Zahra: Bulletin Big data, Data Science and Artificial Intelligence
Vol. 3, No. 1, [June 2024], pp. [27–42]
E-ISSN 2830-7984

Research Article Open Access (CC–BY-SA)

Cost-Aware Multi-Objective Optimization of Random Forest for Industrial Machine Failure Prediction

Zahratul Laily binti Edaris[1*], Siti Maryam binti Zainol[2], H.R Merdu Wira Jasa[3], Mesran[4]
[1] *Department of Mechanical Engineering, Politeknik Sultan Abdul Halim Mu'adzam Shah, Kedah, Malaysia*
[2] *Department of Electrical Engineering, Politeknik Tuanku Syed Sirajuddin, Arau, Perlis, Malaysia*
[3] *Informatics, STIKOM Tunas Bangsa, Pematangsiantar, North Sumatra, Indonesia*
[4] *Management Study Program, Sekolah Tinggi Ilmu Manajemen Sukma, Medan, Indonesia*

[1] *zahratul@polimas.edu.my;* [2] *sitimaryam@ptss.edu.my;* [3] *hamdanhrmwj@gmail.com;* [4] *mesran.skom.mkom@gmail.com*
** Corresponding author*

Abstract

Industrial machine failures are relatively rare but potentially costly to overlook, making aggregate classification accuracy insufficient for maintenance decision support. This study assesses whether cost-aware multi-objective Random Forest tuning, combined with an optimized decision threshold, improves failure prediction under asymmetric error penalties. The experiment covers 10,000 synthetic machine-operating observations, including 339 recorded failures, partitioned into 6,000 training, 2,000 development, and 2,000 held-out test records. Four alternatives were evaluated: default Random Forest, threshold adjustment alone, random-search tuning, and the proposed Pareto-based (NSGA-II) forest with threshold optimization. A missed failure was assigned twenty times the penalty of a false alarm, and model selection was restricted to development data. On the final test, the proposed method reduced the assumed average cost from 0.1800 to 0.0945 units per observation and raised failure recall from 73.53% to 94.12%. This improvement relative to the default model accompanied a precision decrease from 100.00% to 36.99% and an F1 decrease from 0.8475 to 0.5311. The simpler threshold-only comparator achieved a lower cost of 0.0925 at the same 94.12% recall. The proposed search therefore improved the default decision rule but did not outperform the strongest controlled alternative. Interpretation remains limited by the synthetic benchmark, a single recorded partition, hypothetical maintenance penalties, and unavailable optimization source modules.

Keywords: predictive maintenance; Random Forest; multi-objective optimization; asymmetric decision cost; failure detection

1. Introduction

Predictive maintenance uses machine-condition information to anticipate faults, reduce unplanned downtime, and support inspection and repair decisions [1]. Reviews of supervised industrial maintenance systems identify persistent challenges involving heterogeneous measurements, data provenance, and evaluation conditions [2]. The relevance of a predictive model also depends on the industrial sector, equipment type, and operational consequences of a failure alert [3]. Random Forest has shown competitive performance on machine-failure benchmarks, making it a useful reference model for studying nonlinear relationships among operating variables [4]. Industrial failure events are frequently uncommon, so a high overall accuracy can coexist with inadequate detection of the minority failure class [5]. Cost-sensitive maintenance studies have consequently examined how unequal penalties for missed failures and false alarms influence model assessment [6]. The operating usefulness of estimated risk additionally depends on probability quality and threshold selection rather than class labels alone [7]. A rigorous comparison must therefore distinguish changes to the learned forest from changes to the decision threshold, particularly under asymmetric error costs [8].

Hyperparameter optimization can alter tree depth, ensemble size, and leaf structure, but the outcome depends on the search domain, objective function, and computational budget [9]. Multi-objective feature-selection research illustrates how improvements in one predictive criterion may conflict with other objectives or model complexity [10]. The same conflict arises in failure detection when increasing sensitivity reduces missed failures but generates more maintenance alerts. Studies of imbalanced classification show that precision, recall, and class-weighted summaries can lead to different model rankings [5]. Evolutionary optimization of forest tree components, although demonstrated on nonindustrial data, shows that tree structure can be tuned beyond default settings [11]. Calibration

analysis further separates the reliability of predicted probabilities from discrimination and threshold-dependent classification outcomes [7]. Industrial validation is also vulnerable to dependence among observations collected from related equipment or time periods, a broader concern illustrated by structured cross-validation research in other scientific fields [12]. Recent predictive-maintenance studies address process-specific fault dynamics, missing information, and the transfer of patterns across operational settings [13]. These considerations make a single fixed split useful for controlled within-benchmark comparison but insufficient to establish the stability or external validity of an optimized decision rule.

The central question of this study is whether multi-objective Random Forest tuning yields an additional cost benefit after the decision threshold has already been optimized. A synthetic 10,000-record machine-maintenance benchmark is used to compare four conditions under identical data partitions: a default forest, threshold-only adjustment, random-search tuning, and a proposed Pareto-based forest combined with threshold selection. The primary endpoint is an assumed decision cost in which a missed failure is assigned twenty times the penalty of a false alarm. The comparison also examines failure precision, recall, F1, balanced accuracy, specificity, confusion counts, and available ranking and probability diagnostics. Threshold-only adjustment is included because changes in the decision policy can alter expected cost without changing the underlying forest, as highlighted by cost-aware classification research [8]. Random search provides a strong optimization control because the benefit of a structured search should be judged against an alternative with a comparable evaluation budget [9]. The resulting design tests the incremental value of combined tuning instead of assuming that a more elaborate optimization procedure is inherently superior. The empirical contribution is a transparent comparison of accuracy-oriented and cost-oriented outcomes, including any unfavorable trade-offs of the proposed method.

2. Method

2.1 Related Work

Predictive-maintenance research has evolved from isolated classification exercises toward systems that connect fault estimates with asset-management decisions [1]. A recent review of supervised industrial learning identifies diverse data sources, modeling practices, and validation settings across published studies [2]. Cross-sector mapping likewise shows that the definition of maintenance success is conditioned by equipment characteristics and operational context [3]. Reviews of maintenance planning distinguish prediction of failure risk from the timing and allocation of subsequent interventions [14]. Breakdown modeling for material-handling systems demonstrates how the availability of additional operational data can affect maintenance predictions [15]. A roadmap of industrial machine diagnosis identifies continuing challenges in generalization across faults and measurement conditions [16]. Research on the transition toward Industry 5.0 further situates predictive algorithms within broader human and technical maintenance systems [17]. These contributions motivate evaluating classification outputs against an explicit decision objective rather than treating predictive accuracy as a complete measure of maintenance utility.

Class imbalance is a recurring obstacle in manufacturing failure detection because the minority class may represent the events of greatest operational interest [5]. A broad imbalanced-learning survey distinguishes data resampling, loss modification, and classifier-level strategies for addressing this problem [18]. Research on evaluation under unequal class distributions demonstrates that the interpretation of aggregate metrics depends on prevalence and error composition [19]. Automotive maintenance experiments evaluate imbalance-handling approaches, although their results are specific to their data and objective functions [20]. A cost-optimized maintenance comparison using industrial records explicitly accounts for unequal classification consequences when contrasting learning algorithms [6]. Cost-aware calibration addresses a related but distinct requirement by aligning probability predictions with asymmetric downstream losses [8]. A survey of calibration methods emphasizes that probability reliability should not be conflated with discrimination or a favorable confusion matrix [7]. These findings support a multi-metric assessment in which cost, failure recall, false-alarm burden, and precision are evaluated together.

Random Forest remains a practical baseline for structured machine-failure prediction because it can model nonlinear feature interactions without requiring a deep architecture [4]. Alternative predictive-maintenance studies examine remaining-useful-life estimation and motor-fault analysis under different data and modeling assumptions [21]. Comparative experiments with machine-failure classifiers show that the relative performance of tree-based and neural approaches depends on the evaluated task [22]. Optimization of forest tree components has also been investigated in nonindustrial structured data, illustrating methodological possibilities without establishing industrial transferability [11]. Symbolic feature ranking within a random-forest family offers another route to altering

predictive representations rather than operating thresholds [23]. Evolutionary multi-objective feature selection addresses competing classification and feature-subset criteria, providing relevant optimization context while targeting a different intervention [10]. Hyperparameter-optimization research recommends matched budgets, transparent search domains, and independent performance assessment when comparing strategies [9]. The present experiment applies this principle by contrasting the proposed Pareto-based tuning with random search and a threshold-only control on the same final test partition.

