Supplementary Material for: FusionXNet-Edge: A Physics-Guided Lightweight Framework for Predictive Maintenance Under Cross-Domain Deployment Shift

Aman Sharma[1*], Kwan Yong Sim[2] and Sivachandran Chandrasekaran[1]

[1*]Department of Computing Technologies, Swinburne University of Technology, John Street, Melbourne, Victoria, 3122, Australia.
[2]Faculty of Engineering, Computing and Science, Swinburne University of Technology Sarawak, Jalan Simpang Tiga, Kuching, Sarawak, 93350, Malaysia.

*Corresponding author(s). E-mail(s): 104893880@student.swin.edu.au;
Contributing authors: ksim@swinburne.edu.my;
schandrasekaran@swin.edu.au;

This Supplementary Material provides the extended evidence base for the main manuscript. Selected compact tables and diagnostic figures are also included in the main manuscript and Appendix A to improve readability and reviewer traceability. The supplementary file retains the broader supporting material, including raw class distributions, leakage-controlled split-policy checks, extended baseline plots, full source-wise transfer tables, ablation details, threshold-sensitivity evidence, and reproducibility notes.

1 Broader AI adoption gaps in smart manufacturing

Table S1 summarises broader manufacturing-level gaps commonly discussed in the AI and Industry 4.0 literature. These gaps provide high-level context for the present study, but they do not directly drive the design of the proposed model. Instead, they are used to motivate the need for compact, standardised, and deployment-oriented PdM models capable of operating under heterogeneous sensing and operational conditions.

Table 1 Broader AI adoption gaps in smart manufacturing

Gap category	Summary	Representative studies
Sustainability and resource efficiency	Manufacturing systems continue to face challenges in integrating circular-economy principles, real-time sustainability indicators, and resource-aware optimisation at scale.	(Priore et al. 2001; Bag et al. 2021; Jamwal et al. 2022; Yang et al. 2023)
Customisation and production flexibility	High-mix and low-volume production remains constrained by rigid process flows and limited adaptability in current automation frameworks.	(Tu et al. 2001; Radziwon et al. 2014; Kotsiopoulos et al. 2021; Psarommatis et al. 2022; Parvanda and Kala 2023)
Human–machine collaboration and workforce skills	Digital transformation requires closer human–AI collaboration, but adoption is slowed by skill gaps, interface misalignment, and limited operator trust.	(Zhong et al. 2017; Sony and Naik 2020; Ahmed et al. 2022; Leng et al. 2022)
Scalability and standardisation of AI/DL models	AI and DL models remain difficult to standardise across heterogeneous manufacturing environments due to differences in data schemas, machine types, and operating regimes.	(Park et al. 2016; Scime and Beuth 2018; Ruiz-Sarmiento et al. 2020; Tercan and Meisen 2022; Kumar et al. 2023)
Data-driven decision-making and real-time analytics	Supporting reliable real-time decision-making from heterogeneous, high-frequency industrial data remains a major challenge in intelligent manufacturing workflows.	(Priore et al. 2001; Diez-Olivan et al. 2019; Ren et al. 2019)

2 Dataset and Feature-Support Diagnostics

This section provides the extended dataset-level evidence supporting the preprocessing and feature-engineering strategy described in the main manuscript. Representative physics-guided feature diagnostics are summarised as a compact multi-panel figure in the main manuscript, while this supplementary section retains the individual raw class-distribution plots, leakage-controlled split-policy illustrations, feature-diagnostic plots, and broader dataset-quality checks for traceability.

2.1 Raw Class Distributions

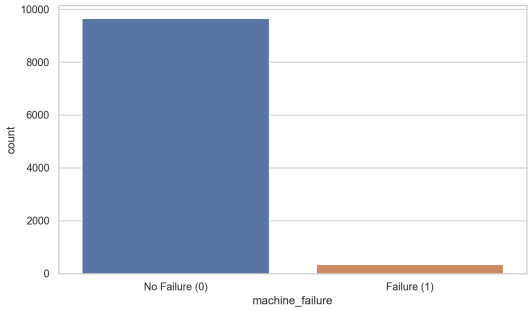

Fig. 1 Raw class distribution in the AI4I dataset.

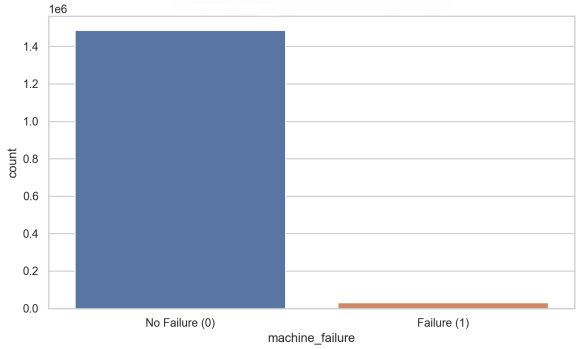

Fig. 2 Raw class distribution in the MetroPT3 dataset.

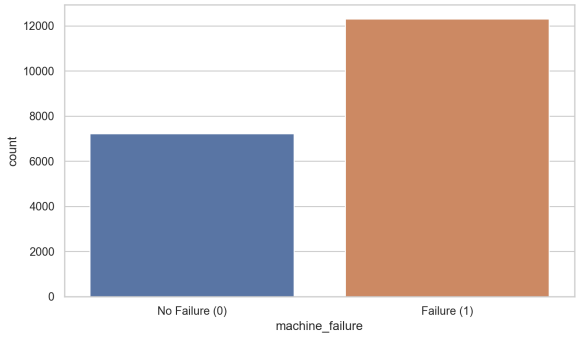

Fig. 3 Raw class distribution in the Engine dataset.

2.2 leakage-controlled Split Policy

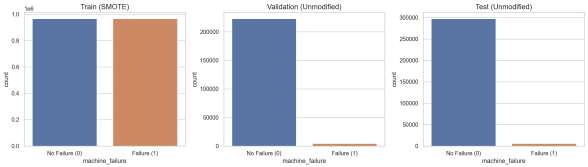

Fig. 4 leakage-controlled data split for the MetroPT3 dataset.

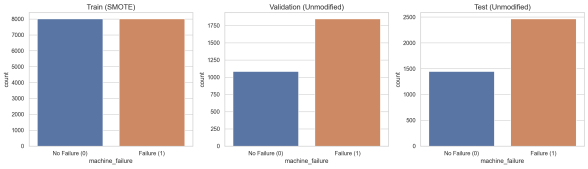

Fig. 5 leakage-controlled data split for the Engine dataset.

2.3 Representative Physics-Guided Feature Diagnostics

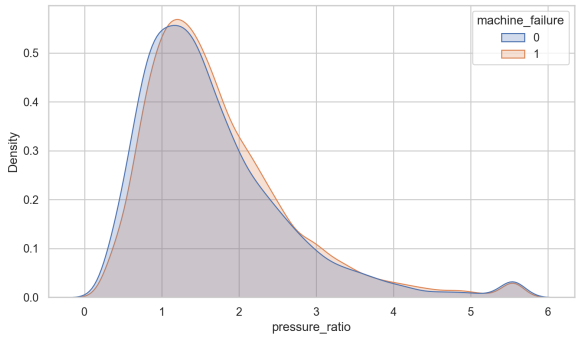

Fig. 6 Class-wise distribution of pressure ratio ($p_{\text{lub}}/p_{\text{cool}}$) in the Engine dataset.

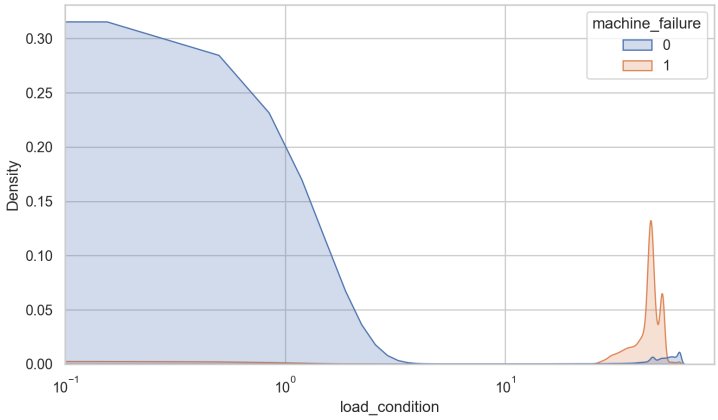

Fig. 7 Class-wise distribution of load condition ($I_{\text{motor}} \times P_{\text{tp2}}$) in the MetroPT3 dataset.

