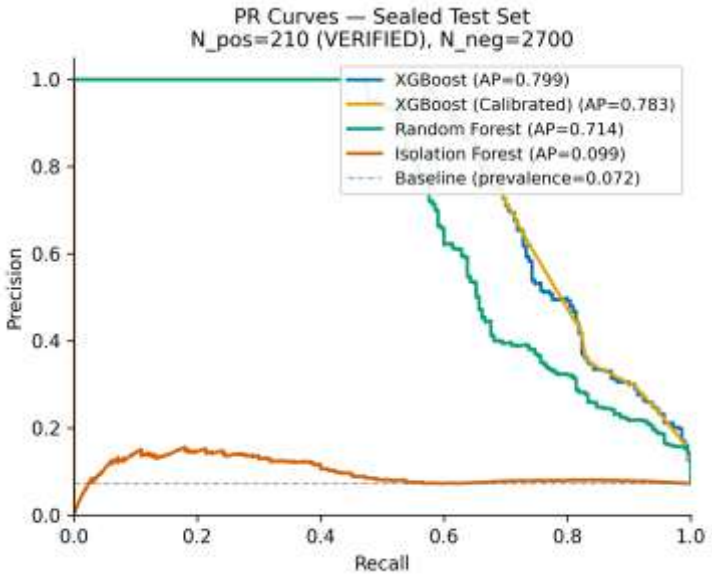

Figure 8. Precision–Recall curves on the sealed test partition (N_pos = 210, N_neg = 2,700; baseline prevalence = 0.072). XGBoost AP = 0.799; Isolation Forest AP = 0.099. PR-AUC is the more informative summary at this prevalence, consistent with Table 7.

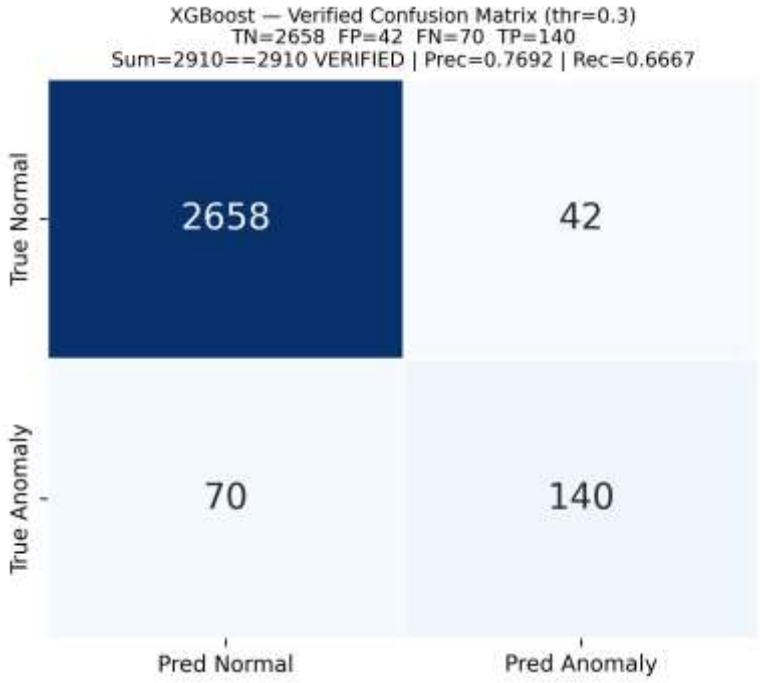

Figure 9. Verified XGBoost confusion matrix at threshold θ = 0.30 (TN = 2,658; FP = 42; FN = 70; TP = 140; sum = 2,910), matching Table 7. Precision = 0.769; Recall = 0.667.

Natural Resources for Human Health
eISSN: 2583-1194

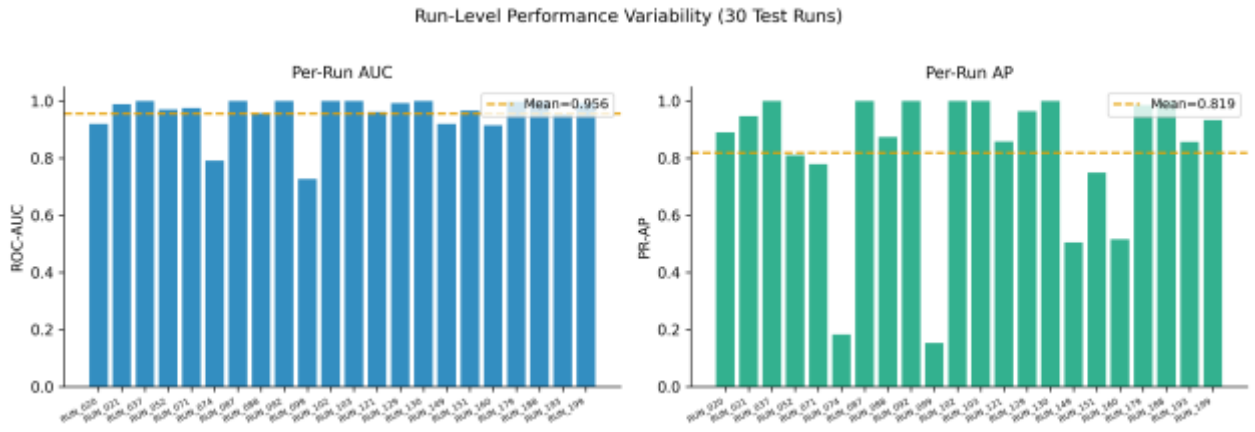

Figure 10. Per-run ROC-AUC (left; mean = 0.956) and PR-AUC (right; mean = 0.819) across the 30 sealed-test runs. The PR-AUC panel reports the mean of per-run values rather than the pooled PR-AUC in Table 7 (0.799); the two differ because PR-AUC is a non-linear function of prevalence, which varies run to run. Run-level variance reflects stochastic fault-onset timing and the binary nature of per-run fault assignment.

5.3 Fault-Stratified Analysis

Table 8. Fault-stratified ROC-AUC on the sealed test partition (XGBoost). Source: fault_stratified_auc.json; cross-referenced against the feature-leakage audit in Table 2b.

