Proceedings of CONF-MLA 2026 Symposium: Intelligent Systems and Automation: AI Models, IoT, and Robotic Algorithms
DOI: 10.54254/2755-2721/2026.LD38808

Cost-Sensitive Evaluation of Machine Learning Models for Manufacturing Equipment Failure Classification

Xu Yang

School of Civil Engineering and Architecture, Beijing Jiaotong University, Beijing, China
2981134736@qq.com

Abstract. High accuracy can conceal missed failures in imbalanced equipment data. This study compares logistic regression, a decision tree, and a random forest on 10,000 synthetic AI4I 2020 records using an 80:20 stratified split, five-fold training cross-validation, and a relative cost that weights missed failures more heavily than false alarms. Under default decision rules, the random forest achieves the highest test accuracy (0.9805) and ROC-AUC (0.9653), while the decision tree achieves the highest failure recall (0.6765). At false-negative-to-false-positive cost ratios of 5, 10, and 20, the decision tree has the lowest default-rule costs: 131, 241, and 461. Thresholds selected from out-of-fold training probabilities, without test-label tuning, reverse this comparison: the random forest achieves test costs of 95, 162, and 237. The results show why model selection should consider decision thresholds and error costs together. Because the dataset contains simulated contemporaneous failure labels, these findings concern failure-state classification rather than validated advance warning or demonstrated industrial savings.

Keywords: equipment failure classification, class imbalance, predictive maintenance, decision threshold, cost-sensitive evaluation

1. Introduction

Condition monitoring supports inspection and repair decisions, but a classifier's accuracy alone does not establish its maintenance value. Reviews of condition-based maintenance and machine learning describe applications in diagnostics and prognostics [1-3]. In imbalanced data, a classifier can achieve high accuracy by predicting the majority class while missing most failures [4]. The practical question is how its missed failures and false alarms change the decision to inspect equipment.

This study compares logistic regression (LR), a decision tree (DT), and a random forest (RF) under a common evaluation protocol. It asks whether the model preferred at its default decision rule remains preferable when thresholds are selected using training predictions and explicit error costs. The contribution is a reproducible comparison of evaluation choices, rather than a new learning algorithm. The target is failure status in a simulated record; no future warning horizon is defined.

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.LD38808

2. Related work

Jardine et al. [1] reviewed machinery diagnostics and prognostics, and Susto et al. [2] linked classifiers at different prediction horizons to maintenance costs. Carvalho et al. [3] surveyed machine learning methods for predictive maintenance. For imbalanced classification, He and Garcia [4] explained why aggregate performance can obscure minority-class errors. ROC analysis separates score discrimination from a particular operating point [5], while cost-sensitive learning makes the consequences of classification errors explicit [6]. The present comparison applies these distinctions to LR, a single DT, and the RF ensemble [7].

3. Data and methods

3.1. Data, features and split

The AI4I 2020 Predictive Maintenance Dataset contains 10,000 synthetic equipment records [8, 9], including 339 failures (3.39%) and 9,661 nonfailures (96.61%), with no missing values. Inputs comprise product type (L, M, or H), air temperature (K), process temperature (K), rotational speed (rpm), torque (N m), and tool wear (min). UDI and Product ID are excluded as identifiers. The failure-mode labels TWF, HDF, PWF, OSF and RNF are also excluded because they contain outcome information. A stratified 80:20 split with random_state=42 yields 8,000 training records, including 271 failures, and 2,000 test records, including 68 failures.

3.2. Preprocessing, models, and metrics

All models use the same preprocessing pipeline: StandardScaler for the five numerical inputs and one-hot encoding for product type. Preprocessing is fitted within each training fold. LR uses max_iter=1000 and random_state=42; DT uses default scikit-learn settings with random_state=42; RF uses 100 trees, random_state=42, and n_jobs=-1. Neither class weighting nor resampling is applied. Five-fold stratified cross-validation uses shuffle=True and random_state=42 on the training set only [10]. Reported standard deviations are sample standard deviations across the five folds. Default-rule results use each fitted pipeline's predict method; the test set is excluded from fitting and threshold selection.

Failure is the positive class. Reported metrics are accuracy, precision, recall, F1 score, and the area under the receiver operating characteristic curve (ROC-AUC). Confusion matrices report true negatives (TN), false positives (FP), false negatives (FN), and true positives (TP). Relative cost is defined as $C(r) = FP + r \times FN$, where one false positive costs one unit and r is 5, 10, or 20. These are hypothetical error penalties, not monetary estimates. An all-normal classifier would achieve 0.9661 accuracy on the complete dataset and 0.9660 on this test set, with zero failure recall.