Machine-fault prediction is also affected by the data-generating process and by incomplete or changing condition measurements [13]. Process-aware models have been developed for production settings in which operating regimes vary across product configurations [24]. Multi-sensor fusion has been investigated for interpretable fault diagnosis in rotating machinery [25]. Related work evaluates explanation techniques for machine-maintenance predictions, including feature-attribution and response-profile methods [26]. Domain adaptation and engineered time-domain features have been used to improve tool-wear prediction under changing milling conditions [27]. Deep learning approaches to tool-wear onset and remaining useful life address a distinct prognostic endpoint [28]. Time-series reviews highlight that dependence and temporal ordering are central to many real maintenance applications [29]. These studies underscore why results from a synthetic record-level classification benchmark cannot alone establish robustness to chronological drift or unseen machines.

Maintenance-oriented machine learning increasingly connects predictive outputs with production planning, interpretability, and resource allocation [30]. An industrial cold-forming press study illustrates the distinction between constructing an algorithm and evaluating its usefulness under operational conditions [31]. Data-efficient active learning has been considered for anomaly detection in industrial time series with limited annotation availability [32]. Research on water-pump failures examines both explanation methods and the influence of sampling choices on model behavior [33]. Uncertainty-aware prognostics connect predictions to downstream maintenance decisions through an explicit decision framework [34]. Structured cross-validation methods from other applied domains emphasize the danger of evaluating dependent observations as though they were independently sampled [12]. Related imbalanced-transaction research also explores optimization of loss and probability-based decisions, although its domain and cost structure differ from industrial maintenance [35]. Table 1 synthesizes the most closely related methodological directions and identifies the remaining need to test whether a more complex Random Forest optimization improves upon simpler cost-aware decisions.

Table 1. Comparative synthesis of selected related studies

Author / year	Contribution	Research-specific limitation
Hamdi Alaoui et al. (2026)	Reviewed AI predictive maintenance models and industrial challenges	Review does not isolate a matched random-forest threshold ablation
Bischl et al. (2023)	Summarized hyperparameter-search principles and evaluation practice	General methods; not the present industrial holdout test
de Menezes e Silva Filho et al. (2023)	Reviewed discrimination and probability calibration distinctions	Calibration evidence is not a maintenance-cost comparison
de Giorgio et al. (2023)	Analyzed imbalanced manufacturing classification	Does not isolate the present four-arm cost-based comparison
Yang and Iqbal (2025)	Compared cost-optimized learning for predictive maintenance	Different maintenance data and economic assumptions
Yang and Wang (2025)	Compared random-forest machine-failure approaches	A strong decision-threshold-only comparator remains necessary

The literature synthesis in Table 1 situates this study among recent work on predictive maintenance, model selection, imbalance, and probability-based decisions. Maintenance reviews establish the significance of failure prediction but do not isolate the contribution of a decision threshold from the underlying learning algorithm [1]. Methodological studies of hyperparameter search emphasize fair comparisons because additional search evaluations can increase the opportunity for validation overfitting [9]. Research on calibrated probabilities identifies a separate source` of decision error when estimated risks are interpreted as reliable event probabilities [7]. Manufacturing studies show that class imbalance changes the balance between missed failures and unnecessary alarms [5]. A cost-optimized industrial maintenance comparison provides a precedent for treating the consequences of errors as part of model evaluation [6]. Direct Random Forest benchmarking on machine-failure data provides an especially relevant algorithmic reference but does not replace the present four-arm cost-based comparison [4]. The specific contribution

tested here is the incremental value of Pareto-based forest tuning beyond threshold adjustment under a matched held-out evaluation.

2.2. Dataset

The experiments use the 10,000-record machine predictive-maintenance dataset distributed through the Kaggle resource named in the study. Its observations are synthetic records designed to represent machine-operating conditions, rather than a longitudinal sample from a deployed industrial fleet. The binary response, Target, identifies whether a record corresponds to machine failure. Among the available records, 339 have a positive failure label and 9,661 are labeled as nonfailures, giving a failure prevalence of 3.39%. Identification fields UDI and Product ID are excluded from predictor construction, together with Failure Type because that field is linked to the target outcome. The feature representation contains product category, measured temperatures, rotational speed, torque, tool wear, and engineered operating interactions. Table 2 summarizes the dataset properties and the split used for model development and evaluation. The small positive class motivates explicit analysis of minority-class detection and its consequences under unequal error penalties [5].

Table 2. Dataset characteristics and experimental partition

Dataset attribute	Study-specific protocol
Public source	Kaggle: shivamb/machine-predictive-maintenance-classification; AI4I-derived synthetic benchmark
Research endpoint	Target = binary machine failure (339/10,000 = 3.39% positive)
Analysis unit	One synthetic machine operating record; machine-level independence not evidenced
Executed partition	Stratified 6,000 training / 2,000 development / 2,000 locked test; seed = 42
Processing / quality audit	Exclude UDI, Product ID, Failure Type; use process/air temperature, speed, torque, wear, type and engineered interactions
Annotation / scope check	Assumed costs: FN=20, FP=1 (unitless); synthetic labels; no observed monetary costs

The dataset summary in Table 2 provides the class distribution, admissible inputs, and evaluation partition for the synthetic benchmark. The unit of analysis is a machine-operating record, not an independently observed maintenance event from a distinct physical machine. The absence of verified equipment identifiers and time-order information prevents the partition from representing machine-level or temporal external validation. Identifier and outcome-derived columns were omitted to reduce the possibility of using information unavailable when a maintenance decision is made. The retained variables describe operating temperature, speed, torque, wear, machine type, and interactions between physical measurements. The binary failure rate makes precision and recall sensitive to relatively small changes in the number of positive predictions. This consideration is consistent with broader findings on minority-class evaluation in manufacturing applications [5]. The cost values used subsequently are analytical penalties, not amounts measured from an industrial maintenance ledger.

The separation of training, development, and final-test observations is summarized in Table 3. A stratified split assigns 6,000 records to model fitting, 2,000 to model and threshold development, and 2,000 to the held-out comparison. All reported alternatives use the same random seed, 42, and the same final-test partition. Search configurations and operating thresholds are selected from development outcomes rather than from final-test scores. The test partition contains 68 failures and 1,932 nonfailures, permitting an explicit account of missed failures and false alarms. This fixed partition produces a controlled comparison of the four model configurations but does not quantify variation across alternative random splits. Methodological research on hyperparameter optimization identifies this separation between selection and assessment as important for reducing optimistic performance estimates [9]. The resulting inference is limited to the dataset and split that were actually evaluated.

Table 3. Data partition, model selection, and evaluation procedure

Stage	Experimental procedure
Source profiling	10,000 synthetic records, including 339 failures; identifiers and target-derived predictors excluded.
Data partition	Stratified 60:20:20 allocation with seed 42: training 6,000; development 2,000; test 2,000.
Model fitting	Random Forest alternatives fitted using training data; search and threshold choice use development outcomes.
Decision-rule selection	Model configurations and cutoffs fixed before the held-out test comparison.
Held-out assessment	False-negative and false-positive counts, cost, precision, recall, F1, balanced accuracy, MCC and specificity.

The procedure outlined in Table 3 distinguishes the sequential roles of source profiling, model fitting, configuration selection, and independent evaluation. The same source file is used across all experimental branches. A fixed seed of 42 determines the reported 60:20:20 stratified partition into training, development, and test sets. Predictive transformations and model fitting belong to the training-development boundary, whereas the 2,000-record final test is reserved for comparing selected decision rules. Each model's confusion counts and associated cost are calculated from this common partition. The design avoids changing test samples between alternatives, although it does not address dependence among machine observations or unobserved operating regimes. Studies of structured data validation illustrate why a random-record split alone should not be interpreted as temporal or equipment-independent generalization [12]. The reported cost comparison is therefore a within-benchmark result under a fixed assumed error-penalty ratio.

2.3. Research Framework

Figure 1 illustrates the experimental architecture shared by the default and cost-aware Random Forest branches. The baseline uses the standard forest decision rule with a fixed probability cutoff of 0.5. The proposed branch combines forest configuration search with a decision threshold selected on development observations. The same training, development, and test partitions are retained across the two branches. Two additional controls, threshold-only adjustment and random-search tuning, are evaluated to isolate the contribution of each intervention. This separation is important because threshold selection can alter a cost-sensitive classification outcome without modifying the learned probability ranking [8]. The four-arm framework therefore enables an incremental comparison against a simpler decision adjustment as well as the default model. Figure 1 precedes the formal description of the cost objective and the executed training protocol.