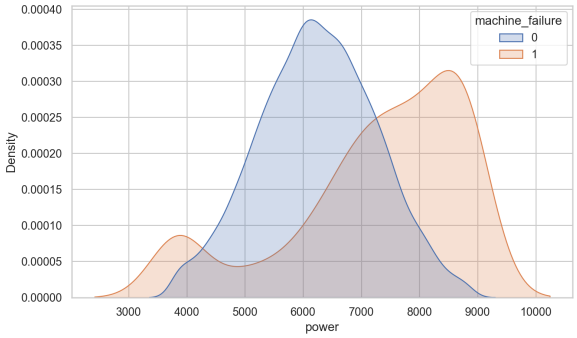

Fig. 8 Class-wise distribution of power ($\tau \cdot \omega$) in the AI4I dataset.

2.4 Data Quality and Integrity Checks

Before any learning, each dataset undergoes a brief quality audit to ensure the shared pipeline is both safe and comparable. We verify the label definitions and class priors, scan for duplicate rows and impossible values (e.g., negative pressures or temperatures), and compute simple distributional summaries for each feature. Missingness is modest across all three domains and is handled by the train-fitted mean imputer described above. To surface obvious multicollinearity, we produce a correlation map per dataset, which helps explain benign differences in downstream feature importances without changing the procedure itself. Finally, we confirm that post-engineering feature statistics (mean and variance after scaling) remain well-behaved; this matters because our calibrations assume unimodal, standardised inputs rather than heavy-tailed outliers.

3 Shared Experimental Settings and Model Consistency

This section defines the shared experimental configuration used across the AI4I, MetroPT3, and Engine datasets to ensure controlled, comparable evaluation. All datasets follow an identical preprocessing pipeline, fixed input representation, and consistent optimisation strategy. By enforcing uniform training conditions, the observed performance differences are attributable to dataset characteristics rather than to variations in model design or training procedures. This controlled setup follows common reproducible benchmarking principles used in PdM and industrial PdM evaluation (Tao et al. 2018; Wu et al. 2024).

In addition to performance consistency, computational efficiency is treated as a primary design constraint. The proposed FusionXNet-Edge model maintains a fixed, lightweight architecture ($\approx 16k$ parameters) across all datasets, ensuring comparable capacity while enabling deployment in resource-constrained settings. Detailed analyses of complexity, latency, and footprint are provided in the supplementary tables to support resource-aware evaluation.

3.1 Shared Hyperparameters

Table 2 Hyperparameters used uniformly across all datasets.

Component	Setting
Initial learning rate	2×10^{-4}
Learning rate decay	Steps: 1000; decay rate: 0.9; staircase = True
Batch size	64
Loss function	Focal Loss ($\gamma = 1.5$, $\alpha = 0.4$); fallback to BCE if unstable
Early stopping	Monitor validation AUC; patience = 8; restore best weights
Threshold selection	G-Mean optimised on validation fold; fixed for test and cross-domain transfer
Latency reporting	Mean $\pm$ std; p50 and p95 after warm-up

3.2 Main Symbols

3.3 Pipeline Consistency Across Datasets

All datasets share an identical structural configuration, including a fixed input dimensionality (1×14), a binary output formulation, and a consistent model topology. This strict alignment ensures that architectural capacity, feature representation, and learning dynamics remain invariant across domains, enabling fair cross-domain comparison without confounding factors.

Table 3 Main symbols used in the methodology.

Symbol	Definition
$x \in \mathbb{R}^{1\times 14}$	Single-step input vector in the fixed schema
x_{phys}	Shared physics-guided slice of x (indices [5:8))
f_θ	FusionXNet-Edge mapping (model)
$\hat{y} = p_\theta(y{=}1\|x)$	Predicted failure probability
$\tilde{y}$	Final binary decision after thresholding
α, γ	Focal-loss balance and focusing parameters
τ^*	Validation-optimised threshold
$\text{GMean}(\tau)$	$\sqrt{\text{TPR}(\tau)\text{TNR}(\tau)}$

3.4 Implementation details

Each experiment stores the trained model, feature scaler, calibrated threshold, and evaluation outputs. Performance diagnostics include ROC and precision–recall curves, confusion matrices, and prediction distributions. All results are aggregated into a unified summary file, and schema validation ensures consistent feature ordering and dimensionality during both training and inference.

3.5 Feature schema, units, and derivations

Harmonising the three domains into a single $(1, 14)$ interface requires agreeing on units and small numeric conventions. Raw rpm is left as-is for the base channels, but torque–speed power uses a rad/s conversion to respect physical units; ratios add a 10^{-6} term in the denominator purely to avoid undefined values.

Table 4 Pipeline consistency across datasets.

Aspect	MetroPT3	AI4I	Engine
Input shape	(1, 14)	(1, 14)	(1, 14)
Output shape	(1)	(1)	(1)
# Parameters	16,061	16,061	16,061
Layer sequence	Identical	Identical	Identical
Features (count/type)	14 / numeric	14 / numeric	14 / numeric
Target type	Binary failure	Binary failure	Binary failure

3.6 Shared Experimental Settings Across Datasets

3.7 Calibration and operating point

Beyond reporting rank-based metrics (AUROC and AUPRC), a single operating threshold is calibrated per source domain to reflect balanced error costs. Specifically, the threshold $\tau \in [0, 1]$ is swept on the validation split, and the selected operating

Table 5 Shared experimental settings across all datasets and dataset-specific variations arising only from sensor availability. Model capacity and evaluation protocol remain fixed.

Shared across all domains	**Dataset-specific variations**
Preprocessing order; $(1, 14)$ input; shared physics-guided slice [5:8) fixed	AI4I: torque–speed power and simple stress proxy; MetroPT3: pressure gradient and load proxies; Engine: coolant–oil ΔT and pressure ratios.
Adam with exponential decay; batch size 64; focal loss; early stopping on validation AUC	Unit conversions and 10^{-6} stabilising terms are numerical conventions and do not affect rank-based metrics.
Threshold via G-Mean per source; identical latency procedure	One notebook variant monitored `val_loss`; internal checks indicated no material effect on AUC/F1.

point is defined as

$$\tau^\star = \arg\max_{\tau \in [0,1]} \sqrt{\mathrm{TPR}(\tau)\,\mathrm{TNR}(\tau)}. \tag{1}$$

This validation-based rule is the same operating-threshold criterion used throughout the main paper.

The criterion is suitable when class priors differ across datasets because it balances sensitivity and specificity symmetrically. In scenarios with asymmetric costs, such as when missed failures are more costly than false alarms, we also report cost-weighted operating points for $\lambda \in \{0.6, 0.7\}$ in the Appendix. These supplementary results do not alter the primary analysis, but document alternative operating points that practitioners may prefer. This consistent thresholding strategy maintains comparable deployment decisions across domains without post-hoc target adjustment, aligning with deployment-oriented evaluation.

4 Performance of the Meta-Model

Table 6 Performance of the meta-model baseline (CNN + BiLSTM + RF blending) across the AI4I, Engine, and MetroPT3 datasets. The meta-model aggregates predictions from heterogeneous base learners using a fixed evaluation threshold.

Dataset	Model	Acc.	AUROC	AUPRC	F2	Params	Size (MB)	ms/sample
AI4I	Meta (CNN+BiLSTM+RF)	0.9725	0.9717	0.6790	0.5438	233,862	11.535	8.632
Engine	Meta (CNN+BiLSTM+RF)	0.6561	0.6938	0.7956	0.8316	233,862	49.210	8.415
MetroPT3	Meta (CNN+BiLSTM+RF)	0.9987	0.99994	0.99682	0.9823	233,862	11.054	8.248

Table 7 Performance of the lightweight FusionXNet baseline (16k parameters, projection-only fusion, without the auxiliary physics-guided feature branch). This model serves as a compact reference without physics-guided inputs.

Dataset	Model	Acc.	AUROC	AUPRC	F2	Params	Size (MB)	ms/sample
AI4I	FusionXNet (16k_base)	0.9380	0.9617	0.6381	0.6279	16,041	0.074	11.537
Engine	FusionXNet (16k_base)	0.6579	0.6927	0.7933	0.7058	16,013	0.074	11.562
MetroPT3	FusionXNet (16k_base)	0.9928	0.99976	0.98987	0.9294	16,139	0.074	1.182

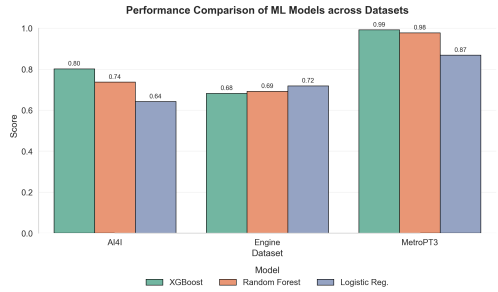

Fig. 9 Performance comparison of classical ML models across the AI4I, Engine, and MetroPT3 datasets.

