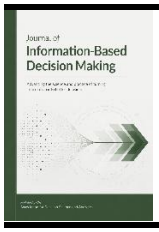
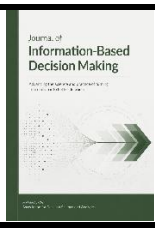


Journal of Information-Based Decision Making

Journal homepage: www.jibdm.journal-publishing.org
ISSN: XXXX-XXXX



Cost-Sensitive and Explainable Machine Learning for Predictive Maintenance Decision Support in Industry 4.0

Yavuz Selim Balcioglu[1,*]

[1] Department of Management Information, Business Administration Faculty, Dogus University, Istanbul, Türkiye

ARTICLE INFO

Article history:
Received 8 May 2026
Received in revised form 30 July 2026
Accepted 4 September 2026
Available online 6 September 2026

Keywords:
Predictive Maintenance; Explainable Artificial Intelligence; Class Imbalance; Cost-sensitive Learning; Probability Calibration; Gradient Boosting; Industry 4.0

ABSTRACT

Unplanned equipment failure remains a major controllable cost driver in discrete manufacturing, while Industry 4.0 sensor infrastructure has made data-driven predictive maintenance a practical alternative to reactive and calendar-based policies. However, published benchmarks are often dominated by threshold-free accuracy reporting, potentially misleading evaluation under severe class imbalance, routine use of synthetic resampling, arbitrary classification thresholds, and limited explanation of model predictions. This study develops and validates an integrated decision-support framework combining classifier selection, imbalance handling, probability calibration, decision-theoretic thresholding, and post-hoc explanation within a reproducible pipeline. The framework is evaluated on the AI4I 2020 predictive maintenance benchmark, containing 10,000 machining observations with a 3.39% failure rate. Four physics-informed descriptors are derived from raw signals, and eleven classifiers are evaluated using ten-fold stratified cross-validation with the area under the precision-recall curve (PR-AUC) as the primary criterion. Seven resampling schemes are compared with an unresampled baseline, with statistical differences assessed using the Friedman test followed by Holm-corrected Wilcoxon signed-rank comparisons. Gradient boosting achieves the best cross-validated performance (PR-AUC = 0.9145) and a hold-out PR-AUC of 0.8987 with a Matthews correlation coefficient of 0.888. It is statistically indistinguishable from random forest and LightGBM but superior to the remaining eight classifiers. None of the resampling schemes improves ranking quality, while all substantially reduce precision. Isotonic calibration reduces the Brier score from 0.00726 to 0.00654, while calibrated probabilities combined with a cost-minimizing threshold reduce expected misclassification cost by 37.2% relative to the default 0.50 threshold and by 87.0% relative to a run-to-failure policy. Shapley attributions identify tool wear, rotational speed, process-air temperature difference, and torque as the dominant drivers, consistent with the benchmark's documented physical failure mechanisms and supporting engineer-facing justification of individual alarms.

1. Introduction

Maintenance expenditure is one of the few large cost items in discrete manufacturing that remains genuinely controllable by better information. Reactive strategies repair equipment only after it has stopped, and therefore absorb the full cost of unplanned downtime, scrapped work-in-progress

* *Corresponding author.*
E-mail address: ysbalcioglu@dogus.edu.tr

and expedited spare parts; calendar-based preventive strategies avoid part of that exposure but pay for it by discarding useful component life and by taking healthy machines out of production. Predictive maintenance (PdM) occupies the middle ground: an intervention is scheduled only when condition data indicate that failure is imminent. The instrumentation that makes this possible embedded torque, speed, temperature and vibration sensing, together with the connectivity and storage of Industry 4.0 architectures is now standard on new machine tools, and the systematic reviews of Zonta*et al.,* [1] and Carvalho *et al.,* [2] document a corresponding growth in the volume of machine learning research addressing the problem.

The methodological core of most of this literature is a supervised classification problem: given a vector of process measurements, predict whether the machine is in a state that will lead to failure. Susto *et al.,* [3] framed this explicitly as a maintenance decision problem in semiconductor manufacturing, training multiple classifiers with different prediction horizons and selecting among them on the basis of expected operating cost. Matzka [4] released the AI4I 2020 benchmark specifically to give this community a public dataset with realistic failure physics, and it has since become one of the most widely used reference problems in the field, with subsequent work applying tree ensembles, neural networks and explanation methods to it [5, 6].

Three methodological difficulties recur across this body of work. The first is class imbalance. Failures are rare by construction—3.39% of observations in AI4I 2020—so a classifier that never raises an alarm already achieves 96.6% accuracy. He and Garcia [7] surveyed the resulting learning pathologies, and the standard remedy has been synthetic minority oversampling in the tradition of SMOTE [8] and its borderline [9] and adaptive [10] variants, often combined with cleaning steps such as Tomek-link or edited-nearest-neighbour removal [11]. Application papers frequently adopt one of these schemes without testing whether it helps, even though the comparative evidence is mixed and oversampling is known to distort the calibration of the resulting probability estimates.

The second difficulty is evaluation. Accuracy is uninformative at this prevalence, and the area under the receiver operating characteristic curve is only marginally better: because the false positive rate is normalised by a very large negative class, ROC curves remain visually and numerically optimistic even when the practical precision of the alarm is poor. Saito and Rehmsmeier [12] showed that the precision-recall curve is the more informative summary for imbalanced problems, and Chicco and Jurman [13] made the parallel argument for the Matthews correlation coefficient over the F1 score, since only the former uses all four cells of the confusion matrix. Fawcett [14] had already cautioned that a single threshold-free scalar cannot substitute for an explicit operating point.

The third difficulty is the gap between a probability and a maintenance action. A model output of 0.31 is not a decision; it becomes one only when compared with a threshold, and the near-universal default of 0.50 is optimal only when the two error types cost the same. In maintenance they never do: an undetected failure typically costs one to two orders of magnitude more than an unnecessary inspection. Converting probabilities into cost-optimal actions additionally requires those probabilities to be calibrated, which tree ensembles are generally not without post-processing [15]. Finally, an alarm that a maintenance engineer cannot interrogate will not be trusted on the shop floor, which is why Shapley-value attribution [16], local surrogate models [17] and the broader explainable artificial intelligence agenda [18] have become prominent in industrial deployments.

These three difficulties are usually addressed when they are addressed at all in isolation. Papers that compare classifiers rarely calibrate them; papers that apply resampling rarely test it against an unresampled baseline under a metric that is sensitive to precision; papers that report explanations rarely connect them to the cost structure of the decision they support. The gap addressed here is therefore not the absence of accurate classifiers for AI4I 2020, which are abundant, but the absence of an end-to-end account in which model selection, imbalance treatment, calibration, thresholding

and explanation are evaluated jointly under a single protocol, so that the marginal contribution of each stage to operational cost can be measured rather than assumed.

The significance of closing this gap is practical. If, as this study finds, the customary resampling step contributes nothing while the neglected calibration and thresholding steps contribute a 37% reduction in expected cost, then the effort allocated to building PdM systems is currently being spent in the wrong place. The objectives of the study follow directly. First, to establish which classifier family performs best on the AI4I 2020 benchmark under a precision-sensitive metric with formal statistical testing rather than point comparison. Second, to quantify the marginal contribution of seven resampling schemes against an unresampled baseline. Third, to measure how much of the achievable operational gain is delivered by probability calibration and decision-theoretic thresholding. Fourth, to establish whether the model's internal logic, recovered through Shapley attribution, is consistent with the documented failure physics of the benchmark and therefore defensible to an engineer.

2. Methodology

2.1 Dataset

The study uses the AI4I 2020 Predictive Maintenance Dataset introduced by Matzka [4] and archived in the UCI Machine Learning Repository under a Creative Commons Attribution 4.0 licence. The dataset is synthetic but is generated from the documented physics of a milling process, which is an advantage for the present purpose: the ground-truth failure mechanisms are known, so the plausibility of a model's explanations can be assessed against them rather than against intuition. It contains 10,000 observations and 14 columns, described in Table 1. Of these, 339 observations (3.39%) are labelled as machine failures, giving an imbalance ratio of approximately 28:1.

Table 1
Variables of the AI4I 2020 predictive maintenance dataset

Variable	Type	Unit / values	Role in this study
UDI	Integer	1–10,000	Identifier; removed
Product ID	String	L/M/H + serial	Identifier; removed
Type	Categorical	L, M, H	Predictor (quality variant)
Air temperature	Continuous	K	Predictor
Process temperature	Continuous	K	Predictor
Rotational speed	Integer	rpm	Predictor
Torque	Continuous	Nm	Predictor
Tool wear	Integer	min	Predictor
Machine failure	Binary	0 / 1	Target
TWF, HDF, PWF, OSF, RNF	Binary	0 / 1	Failure-mode flags; removed (leakage)

Note: TWF = tool wear failure; HDF = heat dissipation failure; PWF = power failure; OSF = overstrain failure; RNF = random failure.