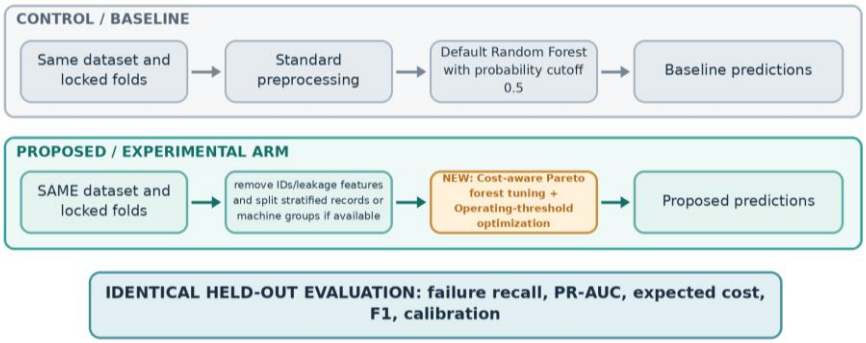


Figure 1. Controlled baseline-versus-proposed research framework

The research framework in Figure 1 separates common data handling from the aspects of the classifier that differ between experimental branches. Both branches operate on the same observations and use the same binary failure label and assumed error penalties. The default forest retains its conventional cutoff, whereas the proposed method adjusts both forest configuration and the operating point. Threshold-only optimization tests whether the decision layer is sufficient to produce most of the reduction in assumed cost. Random-search tuning tests the benefit of a competing hyperparameter-selection strategy under an equal recorded number of evaluations. Such matched comparisons reduce the risk of attributing improvement solely to the more elaborate optimization design [9]. Configuration choices are fixed using development data before the final test is assessed. The framework provides a basis for interpreting the eventual cost and detection trade-offs without implying that the combined method must dominate all controls.

Table 4 specifies the four experimental configurations used in the ablation comparison. The default model provides the fixed-threshold Random Forest reference. Threshold-only adjustment retains the fitted forest while changing the probability cutoff on development data. Random search explores forest configurations and uses a separately selected operating threshold. The proposed arm combines the recorded Pareto-based forest search with the same type of threshold selection. All branches are compared under identical failure labels, data partitions, and assumed error penalties. A common evaluation protocol is essential when testing whether a structured search yields an improvement beyond conventional tuning [9]. The table makes the distinction between classifier optimization and decision-rule optimization explicit.

Table 4. Controlled model configurations and comparison criteria

Component	Experimental configuration
Default baseline	Random Forest with cutoff 0.5; no cost-driven threshold adjustment.
Proposed method	Pareto-based forest configuration search followed by development-selected decision threshold.
Matched conditions	Identical 6,000/2,000/2,000 split, positive label, leakage exclusions and assumed cost ratio.
Comparator models	Default forest; threshold-only forest; random-search tuned forest; Pareto-based forest.
Evaluation endpoints	Held-out cost, failure recall, precision, specificity, F1, MCC, accuracy and balanced accuracy.

The design summarized in Table 4 establishes the controlled contrasts used to evaluate the proposed method. The threshold-only branch isolates the effect of lowering the classification cutoff in response to an asymmetric loss function. The random-search branch assesses whether another hyperparameter-selection method achieves a comparable operating cost. The proposed combined branch is evaluated against both alternatives rather than only against the fixed-cutoff baseline. The shared test records make the reported error counts directly comparable within the benchmark. The adequacy of a search strategy should be judged against meaningful conventional controls rather than its own training or development objective alone [9]. A lower test cost for a simpler comparator would indicate that the additional forest-search component provides no demonstrated cost advantage in this experiment. This criterion keeps the empirical assessment distinct from any claim of inherent methodological novelty.

2.4. Proposed Method

Figure 2 depicts the proposed cost-aware Random Forest pipeline and its two adjustable components. The first component searches forest configurations using NSGA-II, a multi-objective evolutionary algorithm run through Optuna with a population size of 8 and a budget of 36 trials; development cost and a model-complexity criterion are treated as competing objectives, and the configuration carried forward is selected from the recorded Pareto front before the test partition is scored. The second component selects an operating probability threshold using development observations. The decision objective assigns a penalty of 20 to each false negative and a penalty of 1 to each false positive, without attaching a monetary unit to either quantity. This asymmetric cost formulation reflects the distinction between missed machine failures and unnecessary maintenance alerts explored in cost-sensitive classification research [6]. The two searched forest alternatives each have 36 recorded objective evaluations. A threshold-only model is retained because a revised operating point may be more consequential than a more complex search mechanism [8]. The final analysis compares all candidate decisions on the same held-out observations.

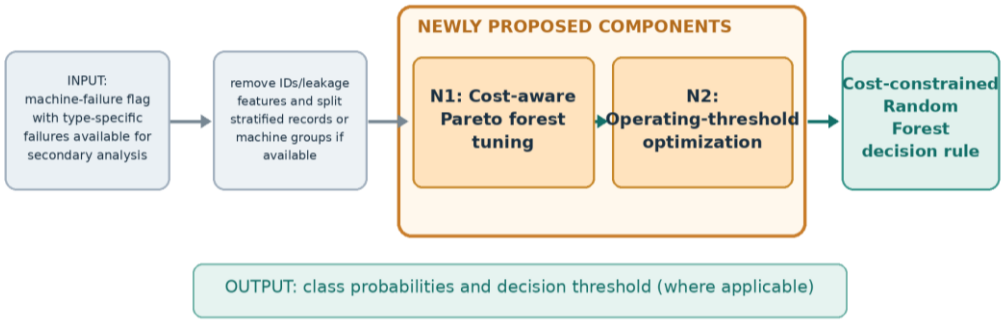


Figure 2. Cost-aware forest tuning and operating-threshold architecture

The architecture in Figure 2 distinguishes forest fitting and hyperparameter exploration from the final threshold-based decision rule. The proposed branch uses a development-selected threshold of 0.043780, whereas the random-search branch uses 0.044468. The threshold-only branch uses a cutoff of 0.072770 with the default forest, avoiding additional forest search. The lowered cutoffs reflect the chosen asymmetry between missed-failure and false-alarm penalties. The reliability visualization does not establish that a separate probability-calibration model was fitted during execution. Classifier calibration and threshold optimization are distinct operations, and neither should be inferred from the other [7]. The approach therefore evaluates a decision-stage adjustment alongside forest search without claiming an independently verified calibration stage. The tested effectiveness of the combined branch is determined by its held-out cost and associated error profile.

Table 5 states the operational definitions of the cost-sensitive classifier and its principal evaluation criteria. The learning task is binary failure prediction using the admissible operating measurements and engineered features. The proposed branch combines a recorded forest search with a development-selected probability threshold. A false

negative carries twenty assumed cost units, whereas a false positive carries one. Candidate selection uses development results, leaving the test partition for comparison of the frozen models. The objective is consistent across search branches so that a lower cost reflects different error counts rather than a changed loss definition. Cost-aware decision studies emphasize that the relative magnitude of error penalties determines the preferred operating point [8]. The table also identifies the strong comparator branches needed to interpret the incremental contribution of the proposed optimization.

Table 5. Cost-sensitive model components and evaluation definitions

Design element	Experimental specification
Input data	Synthetic machine observations; identifiers and outcome-derived fields excluded from predictors.
Forest search	Pareto-based Random Forest search; 36 recorded candidate evaluations.
Threshold selection	Development-only threshold optimization; selected Pareto branch cutoff 0.043780.
Model output	Binary failure decision from Random Forest probability and operating cutoff.
Decision cost	C(tau) = [20 FN(tau) + FP(tau)]/N; unitless assumed cost per evaluated record.
Comparator models	Default RF; threshold-only RF; random-search tuned RF; combined Pareto RF.
Evaluation endpoints	Cost against the strongest control; failure recall, false alarms, precision, F1 and specificity.

The specification in Table 5 defines the average assumed decision cost as the sum of weighted false-negative and false-positive counts divided by the number of observations. Under the selected scenario, each missed failure contributes twenty units and each false alarm contributes one. The resulting quantity is a model-evaluation proxy and must not be described as observed financial loss. Forest hyperparameters and probability cutoffs are selected using development observations before the fixed test comparison. The threshold-only branch measures the effect of changing the decision rule without changing the baseline forest. Cost-aware calibration research provides a theoretical reason to consider the consequences of probability-based decisions separately from general classification accuracy [8]. The four branches share the same denominator and penalty values, which makes their reported test costs directly comparable. The relative cost reduction is interpreted with reference to the default forest, while incremental optimization benefit is assessed against the strongest alternative.