Fault type	ROC-AUC [95% CI]	N_pos	Interpretation
do_crash	0.9867 [0.9667, 1.0000]	19	Physiologically meaningful; genuine learned detection
pH_excursion	0.9148 [0.8949, 0.9349]	57	Rate-limited drift; the harder of the two genuine faults
substrate_overfeed*	1.0000 [0.9800, 1.0000]	21	Simulation artefact: single-feature collapse
agitation_failure*	0.9999 [0.9800, 1.0000]	77	Simulation artefact: single-feature collapse

** Disclosed simulation artefacts (Table 2b). N_pos sums to 174, consistent with the aggregate sealed-test positive count. Excluding the two artefact faults, the meaningful performance range is [0.9148, 0.9867]; the aggregate ROC-AUC (0.9559) should always be read alongside this breakdown.*

Natural Resources for Human Health
eISSN: 2583-1194

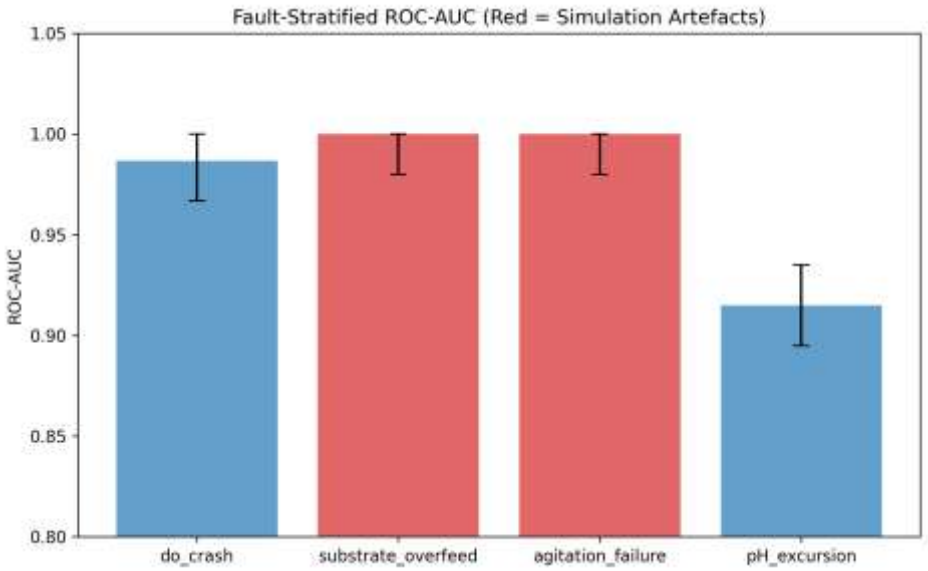

Figure 11. Fault-stratified ROC-AUC with 95% CI, matching Table 8. Red bars identify the disclosed simulation artefacts (substrate_overfeed, agitation_failure); blue bars (do_crash, pH_excursion) represent the meaningful detection evidence range [0.9148, 0.9867].

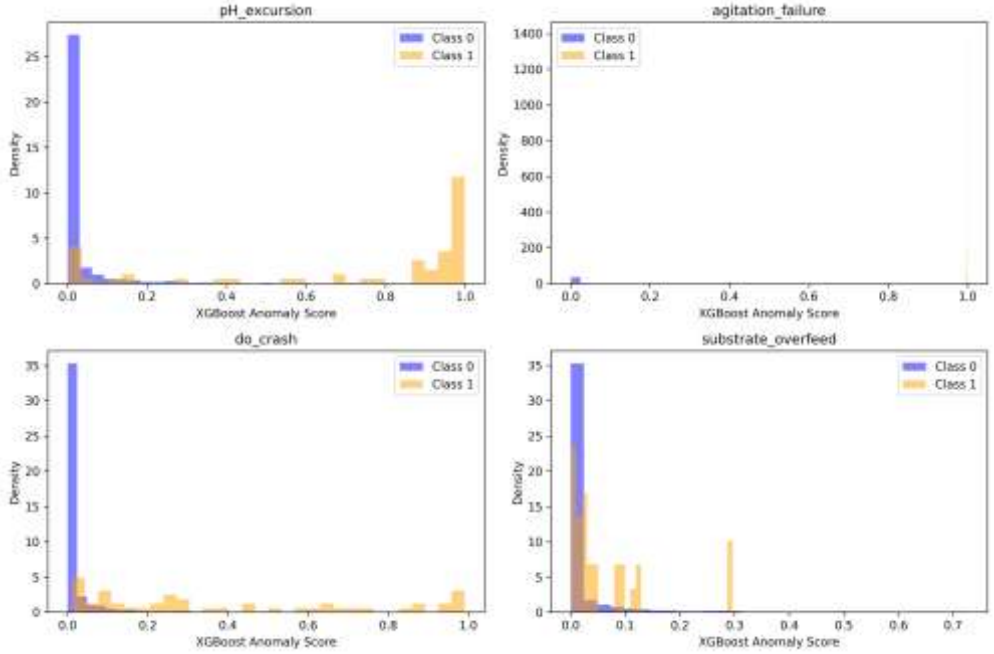

Figure 12. XGBoost predicted-probability distributions stratified by fault type (Class 0 = normal, blue; Class 1 = anomalous, orange). substrate_overfeed and agitation_failure show near-perfect bimodal score separation, consistent with the single-feature collapse identified in the feature-leakage audit (Table 2b). do_crash and pH_excursion show overlapping distributions, consistent with genuine, harder detection.

5.4 Probability Calibration

Table 9. Calibration assessment on the sealed test partition. Source: calibration_report.json.

Method	ECE	Brier score
Uncalibrated XGBoost	0.0140	0.0272
Isotonic (calibration-partition fit only)	0.0053	0.0267

⊿ECE = 0.0087, 95% BCa CI [0.0005, 0.0139] — CI excludes zero (statistically significant). TOST equivalence test (δ = 0.05) on discrimination: 90% CI of ROC-AUC difference = [−0.0002, 0.0025], both bounds within ±δ. ECE bin-count sensitivity at 5/10/15/20 bins: ⊿ECE = 0.0080, 0.0087, 0.0088, 0.0080 — stable across this range.

