Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Application of Machine Learning in Optimizing Industrial Assembly Lines: A Chronologically Validated Study of Predictive Quality Control

Aida Allahverdi[1]

aida.allahverdi@students.uniroma2.eu

University of Rome Tor Vergata

Siavash Akbarzadeh[1]

University of Rome Tor Vergata

Research Article

Keywords: Machine learning, Industrial assembly lines, Predictive quality control, Concept drift, Chronological validation, SHAP interpretability

Posted Date: October 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11317815/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

Industrial assembly lines generate large volumes of in-process sensor data whose predictive value for quality control is routinely assessed with random cross-validation, a protocol that ignores the chronological structure of production and can therefore overstate deployment performance. This study presents a chronologically validated assessment of machine learning for optimizing an industrial assembly line, using the public SECOM semiconductor manufacturing dataset (1,567 units, 590 process signals, 6.64% defect rate). Five model families — logistic regression, support vector machines, random forests, extreme gradient boosting, and multilayer perceptrons — were tuned by nested grid search and compared under three evaluation protocols. Under 5×5-fold random cross-validation, random forests achieved an area under the ROC curve (AUC) of 0.768 ± 0.010, suggesting strong predictability. However, a rolling-origin temporal evaluation across five production blocks collapsed the mean AUC of all models to 0.514–0.599, demonstrating that concept drift, not learnable signal, drives the optimistic random-CV estimates. On a final 30% future hold-out, an F2-optimal decision threshold converted the gradient-boosted model into a practical screening tool: recall 0.885 at precision 0.071, flagging 68.6% of units for inspection while catching 23 of 26 defects, a 30.8% reduction in inspection load. SHAP analysis identified a single dominant process signal. The results argue that chronology-safe validation should be mandatory when reporting machine learning systems for assembly-line optimization, and that even under drift, screening-level utility remains.

Introduction

Modern assembly lines are instrumented with hundreds of sensors whose readings are logged for every unit produced, creating data-rich environments in which quality decisions can, in principle, be made before rather than after end-of-line testing [1, 2]. Within the broader smart-manufacturing agenda, machine learning has been promoted as the natural mechanism for converting this in-process data into predictions of unit-level quality, dynamic inspection prioritization, and evidence-based process adjustment [3, 4, 5].

A substantial literature accordingly reports machine learning classifiers that predict pass/fail outcomes from process signals, with apparently high accuracy. The overwhelming majority of these studies evaluate their models with random k-fold cross-validation, in which units from the same production period appear in both the training and the test folds. This protocol silently violates the temporal structure of manufacturing: a model deployed on an assembly line is always trained on the past and executed on the future. Concept drift — changes in process behaviour caused by maintenance events, material batches, tool wear, and recipe updates — is well documented in industrial monitoring research [6], and class imbalance typical of low-defect-rate lines further complicates both training and honest evaluation [7, 8].

The SECOM dataset, donated by Michael McCann and Adrian Johnston and distributed through the UCI Machine Learning Repository, has become a de facto benchmark for defect prediction on a real

assembly line: 1,567 consecutively produced units described by 590 anonymized process signals, of which only 6.64% fail final quality control [9, 10, 11]. Reported results on SECOM vary widely, and almost uniformly rest on random cross-validation. To our knowledge, the sensitivity of SECOM performance estimates to the choice of validation protocol has received little systematic attention — a gap with direct consequences for how the feasibility of machine-learning-driven line optimization is judged by practitioners.

This study therefore evaluates machine learning for assembly-line optimization under an explicitly deployment-oriented design. Our contributions are fourfold. First, we provide a statistically tested comparison of five model families (logistic regression, support vector machines, random forests, extreme gradient boosting, and multilayer perceptrons) under a common tuned preprocessing pipeline. Second, we quantify the gap between random cross-validation and chronological validation, including a rolling-origin evaluation over five consecutive production blocks, and show that the former overstates deployment performance dramatically. Third, we demonstrate that even under temporal validation, an operating threshold selected on a validation slice yields a screening tool with high defect recall and a quantifiable reduction in inspection load. Fourth, we use SHAP attribution to identify the process signals that dominate the learned decision, connecting the statistical exercise back to line-level process monitoring priorities.

The remainder of the paper is organized as follows. The Methods section describes the dataset, preprocessing, models, and the three validation protocols. The Results and discussion section presents the comparative evaluation, the temporal analysis, the screening operating point, and the interpretability analysis, followed by practical implications and limitations. The Conclusions section summarizes the findings and outlines future work.