The complete benchmark evaluation uses one fixed split with seed 42 and records the SHA-256 checksum of the source file predictive_maintenance.csv (9f0ede0b6fc33edacccfa1924e6430f92bd33ecce6e895543cd228c3a55ff10b) for reproducibility. Each searched method has 36 recorded candidate evaluations under the specified 20-to-1 loss ratio. The execution environment is reported as Windows 10 with Python 3.11.0, scikit-learn 1.9.1, pandas 3.1.0, NumPy 2.4.6, SciPy 1.15.3, and Optuna 5.0.0 on a 16-core AMD64 processor. Recorded search-related fitting times describe that environment and are not measurements of deployment latency: the search took 108.969 s for random search and 87.108 s for the Pareto-based search, the final refit-and-test step took 2.9-3.9 s per model, and inference took 39.0 ms per 1,000 records for three models and 54.5 ms for the random-search model. The original Pareto-search module was not retained with the experiment outputs, limiting independent reproduction of its update rules and search-space construction. The available experiment nevertheless permits an exact comparison of the recorded selected thresholds, held-out predictions, and assumed costs. Statistical variation across alternative seeds or chronological machine groups was not evaluated in this experiment.

The mathematical definitions of the decision rule and evaluation endpoints are summarized in Equations (1)-(5). The symbol p-hat denotes the forest's estimated probability of machine failure, and tau denotes the selected operating cutoff. Equation (1) maps that probability to a binary failure alert. Equation (2) specifies the mean weighted decision cost under the fixed false-negative and false-positive penalties. Equation (3) expresses failure-class precision and recall through the components of the confusion matrix. Equation (4) gives F1 and balanced accuracy, which capture different aspects of the detection and false-alarm trade-off. Equation (5) measures cost reduction relative to the fixed-threshold Random Forest baseline. The use of complementary endpoints follows recommendations to evaluate imbalanced classification beyond a single aggregate accuracy measure [19].

$$\hat{y}_i(\tau) = \mathbb{1}[\hat{p}_i \geq \tau] \tag{1}$$

$$C(\tau) = \frac{20\,\mathrm{FN}(\tau)+\mathrm{FP}(\tau)}{N} \tag{2}$$

$$P = \frac{TP}{TP+FP}, \qquad R = \frac{TP}{TP+FN} \tag{3}$$

$$F_1 = \frac{2PR}{P+R}, \qquad BA = \frac{R+S}{2} \tag{4}$$

$$CR(\%) = 100\ \frac{C_b - C_m}{C_b} \quad (5)$$

Equation (1) returns a failure alert when the estimated probability is at least tau and a nonfailure prediction otherwise. In Equation (2), FN and FP denote false-negative and false-positive counts, while N denotes the number of evaluated observations. The associated coefficients 20 and 1 describe assumed relative penalties rather than currency-valued losses. Equation (3) defines precision as TP/(TP+FP) and recall as TP/(TP+FN). Equation (4) combines precision and recall in F1 and averages sensitivity with specificity in balanced accuracy. Equation (5) compares the cost of an alternative model with that of the default threshold at the same fixed penalty ratio. These definitions expose why a gain in failure sensitivity can coincide with a reduction in precision or overall accuracy. Their simultaneous use avoids attributing an improvement in one metric to unmeasured operational or economic outcomes. Here S denotes specificity, TN/(TN+FP), and C_b and C_m denote the average assumed cost per record of the baseline and of the evaluated model, respectively. The Matthews correlation coefficient (MCC) is computed from the four confusion-matrix counts, average precision summarizes the precision-recall curve, and the Brier score is the mean squared difference between the predicted failure probability and the observed binary outcome.

2.5. Experimental Setup and Implementation

The experimental protocol covers all 10,000 synthetic records in predictive_maintenance.csv rather than a reduced pilot sample. The original dataset version is kept consistent across the model comparisons. The stratified partition consists of 6,000 training records, 2,000 development records, and 2,000 held-out test records under seed 42. The binary target has 339 failures in the complete dataset, including 68 failures in the held-out test set. Input construction uses product category, five measured process variables, and three engineered operating features (the difference between process and air temperature, mechanical power derived from torque and rotational speed, and the product of tool wear and torque), giving nine predictors in total. Fields UDI, Product ID, and Failure Type are excluded from the predictive representation. All candidate configurations are selected from the training-development data before the held-out comparison is calculated. The synthetic source and absence of verified equipment grouping constrain the interpretation to record-level generalization within this benchmark.

A default Random Forest with threshold 0.5 provides the baseline and yields development cost 0.1700. Threshold-only adjustment retains the default forest but selects cutoff 0.072770, giving development cost 0.0735. A 36-evaluation random-search alternative selects threshold 0.044468 and reaches development cost 0.0625. A separate 36-evaluation Pareto-based alternative selects threshold 0.043780 and also reaches development cost 0.0625. The two searches account for 72 recorded objective evaluations, with reported search-related fit times of 108.969 s for random search and 87.108 s for the Pareto-based search. The four selected decision rules are evaluated without further tuning on the same 2,000 held-out observations. The cost equation assigns twenty units to a missed failure and one to a false alarm in every branch. The evaluation does not include repeated random seeds or paired confidence intervals, and its interpretation remains limited to the recorded comparison.

3. Results and Discussion

3.1. Dataset Characteristics and Model Selection

The synthetic dataset contains 10,000 operating records, including 339 failures and 9,661 nonfailures. This marked imbalance makes the failure class the central concern for cost-sensitive evaluation. The observed class prevalence is a characteristic of the benchmark and should not be extrapolated to a production fleet. Figure 3 depicts the class frequencies together with the feature-missingness assessment. The plotted predictors exhibit no recorded missing values in the displayed audit. All four model configurations use the same 6,000/2,000/2,000 data partition. The descriptive analysis therefore supports a controlled comparison of minority-class detection, while the dataset's synthetic provenance limits broader industrial generalization.

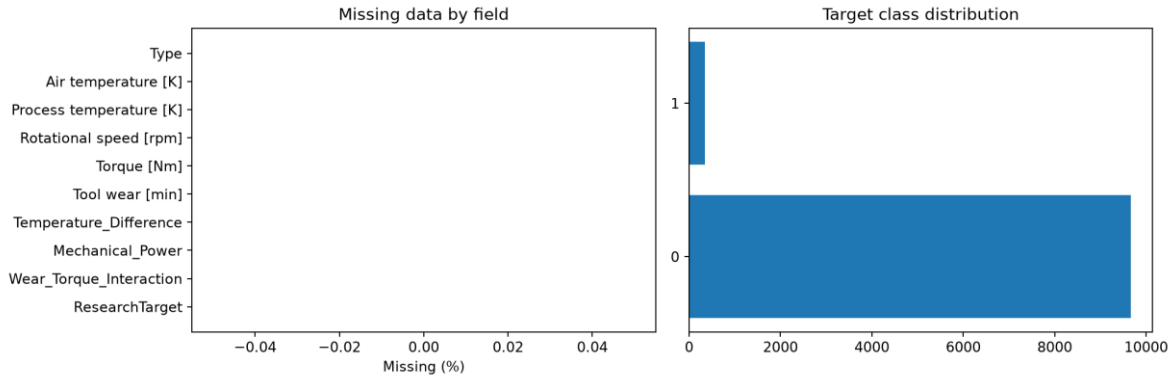


Figure 3. Dataset class distribution and missingness in the executed benchmark

The class distribution in Figure 3 shows that the failure class accounts for only 3.39% of the available records, compared with 96.61% nonfailure observations. The class frequencies explain why accuracy alone can mask substantial failure-detection errors. The feature-missingness panel indicates that the displayed predictors contain no missing observations. This absence of missingness does not establish independence of successive machine records or remove the risk of dataset-specific patterns. A classifier predicting mostly nonfailures could still achieve high overall accuracy while missing events given a large penalty in the study's cost function. Subsequent analyses therefore examine false negatives, false positives, failure recall, precision, and average assumed cost. The visual supports the need for multiple endpoints but does not itself establish predictive performance.