The five failure-mode flags require particular care. Each records the specific mechanism responsible for an observed failure and is therefore available only after the event; retaining them would leak the target and inflate performance to a degree that has no operational meaning. They are removed from the predictor set and retained only for the consistency audit reported in Section 2.2 and for the interpretation of the explanations in Section 3.8.

2.2 Data Screening and Feature Engineering

Screening confirmed that the dataset contains no missing values and no duplicated records. An audit of label consistency, comparing the aggregate failure label with the disjunction of the five mode flags, identified 27 records (0.27%) in which the two disagree; these arise from the random-failure

mechanism of the generating process and are retained unmodified and treated as irreducible label noise, since removing them would alter the benchmark and overstate attainable performance.

Four descriptors were derived from the raw signals, listed in Table 2. Each corresponds to a physical quantity that the documented failure mechanisms of the benchmark depend on, rather than to a purely statistical transformation. Heat dissipation failure depends on the difference between process and air temperature in combination with rotational speed; power failure depends on the mechanical power delivered at the tool, which is the product of torque and angular velocity and is not recoverable by a tree from either factor alone without deep interaction splits; overstrain failure depends on the product of tool wear and torque. Making these quantities explicit converts interactions that a tree must approximate with many axis-aligned splits into single informative features.

Table 2
Physics-informed features derived from the raw process signals

Feature	Definition	Unit	Associated failure mechanism
TempDiff	$\Delta T = T_{process} - T_{air}$	K	Heat dissipation (HDF)
Power	$P = \tau \cdot 2\pi n / 60$	W	Power failure (PWF)
Strain	$S = w \cdot \tau$	min·Nm	Overstrain (OSF)
WearSpeed	$V = n \cdot w / 1000$	10^3 rpm·min	Tool wear (TWF)

Note: τ = torque (Nm); n = rotational speed (rpm); w = tool wear (min); T = temperature (K). The categorical quality variant is encoded ordinally as L = 0, M = 1, H = 2, reflecting its documented monotone relationship with product quality.

The final predictor matrix therefore contains ten features: the ordinally encoded quality variant, five raw process signals and the four derived descriptors.

2.3 Experimental Design

The experimental protocol is summarised in Figure 1. The dataset is first partitioned into a training set of 8,000 observations (271 failures) and a hold-out test set of 2,000 observations (68 failures) by stratified random sampling with a fixed seed. The test partition is used exactly once, for the final evaluation reported in Section 3.5 onwards; every selection decision algorithm choice, resampling scheme, hyper-parameters, calibration method and decision threshold is made using the training partition only, through cross-validation or out-of-fold prediction. This discipline is essential here, because the threshold optimisation of Section 2.7 is precisely the kind of operation that produces optimistic results when tuned on the data used to report them.

Model screening uses ten-fold stratified cross-validation repeated over the training partition, which preserves the 3.39% failure rate in every fold and yields ten performance estimates per model for the significance testing described in Section 2.9.

2.4 Learning Algorithms

Eleven classifiers spanning the principal families of supervised learning were screened: logistic regression, Gaussian naive Bayes, k-nearest neighbours, a support vector machine with a radial basis function kernel [19], a multilayer perceptron, a single decision tree, random forest [20], extremely randomised trees [21], gradient boosting [22], XGBoost [23] and LightGBM. Scale-sensitive learners were preceded by standardisation fitted within each training fold. Where the implementation supports it, class weights inversely proportional to class frequency were applied, and for XGBoost the equivalent positive-class scaling factor of 28.52 was used; this provides a cost-free imbalance correction independent of the resampling comparison in Section 2.6. The best-performing family was then refined by randomised search over 30 configurations of the space in Table 3, scored by five-fold cross-validated average precision.

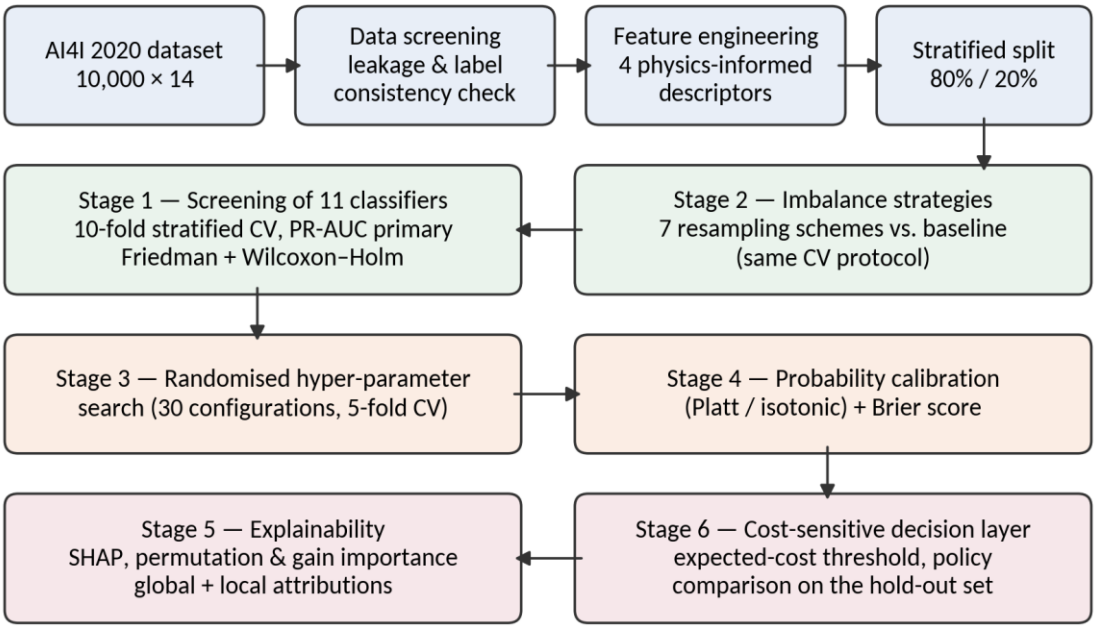


Fig. 1. Six-stage experimental framework linking data preparation, model screening, imbalance treatment, hyper-parameter search, calibration, explanation and the cost-sensitive decision layer

Table 3
Hyper-parameter search space for the gradient boosting classifier

Hyper-parameter	Candidate values
Number of estimators	150, 200, 300, 400
Learning rate	0.03, 0.05, 0.075, 0.10, 0.15
Maximum depth	2, 3, 4, 5
Minimum samples per leaf	1, 3, 5, 10, 20
Subsample fraction	0.7, 0.8, 0.9, 1.0
Maximum features per split	all, √k, 0.5k, 0.7k

Note: k denotes the number of predictors. 30 of the 1,600 possible configurations were sampled without replacement.

2.5 Performance metrics

Let TP, TN, FP and FN denote the four cells of the confusion matrix. Precision and recall are defined in Eqs. (1) and (2), and their harmonic mean in Eq. (3).

$$\text{Precision} = TP / (TP + FP) \tag{1}$$

$$\text{Recall} = TP / (TP + FN) \tag{2}$$

$$F1 = 2 \cdot \text{Precision} \cdot \text{Recall} / (\text{Precision} + \text{Recall}) \tag{3}$$

The Matthews correlation coefficient, Eq. (4), is reported as the principal single-threshold summary because it is the only common measure that uses all four cells and therefore cannot be inflated by a large true-negative count [13]. The geometric mean of recall and specificity, Eq. (5), summarises the balance between the two classes.

$$MCC = (TP \cdot TN - FP \cdot FN) / \sqrt{[(TP+FP)(TP+FN)(TN+FP)(TN+FN)]} \tag{4}$$

$$\text{G-mean} = \sqrt{(\text{Recall} \cdot \text{Specificity})} \tag{5}$$

The primary model-selection criterion is the area under the precision-recall curve, estimated by average precision as in Eq. (6), where P_n and R_n are precision and recall at the n-th threshold. Unlike ROC-AUC it does not reward performance on the majority class and is therefore the appropriate ranking metric at this prevalence [12, 14]. ROC-AUC and the Brier score are reported alongside it for comparability with the literature and for the calibration analysis respectively.

$$AP = \Sigma_n (R\_n - R_{n-1}) \cdot P\_n \quad (6)$$

2.6 Class Imbalance Strategies

Seven strategies were compared against an unresampled baseline under the identical cross-validation protocol: random oversampling of the minority class, SMOTE [8], borderline-SMOTE [9], ADASYN [10], the hybrid SMOTE-Tomek and SMOTE-ENN schemes [11], and random undersampling of the majority class. In every case resampling is applied inside the cross-validation loop, to the training folds only, so that no synthetic observation can influence a validation fold. Applying resampling before splitting is a common and severe source of optimistic bias in the applied literature, since synthetic minority points are interpolations of their neighbours and will otherwise appear on both sides of the split.