Methods

Dataset

The SECOM dataset records a fabrication line of a semiconductor plant over roughly three months of production (19 July to 17 October 2008). Each of the 1,567 units is described by 590 process sensor readings and a binary quality outcome recorded at final electrical test: 1,463 units pass and 104 fail (6.64% defect rate) [9, 10, 11]. The production file is ordered chronologically by completion time, which we verified from the accompanying timestamps. Sensor identifiers are anonymized, so features are referenced by their column index (Sensor 1–590). Figure 2 characterizes the dataset: the severe class imbalance, the distribution of missing values across sensors (4.5% of all sensor cells are missing, concentrated in a subset of signals), and the weekly fail-rate trajectory, which already suggests non-stationarity over the production window. Table 1 summarizes the dataset and the data split used throughout the study.

Preprocessing

All preprocessing was fitted on training data only and applied unchanged to validation and test data. Missing sensor values were imputed with the training median of the corresponding feature. Features with zero variance in the training set were removed, retaining 468 of 590 sensors; the surviving features were standardized to zero mean and unit variance. No feature selection based on the outcome variable was applied, so that all models faced an identical learning problem. The complete pipeline is summarized in Fig. 1.

Models

Five model families spanning different inductive biases were compared: L2-regularized logistic regression (LR) with class weighting; a radial-basis-function support vector machine (SVM) with class weighting [12]; a random forest (RF) of 300 trees with balanced sub-sample weights [13]; extreme gradient boosting (XGBoost) with the class ratio as positive-class weight [14, 15]; and a multilayer perceptron (MLP) with early stopping. Hyperparameters of every model were selected by grid search with five-fold stratified cross-validation on the training set, scored by average precision to account for class imbalance. All models were implemented in scikit-learn and XGBoost [16].

Validation protocols

Three complementary protocols were used (Fig. 1). (i) Random cross-validation: 5×5-fold stratified CV on the training set (the first 70% of production time), reported as mean±standard deviation over five repeats; model rankings were tested with the Friedman test and Wilcoxon signed-rank post-hoc comparisons with Bonferroni correction [17, 18]. (ii) Rolling-origin temporal evaluation: the production timeline was divided into five consecutive test blocks; for each block, models were trained on all preceding production and evaluated on the block, mimicking periodic retraining and forward deployment. (iii) Chronological hold-out with deployment threshold: models were trained on the first 60% of production, the decision threshold was selected on the subsequent 10% (validation slice) by maximizing the F2 measure — favouring recall — and the resulting operating point was applied unchanged to the final 30% (future) hold-out.

Evaluation metrics and statistics

Because the defect rate is low, we report threshold-free metrics (area under the ROC curve, AUC; average precision, AP) alongside threshold-dependent operational metrics (recall, precision, and F1 for the fail class; balanced accuracy; Matthews correlation coefficient) [8]. The majority-class baseline (predict pass always) attains an AP equal to the defect rate and zero recall and is included in all comparisons. Statistical significance of model ranking differences was assessed with the Friedman test over the repeated CV rounds [17], followed by pairwise Wilcoxon tests with Bonferroni correction [18].

Interpretability

Global feature attribution for the best-performing boosted model on the chronological hold-out was computed with SHAP TreeExplainer values [19]. Because the variance filter reindexes features, attributions were mapped back to the original sensor identifiers to support line-level interpretation.

Table 1
SECOM dataset summary and data split.

Item	Value
Units (total)	1,567
Pass / fail units	1,463 / 104 (6.64% fail)
Process sensors (raw)	590
Missing sensor cells	4.5%
Sensors retained (variance filter)	468
Production period	19 Jul – 17 Oct 2008
Chronological split (train / test)	1096 / 471 units (78 / 26 fails)

Results and discussion

Model comparison under random cross-validation

Table 2 and Fig. 3 report the 5×5-fold random CV results on the training period. Random forests achieved the best mean AUC (0.768 ± 0.010) and average precision (0.234 ± 0.018), followed by XGBoost (AUC 0.748 ± 0.021); the MLP performed clearly worst. The Friedman test rejected the hypothesis of equivalent model performance ($\chi^2(4) = 19.36$, $p = 0.0007$). Pairwise Wilcoxon tests with Bonferroni correction did not separate the random forest from XGBoost ($p = 0.75$), while the advantage over the MLP, LR, and SVM was significant or near-significant at this number of repeats. Taken at face value — and this is exactly how such results are usually read — the ensemble models appear to deliver usable defect prediction on this line.

Table 2
5×5-fold random cross-validation on the training production period (mean ± std over five repeats).