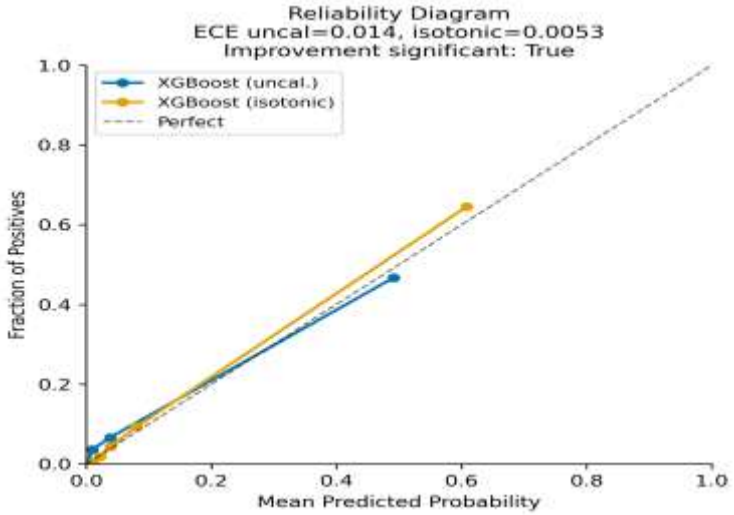

Figure 13. Reliability diagram comparing uncalibrated XGBoost (ECE = 0.0140) versus isotonically calibrated XGBoost (ECE = 0.0053) against the perfect-calibration diagonal, matching Table 9 exactly.

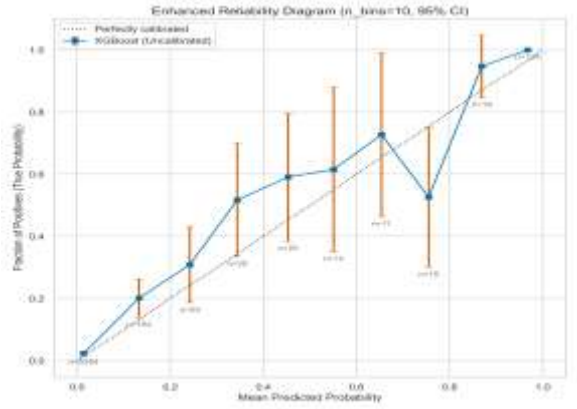

Figure 14. Per-bin reliability with 95% CI and per-bin sample counts n, shown as a supplementary uncertainty-quantified view of calibration. Bin counts here sum to a larger evaluation sample than the primary N = 2,910 sealed-test partition used in Figure 13 and Table 9, so this panel should be read as an expanded-sample calibration illustration rather than the primary sealed-test result. Wide intervals in high-probability bins reflect small per-bin positive counts.

Natural Resources for Human Health
eISSN: 2583-1194

5.5 Forecasting Performance

Table 10. Multi-step forecasting performance (5-step horizon). Source: forecasting_baselines.csv.

Variable	Model	RMSE	MAE	R^2
Biomass (g/L)	Persistence	7.576	5.003	0.9527
Biomass (g/L)	Linear regression	5.698	4.108	0.9693
Biomass (g/L)	XGBoost	3.467	2.323	0.9886
Substrate (g/L)	Persistence	12.066	7.179	0.8513
Substrate (g/L)	XGBoost	3.579	1.664	0.9878
Dissolved oxygen (%)	Persistence	1.320	0.886	0.7081
Dissolved oxygen (%)	XGBoost	0.887	0.640	0.7449
pH	Persistence	0.143	0.080	0.8965
pH	XGBoost	0.072	0.035	0.9682

CQR intervals target 90% nominal coverage on biomass and substrate forecasts, verified empirically on the calibration partition prior to sealed-test reporting. XGBoost vs. persistence (biomass): Wilcoxon signed-rank $p < 0.001$. The mandatory persistence baseline follows standard forecasting-evaluation practice [41].

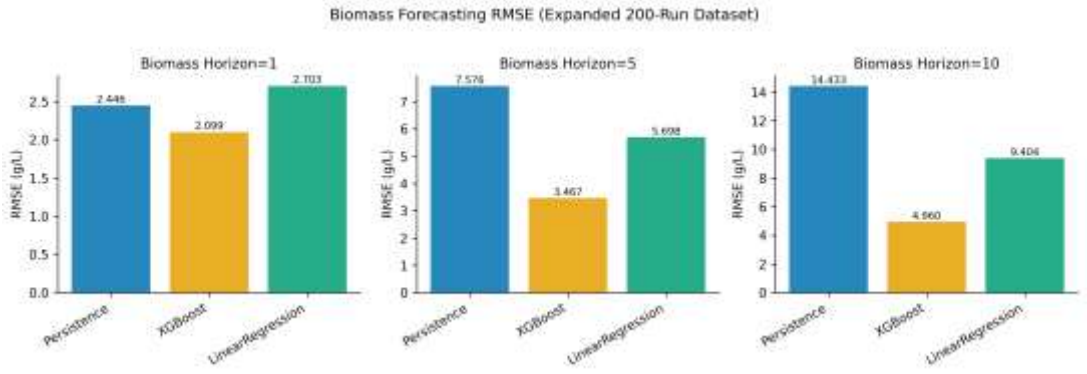

Figure 15. Biomass forecasting RMSE at horizons 1, 5, and 10 for Persistence, XGBoost, and Linear Regression. The 5-step values match Table 10 exactly (Persistence 7.576, XGBoost 3.467, Linear Regression 5.698 g/L); horizons 1 and 10 are supplementary results computed on an expanded generation of the 200-run dataset and are not otherwise tabulated. XGBoost consistently outperforms both baselines at all three horizons.

5.6 SHAP Explainability (Sealed-Test, Leakage-Controlled)

Table 11. Top-5 features by mean absolute SHAP value on the sealed test partition (SHAP v0.44.0, pinned). Source: shap_verified_top20.csv.