2.7 Probability Calibration and the Cost-Sensitive Decision Layer

Boosted trees optimise a ranking-sensitive loss and their raw scores are not reliable probability estimates; Platt scaling and isotonic regression, both fitted by internal five-fold cross-validation on the training partition, were therefore compared using the Brier score and reliability diagrams [15]. The maintenance decision is then modelled explicitly. Let C_{FP} be the cost of an unnecessary inspection and C_{FN} the cost of an undetected failure. The expected cost of operating the classifier at threshold t is given by Eq. (7), and for calibrated probabilities the cost-minimising threshold has the closed form of Eq. (8).

$$E[C(t)] = C_{FP} \cdot FP(t) + C_{FN} \cdot FN(t) \quad (7)$$

$$t^* = C_{FP} / (C_{FP} + C_{FN}) \quad (8)$$

A base case of C_{FP} = 250 and C_{FN} = 5,000 currency units is adopted, a 1:20 ratio consistent with the maintenance costing literature, in which an unplanned stoppage carries lost production, expedited parts and secondary damage while a false alarm carries only inspection labour. Because this ratio is plant-specific, a sensitivity analysis across ratios from 1:2 to 1:100 is reported in Section 3.7. Two threshold-selection routes are compared: the analytical threshold of Eq. (8), and an empirical threshold obtained by minimising Eq. (7) over out-of-fold training predictions. Both are then applied unchanged to the hold-out set, and both are benchmarked against a run-to-failure policy and an inspect-everything policy.

2.8 Explainability

Model behaviour was interrogated with three complementary tools. Shapley additive explanations were computed with the exact tree estimator of Lundberg *et al.,* [16], which decomposes each individual prediction as in Eq. (9), where ϕ_0 is the expected model output and ϕ_i is the contribution of feature i in log-odds units. Permutation importance measured on the hold-out set with 20 repetitions gives a model-agnostic view anchored in predictive loss, and the impurity-based gain of the fitted ensemble is reported for contrast. Reporting all three is deliberate: they answer different questions and disagree systematically in the presence of correlated features, as Section 3.8 demonstrates.

$$g(x) = \phi_0 + \Sigma_{i=1..k}\, \phi_i \quad (9)$$

2.9 Statistical Testing

Differences between classifiers were tested with the Friedman test on the ten cross-validated PR-AUC values per model, followed by Wilcoxon signed-rank comparisons between the best-ranked control model and each competitor, with Holm correction for multiple comparisons. Feature-level differences between failed and normal observations were tested with the Mann-Whitney U test, with the rank-biserial correlation reported as effect size, since none of the variables is normally distributed within class.

2.10 Reproducibility

All experiments were executed in Python 3.12 with scikit-learn 1.8.0, XGBoost 3.4.1, LightGBM 4.7.0, imbalanced-learn 0.14.2 and SHAP 0.52.0 on a single-core Linux environment. A fixed random seed of 42 governs the train-test split, every cross-validation partition, every stochastic learner and every resampling procedure, so that all reported figures are exactly reproducible from the public dataset.

3. Results

3.1 Exploratory Analysis

Figure 2 summarises the structure of the target. The 339 failures represent 3.39% of observations, and the failure rate declines monotonically with product quality, from 3.92% for the low-quality variant to 2.77% for medium and 2.09% for high quality. Heat dissipation is the most frequent mechanism (115 occurrences), followed by overstrain (98), power failure (95), tool wear (46) and the random mechanism (19); the mode counts sum to more than the failure total because a single event may trigger several mechanisms simultaneously.

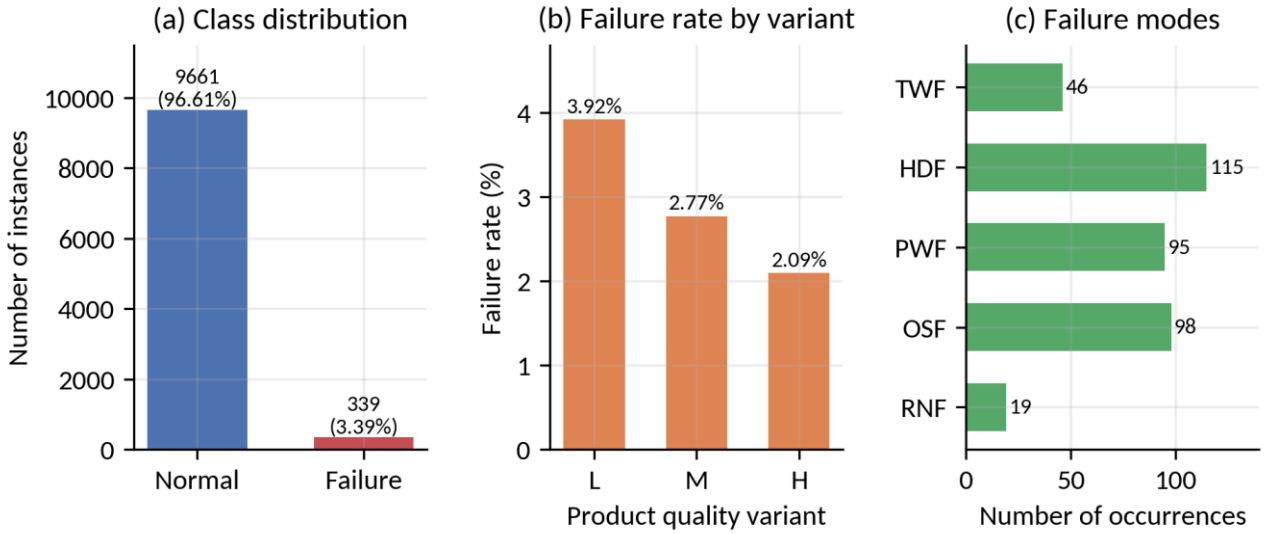


Fig. 2. Structure of the target variable: (a) class distribution, (b) failure rate by product quality variant, (c) frequency of the five documented failure mechanisms

Table 4 reports the distributional differences between the two classes. Every predictor differs significantly at the 0.1% level, but the effect sizes separate them sharply. Torque and rotational speed show the largest rank-biserial correlations (0.541 and 0.533), followed by the derived power descriptor (0.491) and the overstrain proxy (0.409). The two raw temperatures, by contrast, have

small effects (0.265 and 0.127) while their difference has a substantially larger one (0.325) and moves in the opposite direction: failures occur when the process-air gap narrows to a mean of 9.40 K against 10.02 K for normal operation, which is precisely the documented heat-dissipation condition. This is the first indication that the informative signal resides in derived quantities rather than in the raw channels.

Table 4
Distributional comparison of predictors between normal and failed observations

Feature	Normal, mean (SD)	Failure, mean (SD)	p	|r_rb|
Air temperature (K)	299.97 (1.99)	300.89 (2.07)	< 0.001	0.265
Process temperature (K)	310.00 (1.49)	310.29 (1.36)	< 0.001	0.127
Rotational speed (rpm)	1540.3 (167.4)	1496.5 (384.9)	< 0.001	0.533
Torque (Nm)	39.63 (9.47)	50.17 (16.37)	< 0.001	0.541
Tool wear (min)	106.7 (62.9)	143.8 (72.8)	< 0.001	0.324
TempDiff (K)	10.02 (0.99)	9.40 (1.16)	< 0.001	0.325
Power (W)	6244.5 (1011.4)	7282.8 (1851.1)	< 0.001	0.491
Strain (min·Nm)	4213.8 (2709.7)	7187.9 (4234.0)	< 0.001	0.409
WearSpeed (10^3 rpm·min)	164.5 (99.6)	212.1 (114.9)	< 0.001	0.250

Note: p-values from the Mann-Whitney U test; |r_rb| is the absolute rank-biserial correlation as effect size. n = 9,661 normal and 339 failed observations.

Figure 3 shows why the classification problem is not linearly separable. In the torque-speed plane the failures form two disjoint clusters at opposite corners—high torque with low speed, and low torque with high speed—which is the signature of the power-failure envelope acting on the product of the two variables rather than on either individually. The tool-wear plane shows the overstrain boundary as a hyperbolic frontier, and the third panel shows the heat-dissipation cluster at low temperature difference and low speed.