Model	Selected hyperparameters	ROC-AUC	Average precision
LR	C: 1	0.664 ± 0.030	0.181 ± 0.031
SVM	C: 1, gamma: 0.01	0.682 ± 0.034	0.182 ± 0.015
RF	max_depth: 20, min_samples_leaf: 5	0.768 ± 0.010	0.234 ± 0.018
XGB	learning_rate: 0.05, max_depth: 5	0.748 ± 0.021	0.224 ± 0.021
MLP	alpha: 0.001, hidden_layer_sizes: (128, 64)	0.554 ± 0.020	0.139 ± 0.008

Rolling-origin temporal evaluation: the optimism gap

Table 3 and Fig. 6 show what happens when the same tuned models are evaluated strictly forward in time. Mean temporal AUC collapses to 0.599 for XGBoost, 0.564 for the random forest, 0.530 for LR, 0.533 for the MLP, and 0.514 for the SVM — values at or barely above the 0.5 chance level. Per-block results are strikingly unstable: LR ranges from 0.285 to 0.828 across blocks, and the best model changes from block to block. The gap between the random-CV estimate (0.768) and the temporal estimate (0.564) for the random forest is approximately 0.20 AUC points.

Two mechanisms explain this collapse. First, random CV allows near-duplicate neighbouring units (produced minutes apart under identical conditions) to appear in both training and test folds, which leaks short-term process state and inflates apparent generalization. Second, the line is genuinely non-stationary — visible already in the weekly fail-rate trajectory (Fig. 2c) — so that relationships learned in one production period transfer only weakly to the next. Both mechanisms are precisely what deployment faces, and neither is visible under random CV. We conclude that random-CV performance on this benchmark (and, by extension, on similarly collected line data) should be treated as an upper bound produced by temporal leakage rather than as evidence of deployable accuracy.

Table 3
Rolling-origin temporal evaluation: ROC-AUC when training on all production preceding each test block.

Test block (2008)	LR	SVM	RF	XGB	MLP
08–17–08–28	0.475	0.465	0.516	0.545	0.556
08–28–09 – 02	0.534	0.558	0.443	0.577	0.535
09 – 02–09–20	0.531	0.512	0.668	0.662	0.420
09–20–09–29	0.285	0.567	0.555	0.641	0.315
10 – 05–10–17	0.828	0.467	0.637	0.570	0.839
Mean	0.530	0.514	0.564	0.599	0.533

Deployment operating point on the future hold-out

Although ranking quality on the future hold-out is weak (Fig. 4 shows ROC and precision–recall curves close to the diagonal), a deployed classifier is operated at a chosen threshold, and the threshold can be chosen for screening rather than discrimination. Table 4 reports the hold-out results at the F2-optimal thresholds selected on the validation slice. The XGBoost screening rule attains recall 0.885 (catching 23 of 26 defects) at precision 0.071, flagging 323 of 471 units (68.6%) for inspection. Relative to screening every unit, this spares 145 units (30.8% of the hold-out) from inspection at the cost of 3 missed defects; the random forest yields a similar operating point (recall 0.846). The precision achieved (0.071) exceeds the 5.5% base rate of the hold-out period only modestly — an honest characterization of the screening value that survives drift on this line.

The practical reading is nuanced. A recall-first screening rule that still inspects roughly three quarters of units offers limited but real savings, and — more importantly — concentrates attention on the highest-risk quarter of production. Whether this is economically useful depends on the relative costs of inspection, escapes, and false alarms, which vary by line; the transparent reporting of the full confusion matrix (Table 4) is intended to let a plant derive its own cost-based verdict.

Table 4
Future hold-out at deployment thresholds selected on the chronological validation slice (F2-optimal).

Model	Threshold	Recall (fail)	Precision (fail)	F1 (fail)	Flagged units	Missed defects
LR	0.100	0.000	0.000	0.000	21 (4.5%)	26 of 26
SVM	0.084	0.038	0.027	0.032	37 (7.9%)	25 of 26
RF	0.091	0.846	0.060	0.113	365 (77.5%)	4 of 26
XGB	0.002	0.885	0.071	0.132	323 (68.6%)	3 of 26
MLP	0.123	0.192	0.069	0.102	72 (15.3%)	21 of 26

Which process signals drive the model

Figure 5 and Table 5 report the SHAP attribution of the XGBoost model on the future hold-out. One signal — Sensor 60 — dominates the model output, with a mean absolute SHAP value of 1.46, roughly 5× that of the next most influential sensor; the remaining attributions decay smoothly. Because SECOM sensor identities are anonymized we cannot attribute this dominance to a specific physical process step, but the shape of the attribution profile — one dominant signal followed by a long, thin tail — matches the typical finding of manufacturing feature-importance studies and gives the line engineer a concrete monitoring priority: the process step behind the dominant sensor, and its calibration drift over time, deserve first attention when models of this kind degrade.