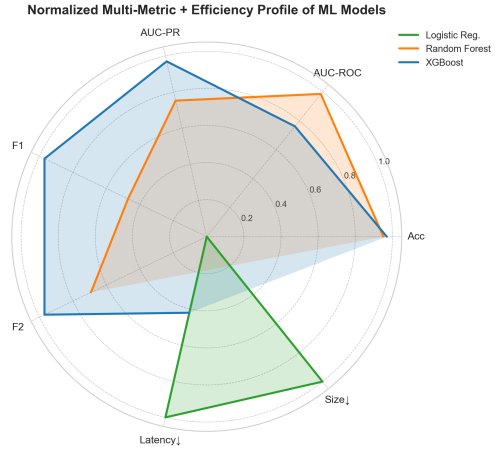

Fig. 10 Radar comparison of normalised performance and efficiency metrics for ML models.

5 Extended In-Domain Diagnostics

This section reports supplementary in-domain ROC, precision-recall, and prediction-probability diagnostics for AI4I, Engine, and MetroPT3. These plots support the consolidated in-domain results discussed in the main manuscript.

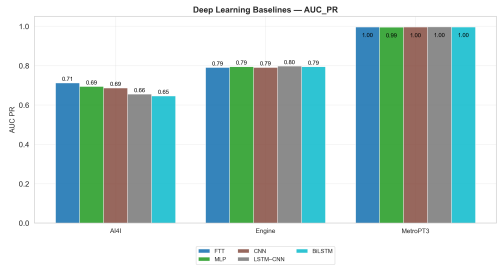

Fig. 11 AUPRC performance of DL baseline models across the AI4I, Engine, and MetroPT3 datasets.

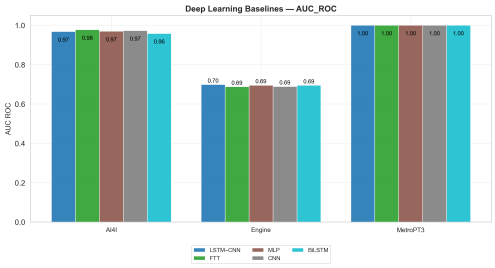

Fig. 12 AUROC performance of DL baseline models across the AI4I, Engine, and MetroPT3 datasets.

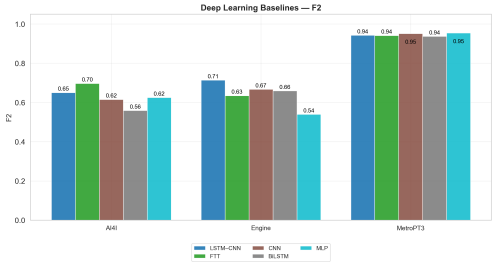

Fig. 13 F2-score performance of DL baseline models across the AI4I, Engine, and MetroPT3 datasets.

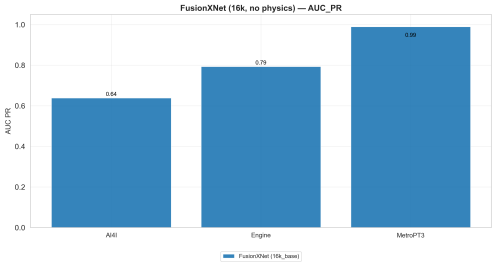

Fig. 14 AUPRC performance of the lightweight FusionXNet (16k parameters, no physics-guided features).

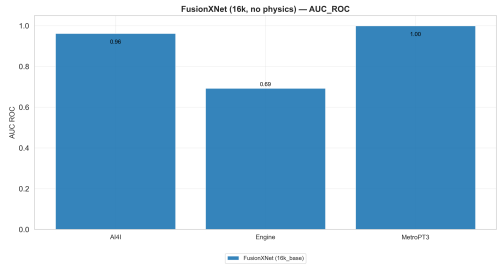

Fig. 15 AUROC performance of the lightweight FusionXNet (16k parameters, no physics-guided features).

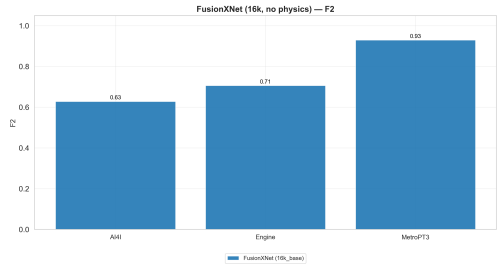

Fig. 16 F2-score of the lightweight FusionXNet (16k parameters, no physics-guided features).

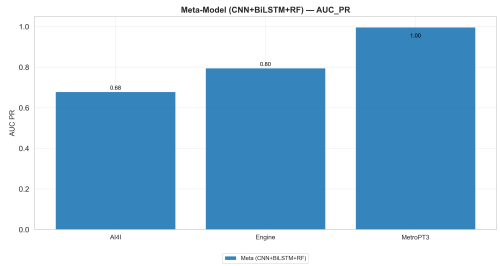

Fig. 17 AUPRC performance of the meta-model (CNN + BiLSTM + RF).

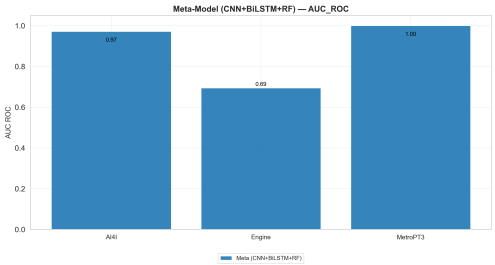

Fig. 18 AUROC performance of the meta-model (CNN + BiLSTM + RF).

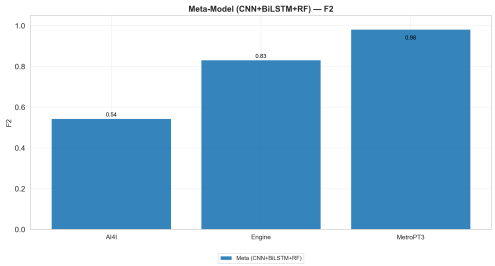

Fig. 19 F2-score of the meta-model (CNN + BiLSTM + RF).

5.1 AI4I Diagnostics

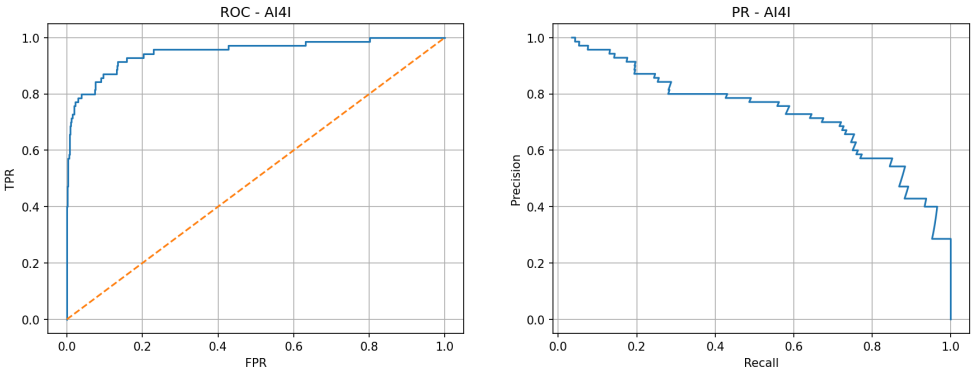

Fig. 20 Supplementary ROC and precision–recall curves of FusionXNet-Edge on the AI4I dataset.

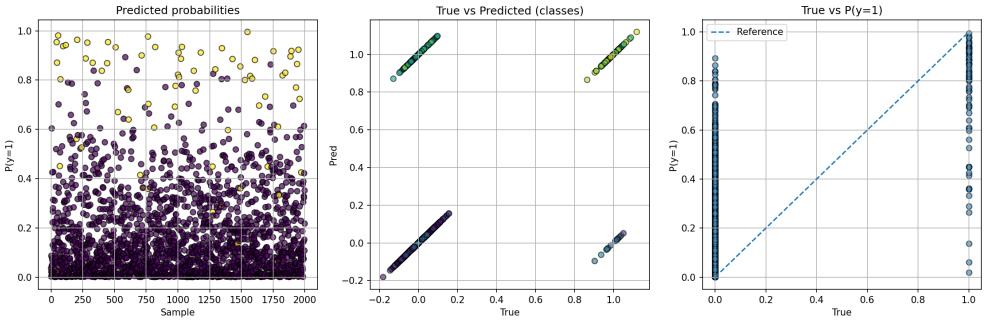

Fig. 21 Supplementary prediction-probability diagnostics on AI4I: (a) predicted probability distribution, (b) true vs. predicted classes, and (c) true vs. $P(y{=}1)$.