3.3. Threshold selection without test-set tuning

For each classifier, a supplementary analysis generates out-of-fold probabilities for all 8,000 training records using five-fold stratified cross-validation with shuffle=True and random_state=42. For each cost ratio, it searches thresholds from 0.01 to 0.99 in increments of 0.01 and minimizes $FP + r \times FN$ on the pooled out-of-fold predictions. Ties favor the larger threshold. The pipeline is then fitted on the complete training set, and the selected threshold is applied to the held-out test probabilities. A record is classified as a failure only when its probability is strictly greater than the threshold. This

DOI: 10.54254/2755-2721/2026.LD38808

strict comparison matters for discrete scores, particularly those from RF. The procedure changes the decision rule, not the training loss or model hyperparameters.

4. Experimental results

4.1. Performance under default decision rules

Table 1. Held-out test metrics under default decision rules

Model	Accuracy	Precision	Recall	F1	ROC-AUC
LR	0.9675	0.6364	0.1029	0.1772	0.8994
DT	0.9785	0.6866	0.6765	0.6815	0.8328
RF	0.9805	0.8919	0.4853	0.6286	0.9653

At the default rules, RF leads in accuracy, precision, and ROC-AUC, whereas DT leads in recall and F1 score (Table 1). RF produces fewer false alarms but misses more failures than DT (Table 2). LR barely improves on the all-normal accuracy baseline and detects few failures. The ROC curves in Figure 1 show that RF ranks failure records well even though its default decision rule detects fewer failures than DT.

Table 2. Held-out confusion matrices and five-fold training summary

Model	TN	FP	FN	TP	CV F1 (mean ± SD)	CV ROC-AUC (mean ± SD)
LR	1928	4	61	7	0.3119 ± 0.0291	0.8938 ± 0.0188
DT	1911	21	22	46	0.6868 ± 0.0526	0.8308 ± 0.0313
RF	1928	4	35	33	0.5889 ± 0.0571	0.9665 ± 0.0135

The cross-validation summaries show the same broad recall-related pattern through F1 score (Table 2). They describe variability across training folds, not uncertainty in the held-out model ranking. With only 68 positive test records, small accuracy differences should be interpreted cautiously.

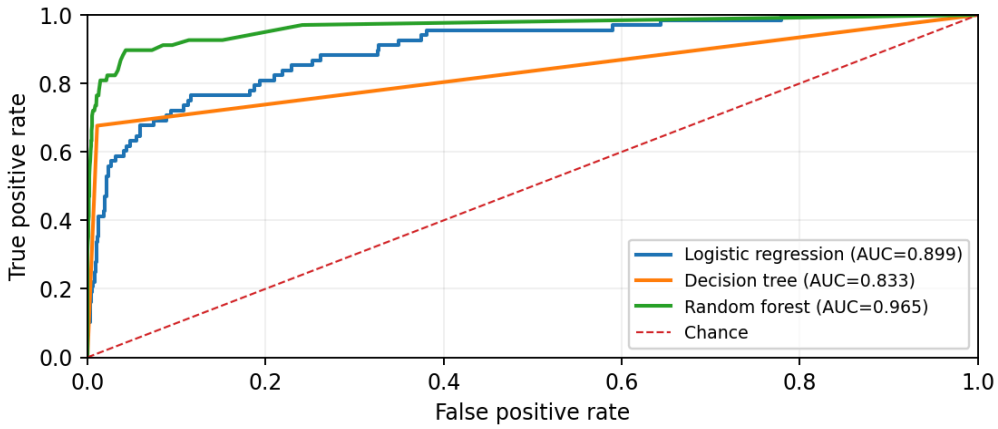


Figure 1. Held-out ROC curves computed from the saved per-record probabilities

DOI: 10.54254/2755-2721/2026.LD38808

4.2. Default-threshold cost comparison

At r = 5, 10, and 20, respectively, default-rule costs are 309, 614, and 1,224 for LR; 131, 241, and 461 for DT; and 179, 354, and 704 for RF. DT therefore has the lowest cost in all three scenarios. Comparing its cost, 21 + 22r, with the RF cost, 4 + 35r, gives a crossover at r = 17/13, approximately 1.31. This crossover describes the observed test confusion matrices; it is not a general maintenance decision threshold.

4.3. Training-selected thresholds and held-out cost

Table 3. Training-selected thresholds and held-out errors and costs

Model	Cost ratio r	Threshold	FP	FN	Cost
LR	5	0.20	39	36	219
LR	10	0.11	88	26	348
LR	20	0.06	190	19	570
DT	5	0.99	21	22	131
DT	10	0.99	21	22	241
DT	20	0.99	21	22	461
RF	5	0.24	30	13	95
RF	10	0.13	72	9	162
RF	20	0.12	77	8	237

Threshold selection changes the preferred classifier: RF has the lowest held-out cost at every examined ratio (Table 3). For r = 5, its selected threshold of 0.24 lowers the cost by 84 relative units, from 179 to 95. LR also improves after threshold adjustment, while the selected DT thresholds leave its test classifications unchanged. The cost ranking under a default rule therefore does not establish the ranking after threshold selection.