Table 5
Top-10 process signals by mean absolute SHAP value (XGBoost, future hold-out).

Sensor (original index)	Mean |SHAP|
Sensor 60	1.459
Sensor 461	0.272
Sensor 334	0.228
Sensor 426	0.179
Sensor 34	0.168
Sensor 103	0.156
Sensor 65	0.155
Sensor 478	0.153
Sensor 130	0.142
Sensor 1	0.137

Practical implications and limitations

For practitioners optimizing assembly lines with machine learning, our results support three concrete recommendations. First, insist on chronological validation before believing any reported accuracy: the random-CV-to-temporal gap observed here (roughly 0.20 AUC points) is large enough to change go/no-go decisions. Second, evaluate models at deployment thresholds selected on a chronological validation slice, not at the default 0.5 — under drift, calibrated thresholds are what convert weak rankings into usable screening rules. Third, use attribution profiles to prioritize process monitoring and sensor maintenance, since a small number of signals carries most of the learned signal.

The study has limitations. It concerns a single production line over three months, with only 104 defect events in total and as few as 4 defects in some temporal blocks, which limits both statistical power and

the complexity of models that can be justified. Sensor anonymization prevents physical attribution of the dominant feature. Finally, the study is retrospective; although the rolling-origin design mimics periodic retraining, genuinely adaptive online schemes were out of scope and are natural future work.

Conclusions

We evaluated five machine learning model families for defect prediction on the SECOM assembly-line benchmark under random cross-validation, rolling-origin temporal evaluation, and a threshold-based future hold-out. Random cross-validation suggested strong performance (random forest AUC 0.768), but chronological validation showed that this apparent performance is largely an artifact of temporal leakage: forward-in-time AUC fell to 0.599 at best, with pronounced block-to-block instability. A recall-oriented operating threshold nonetheless yielded a screening rule catching 23 of 26 future defects while sparing 30.8% of units from inspection, and SHAP attribution identified a single dominant process signal. The central message for the machine-learning-in-manufacturing community is methodological: validation protocol, not model choice, is the decisive factor in honest estimates of assembly-line optimization performance. Future work should target drift-adaptive and online learning, transfer across lines and recipes, cost-sensitive threshold optimization integrated with line balancing and inspection scheduling, and attribution tied to identifiable process physics.

Declarations

Competing interests

The authors declare that they have no competing interests.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author Contribution

Author Contributions StatementManuscript: Application of Machine Learning in Optimizing Industrial Assembly Lines: A Chronologically Validated Study of Predictive Quality ControlAida Allahverdi[1]* and Siavash Akbarzadeh[11] Department of Industrial Engineering, University of Rome Tor Vergata, Rome, Lazio, ItalyThe individual contributions of the authors to this work, stated in accordance with the CRediT (Contributor Roles Taxonomy) classification, are as follows.Aida Allahverdi (AA): Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization.Siavash Akbarzadeh (SA): Conceptualization, Methodology, Supervision, Writing – review & editing.Both authors contributed to Writing – review & editing of the revised manuscript, read and

approved the final version, and agree to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

Acknowledgements

Not applicable.

Availability of data and materials

The SECOM dataset analysed during the current study is publicly available from the UCI Machine Learning Repository (DOI: 10.24432/C54305) under a Creative Commons BY 4.0 licence. The analysis code generated during the current study is available from the corresponding author on reasonable request.

References

1. Zhong RY, Xu X, Klotz E, Newman ST (2017) Intelligent manufacturing in the context of industry 4.0: a review. Engineering 3(5):616–630. https://doi.org/10.1016/J.ENG.2017.05.015
2. Wang J, Ma Y, Zhang L, Gao RX, Wu D (2018) Deep learning for smart manufacturing: methods and applications. J Manuf Syst 48:144–156. https://doi.org/10.1016/j.jmsy.2018.01.003
3. Qin SJ (2012) Survey on data-driven industrial process monitoring and diagnosis. Annu Rev Control 36(2):220–234. https://doi.org/10.1016/j.arcontrol.2012.09.004
4. Reis MS, Gins G (2017) Industrial process monitoring in the big data/industry 4.0 era: from detection, to diagnosis, to prognosis. Processes 5(3):48. https://doi.org/10.3390/pr5030048
5. Kang Z, Catal C, Tekinerdogan B (2020) Machine learning applications in production lines: a systematic literature review. Comput Ind Eng 149:106773. https://doi.org/10.1016/j.cie.2020.106773
6. Gama J, Žliobaitė I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. ACM Comput Surv 46(4):1–37. https://doi.org/10.1145/2523813
7. He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284. https://doi.org/10.1109/TKDE.2008.236
8. 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
9. Dua D, Graff C (2019) UCI machine learning repository. University of California, Irvine. https://archive.ics.uci.edu