5.2 Engine Diagnostics

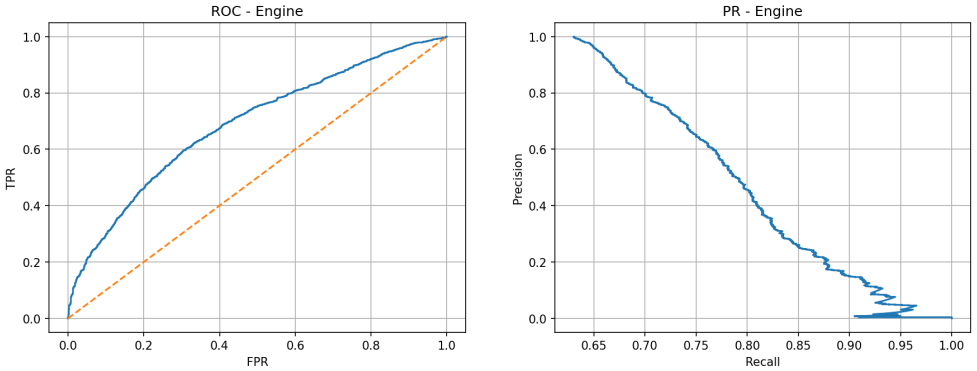

Fig. 22 Supplementary ROC and precision–recall curves of FusionXNet-Edge on the Engine dataset.

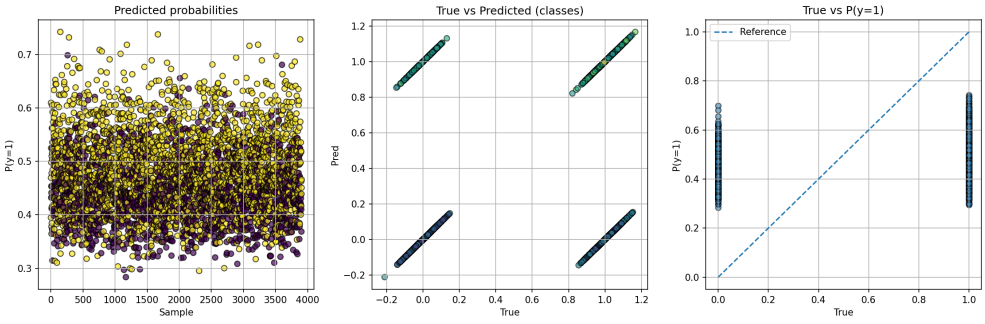

Fig. 23 Supplementary prediction-probability diagnostics on the Engine dataset.

5.3 MetroPT3 Diagnostics

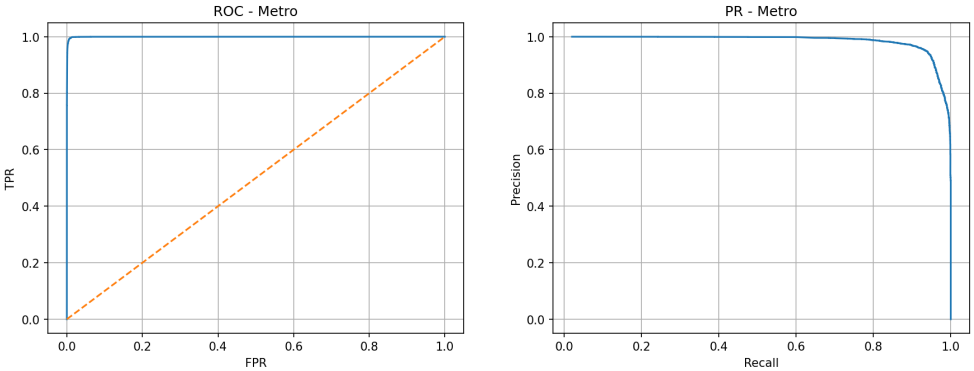

Fig. 24 Supplementary ROC and precision–recall curves of FusionXNet-Edge on the MetroPT3 dataset.

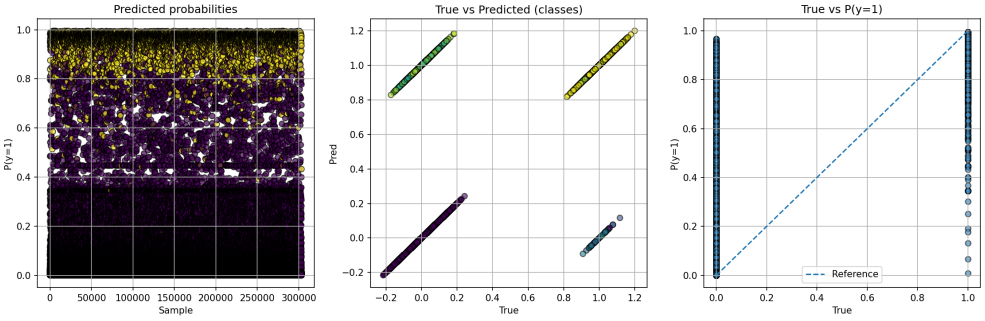

Fig. 25 Supplementary prediction-probability diagnostics on MetroPT3.

6 Correlation Structure of Leak-Free Engineered Features

These correlation maps provide additional feature-space context for the engineered variables used in the shared $(1, 14)$ input schema. They are included to support the methodology discussion without increasing the visual load of the main paper.

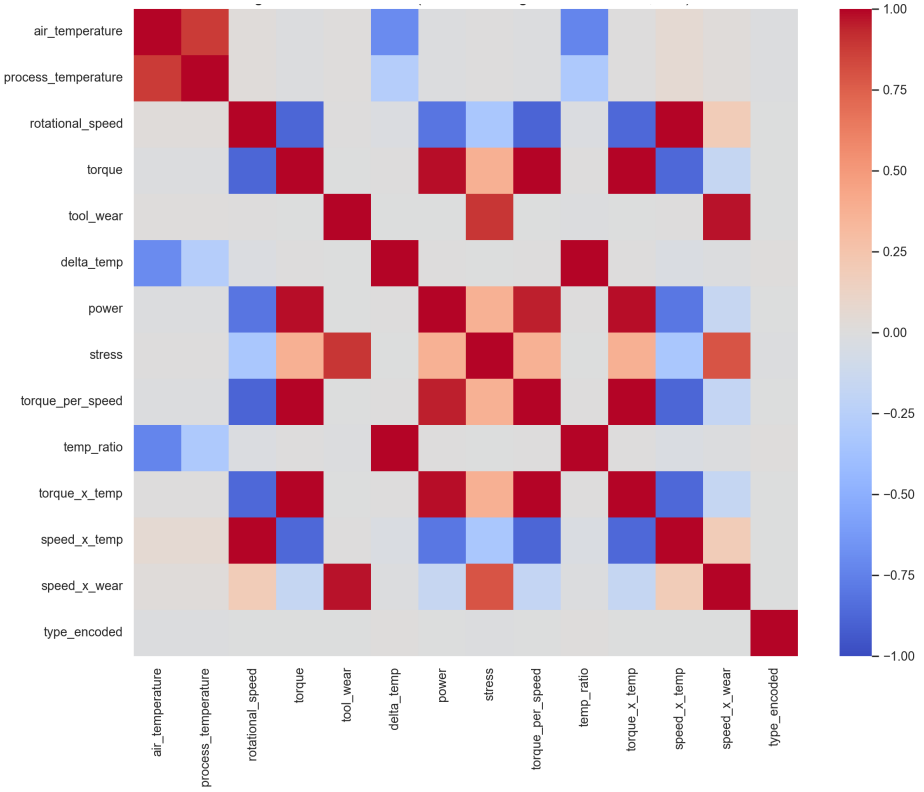

Fig. 26 Correlation matrix of engineered features for the AI4I dataset.

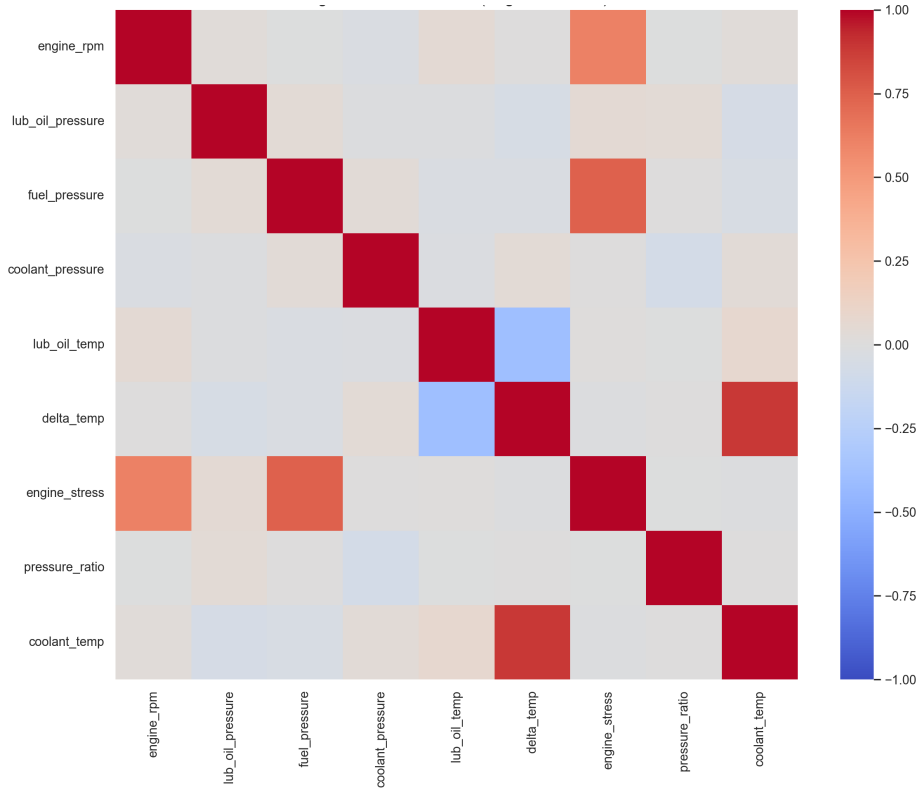

Fig. 27 Correlation matrix of engineered features for the Engine dataset.