4.4. Interpretation of the cost comparison

The RF improvement reflects a change in the trade-off between false alarms and missed failures. At r = 5, RF at the selected threshold detects nine more failures and generates nine more false alarms than DT. The reduction of nine missed failures outweighs those additional false alarms under the assigned penalty. This comparison would not necessarily satisfy a maintenance team with a strict limit on inspection capacity.

The selected LR and RF thresholds decrease as missed failures receive greater weight. The DT assigns discrete leaf probabilities and offers fewer effective operating points: its selected threshold is 0.99 for all three ratios under the strict greater-than rule and the stated tie-breaking convention. These values are grid-search outcomes, not universal optimal thresholds. The test costs evaluate thresholds selected on training data; they do not establish how stable those thresholds would be in another sample.

DOI: 10.54254/2755-2721/2026.LD38808

5. Discussion and limitations

Accuracy, recall, ROC-AUC and relative cost answer different questions. Accuracy is dominated by normal records; recall measures failure detection at one operating point; ROC-AUC describes score discrimination across operating points. Cost additionally depends on the consequences assigned to each error. Here, RF combines strong ranking performance with a default rule that misses many failures, so threshold adjustment changes its cost more than the initial model comparison suggests.

Several limitations restrict the result. AI4I is synthetic, and its target labels the current record rather than a future breakdown. A single stratified holdout and five-fold training procedure do not quantify threshold-selection uncertainty across repeated splits or independent equipment populations. Only three baseline models and a finite threshold grid are examined, without probability calibration or structural hyperparameter tuning. The cost model assigns the same penalty to every missed failure and omits intervention effectiveness, inspection capacity, maintenance scheduling, and actual downtime costs. RF's lower observed cost therefore applies to this split and these assumptions.

A deployment study would require longitudinal observations, a defined warning horizon, and evaluation on equipment or time periods excluded from development. Site-specific inspection and failure costs would then allow alarm frequency, available maintenance capacity, and intervention outcomes to be evaluated together. Repeated or nested validation would also help assess how sensitive the selected thresholds are to the training sample.

6. Conclusion

Using the public AI4I 2020 dataset, we compared logistic regression, a decision tree, and a random forest for binary classification of equipment failure. The comparison considered standard predictive metrics as well as a simple representation of maintenance costs.

Class imbalance makes accuracy look reassuring even when failures are missed. All three models exceed 96% test accuracy, yet their failure recall ranges from 0.1029 to 0.6765. Accuracy alone does not describe their usefulness for identifying faults.

The metrics do not rank the models in the same way. Random forest achieves the highest accuracy, precision, and ROC-AUC, whereas the decision tree has the highest recall and F1. Logistic regression produces few false alarms but misses 61 failures. A high accuracy score therefore does not by itself imply greater value for maintenance.

When a missed failure is assigned 5, 10, or 20 times the cost of a false alarm, the decision tree has the lowest relative cost among the three models at the fixed threshold. In such a setting, finding more failures can be more valuable than a small gain in accuracy.

Model choice should follow the maintenance objective. Failure consequences, inspection costs, and production constraints determine which errors deserve the most weight; none of these three classifiers can be called universally optimal from this experiment.

This basic, reproducible experiment shows that a classifier's value depends on how its errors relate to the maintenance task, as well as on its predictive metrics. With operational data, future studies could use actual costs to define the evaluation criteria and investigate threshold selection and model explanations.

DOI: 10.54254/2755-2721/2026.LD38808

References

[1] Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing, 20*(7), 1483–1510. https://doi.org/10.1016/j.ymssp.2005.09.012

[2] Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2015). Machine learning for predictive maintenance: A multiple classifiers approach. *IEEE Transactions on Industrial Informatics, 11*(3), 812-820. https://doi.org/10.1109/TII.2014.2349359

[3] Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., et al. (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

[4] 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

[5] 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

[6] Elkan, C. (2001). The foundations of cost-sensitive learning. *Proceedings of IJCAI 2001*, 973–978. https://dblp.org/rec/conf/ijcai/Elkan01

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

[8] UCI Machine Learning Repository. (2020). AI4I 2020 predictive maintenance dataset. https://doi.org/10.24432/C5HS5C

[9] Matzka, S. (2020). Explainable artificial intelligence for predictive maintenance applications. *2020 Third International Conference on Artificial Intelligence for Industries*, 69–74. https://doi.org/10.1109/AI4I49448.2020.00023

[10] Pedregosa, F., Varoquaux, G., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830. https://www.jmlr.org/papers/v12/pedregosa11a.html