10. McCann M, Johnston A (2008) SECOM dataset. https://doi.org/10.24432/C54305. UCI Machine Learning Repository
11. May GS, Spanos CJ (2006) Fundamentals of semiconductor manufacturing and quality control. Wiley-Interscience, Hoboken
12. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297. https://doi.org/10.1007/BF00994018
13. Breiman L (2001) Random forests. Mach Learn 45(1):5–32. https://doi.org/10.1023/A:1010933404324
14. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29(5):1189–1232. https://doi.org/10.1214/aos/1013203451
15. 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. https://doi.org/10.1145/2939672.2939785
16. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay É (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
17. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701. https://doi.org/10.1080/01621459.1937.10503522
18. Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics Bull 1(6):80–83. https://doi.org/10.2307/3001968
19. Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. Adv Neural Inf Process Syst 30:4765–4774

Figures

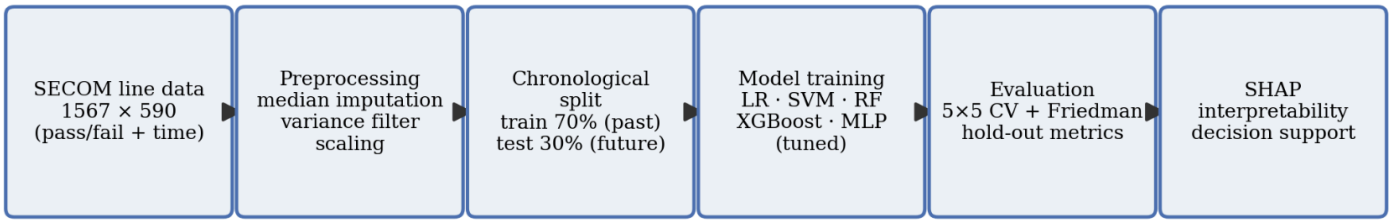

Figure 1

End-to-end analysis pipeline, from raw line data to chronologically validated evaluation and interpretability.

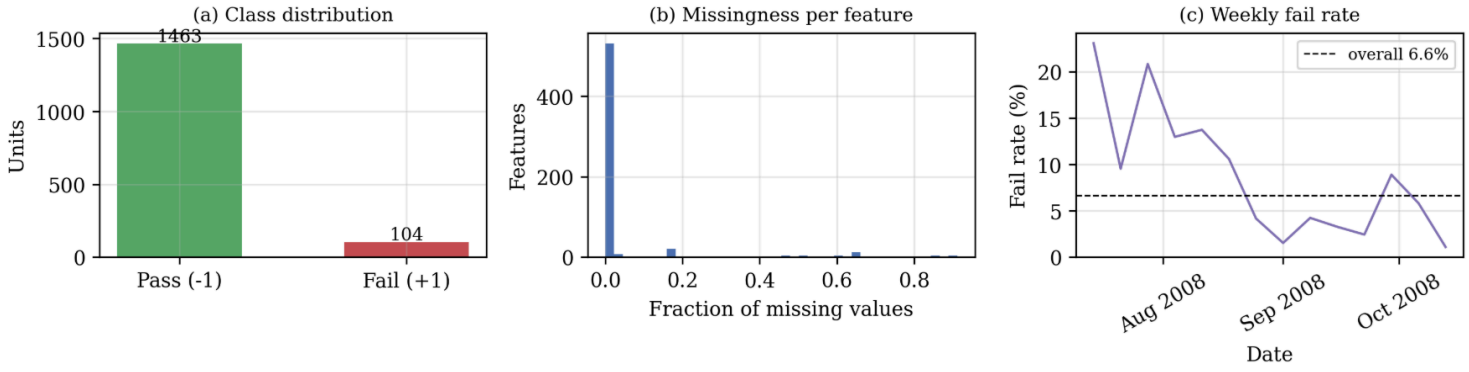

Figure 2

Dataset characterization: (a) class distribution; (b) fraction of missing values per sensor; (c) weekly fail rate with the overall mean (dashed).