Table 6 presents the development-stage selections made before the final test evaluation. The fixed-threshold baseline has a development cost of 0.1700 at cutoff 0.500000. Threshold-only adjustment reduces that cost to 0.0735 at cutoff 0.072770. Random search and the Pareto-based search each obtain development cost 0.0625, using selected thresholds of 0.044468 and 0.043780. Both searched branches have 36 recorded objective evaluations, indicating parity in the number of reported search calls. Their underlying search mechanisms may nevertheless differ in computational effort and parameter exploration. The final choice between these approaches consequently depends on the held-out results rather than on their tied development costs.

Table 6. Development-stage selections and recorded objective budgets

Model	Cutoff	Development cost	Search evaluations
Default RF	0.500000	0.1700	Not applicable
Threshold-only RF	0.072770	0.0735	Not applicable
Random-search RF	0.044468	0.0625	36
Proposed Pareto RF	0.043780	0.0625	36

The development results in Table 6 indicate that adjusting the operating threshold produces a substantial reduction in development cost without additional forest tuning. The threshold-only branch improves the development objective by 0.0965 cost units relative to the fixed-threshold baseline. The two searched branches improve the development cost by another 0.0110 relative to threshold-only adjustment. Their equal development costs do not distinguish random search from Pareto-based tuning on this endpoint. The differences among selected cutoffs reflect the respective predicted probability distributions and the imposed asymmetric loss. The development partition was used to select these decision rules, so its cost estimates are not independent measures of future performance. The held-out results in Table 7 provide the decisive comparison among the four frozen configurations.

3.2. Held-Out Comparison of Baseline and Optimized Models

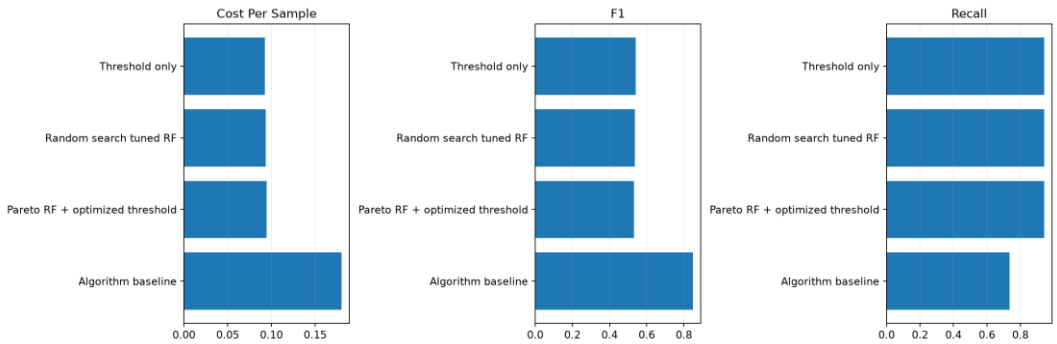
Table 7 compares the default Random Forest, threshold-only adjustment, random-search tuning, and the proposed Pareto-based forest on the same held-out observations. The final set contains 68 failure records and 1,932 nonfailure records. All four costs use the fixed penalty ratio of twenty per missed failure to one per false alarm. The table reports confusion counts together with precision, recall, specificity, balanced accuracy, F1, MCC, and overall accuracy. The fixed-threshold baseline has the highest accuracy, precision, and F1 among the compared configurations. All adjusted decision rules achieve substantially higher failure recall but produce more false alarms. Threshold-only adjustment attains the smallest cost per record in the final test, which is the primary result of the cost-based comparison.

Table 7. Final held-out test: default baseline versus optimization and strong controls

Indicator	Default RF	Threshold only	Random search	Proposed Pareto
Operating threshold	0.5000	0.0728	0.0445	0.0438
Accuracy	0.9910	0.9455	0.9445	0.9435
Balanced accuracy	0.8676	0.9434	0.9429	0.9424
Precision (failure)	1.0000	0.3787	0.3743	0.3699
Recall (failure)	0.7353	0.9412	0.9412	0.9412
Specificity	1.0000	0.9457	0.9446	0.9436
F1 (failure)	0.8475	0.5401	0.5356	0.5311
MCC	0.8535	0.5778	0.5741	0.5704
False positives	0	105	107	109
False negatives	18	4	4	4
True positives	50	64	64	64
True negatives	1932	1827	1825	1823
Cost / record	0.1800	0.0925	0.0935	0.0945
Cost reduction vs default	-	48.61%	48.06%	47.50%

The held-out comparison in Table 7 shows that the proposed Pareto-based configuration lowers the assumed cost from 0.1800 to 0.0945 per record, a 47.50% reduction relative to the default forest. The baseline detects 50 of 68 failures, whereas the proposed branch detects 64, increasing failure recall from 73.53% to 94.12%. The improved detection is accompanied by 109 false alarms instead of zero, reducing failure precision from 100.00% to 36.99%. F1 declines from 0.8475 to 0.5311, while balanced accuracy increases from 0.8676 to 0.9424, illustrating the disagreement between performance criteria. The threshold-only branch produces 105 false alarms and lower cost of 0.0925 while matching the proposed branch's failure recall. Random-search tuning also has a lower test cost of 0.0935 and fewer false alarms than the proposed branch. The proposed approach therefore improves the fixed-threshold default under the stipulated loss but does not improve upon the strongest simpler control. No paired confidence intervals or repeated-split results were available, so the observed differences are descriptive and cannot establish statistical superiority. For orientation, Wilson 95% confidence intervals for failure recall, based on the 68 test failures, are approximately 62.0-82.6% for the default forest (50/68) and 85.8-97.7% for each adjusted forest (64/68). The three adjusted rules differ by only two to four false alarms among 1,932 nonfailure records, so their cost ordering should not be regarded as established.

Figure 4 compares the held-out cost, F1, failure recall, average precision, and Brier score for the four alternatives. The displayed indicators capture operational loss, threshold-dependent classification, ranking, and probability quality. These criteria need not rank the same fitted classifiers in the same order. The fixed-threshold baseline retains a strong F1 score while incurring a relatively large assumed cost. Lower-threshold models produce greater failure sensitivity but weaker precision-related summaries. The graph provides a complementary view of model behavior to the confusion counts in Table 7. Its outcomes arise from the same synthetic test records and do not imply performance stability across other equipment populations.



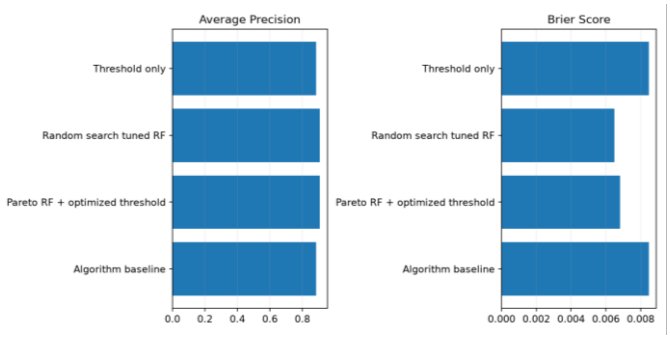


Figure 4. Model-level metric comparison from the locked test

The metric panels in Figure 4 display a pronounced trade-off between conventional classification fit and the stipulated asymmetric decision loss. The default forest misses 18 failures but generates no false alarms, resulting in its comparatively high F1 score. The three lower-threshold alternatives each detect 64 failures but flag more than 100 nonfailure observations. The proposed Pareto-based branch does not dominate threshold-only adjustment across the plotted test indicators. Average precision describes aspects of ranking quality, while the Brier score summarizes probability-error behavior. Neither metric alone establishes that a separate calibration model was fitted during optimization. The combined evidence indicates that an adjusted decision cutoff supplies most of the demonstrated cost benefit under the adopted penalty ratio.

Figure 5 shows the confusion matrices underlying the final-test performance measures. Each matrix contains the same 68 actual failures and 1,932 nonfailures. The default forest identifies 50 failures and misses 18 while producing no false alarms. The threshold-only forest identifies 64 failures, misses four, and generates 105 false alarms. Random-search tuning detects the same 64 failures with 107 false alarms. The proposed Pareto-based branch likewise detects 64 failures but generates 109 false alarms. These differences in false positives fully explain the cost ordering among the three adjusted decisions when their false-negative counts are identical.