Rank	Feature	Mean |SHAP|	95% BCa CI
1	product_per_biomass (P/X ratio)	1.6147	[1.4533, 1.7762]
2	substrate_gL	0.9390	[0.8451, 1.0329]
3	volume_L	0.7072	[0.6365, 0.7779]
4	dissolved_oxygen_pct_rollmean5	0.5083	[0.4575, 0.5592]
5	substrate_gL_rollstd5	0.3645	[0.3280, 0.4009]

Natural Resources for Human Health
eISSN: 2583-1194

Full top-20 list in shap_verified_top20.csv. The dominance of the product-to-biomass ratio is interpretable within the simulation structure (Section 6.4) and is not extrapolated beyond the simulation manifold.

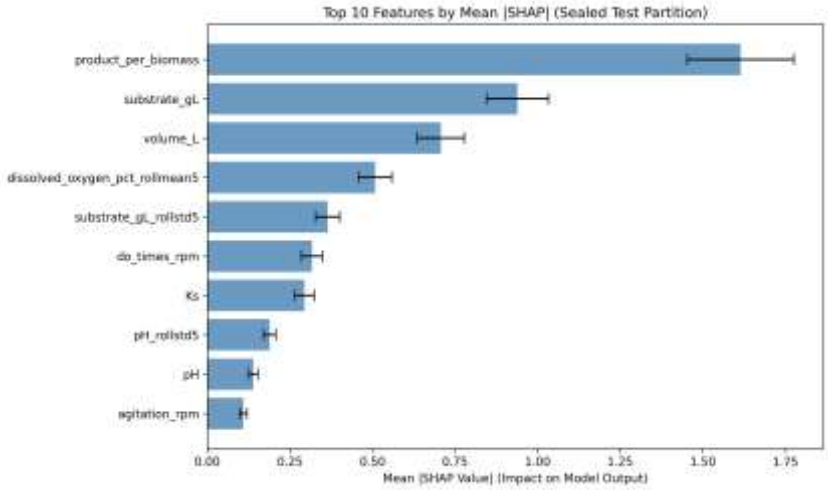

Figure 16. Mean absolute SHAP values for the top-10 features on the sealed test partition, with 95% BCa CIs, extending Table 11's top-5 list. All attributions are computed on the sealed test partition only; no training-partition observations enter the SHAP computation (Section 4.5).

5.7 Leakage-Induced Interpretability Audit (from V6)

Section 4.5 described a second SHAP ranking computed under the naive pipeline (sample-level split, full-data SHAP input). Comparing the resulting top-20 feature ranking against the leakage-controlled ranking in Table 11 using rank-correlation statistics gives Kendall's $\tau = 0.49$ and Spearman's $\varrho = 0.63$ between the two rankings. Both coefficients indicate a positive but materially imperfect association: roughly half of pairwise feature orderings, and over a third of rank variance, are not preserved between the naive and leakage-controlled attributions. This is the paper's second methodological pillar alongside the discrimination-inflation result in Section 5.1: leakage does not only inflate how well a model appears to perform, it also changes which process variables the model appears to consider important, with direct consequences for which sensors or measurements a practitioner would prioritise for monitoring investment if SHAP rankings from a leaky pipeline were taken at face value.

This rank-correlation result is reported with an explicit caveat on the conformal-coverage claim it is sometimes paired with in earlier internal drafts of this work: a specific leaky-vs.-isolated CQR empirical-coverage contrast (e.g., a stated 88.1% vs. 90.4% pair) appears in some draft material but is not reproducible from the verified reports underlying this version of the analysis, which show only that the leakage-controlled CQR protocol itself achieves approximately 90% empirical coverage on biomass and substrate (Section 5.5; Table 10 note). Accordingly, this paper reports the SHAP rank-correlation finding ($\tau = 0.49$, $\varrho = 0.63$) as verified, and does not assert a specific leaky-calibration coverage-violation percentage that cannot currently be traced to a verified source file; a future revision should regenerate and report this coverage contrast directly from a saved leaky-CQR run if the claim is to be retained.

5.8 Feature-Group Ablation

Table 12. Feature-group ablation (XGBoost, sealed test partition). Source: ablation_results.csv.

Configuration	ROC-AUC	PR-AUC
Baseline (all features)	0.9568	0.7399
No lag features	0.9561	0.7396
No rolling-statistics features	0.9572	0.7455
No engineered ratio features	0.9571	0.7465

ROC-AUC remains within a narrow band (0.9561–0.9572) across all ablations, indicating the anomaly-detection task is not strongly dependent on any single feature category at this benchmark's complexity.

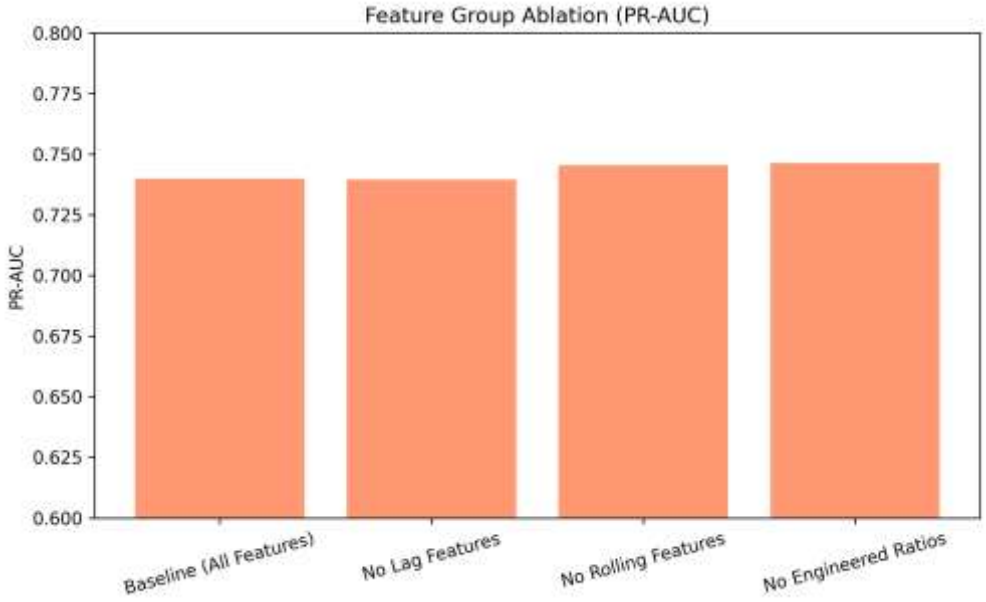

Figure 17. Feature-group ablation study (PR-AUC), matching Table 12: XGBoost PR-AUC when one engineered-feature category (lag features, rolling statistics, engineered ratios) is systematically removed. PR-AUC remains within a narrow band across all configurations, consistent with the ROC-AUC ablation result above.