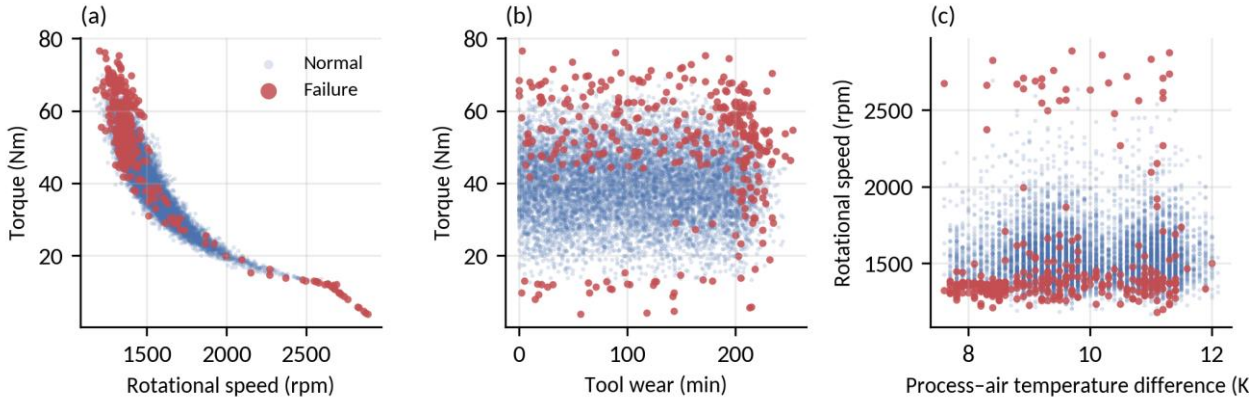


Fig. 3. Failure geometry in three projections of the feature space: (a) torque against rotational speed, (b) torque against tool wear, (c) rotational speed against the process-air temperature difference

Figure 4 reports Spearman correlations. Air and process temperature are almost perfectly collinear (ρ = 0.86) by construction, torque and rotational speed are strongly negatively related (ρ = −0.87), and the derived features inherit substantial correlation with their constituents. No individual feature correlates strongly with the target, the largest coefficient being 0.16 for torque, which is expected given that the failure conditions are conjunctions rather than marginal effects.

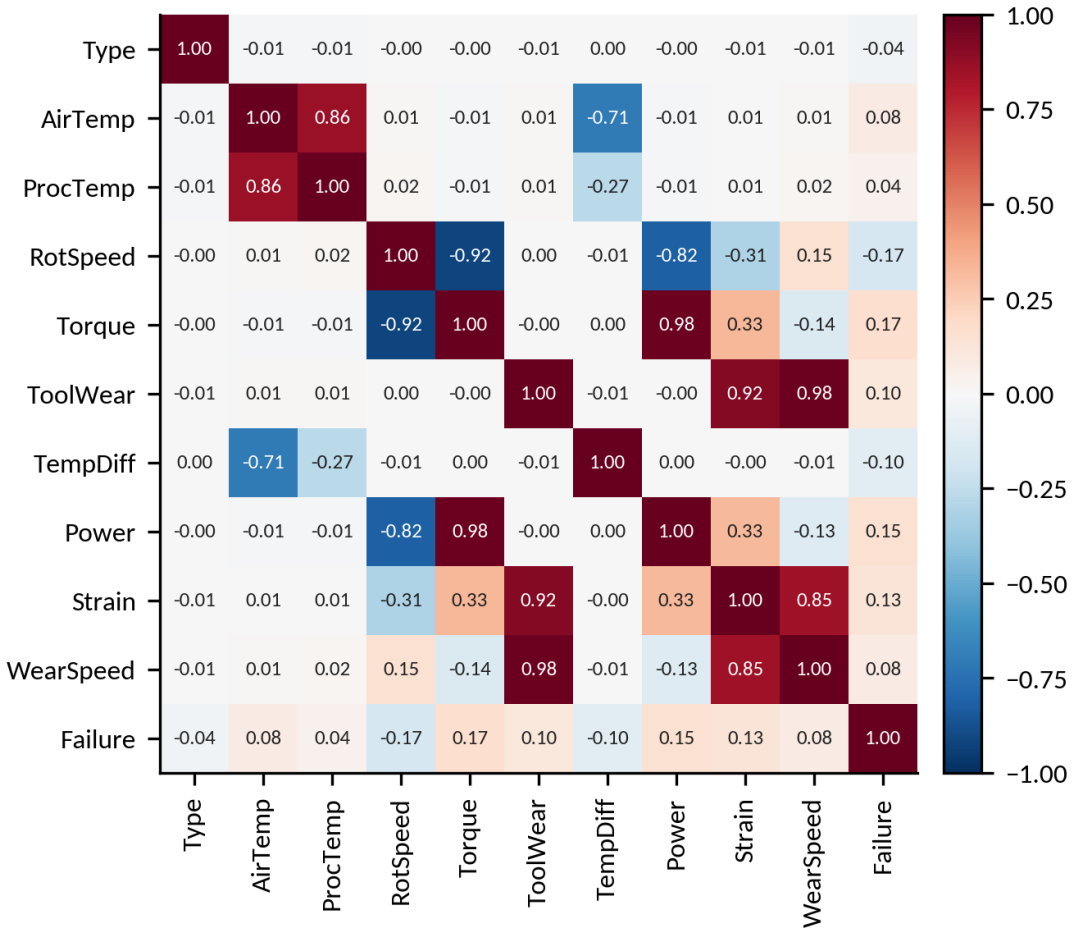


Fig. 4. Spearman rank correlation matrix of the predictor set and the target

3.2 Comparative Model Performance

Table 5 and Figure 5 report the ten-fold cross-validated performance of the eleven classifiers, ordered by the primary criterion. Gradient boosting attains the highest PR-AUC (0.9145), followed by random forest (0.8956) and LightGBM (0.8809). The gap between the tree ensembles and the remaining families is large: the multilayer perceptron reaches 0.8021, and the linear and instance-based learners fall below 0.55.

Table 5
Ten-fold cross-validated performance on the training partition (mean ± SD)

Model	PR-AUC	ROC-AUC	F1	MCC	G-mean	Recall	Precision
Gradient boosting	0.9145 ± 0.036	0.9811	0.8824	0.8820	0.9040	0.8190	0.9587
Random forest	0.8956 ± 0.041	0.9743	0.8702	0.8734	0.8830	0.7820	0.9856
LightGBM	0.8809 ± 0.042	0.9718	0.8494	0.8478	0.8961	0.8079	0.9037
XGBoost	0.8656 ± 0.041	0.9725	0.8171	0.8128	0.8914	0.8004	0.8411
Extra trees	0.8558 ± 0.043	0.9756	0.7512	0.7673	0.7815	0.6130	0.9778
Multilayer perceptron	0.8021 ± 0.055	0.9628	0.7394	0.7389	0.8177	0.6761	0.8318
Support vector machine	0.6762 ± 0.064	0.9633	0.4641	0.5029	0.9063	0.8817	0.3156
Decision tree	0.6672 ± 0.071	0.8954	0.8101	0.8049	0.8893	0.7968	0.8283
k-nearest neighbours	0.5384 ± 0.067	0.8723	0.4631	0.5056	0.5670	0.3247	0.8214
Logistic regression	0.4695 ± 0.074	0.9241	0.2715	0.3270	0.8432	0.8414	0.1619
Gaussian naive Bayes	0.4649 ± 0.054	0.9203	0.4748	0.4588	0.7209	0.5349	0.4328

Note: single-threshold metrics evaluated at t = 0.50. Best value in each column in the upper rows; PR-AUC is the primary selection criterion.

The table also illustrates why the choice of metric matters. Logistic regression records a ROC-AUC of 0.9241, which in isolation would suggest a serviceable model; its PR-AUC of 0.4695 and precision of 0.162 reveal that roughly five of every six alarms it raises are false. The support vector machine shows the same pattern, with the highest G-mean of the whole table (0.9063) but a precision of 0.316. Class weighting drives both models to a high-recall, low-precision operating point that flatters every metric normalised by the majority class. Conversely the single decision tree, whose ROC-AUC of 0.8954 is the second-worst in the table, achieves an MCC of 0.8049 that exceeds that of the far better ranked support vector machine, because its hard axis-aligned partitions match the rectangular failure envelopes of Figure 3 even though its probability estimates are coarse.