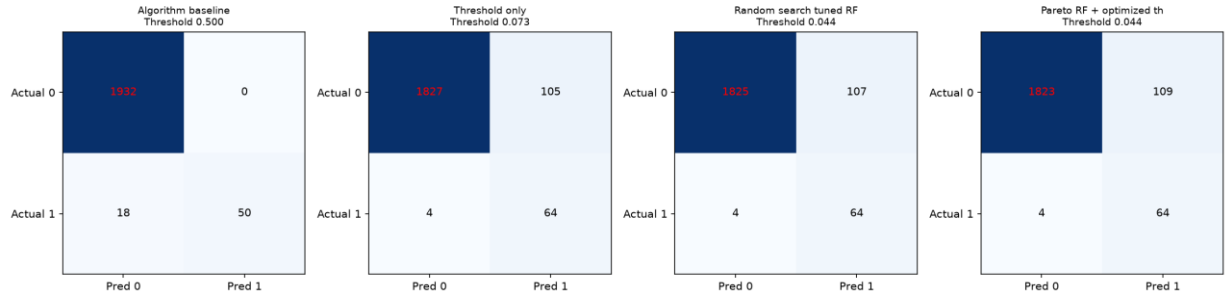


Figure 5. Confusion matrices for the four final-test decision rules

The confusion matrices in Figure 5 confirm that the main improvement of the cost-adjusted decisions is the reduction in missed failures rather than a uniform improvement across classification criteria. The fixed-threshold baseline has 18 false negatives and no false positives. Each of the three lower-threshold alternatives reduces false negatives to four. The proposed branch does not detect more failures than the threshold-only or random-search comparators. Its four additional false alarms compared with threshold-only adjustment explain the 0.0020 difference in average test cost. Such additional alarms could require unnecessary inspections if the model were translated into a maintenance policy, although no actual inspection costs were measured here. The confusion matrices therefore clarify both the detection gain and the practical trade-off accompanying that gain.

3.3. Ranking Performance and Cost Sensitivity

Figure 6 presents precision-recall curves for the four forest configurations on the final test set. A precision-recall analysis is informative when failures represent only a small fraction of observations. The curves characterize ranking behavior across classification thresholds rather than performance at a single operating point. The default threshold gives high precision but leaves more failures undetected. The selected lower thresholds increase detection sensitivity while admitting many more nonfailure records as alerts. This trade-off is consistent with the error counts in Figure 5 and the threshold-dependent scores in Table 7. The curve comparison does not alter the study's primary decision-cost endpoint or the development-selected cutoffs.

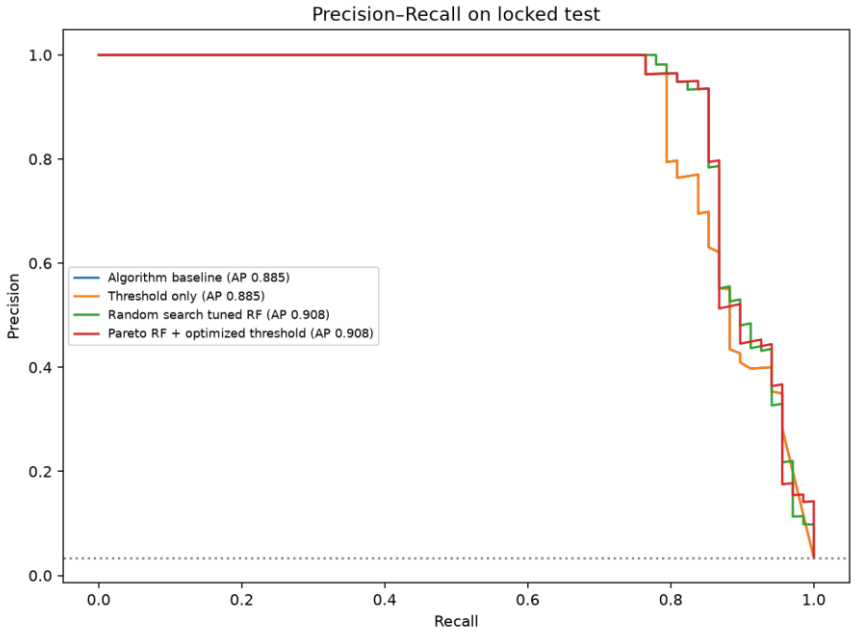


Figure 6. Precision-recall curves from the locked final-test predictions

The precision-recall curves in Figure 6 display high precision for the most selective decision regions and decreasing precision as more observations are classified as failures. The large number of nonfailure records makes false alarms accumulate as the cutoff is lowered. The default operating point has perfect observed precision but detects only 50 of the 68 failures. The alternative cutoffs detect 64 failures and therefore occupy more sensitive decision regions. This increase in recall is not cost-free because the adjusted rules incorrectly flag between 105 and 109 nonfailures. The precision-recall curves describe discriminatory ranking but cannot establish whether the assumed cost weights match a real maintenance context. The interpretation of their operating points must therefore remain linked to the explicit error counts and cost formula.

Figure 7 examines the sensitivity of average assumed cost to alternative classification thresholds. The final model comparison uses decision cutoffs selected from development observations rather than from the held-out test. The test-set threshold sweep is presented only as a diagnostic analysis of the cost curve. Its minimum is not used to revise the frozen thresholds or rerank the models through additional selection. Every point in the curve retains the twenty-to-one weighting between missed failures and false alarms. The shape of the curve reflects the changing balance between detection sensitivity and false-alarm burden. This analysis demonstrates how strongly the cost endpoint depends on the operating decision rule without representing observed financial savings.

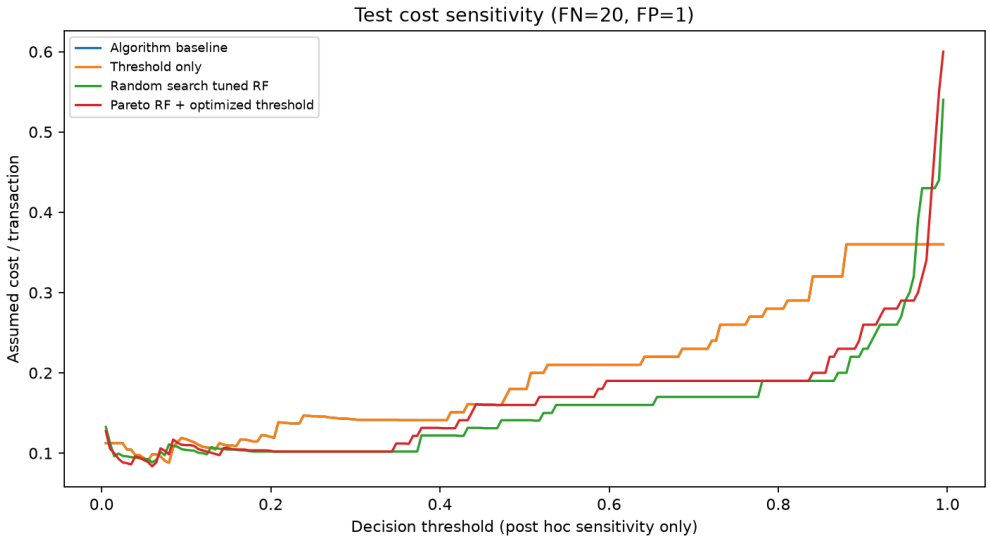


Figure 7. Locked-test sensitivity of the assumed cost to decision threshold

The threshold sweep in Figure 7 demonstrates that assumed cost varies nonmonotonically as the decision threshold changes. Very restrictive thresholds produce missed failures, whereas permissive thresholds create numerous false alarms. The default cutoff of 0.5 incurs a relatively high cost under the selected missed-failure penalty. The three adjusted cutoffs lie in a more sensitive region with fewer false negatives. The plotted sensitivity analysis is diagnostic and does not change the frozen selected operating points. Its shape depends on both the class prevalence and the hypothetical penalty values used in this experiment. Calibration assessment, external validation, and measured maintenance economics would be required before a threshold could be recommended for a deployed machine fleet.

Figure 8 evaluates the consequences of varying the assumed missed-failure penalty after the primary analysis. The horizontal axis represents a hypothetical penalty ratio rather than a monetary amount observed in production. The default forest has 18 false negatives, compared with four for each adjusted-threshold branch. Its estimated loss therefore rises faster as the penalty for missed failures increases. The adjusted branches incur persistent false-alarm penalties that influence their relative costs when missed failures are assigned less weight. The selected decision thresholds remain unchanged throughout this scenario analysis. The primary comparison continues to use the prespecified twenty-to-one ratio, while the figure reveals how alternative economic assumptions might change the preferred decision policy.