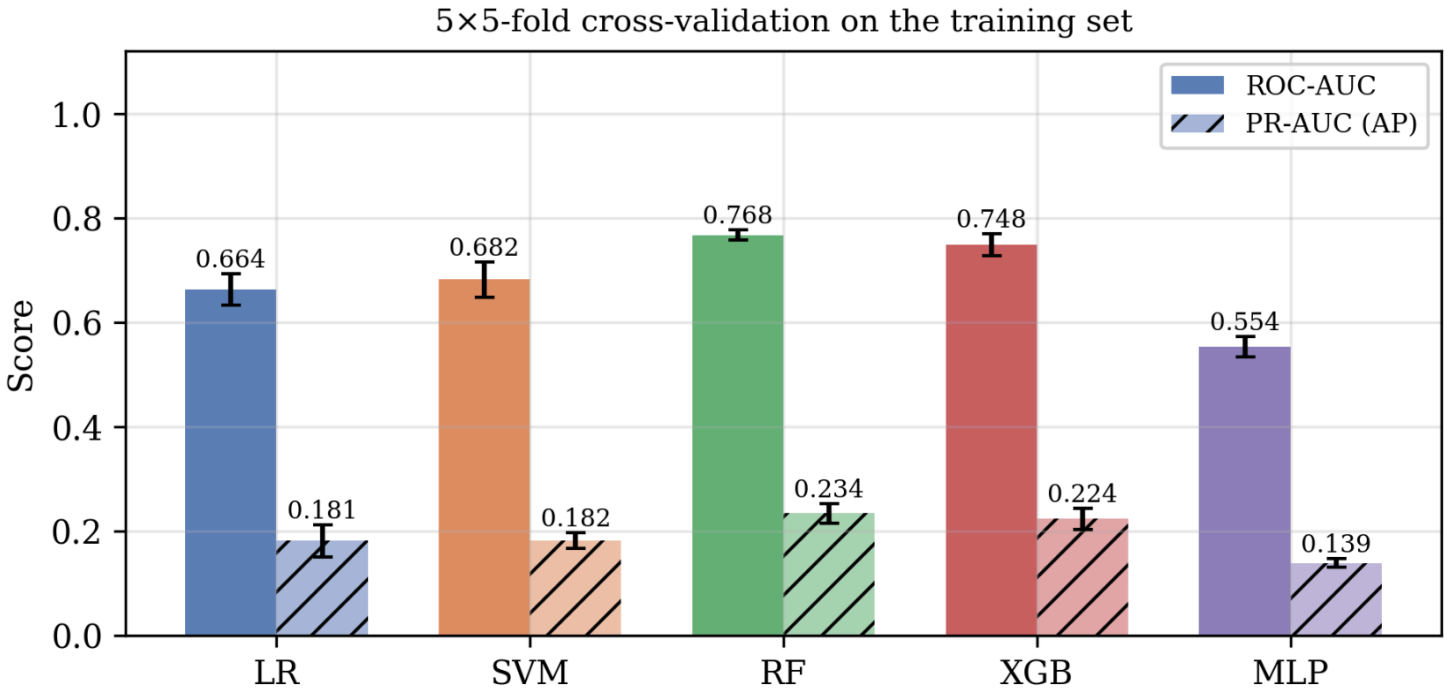

Figure 3

Random cross-validation comparison (ROC-AUC and average precision, mean±std over five repeats).