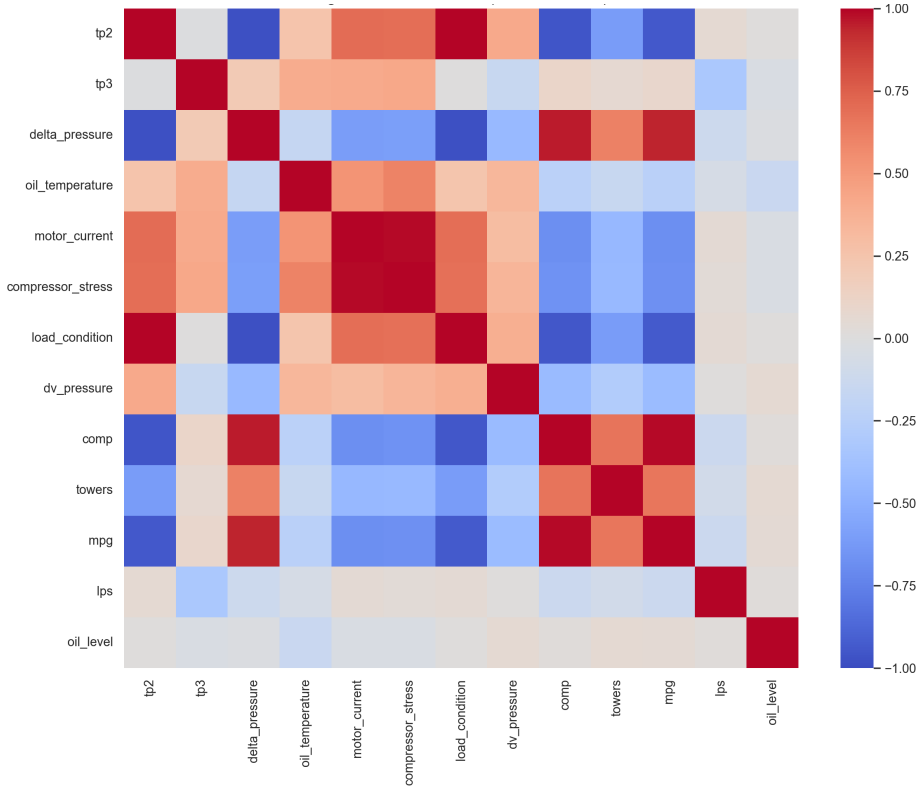

Fig. 28 Correlation matrix of engineered features for the MetroPT3 dataset.

7 Supplementary Benchmark-Support Tables

This section provides supplementary benchmark-support tables for directly verifiable baseline and comparison models. These tables are included for traceability and completeness and are not intended to replace the main comparative results reported in the manuscript body.

7.1 Threshold Optimisation and G-Mean Summary

The threshold-adjusted results are reported for sensitivity analysis only. The primary cross-domain protocol retains the source-domain threshold to represent deployment without target-domain recalibration or target-label tuning.

Table 8 Threshold optimisation and G-Mean evaluation for directly verifiable baseline and comparison models. The table reports the fixed inference threshold, the G-Mean-optimised threshold, and the resulting accuracy, recall, F2-score, and specificity.

Dataset	Model	Thr. (fixed)	Thr. (G-Mean)	Acc.	Recall	F2	Spec.
AI4I	FTT	0.43	0.11	0.934	0.912	0.674	0.935
AI4I	LSTM–CNN	0.43	0.15	0.869	0.912	0.525	0.867
AI4I	MLP	0.43	0.13	0.901	0.868	0.567	0.902
AI4I	CNN	0.43	0.27	0.868	0.941	0.537	0.865
AI4I	BiLSTM	0.43	0.12	0.852	0.926	0.503	0.849
AI4I	XGBoost	0.43	0.026	0.912	0.897	0.608	0.913
AI4I	Random Forest	0.43	0.214	0.921	0.912	0.637	0.921
AI4I	Logistic Regression	0.43	0.436	0.876	0.794	0.482	0.879
Engine	LSTM–CNN	0.46	0.51	0.645	0.625	0.650	0.678
Engine	CNN	0.46	0.48	0.632	0.615	0.639	0.662
Engine	BiLSTM	0.46	0.45	0.652	0.653	0.672	0.651
Engine	FTT	0.46	0.48	0.624	0.584	0.613	0.692
Engine	MLP	0.46	0.38	0.646	0.641	0.662	0.656
Engine	Logistic Regression	0.46	0.542	0.636	0.592	0.621	0.711
Engine	Random Forest	0.46	0.514	0.624	0.620	0.641	0.631
Engine	XGBoost	0.46	0.528	0.604	0.596	0.618	0.618
MetroPT3	MLP	0.41	0.73	0.997	0.994	0.968	0.997
MetroPT3	CNN	0.41	0.87	0.998	0.994	0.973	0.998
MetroPT3	LSTM–CNN	0.41	0.45	0.994	0.999	0.944	0.994
MetroPT3	FTT	0.41	0.52	0.997	0.995	0.967	0.997
MetroPT3	BiLSTM	0.41	0.54	0.997	0.995	0.964	0.997
MetroPT3	XGBoost	0.41	0.08	0.995	1.000	0.950	0.995
MetroPT3	Random Forest	0.41	0.53	0.997	0.996	0.970	0.997
MetroPT3	Logistic Regression	0.41	0.75	0.985	0.991	0.859	0.984

7.2 Training and Testing Sample Utilisation

These threshold-adjusted values are provided for sensitivity analysis only. The primary cross-domain protocol retains the source-domain threshold to represent deployment without recalibrating the target domain.

Table 9 Training and testing sample utilisation for directly verifiable baseline and comparison models, including row-capping parameters, total training samples used, test-set size, and minority-class prevalence.

Dataset	Model	Train Used	Cap Rows	Test Size	Prevalence
AI4I	FTT	12560	200000	2000	0.034
AI4I	LSTM–CNN	12560	200000	2000	0.034
AI4I	CNN	12560	200000	2000	0.034
AI4I	MLP	12560	200000	2000	0.034
AI4I	BiLSTM	12560	200000	2000	0.034
AI4I	XGBoost	12560	250000	2000	0.034
AI4I	Random Forest	12560	250000	2000	0.034
AI4I	Logistic Regression	12560	400000	2000	0.034
Engine	LSTM–CNN	16012	200000	3908	0.631
Engine	CNN	16012	200000	3908	0.631
Engine	BiLSTM	16012	200000	3908	0.631
Engine	FTT	16012	200000	3908	0.631
Engine	MLP	16012	200000	3908	0.631
Engine	Logistic Regression	16012	400000	3908	0.631
Engine	Random Forest	16012	250000	3908	0.631
Engine	XGBoost	16012	250000	3908	0.631
MetroPT3	MLP	300000	300000	303390	0.020
MetroPT3	CNN	300000	300000	303390	0.020
MetroPT3	LSTM–CNN	300000	300000	303390	0.020
MetroPT3	FTT	300000	300000	303390	0.020
MetroPT3	BiLSTM	300000	300000	303390	0.020
MetroPT3	XGBoost	250000	250000	303390	0.020
MetroPT3	Random Forest	250000	250000	303390	0.020
MetroPT3	Logistic Regression	400000	400000	303390	0.020

7.3 Model Complexity Summary

Table 10 Model complexity summary for directly verifiable baseline and comparison models, including total parameters, trainable parameters, parameter footprint in millions, and total model size on disk.