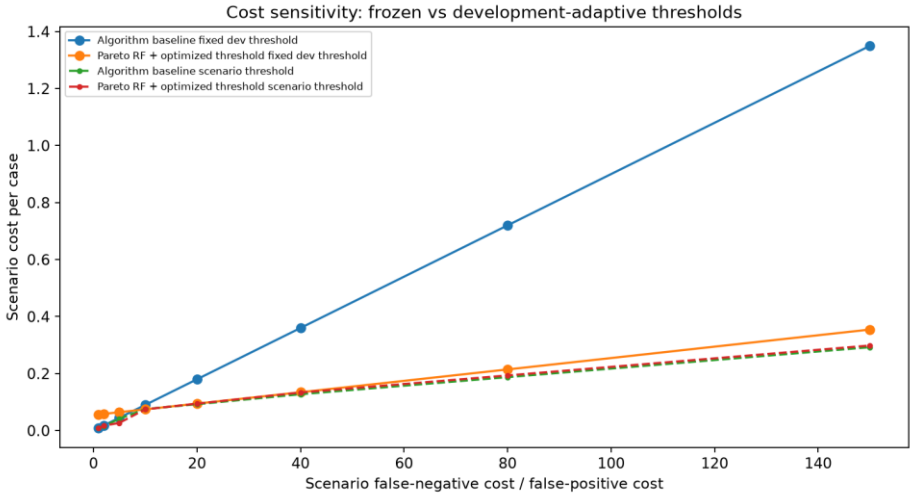


Figure 8. Cost scenarios under frozen, development-selected decision thresholds

The sensitivity curves in Figure 8 show that the relative value of the four decisions is sensitive to the assumed consequences of missed failures and unnecessary inspections. The fixed-threshold forest incurs no false-alarm cost in this test but misses substantially more failures than the adjusted alternatives. Increasing the missed-failure penalty therefore produces a steeper rise in its total assumed loss. The three adjusted branches display similar responses because each misses four failures and differs only slightly in false-alarm count. At the primary twenty-to-one penalty ratio, the threshold-only branch retains the lowest average cost. The proposed Pareto-based branch remains slightly less economical because it generates four more false alarms at equal recall. These are scenario costs derived from model errors, not empirically measured reductions in downtime or maintenance expenditure.

3.4. Discussion and Limitations

The strongest observed reduction in assumed maintenance cost is achieved by changing the Random Forest decision threshold without additional forest hyperparameter search. Threshold-only adjustment lowers test cost by 48.61% relative to the default and increases failure recall from 73.53% to 94.12%. The proposed Pareto-based branch also improves substantially over the fixed-cutoff baseline, but it does not outperform threshold-only adjustment under the prespecified penalty ratio. Its development objective equals that of random search, yet both searched branches are marginally more costly than the simpler threshold-only rule on the held-out test. These findings separate the effect of cost-sensitive decision selection from the incremental effect of forest configuration search. An improvement against the default alone would overstate the proposed approach's comparative benefit because a stronger and less complex alternative is available. The results consequently support the importance of threshold optimization while providing no observed incremental cost advantage for the added Pareto-based search. This pattern is consistent with cost-aware classification research that ties the preferred operating point to the ratio of

error penalties [8] and with cost-oriented maintenance comparisons that evaluate models by the consequences of their errors rather than by accuracy alone [6]. As a reference point, if the estimated probabilities were perfectly calibrated, minimizing the expected cost in Equation (2) would imply raising an alert once the failure probability reaches 1/(1+20), or about 0.048. The selected cutoffs of 0.044 to 0.073 are of the same order, which is compatible with the view that the gain comes mainly from the decision rule; this value is derived from the assumed penalties and is not evidence that the forests are calibrated.

The reduction in accuracy under the adjusted cutoffs is compatible with their lower assumed decision cost because the two quantities penalize different outcomes. The default model achieves 99.10% accuracy by correctly identifying all 1,932 nonfailures, but it misses 18 of the 68 actual failures. The proposed model achieves 94.35% accuracy and misses only four failures while producing 109 false alarms. This change lowers failure precision to 36.99% and F1 to 0.5311. The false-alarm burden could translate into additional inspection demand, but the experiment does not provide data on corresponding maintenance activities or their monetary consequences. The twenty-to-one penalty is an explicit scenario assumption rather than a cost ratio estimated from industrial accounting. A cost-focused interpretation must therefore retain the observed precision, F1, and confusion-matrix trade-offs rather than treating the reduction in assumed cost as universal operational benefit.

Several limitations restrict the generality of the findings. The executed record includes a dataset checksum and model outputs, but the external optimization modules were unavailable for independent examination of their internal search-space definitions. A Pareto-based designation cannot by itself establish a distinctive multi-objective mechanism without inspectable objective components and nondominated-solution records. The benchmark documentation describes failure labels as arising from rule-based failure modes tied to temperature difference, power, and wear-torque thresholds, and the three engineered features mirror these quantities, so the models may partly recover label-generating rules and the task may be easier than real fault prediction. The comparison uses one random seed and one stratified partition, so its stability across alternative data realizations remains unknown. The synthetic observations also do not represent a prospective deployment trial with independent machines, evolving operating conditions, or measured failure consequences. Confidence intervals for paired model differences were unavailable, precluding a claim of statistically established superiority. Future work should apply an auditable search implementation to chronological multi-machine data, quantify uncertainty across repeated trials, verify probability calibration, and estimate realistic failure and inspection costs. Within the tested benchmark and hypothetical cost structure, the simpler threshold-only decision remains the best-performing observed alternative.

4. Conclusion

This study compared a default Random Forest, threshold-only adjustment, random-search tuning, and the proposed Pareto-based forest on a 2,000-record held-out set from a synthetic maintenance benchmark. The test contained 68 failures and 1,932 nonfailures. The default model produced an assumed cost of 0.1800 per record, detected 50 failures, and achieved 73.53% recall. The proposed model detected 64 failures and reduced the cost to 0.0945, corresponding to a 47.50% decrease. This gain involved 109 false alarms and lower precision and F1. Threshold-only adjustment attained the lowest test cost, 0.0925, at the same 94.12% recall. The proposed search therefore did not demonstrate an incremental advantage over the strongest control.

These findings identify decision-threshold selection as the principal source of improved cost-sensitive failure detection under the specified twenty-to-one penalty. The additional Pareto-based search offered no improvement over threshold-only adjustment despite reducing cost relative to the default. The penalties are hypothetical, and the synthetic dataset cannot quantify actual inspection or downtime losses. A single record-level partition also leaves temporal, equipment-level, and statistical robustness untested. Independent verification of the optimizer's internal multi-objective procedure was not possible from the available implementation records. Future research should evaluate auditable search mechanisms with repeated paired trials, chronological multi-machine validation, calibrated probabilities, and empirically elicited maintenance costs.

References

[1] A. Hamdi Alaoui, A. Meddaoui, and M. Hain, “AI-driven predictive maintenance for industry 4.0: A systematic review of models, methods, and challenges,” *The International Journal of Advanced Manufacturing Technology*, vol. 143, pp. 4861–4876, 2026, doi: 10.1007/s00170-026-17531-w.

[2] D. Guidotti, L. Pandolfo, and L. Pulina, “A systematic literature review of supervised machine learning techniques for predictive maintenance in Industry 4.0,” *IEEE Access*, vol. 13, pp. 102479–102504, 2025, doi: 10.1109/ACCESS.2025.3578686.

[3] P. Mallioris, E. Aivazidou, and D. Bechtsis, “Predictive maintenance in Industry 4.0: A systematic multi-sector mapping,” *CIRP J. Manuf. Sci. Technol.*, vol. 50, pp. 80–103, 2024, doi: 10.1016/j.cirpj.2024.02.003.

[4] Y. Yang and H. Wang, “Random forest-based machine failure prediction: A performance comparison,” *Applied Sciences*, vol. 15, no. 16, p. 8841, 2025, doi: 10.3390/app15168841.

[5] A. de Giorgio, G. Cola, and L. Wang, “Systematic review of class imbalance problems in manufacturing,” *J. Manuf. Syst.*, vol. 71, pp. 620–644, 2023, doi: 10.1016/j.jmsy.2023.10.014.

[6] Y. Yang and M. Z. Iqbal, “Cost-optimised machine learning model comparison for predictive maintenance,” *Electronics (Basel).*, vol. 14, no. 12, p. 2497, 2025, doi: 10.3390/electronics14122497.