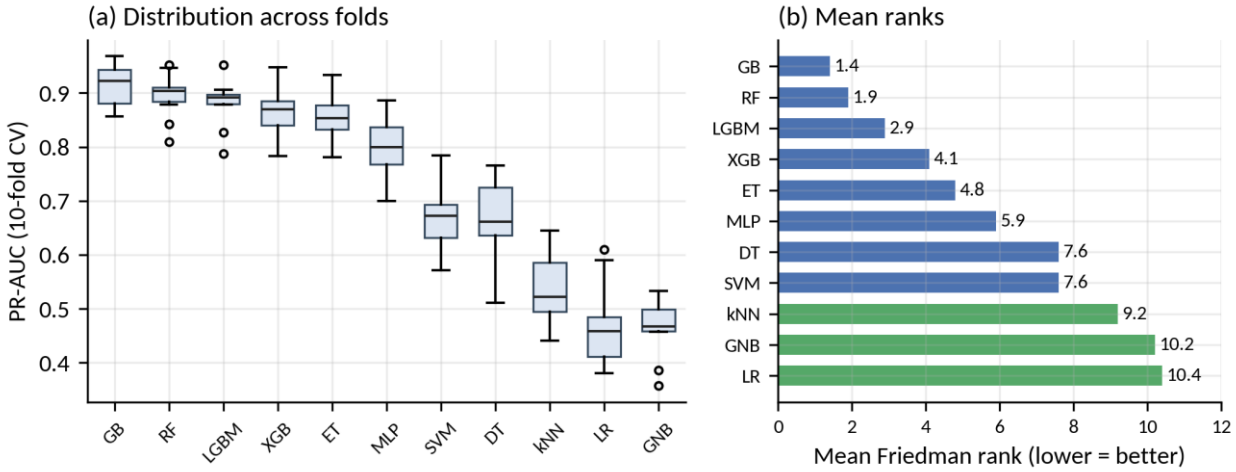


Fig. 5. Cross-validated model comparison: (a) distribution of PR-AUC across the ten folds, (b) mean Friedman ranks

3.3 Statistical Significance

The Friedman test rejects the hypothesis of equal performance across the eleven models (χ^2 = 95.45, df = 10, $p < 0.001$). Gradient boosting obtains the best mean rank (1.4), ahead of random forest (1.9) and LightGBM (2.9). Table 6 reports the Holm-corrected Wilcoxon comparisons between gradient boosting and each competitor. The control model is significantly superior to eight of the ten alternatives, but the differences from random forest and LightGBM are not significant after correction. The honest conclusion is therefore that the three leading gradient-based and bagged ensembles are statistically indistinguishable on this benchmark, and that the selection of gradient boosting for the subsequent analysis is a defensible but not a uniquely determined choice.

Table 6
Holm-corrected Wilcoxon signed-rank comparisons against gradient boosting (control)

Comparison	Mean ΔPR-AUC	W	p	Holm-adjusted p	Significant
vs. Gaussian naive Bayes	0.4495	0.0	0.0020	0.0195	Yes
vs. Logistic regression	0.4450	0.0	0.0020	0.0195	Yes
vs. k-nearest neighbours	0.3761	0.0	0.0020	0.0195	Yes
vs. Decision tree	0.2473	0.0	0.0020	0.0195	Yes
vs. Support vector machine	0.2383	0.0	0.0020	0.0195	Yes
vs. Multilayer perceptron	0.1124	0.0	0.0020	0.0195	Yes
vs. Extra trees	0.0587	0.0	0.0020	0.0195	Yes
vs. XGBoost	0.0488	0.0	0.0020	0.0195	Yes
vs. LightGBM	0.0336	6.0	0.0273	0.0547	No
vs. Random forest	0.0188	10.0	0.0840	0.0840	No

Note: n = 10 paired fold-level PR-AUC values; $\alpha = 0.05$.

3.4 Class Imbalance Strategies and Feature Ablation

Table 7 reports the central negative result of the study. None of the seven resampling schemes improves on the unresampled baseline under the primary criterion, and the ordering is monotone in the aggressiveness of the intervention: random oversampling costs 1.6 PR-AUC points, the SMOTE family between 4.8 and 7.8, and random undersampling 16.8. The pattern in the component metrics explains the mechanism. Every scheme raises recall, from 0.819 at baseline to as much as 0.960 under undersampling, and every scheme collapses precision, from 0.959 to as little as 0.321. Resampling does not create information; it relocates the decision boundary by inflating the prior of the minority class, which is exactly what adjusting the decision threshold does—but it does so implicitly, at a point the analyst cannot control, while simultaneously destroying the calibration of the probability estimates that the decision layer of Section 3.7 requires.

Table 7
Effect of class imbalance strategies on gradient boosting (ten-fold CV)

Strategy	PR-AUC	MCC	F1	G-mean	Recall	Precision
None (baseline)	0.9145	0.8820	0.8824	0.9040	0.8190	0.9587
Random oversampling	0.8984	0.6770	0.6692	0.9240	0.8782	0.5453
SMOTE-Tomek	0.8671	0.5979	0.5770	0.9191	0.8820	0.4316
SMOTE	0.8640	0.6020	0.5832	0.9158	0.8746	0.4413
Borderline-SMOTE	0.8590	0.6598	0.6519	0.9157	0.8635	0.5292
ADASYN	0.8364	0.5564	0.5296	0.9110	0.8746	0.3818
SMOTE-ENN	0.7704	0.5519	0.5175	0.9237	0.9041	0.3632
Random undersampling	0.7464	0.5296	0.4775	0.9421	0.9595	0.3208

Note: resampling applied inside the cross-validation loop, to training folds only.

Table 8 isolates the contribution of the derived features. Restricting the model to the six raw signals costs 9.5 PR-AUC points relative to the full set, and the four derived descriptors used alone recover 0.7925 from only four variables, within 1.6 points of the full raw set. Adding the mechanical descriptors alone recovers most of the gain (0.8919), while adding only the thermal difference recovers less (0.8449), consistent with the dominance of the power and overstrain mechanisms in the failure counts of Figure 2(c). Feature construction grounded in the physics of the process is thus worth considerably more here than either resampling or hyper-parameter tuning.

Table 8
Feature ablation under ten-fold cross-validation

Feature set	Number of features	PR-AUC (mean ± SD)
Raw signals + engineered (full)	10	0.9025 ± 0.027
Raw signals + mechanical descriptors	9	0.8919 ± 0.031
Raw signals + thermal descriptor	7	0.8449 ± 0.043
Raw signals only	6	0.8080 ± 0.047
Engineered descriptors only	4	0.7925 ± 0.067

Randomised search over 30 configurations selected 200 estimators, a learning rate of 0.075, a maximum depth of 3, a minimum of five samples per leaf, no subsampling and all features considered at each split, achieving a five-fold cross-validated average precision of 0.9119. Applied to the hold-out set this tuned configuration scores 0.8941 against 0.8987 for the default configuration, a difference well inside the fold-level standard deviation. Tuning is therefore reported as delivering no measurable benefit on this benchmark, which is worth stating plainly given how routinely search budgets are expanded in the applied literature without such a comparison.

3.5 Hold-out Performance

Table 9 reports performance on the 2,000 untouched test observations. Gradient boosting generalises with a PR-AUC of 0.8987 and an MCC of 0.8879, and at the default threshold it produces no false alarms at all (precision 1.000) while detecting 54 of the 68 failures. LightGBM attains the best recall among the ensembles (0.838) and XGBoost the best ROC-AUC (0.9847). The ranking of the leading models is consistent with the cross-validation, and the absolute values are close to it, indicating that the selection procedure did not overfit the training partition.

Table 9
Hold-out test performance at the default threshold of 0.50 (n = 2,000; 68 failures)

Model	Accuracy	Precision	Recall	F1	MCC	ROC-AUC	PR-AUC	Brier	FP	FN
Gradient boosting	0.9930	1.0000	0.7941	0.8852	0.8879	0.9646	0.8987	0.0064	0	14
LightGBM	0.9925	0.9344	0.8382	0.8837	0.8812	0.9800	0.8926	0.0075	4	11
GB (tuned)	0.9920	0.9643	0.7941	0.8710	0.8712	0.9605	0.8941	0.0073	2	14
XGBoost	0.9900	0.8871	0.8088	0.8462	0.8420	0.9847	0.8895	0.0090	7	13
Random forest	0.9900	0.9615	0.7353	0.8333	0.8362	0.9725	0.8728	0.0091	2	18
Extra trees	0.9835	0.9487	0.5441	0.6916	0.7118	0.9757	0.8388	0.0129	2	31
Multilayer perceptron	0.9855	0.8197	0.7353	0.7752	0.7689	0.9829	0.8365	0.0117	11	18
Decision tree	0.9850	0.8065	0.7353	0.7692	0.7624	0.8645	0.6020	0.0150	12	18
Support vector machine	0.9725	0.6444	0.4265	0.5133	0.5110	0.9715	0.6246	0.0188	16	39
k-nearest neighbours	0.9740	0.8333	0.2941	0.4348	0.4861	0.8823	0.5286	0.0200	4	48
Gaussian naive Bayes	0.9605	0.4286	0.4853	0.4552	0.4357	0.9335	0.4297	0.0314	44	35
Logistic regression	0.8580	0.1786	0.8824	0.2970	0.3585	0.9393	0.4349	0.1043	276	8