Dataset	Model	Params (total)	Trainable	Params (M)	Size (MB)
AI4I	LSTM–CNN	628,225	628,225	0.628	2.402
AI4I	BiLSTM	529,665	529,665	0.530	2.023
AI4I	XGBoost	39,514	39,514	0.040	1.536
AI4I	Random Forest	137,854	137,854	0.138	10.595
AI4I	Logistic Regression	9	9	0.000009	0.0009
AI4I	FusionXNet (16k_base)	16,041	15,865	0.016	0.074
AI4I	Meta (CNN+BiLSTM+RF)	233,862	233,862	0.234	11.535
Engine	XGBoost	90,002	90,002	0.090	3.174
Engine	Random Forest	630,390	630,390	0.630	48.173
Engine	Logistic Regression	7	7	0.000007	0.0009
Engine	FTT	398,337	398,337	0.398	1.529
Engine	LSTM–CNN	628,225	628,225	0.628	2.402
Engine	CNN	116,737	116,737	0.117	0.455
Engine	BiLSTM	529,665	529,665	0.530	2.023
Engine	MLP	530,433	530,433	0.530	2.032
Engine	FusionXNet (16k_base)	16,013	15,837	0.016	0.074
Engine	Meta (CNN+BiLSTM+RF)	233,862	233,862	0.234	49.210
MetroPT3	XGBoost	39,876	39,876	0.040	1.549
MetroPT3	Random Forest	117,270	117,270	0.117	9.025
MetroPT3	Logistic Regression	16	16	0.000016	0.0009
MetroPT3	FTT	400,641	400,641	0.401	1.538
MetroPT3	LSTM–CNN	628,225	628,225	0.628	2.402
MetroPT3	CNN	116,737	116,737	0.117	0.455
MetroPT3	BiLSTM	529,665	529,665	0.530	2.023
MetroPT3	MLP	532,737	532,737	0.533	2.041
MetroPT3	FusionXNet (16k_base)	16,139	15,963	0.016	0.074
MetroPT3	Meta (CNN+BiLSTM+RF)	233,862	233,862	0.234	11.054

7.4 Inference-Latency Summary

Table 11 Inference-latency statistics for directly verifiable baseline and comparison models, including batch-throughput latency, single-sample latency, throughput in batches per second, and batch size.

Dataset	Model	ms/sample (thr)	ms/sample (single)	Batches/s	Batch Size
AI4I	LSTM–CNN	0.00945	0.61185	108391	1024
AI4I	BiLSTM	0.00910	0.36452	112518	1024
AI4I	XGBoost	0.01073	0.25630	91	1024
AI4I	Random Forest	0.01742	8.01185	56	1024
AI4I	Logistic Regression	0.00050	0.12221	1964	1024
Engine	XGBoost	0.00724	0.20528	135	1024
Engine	Random Forest	0.02526	7.40427	39	1024
Engine	Logistic Regression	0.00013	0.09489	7802	1024
Engine	FTT	0.00448	0.88510	233914	1024
Engine	LSTM–CNN	0.00849	0.57315	123506	1024
Engine	CNN	0.00346	0.64397	302833	1024
Engine	BiLSTM	0.00828	0.43749	126625	1024
Engine	MLP	0.00119	0.82520	878066	1024
MetroPT3	XGBoost	0.00643	0.22710	152	1024
MetroPT3	Random Forest	0.00486	7.05143	201	1024
MetroPT3	Logistic Regression	0.00008	0.09865	12302	1024
MetroPT3	FTT	0.00743	0.79126	134969	1024
MetroPT3	LSTM–CNN	0.01237	0.73984	81060	1024
MetroPT3	CNN	0.00203	0.54899	493248	1024
MetroPT3	BiLSTM	0.01179	0.37870	85016	1024
MetroPT3	MLP	0.00148	0.84604	679004	1024

7.5 Traditional Machine-Learning Baselines

This subsection reports extended in-domain results for the traditional machine-learning baselines used in this study. These results are provided for completeness and traceability and complement the compact comparisons retained in the main manuscript.

Table 12 summarises the in-domain performance of the traditional machine learning baselines, including XGBoost, Random Forest, and Logistic Regression, across the AI4I, Engine, and MetroPT3 datasets. Overall, the tree-based ensemble models (XGBoost and Random Forest) achieve competitive accuracy and AUROC values, particularly on the MetroPT3 dataset, where XGBoost attains near-perfect discrimination (AUROC ≈ 0.999). However, these models exhibit substantially higher latency and larger model sizes than lightweight linear models, and their F2-scores remain consistently lower than those reported for the deep learning and FusionXNet models. This supports the observation that while ML baselines can deliver strong classification margins in specific domains, they lack the temporal and representation-learning capabilities required for robust generalisation across diverse operating conditions and dataset distributions.

Table 12 Performance of traditional machine learning baselines across the AI4I, MetroPT3, and Engine datasets. Each model is trained and evaluated in-domain using fixed-threshold decision rules. Metrics include accuracy, AUROC, AUPRC, F2-score, and computational statistics (latency, parameter count, and model size).

Dataset	**Model**	**Acc.**	**AUROC**	**AUPRC**	**F2**	**Params**	**Size (MB)**	**ms (single)**	**Thr.**
AI4I	XGBoost	0.969	0.9769	0.7477	0.6929	39,514	1.536	0.2563	0.43
AI4I	Random Forest	0.953	0.9717	0.6180	0.6281	137,854	10.595	8.0119	0.43
AI4I	Logistic Reg.	0.874	0.9189	0.4600	0.4787	9	0.0009	0.1222	0.43
Engine	Logistic Reg.	0.662	0.6902	0.7830	0.7479	7	0.0009	0.0949	0.46
Engine	Random Forest	0.645	0.6765	0.7790	0.7138	630,390	48.173	7.4043	0.46
Engine	XGBoost	0.630	0.6617	0.7692	0.6951	90,002	3.174	0.2053	0.46
MetroPT3	XGBoost	0.9959	0.99995	0.9976	0.9590	39,876	1.549	0.2271	0.41
MetroPT3	Random Forest	0.9942	0.99992	0.9958	0.9440	117,270	9.025	7.0514	0.41
MetroPT3	Logistic Reg.	0.9776	0.9931	0.7473	0.8117	16	0.0009	0.0987	0.41

7.6 Traditional Deep-Learning Baselines

Table 13 presents the performance of deep learning baseline architectures across the three datasets. On the AI4I dataset, transformer-based FTT and hybrid LSTM-CNN models achieve the highest AUROC scores ($\approx$ 0.97–0.98), while CNN and BiLSTM provide competitive but slightly lower F2 scores. For the Engine dataset, all deep models exhibit reduced discriminative power due to the challenging class distribution and higher temporal variability, with F2-scores stabilising around 0.63–0.71. On the MetroPT3 dataset, all deep learning models achieve near-perfect AUROC values ($>$ 0.999), with MLP, CNN, and LSTM–CNN models exceeding 0.99 accuracy and maintaining high F2-scores. Despite these strong in-domain results, deeper architectures also incur higher parameter counts and greater inference latency than the CNN baseline and the final FusionXNet model, motivating the design of a lightweight yet high-performing fusion architecture.

7.7 Cross-domain transfer tables under fixed-threshold evaluation

This subsection provides the full source-wise cross-domain transfer results for the FusionXNet-Edge baseline and supporting CORAL variants under fixed-threshold evaluation. These tables complement the compact summaries reported in the main manuscript and are included for traceability, enabling detailed inspection of source–target asymmetry, ranking behaviour, and threshold-dependent performance.

Across all tables, cross-domain behaviour remains strongly source–target dependent and exhibits clear asymmetry. In several cases, AUROC and AUPRC remain non-trivial, while F2 collapses under the fixed threshold, reflecting an operating-point mismatch rather than a complete loss of ranking information. These detailed results support the threshold-sensitivity analysis presented in the main manuscript.

8 Computational Profile and Ablation Support

This section provides supplementary computational and sensitivity support for the proposed framework. It includes branch-wise complexity information together with compact transfer summaries that are not essential to the main narrative but help interpret efficiency and cross-domain behaviour.

8.1 Branch-wise Computational Profile

Latency is only one aspect of deployability; model footprint and computational complexity are also important. Because the convolutional blocks use 1×1 kernels and the recurrent paths are single-layer, the overall parameter count remains small and identical across datasets (see the shared-configuration tables in this supplement). The computational characteristics of FusionXNet-Edge and its ablated variants were measured by instantiating each network and saving it in the same environment. Table 17 reports the total parameters, resulting model size on disk, and analytical multiply–accumulate operations (MACs) for a single-sample input.

Table 13 Performance of deep learning baseline models across the AI4I, Engine, and MetroPT3 datasets. Models are trained and evaluated in-domain using fixed thresholds. Metrics include accuracy, AUROC, AUPRC, F2-score, parameter count, model size, and GPU inference latency (ms/sample).