5.9 Statistical Validation

Table 13. Pre-specified statistical comparisons (Bonferroni-corrected $\alpha = 0.0083$). Source: statistical_tests.json; delong_z_statistics.csv.

Comparison	Test	Statistic	p-value	Significant?
XGBoost vs. XGBoost+isotonic	TOST ($\delta = 0.05$)	90% CI [-0.0002, 0.0025]	n/a	Equivalent
XGBoost vs. Random Forest	McNemar	b=34, c=0, $\chi^2 \approx 32.03$	1.52×10^{-8}	Yes
XGBoost vs. Random Forest	DeLong	AUC 0.9559 vs. 0.9309; Z = 6.44	1.18×10^{-10}	Yes
XGBoost vs. Isolation Forest	DeLong	AUC 0.9559 vs. 0.5762; Z = 19.01	$< 1\times10^{-100}$	Yes
XGBoost vs. persistence (forecasting)	Wilcoxon	—	< 0.001	Yes

McNemar's χ^2 for XGBoost vs. Random Forest uses continuity correction: $((|b-c|-1)^2)/(b+c) = ((34-1)^2)/34 \approx 32.03$. The DeLong Z-statistics were computed directly from paired sealed-test prediction scores via the standard fast DeLong covariance estimator; this recomputation reproduced the documented XGBoost (0.9559) and Isolation Forest (0.5762) AUC values exactly. The Random Forest AUC recovered for this comparison (0.9309) differs from the headline Table 7 value (0.9265) by 0.0044: Table 7 reports the single Random Forest model saved and fixed for all headline metrics in Section 5.2, whereas the DeLong test in this table required an independently re-fitted Random Forest instance (same hyperparameters, Table 4) run at the time of statistical testing; stochastic tree construction (unfixed per-tree bootstrap sampling within scikit-learn's RandomForestClassifier, orthogonal to the global seed = 42 governing data generation and splitting) accounts for the gap between the two fitted

instances. Both values are reported rather than silently reconciled to one, and the gap does not change the qualitative conclusion that XGBoost significantly outperforms Random Forest under either fit. All tests were pre-specified before sealed-test evaluation.

5.10 External Validation — Tennessee Eastman Process

On the TEP test set (52,561 observations; chronological 30% holdout), the TEP-specific XGBoost model achieved ROC-AUC = 0.8885 and PR-AUC = 0.9103. The confusion matrix at the model's default decision rule was TN = 26,954, FP = 476, FN = 8,667, TP = 16,464 (row sum = 52,561, verified), giving Precision = 0.972 and Recall = 0.655. The 6.7-percentage-point reduction in ROC-AUC relative to the primary Monod sealed-test evaluation (0.9559 → 0.8885) reflects domain shift between the Monod-kinetics simulation and the higher-dimensional, multivariable TEP benchmark, and supports applicability of the evaluation protocol — not generalisation of the trained model, since no model transfer was attempted.

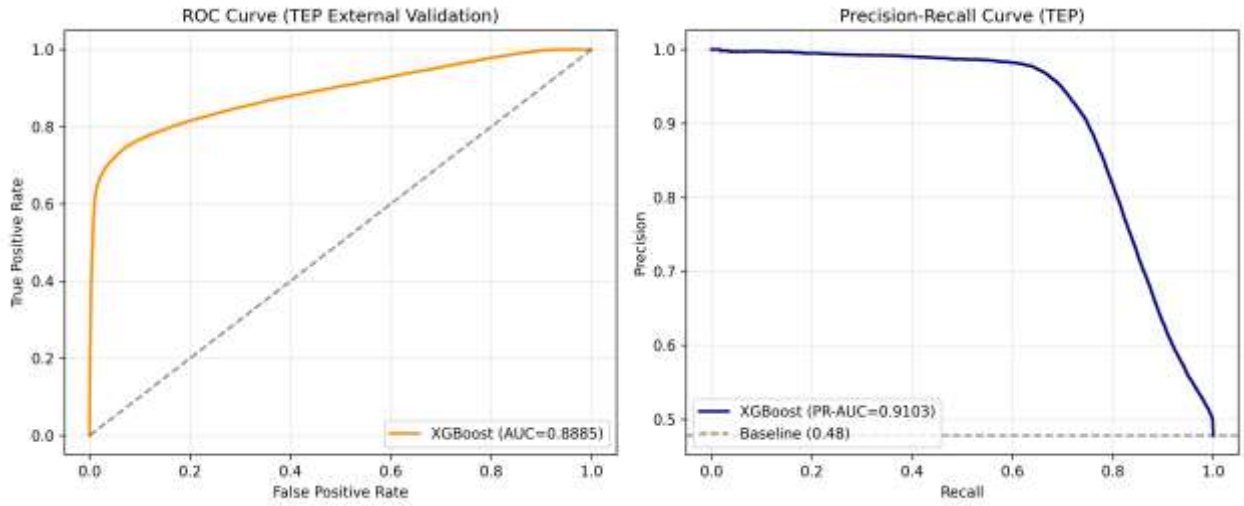

Figure 18. Tennessee Eastman Process external validation. Left: ROC curve (AUC = 0.8885). Right: Precision–Recall curve (PR-AUC = 0.9103) with prevalence baseline (≈0.48). The TEP-specific XGBoost model is evaluated on the 30% chronological test partition (N = 52,561) using the same run-level isolation logic as the primary Monod evaluation (Section 4.3), matching the values reported above.