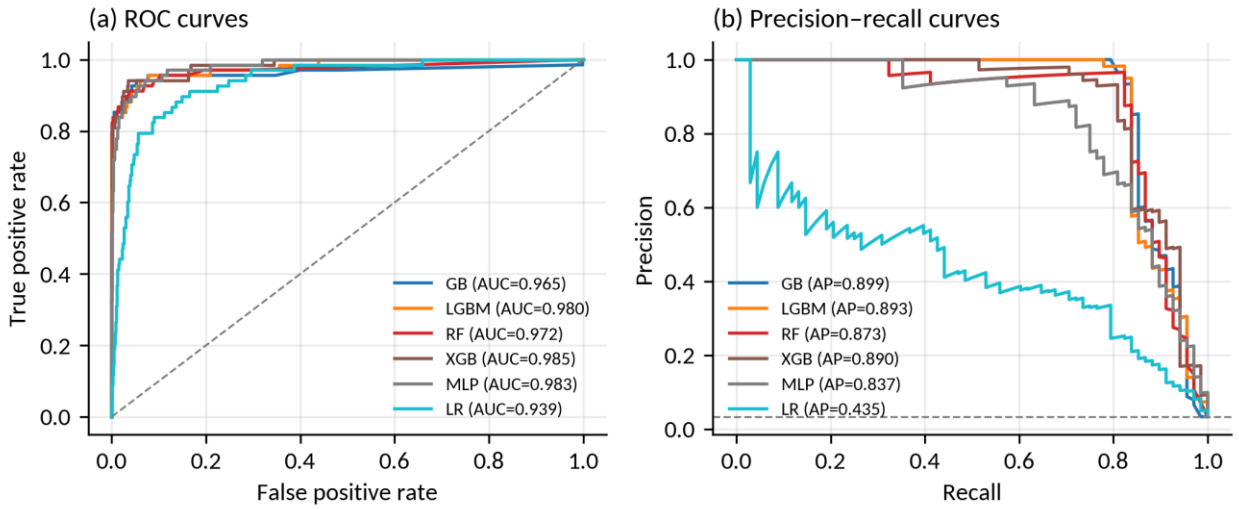


Fig. 6. Hold-out discrimination of the six leading models: (a) receiver operating characteristic curves, (b) precision-recall curves, with the dashed line marking the no-skill baseline at the 3.4% failure prevalence

Figure 6 makes the argument of Section 2.5 visually. In panel (a) all six models trace curves that hug the upper-left corner and are difficult to distinguish; logistic regression, the weakest model in the study, is barely separated from gradient boosting. In panel (b) the same models are clearly ordered, and the collapse of the linear model's precision at moderate recall is immediate. Reporting only ROC curves on a problem at this prevalence would conceal the practically decisive differences.

3.6 Probability Calibration

Table 10 reports the effect of calibration. Both methods improve the Brier score of the raw model, isotonic regression marginally more (0.00654 against 0.00726, a 9.9% reduction), and neither materially changes ranking quality, which is expected since both transformations are monotone. The reliability diagram in Figure 10(a), drawn on logarithmic axes because almost all observations receive very small scores, locates the residual defect. The raw model is mildly under-confident precisely in the low-probability region where the cost-optimal threshold lies: in the bottom nine deciles it predicts a mean of 0.0016 against an observed rate of 0.0022, and in the next band 0.054 against 0.064. Applying Eq. (8) to such scores therefore places the effective operating point higher than the cost structure intends, and alarms that should fire do not.

Table 10
Effect of probability calibration on the gradient boosting model (hold-out set)

Calibration method	Brier score	Reduction vs. uncalibrated	PR-AUC
Uncalibrated	0.00726	—	0.8941
Platt scaling (sigmoid)	0.00657	9.5%	0.8975
Isotonic regression	0.00654	9.9%	0.8929

3.7 Cost-sensitive decision analysis

Table 11 evaluates six maintenance policies on the hold-out set under the base cost structure. The two naive policies bound the problem: running to failure costs 340,000 units, and inspecting every machine costs 483,000. The classifier at the default threshold already reduces this to 70,500, a 79.3% saving, and it is this figure that most published comparisons implicitly report. Moving the threshold, however, delivers a further substantial gain that the default obscures. The best policy combines isotonic calibration with the analytical threshold of Eq. (8) and costs 44,250 units: 37.2% below the default cut-off and 87.0% below run-to-failure. It converts 10 of the 14 missed failures into detections at the price of 97 additional inspections, which the cost ratio makes overwhelmingly worthwhile.

Table 11
Expected cost of alternative maintenance policies on the hold-out set (CFP = 250, CFN = 5,000)

Policy	Threshold	TP	FP	FN	Recall	Precision	Total cost	Saving
Run to failure	—	0	0	68	0.000	—	340,000	—
Inspect every machine	—	68	1,932	0	1.000	0.035	483,000	−42.1%
Classifier, default cut-off	0.500	54	0	14	0.794	1.000	70,500	79.3%
Uncalibrated, analytical t*	0.0476	59	49	9	0.868	0.546	57,250	83.2%
Uncalibrated, empirical t*	0.0082	65	157	3	0.956	0.293	54,250	84.0%
Calibrated, analytical t*	0.0476	64	97	4	0.941	0.398	44,250	87.0%

Note: savings are expressed relative to the run-to-failure policy. Thresholds were selected on the training partition only.

Two features of Table 11 deserve emphasis. First, the analytical threshold applied to uncalibrated scores (57,250) performs worse than the same threshold applied to calibrated ones (44,250), detecting five fewer failures, which is the practical consequence of the under-confidence documented in Figure 10(a): Eq. (8) is valid only for probabilities that mean what they claim. Second, the empirical threshold search is not reliably better than the closed form. It selects an extreme cut-off of 0.0082 and reaches 54,250, but the shape of the cost profile in Figure 7(a) shows why this is fragile—the curve is nearly flat across two orders of magnitude of threshold, so the empirical minimum is determined largely by sampling noise in the out-of-fold predictions. The analytical route is preferable because it is stable, interpretable and directly tied to the plant's own cost parameters.

Because those parameters vary between plants, Table 12 repeats the analysis across cost ratios. The selected threshold falls monotonically as the penalty for missed failures rises, and the relative advantage over the default cut-off grows from nothing at a 1:2 ratio to a 67% cost reduction at 1:100. Below approximately 1:5 the default threshold is close to optimal, which delimits the conditions under which the decision layer developed here is worth implementing at all: it matters when failures are expensive relative to inspections, and not otherwise.

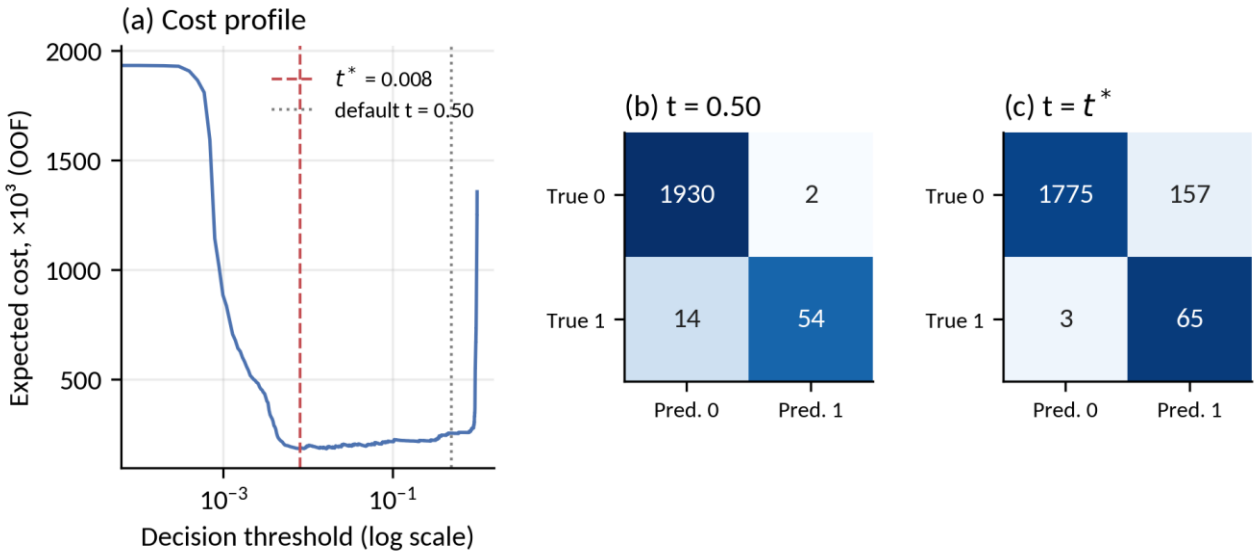


Fig. 7. Cost-sensitive operating point: (a) expected cost as a function of the decision threshold estimated from out-of-fold training predictions, (b) hold-out confusion matrix at the default cut-off, (c) hold-out confusion matrix at the cost-minimising threshold

Table 12
Sensitivity of the cost-optimal operating point to the cost ratio CFN : CFP

Cost ratio	Selected t*	TP	FP	FN	Recall	Precision	Cost relative to t = 0.50
1 : 2	0.5970	54	2	14	0.794	0.964	1.000
1 : 5	0.3191	56	5	12	0.824	0.918	0.903
1 : 10	0.2747	57	7	11	0.838	0.891	0.824
1 : 20	0.0082	65	157	3	0.956	0.293	0.770
1 : 30	0.0082	65	157	3	0.956	0.293	0.585
1 : 50	0.0082	65	157	3	0.956	0.293	0.437
1 : 100	0.0082	65	157	3	0.956	0.293	0.326

Note: thresholds obtained by minimising Eq. (7) over out-of-fold training predictions and applied unchanged to the hold-out set.