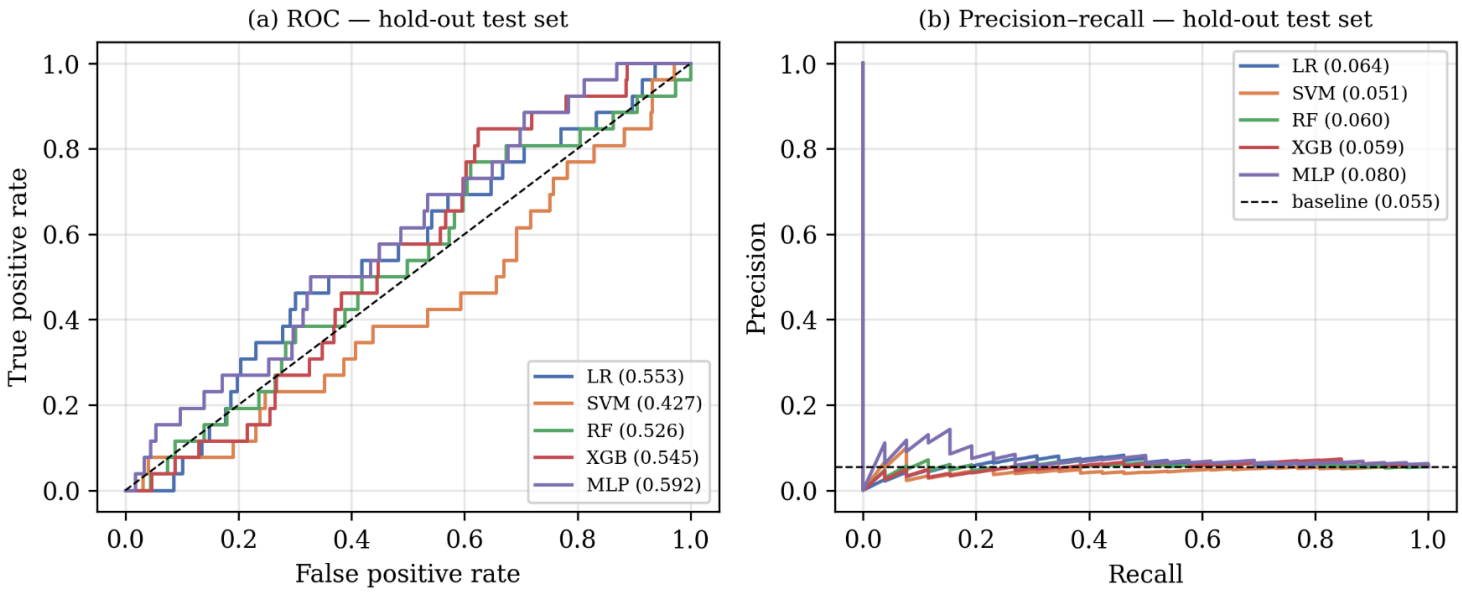

Figure 4

(a) ROC and (b) precision–recall curves of all models on the future hold-out; the majority-class base rate is shown dashed.