[7] T. de Menezes e Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach, “Classifier calibration: A survey on how to assess and improve predicted class probabilities,” *Mach. Learn.*, vol. 112, no. 9, pp. 3211–3260, 2023, doi: 10.1007/s10994-023-06336-7.

[8] M. Yang and X. Bi, “Cost-aware calibration of classifiers,” *INFORMS Journal on Data Science*, vol. 4, no. 2, pp. 101–113, 2024, doi: 10.1287/ijds.2024.0038.

[9] B. Bischl *et al.*, “Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges,” *WIREs Data Mining and Knowledge Discovery*, vol. 13, no. 2, p. e1484, 2023, doi: 10.1002/widm.1484.

[10] L. Xing, J. Li, H. Ma, Z. Su, and J. Wu, “A relevant feature combination assisted bi-level evolutionary multi-objective algorithm for feature selection in classification,” *Sci. Rep.*, vol. 15, p. 39288, 2025, doi: 10.1038/s41598-025-23043-6.

[11] M. Chen and Z. Liu, “Predicting performance of students by optimizing tree components of random forest using genetic algorithm,” *Heliyon*, vol. 10, no. 12, p. e32570, 2024, doi: 10.1016/j.heliyon.2024.e32570.

[12] Y. Wang, M. Khodadadzadeh, and R. Zurita-Milla, “Spatial+: A new cross-validation method to evaluate geospatial machine learning models,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 121, p. 103364, 2023, doi: 10.1016/j.jag.2023.103364.

[13] X. Shao, B. Cai, Z. Zou, H. Shao, C. Yang, and Y. Liu, “Artificial intelligence enhanced fault prediction with industrial incomplete information,” *Mech. Syst. Signal Process.*, vol. 224, p. 112063, 2025, doi: 10.1016/j.ymssp.2024.112063.

[14] I. Hector and R. Panjanathan, “Predictive maintenance in Industry 4.0: A survey of planning models and machine learning techniques,” *PeerJ Comput. Sci.*, vol. 10, p. e2016, 2024, doi: 10.7717/peerj-cs.2016.

[15] S. F. Falkenberg and S. Spinler, “The role of novel data in maintenance planning: Breakdown predictions for material handling equipment,” *Comput. Ind. Eng.*, vol. 169, p. 108230, 2022, doi: 10.1016/j.cie.2022.108230.

[16] G. Vashishtha *et al.*, “A roadmap to fault diagnosis of industrial machines via machine learning: A brief review,” *Measurement*, vol. 242, p. 116216, 2025, doi: 10.1016/j.measurement.2024.116216.

[17] A. A. Murtaza, A. Saher, M. H. Zafar, S. K. R. Moosavi, M. F. Aftab, and F. Sanfilippo, “Paradigm shift for predictive maintenance and condition monitoring from Industry 4.0 to Industry 5.0: A systematic review, challenges and case study,” *Results in Engineering*, vol. 24, p. 102935, 2024, doi: 10.1016/j.rineng.2024.102935.

[18] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. L. P. Chen, “A survey on imbalanced learning: Latest research, applications and future directions,” *Artif. Intell. Rev.*, vol. 57, p. 137, 2024, doi: 10.1007/s10462-024-10759-6.

[19] M. Owusu-Adjei, J. B. Hayfron-Acquah, T. Frimpong, and G. Abdul-Salaam, “Imbalanced class distribution and performance evaluation metrics: A systematic review of prediction accuracy for determining model performance in healthcare systems,” *PLOS Digital Health*, vol. 2, no. 11, p. e0000290, 2023, doi: 10.1371/journal.pdig.0000290.

[20] Y. Mahale, S. Kolhar, and A. S. More, “Enhancing predictive maintenance in automotive industry: Addressing class imbalance using advanced machine learning techniques,” *Discover Applied Sciences*, vol. 7, p. 340, 2025, doi: 10.1007/s42452-025-06827-3.

[21] X. Li, J. Williams, C. Swanson, and T. Berg, “A machine learning approach to predictive maintenance: Remaining useful life and motor fault analysis,” *Comput. Ind. Eng.*, vol. 206, p. 111222, 2025, doi: 10.1016/j.cie.2025.111222.

[22] D. K. Yadav, A. Kaushik, and N. Yadav, “Predicting machine failures using machine learning and deep learning algorithms,” *Sustainable Manufacturing and Service Economics*, vol. 3, p. 100029, 2024, doi: 10.1016/j.smse.2024.100029.

[23] A. Osojnik, P. Panov, and S. Džeroski, "iSOUP-SymRF: Symbolic feature ranking with random forests in online multi-target regression and multi-label classification," *Mach. Learn.*, vol. 114, p. 34, 2025, doi: 10.1007/s10994-024-06718-5.

[24] M. Chen, R. Yu, Z. Liang, K. Li, and H. Qi, "A multiscale process-aware retention network for fault prediction in mixed-model production," *Comput. Ind.*, vol. 170, p. 104313, 2025, doi: 10.1016/j.compind.2025.104313.

[25] S. Gawde, S. Patil, S. Kumar, P. Kamat, and K. Kotecha, "An explainable predictive maintenance strategy for multi-fault diagnosis of rotating machines using multi-sensor data fusion," *Decision Analytics Journal*, vol. 10, p. 100425, 2024, doi: 10.1016/j.dajour.2024.100425.

[26] S. Gawde, S. Patil, S. Kumar, P. Kamat, K. Kotecha, and S. Alfarhood, "Explainable predictive maintenance of rotating machines using LIME, SHAP, PDP, ICE," *IEEE Access*, vol. 12, pp. 29345–29361, 2024, doi: 10.1109/ACCESS.2024.3367110.

[27] V. Warke, S. Kumar, A. Bongale, and K. Kotecha, "Robust tool wear prediction using multi-sensor fusion and time-domain features for the milling process using instance-based domain adaptation," *Knowl. Based. Syst.*, vol. 288, p. 111454, 2024, doi: 10.1016/j.knosys.2024.111454.

[28] P. Kamat, S. Kumar, and K. Kotecha, "DeepTool: A deep learning framework for tool wear onset detection and remaining useful life prediction," *MethodsX*, vol. 13, p. 102965, 2024, doi: 10.1016/j.mex.2024.102965.

[29] M. A. B. Syed, M. R. Hasan, N. I. Chowdhury, M. H. Rahman, and I. Ahmed, "A systematic review of time series algorithms and analytics in predictive maintenance," *Decision Analytics Journal*, vol. 15, p. 100573, 2025, doi: 10.1016/j.dajour.2025.100573.

[30] G. K. Koulinas, P. D. Paraschos, and D. E. Koulouriotis, "A machine learning framework for explainable knowledge mining and production, maintenance, and quality control optimization in flexible circular manufacturing systems," *Flex. Serv. Manuf. J.*, vol. 36, pp. 737–759, 2024, doi: 10.1007/s10696-024-09537-x.

[31] A. Giannoulidis, A. Gounaris, A. Naskos, N. Nikolaidis, and D. Caljouw, "Engineering and evaluating an unsupervised predictive maintenance solution: A cold-forming press case-study," *J. Intell. Manuf.*, vol. 36, pp. 2121–2139, 2025, doi: 10.1007/s10845-024-02352-z.

[32] D. Holtz, C. Kaymakci, D. Leuthe, S. Wenninger, and A. Sauer, "A data-efficient active learning architecture for anomaly detection in industrial time series data," *Flex. Serv. Manuf. J.*, vol. 38, pp. 618–649, 2026, doi: 10.1007/s10696-024-09588-0.

[33] D. Bromberger, A. Neuenfeldt Júnior, V. Garcia, and M. Francescatto, "Interpretable machine learning in industrial water pump failure prediction: The role of Shapley additive explanations values and data sampling techniques," *Eng. Appl. Artif. Intell.*, vol. 165, p. 113558, 2026, doi: 10.1016/j.engappai.2025.113558.

[34] L. Zhuang, A. Xu, and X.-L. Wang, "A prognostic driven predictive maintenance framework based on Bayesian deep learning," *Reliab. Eng. Syst. Saf.*, vol. 234, p. 109181, 2023, doi: 10.1016/j.ress.2023.109181.

[35] J. Yang, D. Zhu, Z. Wang, and Z. Zhang, "Decoupled focal loss and Bayesian temperature fusion XGBoost for imbalanced credit card fraud detection," *Sci. Rep.*, vol. 16, 2026, doi: 10.1038/s41598-026-64196-2.