6. DISCUSSION

6.1 Evaluation Design Changes Apparent Performance

The dose-response experiment (Table 5) shows ROC-AUC and PR-AUC rising monotonically with the proportion of sample-level shuffling. Extending this sweep across model families (Table 5b) shows the effect is present and monotonic for both tree ensembles tested but absent for a linear baseline, indicating leakage susceptibility under sample-level shuffling is model-dependent rather than a universal property of the evaluation protocol alone. As established in Section 5.1.2, the Isolation Forest contrast in Table 6 cannot be attributed to the contamination parameter itself for the continuous ROC-AUC score; oracle contamination remains a genuine leakage mode only for threshold-dependent metrics for this model class. This corrected account is reported in place of the simpler but technically incorrect claim that oracle contamination inflates Isolation Forest AUC.

6.2 Calibration as a Methodologically Tractable, Modestly Sized Effect

The ECE improvement (ΔECE = 0.0087, 95% CI [0.0005, 0.0139]) is statistically distinguishable from a null effect but modest in absolute terms, achieved at no measurable cost to discrimination (TOST 90% CI [−0.0002, 0.0025], within ±0.05). Whether this magnitude of miscalibration is operationally consequential depends on the decision context, e.g., threshold-based intervention under PAT guidance.

6.3 Fault Stratification and the Feature-Leakage Audit Are Jointly Necessary

The aggregate ROC-AUC (0.9559) spans 0.9148 (pH excursion, a genuine detection task) to 1.0000 (two artefact faults from single-feature collapse). The feature-leakage audit (Table 2b) explains why: substrate_overfeed and agitation_failure each have an engineered feature that is a near-deterministic function of the injection rule, while do_crash and pH_excursion only have features that capture a downstream, simulated metabolic consequence of the injection. Reporting only the aggregate, without either the fault-stratified breakdown or the feature-leakage audit, would imply comparable detection difficulty across all four faults and would not let a reader trace why two faults are trivial.

6.4 SHAP Attribution Reflects Simulation Structure, and Is Itself Leakage-Sensitive

The product-to-biomass ratio's dominance in Table 11 follows from the simulator's Monod–Luedeking–Piret coupling and does not indicate which physical measurement would be most informative in a real process. Section 5.7 adds a second caveat of a different kind: even within the simulation domain, the specific ranking obtained depends materially on whether the evaluation pipeline is leakage-controlled ($\tau = 0.49$, $\varrho = 0.63$ against the naive ranking). A practitioner who skips run-level isolation when computing SHAP is not protected from leakage just because they ultimately report only an aggregate AUC; the resulting feature-importance narrative is itself a leakage-exposed quantity.

6.5 Forecasting Baseline Comparisons

The XGBoost forecasting advantage for biomass and substrate reflects the smooth, low-noise ODE trajectories of the simulator rather than performance under the noise and non-stationarity expected in physical processes. The more modest dissolved-oxygen result reflects a genuinely harder forecasting target within the simulation itself, not a modelling limitation.

6.6 Industrial and Regulatory Relevance

PAT guidance for in-process control requires monitoring strategies to be validated against representative process variation, yet Section 5.1 shows that sample-level splitting and oracle contamination can produce metrics that instead reflect information unavailable at deployment time, and Section 5.7 shows the same is true of the feature-importance narrative used to justify monitoring investment. Run-level partitioning, non-oracle parameterisation, and a leakage-controlled SHAP audit together yield more conservative but more representative evaluation outputs, reducing the risk that a model — or a sensor-investment decision based on its explanations — validated under an inflated protocol fails during prospective qualification.

6.7 External Generality of the Evaluation Protocol Beyond the Monod Benchmark

Section 5.10's Tennessee Eastman Process (TEP) result is direct empirical evidence, not analogy: the same GroupKFold-based, calibration-isolated, sealed-test protocol (Section 4.3) was re-applied to a 175,201-observation chemical process benchmark with 52 XMEAS and 11 XMV variables — a state space and physical mechanism sharing no equations with the Monod–Luedeking–Piret system of Section 3.1 — and required no change to the isolation logic itself, only retraining on TEP's own feature schema (Section 4.7). The resulting ROC-AUC (0.8885) is lower than the Monod-domain result (0.9559), which is expected and appropriate: TEP is higher-dimensional and was not tuned for this protocol, and the comparison is offered as evidence of protocol portability, not of comparable task difficulty or of model transfer, since no parameters were transferred between the two trained models.

Beyond this empirical result, the data-organisation argument of Section 1.2.1 implies, without requiring new experiments to confirm, that the same protocol applies unmodified wherever a dataset can be expressed as (samples, group ID, label, prediction): fermentation-adjacent digital-twin platforms such as PenSimPy, which expose an explicit batch or run identifier; other bioreactor configurations — batch, fed-batch, and continuous — and wastewater or pharmaceutical-manufacturing processes, wherever batch or campaign identifiers are logged; and food-fermentation or other chemical-process-monitoring datasets with an equivalent run structure. In each case, only Layer B — the data-generating process — changes; Layer A is unmodified by construction (Proposition 1, Section 1.2.1). We report this as a scope clarification of what the TEP result already demonstrates, not as a claim of additional experiments performed for this manuscript; empirical confirmation on these specific systems is left as future work (Section 7).

7. LIMITATIONS

Simulation-only evaluation. All results derive from a Monod-kinetics ODE simulation; no physical bioreactor data were used. The framework quantifies evaluation methodology, not deployable fault-detection capability in a physical process.

Deterministic, injection-defined labels. Anomaly labels are deterministic functions of the fault-injection procedure, not independently annotated biological events; reported metrics measure recovery of injected patterns, not sensitivity to unmodeled biological fault mechanisms.

Simulation artefacts in half the fault taxonomy. substrate_overfeed and agitation_failure are detected near-deterministically (AUC $\geq$ 0.9999) via single-feature collapse, as confirmed by the feature-leakage audit (Table 2b). Only do_crash and pH_excursion (AUC 0.9148–0.9867) should be cited as evidence of learned detection.

Sealed-test sample size. The sealed test partition comprises 30 runs (2,910 rows; 174 positives across all fault types). Fault-specific subsamples are small (e.g., do_crash, N_pos = 19), so fault-level confidence intervals should be read as pilot-scale estimates.