Dataset	Model	Acc.	AUROC	AUPRC	F2	Params	Size (MB)	ms/sample
AI4I	FTT	0.9545	0.9772	0.7121	0.6968	398,849	1.531	0.8440
AI4I	LSTM–CNN	0.9370	0.9679	0.6550	0.6502	628,225	2.402	0.6119
AI4I	MLP	0.9310	0.9693	0.6938	0.6250	530,945	2.034	0.7734
AI4I	CNN	0.9090	0.9727	0.6865	0.6152	116,737	0.455	0.6439
AI4I	BiLSTM	0.9075	0.9582	0.6461	0.5589	529,665	2.023	0.3645
Engine	LSTM–CNN	0.6622	0.6983	0.7973	0.7136	628,225	2.402	0.5731
Engine	CNN	0.6412	0.6889	0.7906	0.6670	116,737	0.455	0.6439
Engine	BiLSTM	0.6479	0.6946	0.7941	0.6592	529,665	2.023	0.4375
Engine	FTT	0.6313	0.6874	0.7902	0.6348	398,337	1.529	0.8851
Engine	MLP	0.6008	0.6950	0.7946	0.5396	530,433	2.032	0.8252
MetroPT3	MLP	0.9956	0.99988	0.99495	0.9541	532,737	2.041	0.8460
MetroPT3	CNN	0.9951	0.99990	0.99587	0.9509	116,737	0.455	0.5490
MetroPT3	LSTM–CNN	0.9940	0.99987	0.99651	0.9424	628,225	2.402	0.7398
MetroPT3	FTT	0.9939	0.99992	0.99614	0.9412	400,641	1.538	0.7913
MetroPT3	BiLSTM	0.9935	0.99991	0.99606	0.9369	529,665	2.023	0.3787

Table 14 Cross-domain performance of baseline and CORAL-variant models when trained on the Engine dataset and evaluated on unseen domains (AI4I and MetroPT3). All results use fixed-threshold evaluation.

Source	Target	Model Variant	AUROC	AUPRC	F2	Accuracy	AUPRC Lift
Engine	AI4I	Baseline	0.724	0.101	0.237	0.685	2.88
Engine	AI4I	CORAL-Diag (all14)	0.702	0.084	0.210	0.505	2.39
Engine	AI4I	CORAL-Diag (base)	0.716	0.094	0.229	0.559	2.70
Engine	AI4I	CORAL-Shrink (0.2)	0.811	0.151	0.280	0.629	4.33
Engine	MetroPT3	Baseline	0.929	0.125	0.284	0.751	6.33
Engine	MetroPT3	CORAL-Diag (all14)	0.655	0.027	0.113	0.231	1.39
Engine	MetroPT3	CORAL-Diag (base)	0.929	0.125	0.146	0.425	6.35
Engine	MetroPT3	CORAL-Shrink (0.2)	0.929	0.125	0.168	0.513	6.35

Table 15 Cross-domain performance of baseline and CORAL-variant models when trained on the MetroPT3 dataset and evaluated on unseen domains (AI4I and Engine). Results are reported under fixed-threshold conditions.

Source	Target	Model Variant	AUROC	AUPRC	F2	Accuracy	AUPRC Lift
MetroPT3	AI4I	Baseline	0.644	0.063	0.000	0.965	1.80
MetroPT3	AI4I	CORAL-Diag (all14)	0.589	0.130	0.115	0.960	3.71
MetroPT3	AI4I	CORAL-Diag (base)	0.570	0.091	0.154	0.946	2.60
MetroPT3	AI4I	CORAL-Shrink (0.2)	0.563	0.118	0.154	0.945	3.37
MetroPT3	Engine	Baseline	0.659	0.763	0.000	0.369	1.21
MetroPT3	Engine	CORAL-Diag (all14)	0.524	0.661	0.892	0.632	1.05
MetroPT3	Engine	CORAL-Diag (base)	0.604	0.694	0.000	0.369	1.10
MetroPT3	Engine	CORAL-Shrink (0.2)	0.577	0.672	0.000	0.369	1.07

Table 16 Cross-domain performance of baseline and CORAL-variant models when trained on the AI4I dataset and evaluated on unseen domains (Engine and MetroPT3). Results are reported under fixed-threshold conditions.

Source	Target	Model Variant	AUROC	AUPRC	F2	Accuracy	AUPRC Lift
AI4I	Engine	Baseline	0.552	0.671	0.284	0.463	1.06
AI4I	Engine	CORAL-Diag (all14)	0.514	0.659	0.114	0.408	1.05
AI4I	Engine	CORAL-Diag (base)	0.564	0.673	0.198	0.433	1.07
AI4I	Engine	CORAL-Shrink (0.2)	0.584	0.686	0.204	0.440	1.09
AI4I	MetroPT3	Baseline	0.926	0.124	0.222	0.656	6.27
AI4I	MetroPT3	CORAL-Diag (all14)	0.940	0.633	0.827	0.985	32.06
AI4I	MetroPT3	CORAL-Diag (base)	0.926	0.124	0.252	0.708	6.27
AI4I	MetroPT3	CORAL-Shrink (0.2)	0.926	0.124	0.274	0.739	6.28

Table 17 Measured computational profile on the MetroPT3 dataset (single-sample inference).

Model	Params	File size (MB)	MACs (M)
FusionXNet-Edge	16 061	0.149	0.0152
CNN-only	3417	0.053	0.0029
BiLSTM-only	10 049	0.081	0.0095
GRU-only	5105	0.054	0.0046
MLP-only	4465	0.050	0.0041
CNN(+phys)	3741	0.065	0.0033
BiLSTM(+phys)	10 373	0.093	0.0098
GRU(+phys)	5429	0.062	0.0049
MLP(+phys)	4789	0.057	0.0044

Each model was instantiated, saved as a `.h5` file, and evaluated in the same environment. MACs are analytically derived per single-step forward pass (1×14 input).

Table 17 complements the latency statistics reported in the main paper by providing a joint view of parameter count, model size, and MACs. Together, these quantities help explain where speedups originate and which branches dominate computational load. The results show that FusionXNet-Edge remains lightweight, with an on-disk size of approximately 0.15 MB and a single-sample complexity of 0.0152 M MACs, while still retaining a balanced fusion of CNN, BiLSTM, GRU, and MLP components. Adding the physics-guided slice introduces only a small overhead, supporting the model's suitability for resource-constrained deployment.

8.2 Best-variant cross-domain transfer summary

Table 18 Best-variant cross-domain transfer summary. For each ordered source–target pair, the strongest fixed-threshold result among baseline and CORAL variants is reported.

Source	Target	Best variant	AUROC	AUPRC	F2
AI4I	Engine	CORAL-Shrink (0.2)	0.584	0.686	0.284
AI4I	MetroPT3	CORAL-Diag (all14)	0.940	0.633	0.827
Engine	AI4I	CORAL-Shrink (0.2)	0.811	0.151	0.280
Engine	MetroPT3	Baseline / CORAL-base	0.929	0.125	0.284
MetroPT3	AI4I	CORAL-Diag (base)	0.570	0.091	0.154
MetroPT3	Engine	CORAL-Diag (all14)	0.524	0.661	0.892

Table 18 summarises the strongest fixed-threshold transfer result for each ordered source–target pair. The results confirm that cross-domain performance is highly asymmetric and depends strongly on both the source domain and the alignment strategy. Consistent with the threshold-sensitivity analysis in the main paper, these best-case results should not be interpreted as evidence of uniformly robust transfer, since several source–target pairs remain weak even after alignment and alternative operating-point choices.

8.3 Auxiliary physics-guided feature-branch ablation protocol

Algorithm 1 Physics-branch ablation protocol (leakage-controlled)

Require: Frozen splits $\{(X_{\text{train}}^{\text{smote}}, y_{\text{train}}), (X_{\text{val}}, y_{\text{val}}), (X_{\text{test}}, y_{\text{test}})\}_d$
Require: Fixed 14D model input schema and physics-guided feature slice $[5:8)$
Require: Fixed dataset thresholds $\tau_d \in \{0.43, 0.46, 0.41\}$ for AI4I, Engine, and MetroPT3
for each dataset $d \in \{\text{AI4I}, \text{MetroPT3}, \text{Engine}\}$ **do**
  Load frozen train, validation, and test splits for d
  for each variant $v \in \{\text{phys\_enabled}, \text{phys\_zeroed}, \text{base\_variant}\}$ **do**
    if $v = \text{phys\_enabled}$ **then**
      Use the full FusionXNet-Edge variant with the auxiliary physics-guided feature branch enabled
    else if $v = \text{phys\_zeroed}$ **then**
      Use the same architecture, but set the physics-guided feature slice $x[:, 5:8] \leftarrow 0$
    else
      Use the lighter FusionXNet base variant without the auxiliary physics-guided feature branch
    end if
    Train with the same frozen protocol: same splits, optimiser, batch size, epochs, and early stopping on validation PR-AUC
    Evaluate on the test split using the fixed threshold τ_d
    Record AUROC, AUPRC, F2, and Accuracy
  end for
end for
return dataset-wise comparison across ablation variants

The ablation is designed to isolate the contribution of the auxiliary branch linked to physics-guided engineered features while holding identical preprocessing, model capacity, and training conditions constant. The *phys_zeroed* variant preserves the full architecture while zeroing the physics-guided feature slice, whereas the *base_variant* removes the auxiliary physics-guided feature branch entirely, allowing separation of feature-level and architectural effects. No retraining, recalibration, or preprocessing differences are introduced across variants.