3.8 Explainability

Figure 8 compares the three importance measures, and their disagreement is instructive rather than contradictory. Shapley attribution ranks tool wear and rotational speed highest, permutation importance ranks mechanical power and rotational speed highest, and impurity gain concentrates on power, overstrain and the temperature difference. The divergence follows directly from the correlation structure of Figure 4: gain credits whichever correlated feature the tree happened to split on first, permutation importance credits only features whose removal cannot be compensated by a correlated substitute, and Shapley values distribute credit across the correlated group. Air temperature, process temperature and the wear-speed descriptor have permutation importance indistinguishable from zero, confirming that they are redundant once the temperature difference and the mechanical descriptors are available.

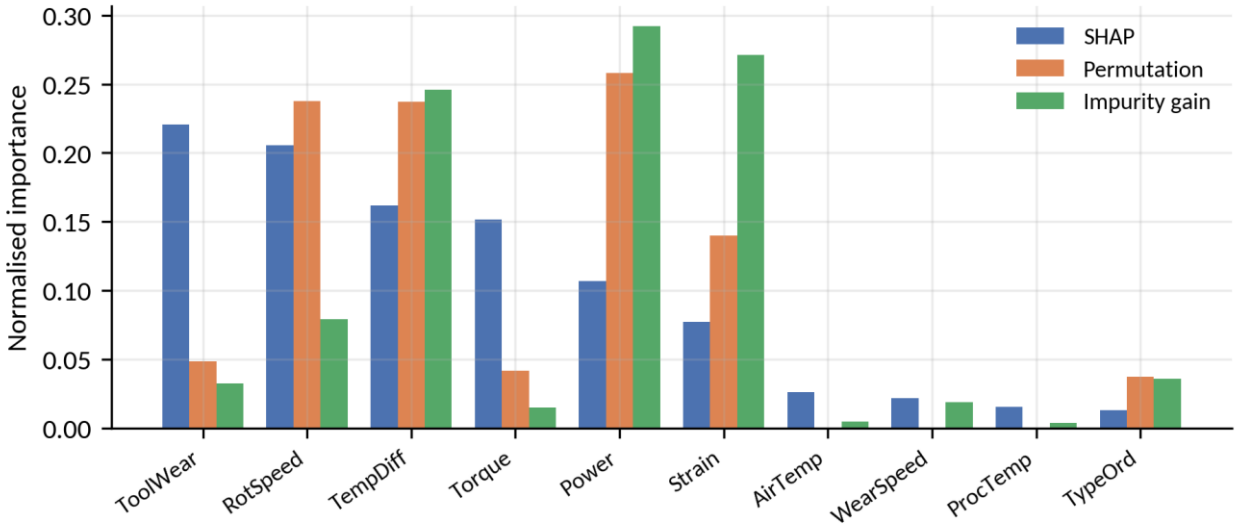


Fig. 8. Normalised feature importance under three attribution schemes: Shapley values, permutation importance and impurity gain

Figure 9(a) shows the distribution of Shapley values across the hold-out set. High tool wear, high torque and low temperature difference push predictions toward failure, and the attributions are strongly asymmetric: most observations receive small negative contributions, while a thin tail of observations receives very large positive ones, which is the expected signature of a rare-event model that abstains by default. Figure 9(b) resolves the interaction underlying the power-failure mechanism. The contribution of torque is not monotone but depends on rotational speed: high torque is attributed as dangerous when it coincides with low speed, and the same torque is attributed as benign at high speed. The model has recovered the product structure of mechanical power from data without being told it, and it agrees with the documented generating mechanism of the benchmark.

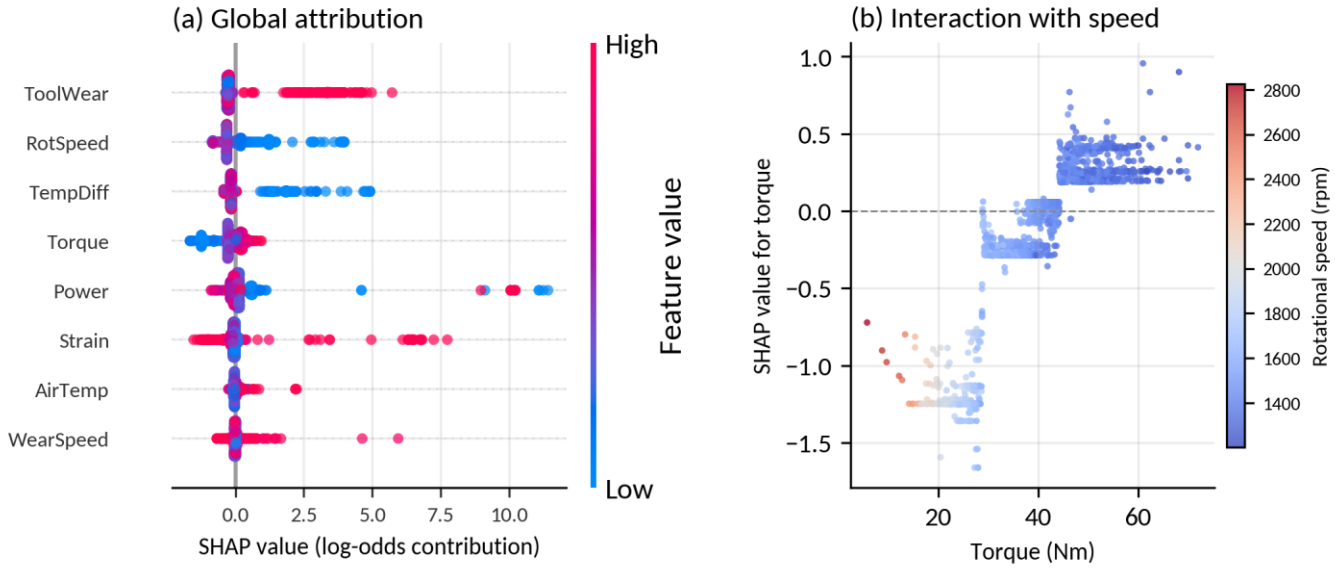


Fig. 9. Shapley explanations of the gradient boosting model: (a) global attribution distribution over the hold-out set, (b) attribution of torque as a function of rotational speed, revealing the power-failure interaction

Figure 10(b) illustrates the case-level output that a maintenance planner would receive. For a low-quality machining unit operating at 51.8 Nm and 1,363 rpm with a process-air temperature gap of 8.4 K, the model returns a failure probability of 0.901 against a base rate of 3.4%. The two dominant contributions are the narrowed thermal gap (+4.6 log-odds) and the low rotational speed (+3.8), with torque and the remaining features contributing marginally. The documented heat-dissipation rule of the benchmark triggers when the temperature difference falls below 8.6 K while rotational speed is below 1,380 rpm, and the ground-truth record for this observation is indeed flagged as a heat-dissipation failure. The explanation therefore does not merely rank features: it recovers the operative physical rule, and in doing so tells the planner that the corrective action is a cooling or ventilation intervention rather than a tool change.