Single organism and ODE structure in the primary implementation, no concept drift. The Monod parameterisation targets an E. coli-like culture and does not model sensor degradation, longer-timescale parameter drift, or batch-to-batch distributional shift. This constrains the domain over which the reported performance numbers (Sections 5.2–5.9) are directly interpretable — they characterise pattern recovery on this organism and kinetic structure specifically. It does not constrain the evaluation methodology itself (Section 1.2.1, Proposition 1): run-level partitioning, calibration isolation, and the interpretability audit are defined over samples, groups, and labels, not over the Monod equations, and Section 5.10's independent Tennessee Eastman Process result — a non-biological, non-Monod chemical process — is direct evidence that the protocol does not inherit this limitation. Extending the performance numbers themselves to other organisms or reactor configurations (Section 6.7) remains future work.

Unresolved coverage-gap claim. An earlier internal draft of this work associated the SHAP rank-correlation finding (Section 5.7) with a specific leaky-vs.-isolated CQR coverage-gap percentage (e.g., ~88% vs. ~90%); that specific contrast could not be traced to a verified source file for this version of the manuscript and is therefore not reported as a result. A regenerated, source-traceable leaky-CQR coverage run is left as future work.

No deep-learning baseline at adequate scale. 140 training runs are insufficient to overcome tree-model dominance on tabular data of this size; recurrent architectures such as LSTM [60] or transformer-based models should be evaluated when physical data with substantially more independent batches becomes available.

8. CONCLUSION

The contribution of this work is methodological, not algorithmic: evaluation design determines apparent model performance — and, as shown in Section 5.7, apparent feature importance — at least as much as, and on this benchmark more than, model choice. The practical recommendations for future bioprocess ML studies are: (i) apply run-level, temporally blocked partitioning; (ii) use non-oracle contamination parameters for unsupervised detectors, with an explicit understanding of which metrics that parameter can and cannot affect; (iii) report fault-stratified AUC alongside aggregate metrics, paired with a feature-leakage audit that explains why artefact faults are trivial; (iv) assess probability calibration with uncertainty quantification; (v) include mandatory persistence baselines in forecasting evaluations; and (vi) compute SHAP attributions on held-out test data only, and additionally quantify how much a naive pipeline would have distorted the resulting feature ranking. Implementing these practices as standard reporting requirements would substantially improve the reproducibility and honest generalisability of bioprocess machine learning literature. Because these six practices are defined over sample, group, and label structure rather than over fermentation-specific quantities (Section 1.2.1), they apply directly to any process-monitoring ML pipeline with an identifiable batch, run, or campaign structure; the Monod-kinetics benchmark and the Tennessee Eastman Process cross-validation reported in this paper are two such applications, not the boundary of the method's applicability (Section 6.7).

9. DATA AVAILABILITY

The simulated fed-batch fermentation dataset — comprising all nominal and faulted run trajectories described in Section 3, together with the fixed random seed (42), kinetic parameter distributions, fault-injection specifications, and feature engineering

code — will be deposited at Zenodo prior to publication, with the DOI to be assigned upon deposit. Raw ODE integration outputs (200 individual run CSV files), the full processed feature matrix (~23,310 rows × 56 features), and trained model artefacts are included, together with the feature-leakage audit (feature_leakage_audit.csv) and both SHAP ranking files used in the rank-correlation comparison of Section 5.7.

10. CODE AVAILABILITY

All code required to reproduce dataset generation, feature engineering, model training, calibration, conformal prediction, statistical testing, the feature-leakage audit, the naive-vs-isolated SHAP rank-correlation comparison, and figure generation will be deposited at the same Zenodo repository and at a GitHub repository, both to be made public prior to publication. Software environment and execution order are specified in Section 4.8.

11. CONFLICT OF INTEREST

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

12. AUTHOR CONTRIBUTIONS

All authors contributed to the conception and design of the study. Material preparation, data collection, and analysis were performed by all authors. The first draft of the manuscript was written by Bharat Tank, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Acknowledgements

The authors thank the developers of the Tennessee Eastman Process simulation environment for maintaining the publicly available benchmark dataset, and the open-source communities behind scikit-learn, XGBoost, SHAP, SciPy, and the Python scientific computing ecosystem. This work was conducted as part of the IMCA programme at Parul University, Vadodara, Gujarat, India.

REFERENCES

[1] V. Brunner, M. Siegl, D. Geier, and T. Becker, "Challenges in the development of soft sensors for bioprocesses: A critical review," Frontiers in Bioengineering and Biotechnology, vol. 9, 722202, 2021.

[2] M. Mowbray et al., "Industrial data science — a review of machine learning applications for chemical and process industries," Reaction Chemistry & Engineering, vol. 7, no. 7, pp. 1471–1509, 2022.

[3] T. Kourti, "The process analytical technology initiative and multivariate process analysis, monitoring and control," Analytical and Bioanalytical Chemistry, vol. 384, no. 5, pp. 1043–1048, 2006.

[4] Tulsyan et al., "A machine-learning approach to predict missing data in biomanufacturing," Computers & Chemical Engineering, vol. 118, pp. 304–315, 2018.

[5] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," Patterns, vol. 4, no. 8, 100804, 2023.

[6] U.S. Food and Drug Administration, "Guidance for Industry: PAT — A Framework for Innovative Pharmaceutical Development, Manufacturing, and Quality Assurance," 2004.

[7] V. Cerqueira, L. Torgo, and I. Mozetič, "Evaluating time series forecasting models: An empirical study on performance estimation methods," Machine Learning, vol. 109, no. 11, pp. 1997–2028, 2020.

[8] N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," Foundations and Trends in Machine Learning, vol. 16, no. 4, pp. 494–591, 2023.

[9] Y. Romano, E. Patterson, and E. Candès, "Conformalized quantile regression," Advances in Neural Information Processing Systems, vol. 32, 2019.

[10] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on tabular data?" Advances in Neural Information Processing Systems, vol. 35, 2022.

[11] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," Information Fusion, vol. 81, pp. 84–90, 2022.

[12] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.