9 Connection to the Main Manuscript

This Supplementary Material extends the main manuscript by providing the broader evidence base behind the compact results reported in the main text and Appendix A. The main manuscript focuses on the core framework, primary in-domain and cross-domain findings, and deployment-oriented conclusions. Appendix A provides compact

benchmark and diagnostic evidence required for interpretation, including complexity, latency, and ROC/precision–recall diagnostics.

The supplementary file retains the extended material that would otherwise overload the main narrative, including raw class distributions, leakage-controlled split-policy plots, individual physics-guided feature diagnostics, full baseline comparisons, threshold-sensitivity results, ablation details, source-wise transfer evidence, and Code Ocean reproducibility notes. In particular, the Code Ocean capsule referenced in the main manuscript is expanded in Section 10, which explains the difference between lightweight capsule validation and full dataset-dependent experimental reruns.

10 Code Availability, Reproducibility, and Artefact Packaging

This section expands the Code availability statement in the main manuscript by describing the contents and intended use of the FusionXNet-Edge reproducibility package. The Code Ocean capsule contains the cleaned source code, validation scripts, archival notebooks with outputs removed, saved lightweight model artefacts, dataset-reference files, reference result tables, and documentation required to reproduce the package structure. The capsule is intended to provide transparent access to the implementation and validation workflow while avoiding redistribution of third-party raw datasets. The Code Ocean capsule is therefore the public source-code record associated with this study.

The default Code Ocean execution is a lightweight validation run Sharma et al. (2026). It verifies the package structure, imports, metadata, dataset-reference files, model artefacts, reference outputs, and reproducibility manifests. By default, it does not load the raw datasets, retrain models, or regenerate the complete scientific results. Full experimental reruns require users to obtain the public datasets from their original repositories and place them according to the dataset-path instructions provided in the capsule.

Every full experiment run exports the artefacts needed to trace results and deployment behaviour: the fitted scaler, the exact $(1, 14)$ feature order and physics-guided feature-slice indices, dataset-specific label mappings, trained weights, selected operating thresholds, software versions, hardware details, and run-level manifests. Loading code refuses to proceed if incoming features do not match the recorded order or count, preventing silent schema drift between training and deployment.

This separation between source-code validation and full dataset-dependent reruns is deliberate. The Code Ocean capsule supports reproducibility of the implementation and execution structure, while the raw datasets remain governed by their original repositories, licences, and citation requirements. The main manuscript reports scientific results from the completed experimental workflow, and the capsule provides the code and instructions needed to reproduce or audit that workflow.

References

Ahmed, I., Jeon, G., Piccialli, F.: From artificial intelligence to explainable artificial intelligence in industry 4.0: A survey on what, how, and where. IEEE Transactions on Industrial Informatics **18**, 5031–5042 (2022) https://doi.org/10.1109/TII.2022.3146552

Bag, S., Gupta, S., Kumar, S.: Industry 4.0 adoption and 10r advance manufacturing capabilities for sustainable development. International journal of production economics **231**, 107844 (2021) https://doi.org/10.1016/j.ijpe.2020.107844

Diez-Olivan, A., Ser, J.D., Galar, D., Sierra, B.: Data fusion and machine learning for industrial prognosis: Trends and perspectives towards industry 4.0. Information Fusion **50**, 92–111 (2019) https://doi.org/10.1016/j.inffus.2018.10.005

Jamwal, A., Agrawal, R., Sharma, M.: Deep learning for manufacturing sustainability: Models, applications in industry 4.0 and implications. International Journal of Information Management Data Insights **2**(2), 100107 (2022) https://doi.org/10.1016/j.jjimei.2022.100107

Kumar, S., Gopi, T., Harikeerthana, N., Gupta, M.K., Gaur, V., Krolczyk, G.M., Wu, C.: Machine learning techniques in additive manufacturing: a state of the art review on design, processes and production control. Journal of Intelligent Manufacturing **34**, 21–55 (2023) https://doi.org/10.1007/s10845-022-02029-5

Kotsiopoulos, T., Sarigiannidis, P., Ioannidis, D., Tzovaras, D.: Machine learning and deep learning in smart manufacturing: The smart grid paradigm. Computer Science Review **40**, 100341 (2021) https://doi.org/10.1016/j.cosrev.2020.100341

Leng, J., Sha, W., Wang, B., Zheng, P., Zhuang, C., Liu, Q., Wuest, T., Mourtzis, D., Wang, L.: Industry 5.0: Prospect and retrospect. Journal of Manufacturing Systems **65**, 279–295 (2022) https://doi.org/10.1016/j.jmsy.2022.09.017

Priore, P., Fuente, D.D.L., Gomez, A., Puente, J.: A review of machine learning in dynamic scheduling of flexible manufacturing systems. Artificial Intelligence for Engineering Design, Analysis and Manufacturing **15**, 251–263 (2001) https://doi.org/10.1017/S0890060401153059

Parvanda, R., Kala, P.: Trends, opportunities, and challenges in the integration of the additive manufacturing with industry 4.0. Progress in Additive Manufacturing **8**, 587–614 (2023) https://doi.org/10.1007/s40964-022-00351-1

Park, J.-K., Kwon, B.-K., Park, J.-H., Kang, D.-J.: Machine learning-based imaging system for surface defect inspection. International Journal of Precision Engineering and Manufacturing-Green Technology **3**(3), 303–310 (2016) https://doi.org/10.1007/s40684-016-0039-x

Psarommatis, F., Sousa, J., Mendonça, J.P., Kiritsis, D.: Zero-defect manufacturing the approach for higher manufacturing sustainability in the era of industry 4.0: a position paper. International Journal of Production Research **60**, 73–91 (2022) https://doi.org/10.1080/00207543.2021.1987551

Radziwon, A., Bilberg, A., Bogers, M., Madsen, E.S.: The smart factory: exploring adaptive and flexible manufacturing solutions. Procedia engineering **69**, 1184–1190 (2014) https://doi.org/10.1016/j.proeng.2014.03.108

Ruiz-Sarmiento, J.-R., Monroy, J., Moreno, F.-A., Galindo, C., Bonelo, J.-M., Gonzalez-Jimenez, J.: A predictive model for the maintenance of industrial machinery in the context of industry 4.0. Engineering Applications of Artificial Intelligence **87**, 103289 (2020) https://doi.org/10.1016/j.engappai.2019.103289

Ren, S., Zhang, Y., Liu, Y., Sakao, T., Huisingh, D., Almeida, C.M.: A comprehensive review of big data analytics throughout product lifecycle to support sustainable smart manufacturing: A framework, challenges and future research directions. Journal of cleaner production **210**, 1343–1365 (2019) https://doi.org/10.1016/j.jclepro.2018.11.025

Scime, L., Beuth, J.: Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. Additive Manufacturing **19**, 114–126 (2018) https://doi.org/10.1016/j.addma.2017.11.009

Sony, M., Naik, S.: Critical factors for the successful implementation of industry 4.0: a review and future research direction. Production Planning & Control **31**, 799–815 (2020) https://doi.org/10.1080/09537287.2019.1691278

Sharma, A., Sim, K.Y., Chandrasekaran, S.: FusionXNet-Edge: A Physics-Guided Lightweight Framework for Predictive Maintenance Under Cross-Domain Deployment Shift. https://www.codeocean.com/ (2026). https://doi.org/10.24433/CO.3751906.v1

Tercan, H., Meisen, T.: Machine learning and deep learning based predictive quality in manufacturing: a systematic review. Journal of Intelligent Manufacturing **33**, 1879–1905 (2022) https://doi.org/10.1007/s10845-022-01963-8

Tao, F., Qi, Q., Liu, A., Kusiak, A.: Data-driven smart manufacturing. Journal of Manufacturing Systems **48**, 157–169 (2018) https://doi.org/10.1016/j.jmsy.2018.01.006

Tu, P.Y., Yam, R., Tse, P., Sun, A.: An integrated maintenance management system for an advanced manufacturing company. The International Journal of Advanced Manufacturing Technology **17**(9), 692–703 (2001) https://doi.org/10.1007/s001700170135

Wu, Y., Sicard, B., Gadsden, S.A.: A review of physics-informed machine learning

methods with applications to condition monitoring and anomaly detection. arXiv preprint arXiv:2401.11860 (2024) https://doi.org/10.48550/arXiv.2401.11860

Yang, Z., Wang, Q., Jia, M.: Integrating industry 4.0 and the internet of things (iot) for eco-friendly manufacturing. The International Journal of Advanced Manufacturing Technology (2023) https://doi.org/10.1007/s00170-023-12331-y

Zhong, R.Y., Xu, X., Klotz, E., Newman, S.T.: Intelligent manufacturing in the context of industry 4.0: a review. Engineering **3**(5), 616–630 (2017) https://doi.org/10.1016/J.ENG.2017.05.015