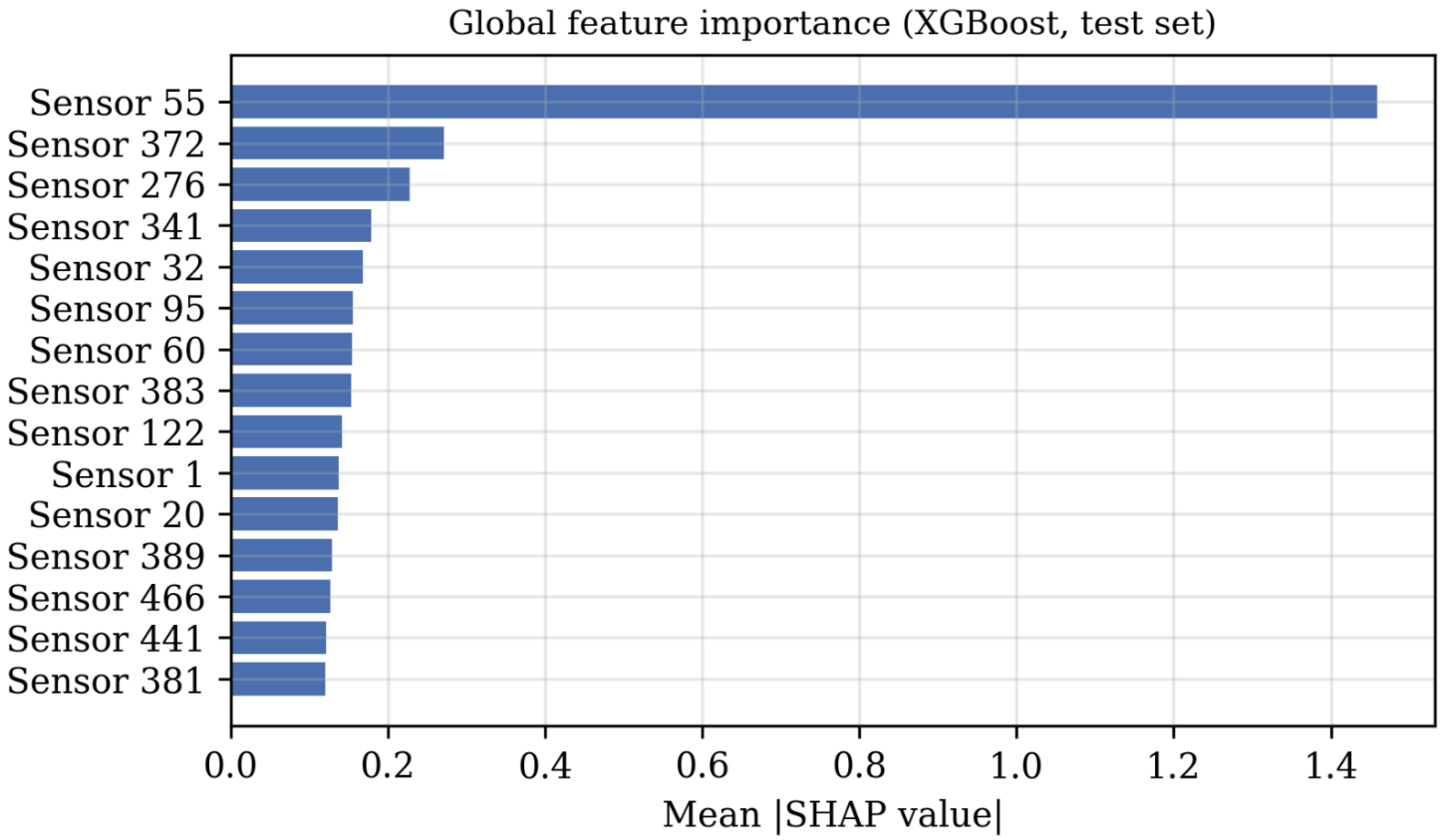

Figure 5

Global SHAP feature attribution of the XGBoost model on the future hold-out (top 15 sensors by mean absolute SHAP value).

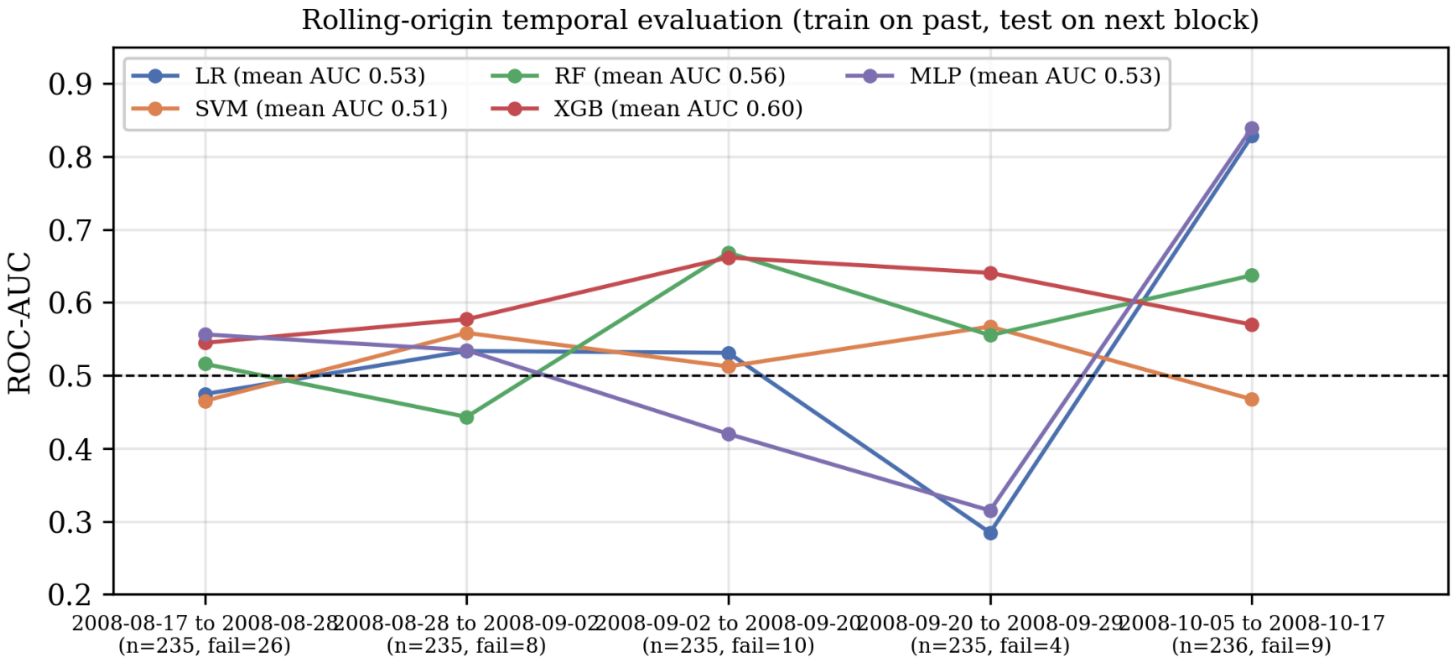

Figure 6

Rolling-origin temporal evaluation across five production blocks: per-block ROC-AUC per model (dashed line: chance level).