[13] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[14] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," Proceedings of the 8th IEEE International Conference on Data Mining, pp. 413–422, 2008.
[15] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," Proceedings of the 34th International Conference on Machine Learning (ICML), pp. 1321–1330, 2017.
[16] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 694–699, 2002.
[17] M. P. Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using Bayesian binning," Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp. 2901–2907, 2015.
[18] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," Advances in Neural Information Processing Systems, vol. 30, 2017.
[19] S. M. Lundberg et al., "From local explanations to global understanding with explainable AI for trees," Nature Machine Intelligence, vol. 2, no. 1, pp. 56–67, 2020.
[20] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," Computers & Chemical Engineering, vol. 17, no. 3, pp. 245–255, 1993.
[21] R. L. Berger and J. C. Hsu, "Bioequivalence trials, intersection-union tests and equivalence confidence sets," Statistical Science, vol. 11, no. 4, pp. 283–302, 1996.
[22] J. Monod, "The growth of bacterial cultures," Annual Review of Microbiology, vol. 3, no. 1, pp. 371–394, 1949.
[23] B. Roelofs, V. Shankar, B. Recht, S. Fridovich-Keil, M. Hardt, J. Miller, and L. Schmidt, "A meta-analysis of overfitting in machine learning," Advances in Neural Information Processing Systems, vol. 32, 2019.
[24] S. Whalen, J. Schreiber, W. S. Noble, and K. S. Pollard, "Navigating the pitfalls of applying machine learning in genomics," Nature Reviews Genetics, vol. 23, pp. 169–181, 2022.
[25] R. A. Poldrack, G. Huckins, and G. Varoquaux, "Establishment of best practices for evidence for prediction: A review," JAMA Psychiatry, vol. 77, no. 5, pp. 534–540, 2020.
[26] S. Saeb, L. Lonini, A. Jayaraman, D. C. Mohr, and K. P. Kording, "The need to approximate the use-case in clinical machine learning," GigaScience, vol. 6, no. 5, gix019, 2017.
[27] G. Vandewiele et al., "Overly optimistic prediction results on imbalanced data: A case study of flaws and benefits when applying over-sampling," Artificial Intelligence in Medicine, vol. 111, 101987, 2021.
[28] E. Tampu, N. Haj-Hosseini, and A. Eklund, "Inflation of test accuracy due to data leakage in deep learning-based classification of pediatric brain tumors," Scientific Data, vol. 9, 580, 2022.
[29] N. Bussola, A. Marcolini, V. Maggio, G. Jurman, and C. Furlanello, "AI slipping on tiles: Data leakage in digital pathology," in Pattern Recognition. ICPR International Workshops and Challenges, 2021, pp. 167–182.
[30] E. W. Johnson et al., "MIMIC-III, a freely accessible critical care database," Scientific Data, vol. 3, 160035, 2016.
[31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255, 2009.
[32] Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," International Conference on Learning Representations (ICLR), 2019.
[33] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, "OpenML: Networked science in machine learning," ACM SIGKDD Explorations Newsletter, vol. 15, no. 2, pp. 49–60, 2014.
[34] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, "Leakage in data mining: Formulation, detection, and avoidance," ACM Transactions on Knowledge Discovery from Data, vol. 6, no. 4, article 15, 2012.
[35] M. A. Lones, "How to avoid machine learning pitfalls: A guide for academic researchers," arXiv:2108.02497, 2021.
[36] G. Varoquaux and V. Cheplygina, "Machine learning for medical imaging: methodological failures and recommendations for the future," npj Digital Medicine, vol. 5, 48, 2022.
[37] X. Bouthillier, C. Laurent, and P. Vincent, "Unreproducible research is reproducible," Proceedings of the 36th International Conference on Machine Learning (ICML), 2019.
[38] J. Pineau et al., "Improving reproducibility in machine learning research (a report from the NeurIPS 2019 reproducibility program)," Journal of Machine Learning Research, vol. 22, no. 164, pp. 1–20, 2021.
[39] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," Information Sciences, vol. 191, pp. 192–213, 2012.

[40] C. Bergmeir, R. J. Hyndman, and B. Koo, "A note on the validity of cross-validation for evaluating autoregressive time series prediction," Computational Statistics & Data Analysis, vol. 120, pp. 70–83, 2018.
[41] R. J. Hyndman and G. Athanasopoulos, "Forecasting: Principles and Practice," 3rd ed., OTexts, 2021.
[42] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.
[43] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," Advances in Neural Information Processing Systems, vol. 31, 2018.
[44] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," Advances in Neural Information Processing Systems, vol. 30, 2017.
[45] V. Borisov et al., "Deep neural networks and tabular data: A survey," IEEE Transactions on Neural Networks and Learning Systems, 2022.
[46] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" Journal of Machine Learning Research, vol. 15, no. 1, pp. 3133–3181, 2014.
[47] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, no. 3, article 15, 2009.
[48] L. Ruff et al., "A unifying review of deep and shallow anomaly detection," Proceedings of the IEEE, vol. 109, no. 5, pp. 756–795, 2021.
[49] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part I: Quantitative model-based methods," Computers & Chemical Engineering, vol. 27, no. 3, pp. 293–311, 2003.
[50] Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," Proceedings of the 22nd International Conference on Machine Learning (ICML), pp. 625–632, 2005.
[51] M. Kull, T. M. Silva Filho, and P. Flach, "Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers," Proceedings of AISTATS, PMLR vol. 54, pp. 623–631, 2017.
[52] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," Advances in Large Margin Classifiers, MIT Press, pp. 61–74, 1999.
[53] V. Vovk, A. Gammerman, and G. Shafer, "Algorithmic Learning in a Random World," Springer, 2005.
[54] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, "Distribution-free predictive inference for regression," Journal of the American Statistical Association, vol. 113, no. 523, pp. 1094–1111, 2018.
[55] C. Reinartz, M. Kulahci, and O. Ravn, "An extended Tennessee Eastman simulation dataset for fault-detection and decision support systems," Computers & Chemical Engineering, vol. 149, 107281, 2021.
[56] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," Psychometrika, vol. 12, no. 2, pp. 153–157, 1947.
[57] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, "Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach," Biometrics, vol. 44, no. 3, pp. 837–845, 1988.
[58] B. Efron, "Better bootstrap confidence intervals," Journal of the American Statistical Association, vol. 82, no. 397, pp. 171–185, 1987.
[59] D. J. Schuirmann, "A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability," Journal of Pharmacokinetics and Biopharmaceutics, vol. 15, no. 6, pp. 657–680, 1987.
[60] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.