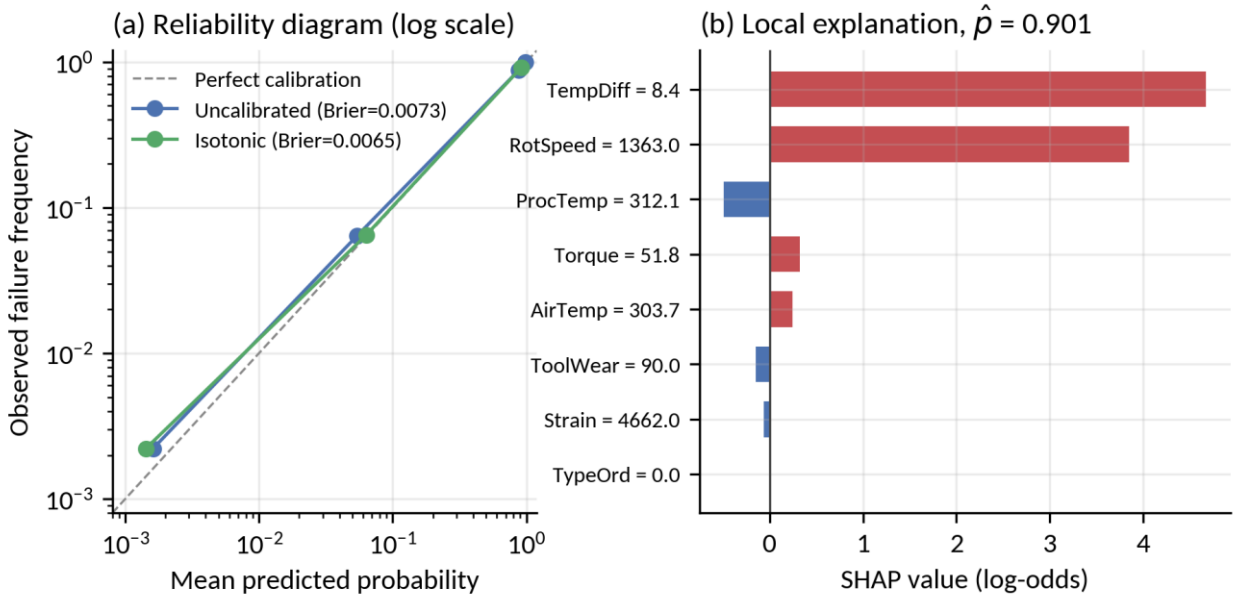


Fig. 10. Probability quality and case-level reasoning: (a) reliability diagram before and after calibration, (b) local Shapley decomposition of a single high-risk prediction

4. Discussion

The results support four claims that bear on how predictive maintenance systems are built rather than merely on how well they score. The first concerns algorithm selection. Gradient boosting, random forest and LightGBM are statistically indistinguishable on this benchmark after correction for multiple comparisons, while all three are clearly superior to linear, instance-based and kernel methods. Reporting a single winning model by point comparison, as much of the applied literature does, therefore overstates what the evidence supports; the defensible conclusion is that the tree-ensemble family is appropriate for problems with this rectangular, interaction-driven failure geometry, and that the choice within that family can be made on secondary grounds such as training cost, where LightGBM was an order of magnitude faster than random forest.

The second claim concerns imbalance. The finding that all seven resampling schemes reduce PR-AUC is not an argument that imbalance is unproblematic, but an argument about where the problem should be solved. Resampling and threshold adjustment address the same underlying issue—that a symmetric decision rule is inappropriate for an asymmetric problem—but resampling does so by corrupting the training distribution, which degrades the probability estimates that the downstream decision requires, and it fixes the implied operating point at whatever the resampling ratio dictates. Thresholding solves the problem at the point of decision, where the cost parameters actually live, and leaves the model's probabilities intact. This is consistent with the cost-based framing of Susto *et*

al., [3] and it suggests that the routine application of SMOTE variants in industrial classification papers deserves more scrutiny than it usually receives.

The third claim concerns where effort pays. Ranked by contribution to expected cost, the interventions examined here are: adopting a classifier at all (79.3% saving over run to failure), calibrating and thresholding it (a further 37.2%), constructing physics-informed features (9.5 PR-AUC points), and hyper-parameter tuning and resampling (no measurable benefit, and a loss respectively). This ordering approximately inverts the effort allocation visible in much of the applied literature, where extensive search over model configurations coexists with an unexamined 0.50 threshold.

The fourth claim concerns explanation. The Shapley attributions recover the power-failure interaction, the overstrain product and the heat-dissipation condition that define the benchmark's generating process, which provides unusually direct evidence that the explanations are faithful to real mechanisms rather than to artefacts. The divergence between attribution schemes under correlated features is a caution against reporting a single importance ranking, a practice that Matzka [4] and subsequent explainable maintenance studies [5, 6] have also had to confront.

Several limitations qualify these conclusions. The benchmark is synthetic, so its failure mechanisms are cleaner and its noise better behaved than in field data, and the absolute performance figures should be read as an upper bound. The observations are treated as independent snapshots, whereas real condition monitoring produces sequences in which degradation is autocorrelated; a temporal formulation predicting failure within a horizon would be closer to deployment and would require blocked rather than random splitting. The cost parameters are illustrative, and although Table 12 maps the sensitivity, a real deployment would need plant-specific costing including the opportunity cost of scheduled downtime, which is not a simple constant. Finally, the analysis is static: it does not address the concept drift that follows tool replacement, process re-qualification or seasonal change, and a production system would require monitoring of both calibration and attribution stability over time.

5. Conclusions

This study set out to establish which learning algorithm best suits the AI4I 2020 predictive maintenance benchmark under a precision-sensitive criterion, to quantify what resampling contributes, to measure the gain available from calibration and decision-theoretic thresholding, and to test whether the resulting model's reasoning is physically defensible. Gradient boosting achieved the best cross-validated PR-AUC of 0.9145 and a hold-out PR-AUC of 0.8987 with a Matthews correlation coefficient of 0.888, but Friedman and Holm-corrected Wilcoxon testing showed it to be statistically indistinguishable from random forest and LightGBM, so the finding is properly stated at the level of the tree-ensemble family rather than of a single winner.

None of the seven resampling schemes improved ranking quality, and each degraded precision severely, whereas physics-informed feature construction raised PR-AUC from 0.8080 to 0.9025 and hyper-parameter search produced no measurable improvement. The decision layer proved decisive: isotonic calibration reduced the Brier score by 9.9%, and applying the analytical cost-minimising threshold to the calibrated probabilities reduced expected misclassification cost by 37.2% relative to the conventional 0.50 cut-off and by 87.0% relative to run-to-failure operation. Shapley attribution recovered the power, overstrain and heat-dissipation mechanisms documented for the benchmark, including the torque-speed interaction, and supported case-level justification of individual alarms.

The practical implication for intelligent decision support in Industry 4.0 settings is that the marginal return on further model refinement is small relative to the return on treating the conversion of probability into action as a first-class modelling problem. Future work should extend the framework to temporal formulations with blocked validation, to field data with genuine sensor noise

and drift, and to multi-class prediction of the failure mechanism itself, which would allow the decision layer to recommend a specific intervention rather than an undifferentiated inspection.

Acknowledgement
This research was not funded by any grant.

Conflicts of Interest
The authors declare no conflicts of interest.

References
[1] Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, 150, 106889. https://doi.org/10.1016/j.cie.2020.106889
[2] Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. P., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137, 106024. https://doi.org/10.1016/j.cie.2019.106024
[3] Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2015). Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11(3), 812–820. https://doi.org/10.1109/TII.2014.2349359
[4] Matzka, S. (2020). Explainable artificial intelligence for predictive maintenance applications. In *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)* (pp. 69–74). IEEE. https://doi.org/10.1109/AI4I49448.2020.00023
[5] Gawde, S., Patil, S., Kumar, S., Kamat, P., Kotecha, K., & Alfarhood, S. (2024). Explainable predictive maintenance of rotating machines using LIME, SHAP, PDP, ICE. *IEEE Access*, 12, 29345–29361. https://doi.org/10.1109/ACCESS.2024.3367110
[6] Ghasemkhani, B., Aktas, O., & Birant, D. (2023). Balanced k-star: An explainable machine learning method for Internet-of-Things-enabled predictive maintenance in manufacturing. *Machines*, 11(3), 322. https://doi.org/10.3390/machines11030322
[7] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239
[8] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357. https://doi.org/10.1613/jair.953
[9] Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing (ICIC 2005)*, LNCS 3644 (pp. 878–887). Springer. https://doi.org/10.1007/11538059_91
[10] He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks* (pp. 1322–1328). IEEE. https://doi.org/10.1109/IJCNN.2008.4633969
[11] Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1), 20–29. https://doi.org/10.1145/1007730.1007735
[12] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3), e0118432. https://doi.org/10.1371/journal.pone.0118432
[13] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21, 6. https://doi.org/10.1186/s12864-019-6413-7
[14] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010
[15] Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning* (pp. 625–632). ACM. https://doi.org/10.1145/1102351.1102430
[16] Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1), 56–67. https://doi.org/10.1038/s42256-019-0138-9

[17] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). ACM. https://doi.org/10.1145/2939672.2939778

[18] Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

[19] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. https://doi.org/10.1007/BF00994018

[20] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

[21] Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

[22] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

[23] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785