

Research Article

An Uncertainty-Calibrated Ensemble Framework for Fault Diagnosis in Industrial Hydraulic Power Units: Rethinking Stacked Neural Architectures

Emmanuel Unekuojo Enejor^{1,4,*} , **Nelson Oluwole Ibigbami⁵**,
Ismail Olusegun Lawal^{1,4} , **Temidayo Adeoti²**, **Ademola Adesoye²**,
Samson Oluwasina Ajibesin², **Adekunle Francis Adeniyi³**,
Kolawole Bamidele Jayeola⁵, **Oludare Michael Shotade⁵**

¹Electrical and Electronic, University of Ibadan, Oyo, Nigeria

²Electrical and Electronic, Federal University of Technology, Ondo, Nigeria

³Control and Instrumentation, University of Derby, Derby, England

⁴Electronic, Instrumentation and Simulation, AAEL/NASRDA, Ondo, Nigeria

⁵Electronic, Instrumentation and Simulation, BAT-CSTP/NASRDA, Lagos, Nigeria

Abstract

Stacked artificial neural network architectures have been proposed for industrial condition monitoring on the premise that adding sequential learning stages improves predictive accuracy, yet this premise is rarely tested against simpler alternatives under leakage-free, cross-validated evaluation. This study conducts a rigorous, block-aware empirical investigation of predictive fault diagnosis on the benchmark UCI Condition Monitoring of Hydraulic Systems dataset (2,205 operating cycles, 17 multi-rate sensors, four superimposed component-degradation labels). We first demonstrate that naive random cross-validation on this dataset inflates reported accuracy by up to 26 percentage points relative to a block-aware (leakage-free) protocol, because the dataset is organised as long contiguous condition blocks. Under the corrected protocol, we reproduce and stress-test the stacking mechanism used by a representative elaborative stepwise stacked neural network (ESSANN) architecture from the recent literature, implementing it faithfully as a deep-ensemble multilayer perceptron (MLP) with feature-concatenation stacking and a boosting-style hard-example variant, and evaluate all architectures under 4-fold block-aware cross-validation with paired per-fold robustness checks. Both neural stacking variants degrade accuracy relative to a single-stage MLP ensemble consistently across every fold on both primary targets (valve condition, accumulator pressure), revealing that naive depth is not a principled correction mechanism and is vulnerable to training-data starvation. In contrast, a two-stage Random Forest architecture and Gradient Boosting, which stage learners on the complete training set every round, improve on the single-stage baseline consistently across all folds on valve condition (Gradient Boosting: $98.1\% \pm 1.1\%$, +11.3 points over Random Forest in every fold), but the same improvement is not sign-consistent across folds for accumulator pressure, where no tested architecture reliably outperforms a plain Random Forest baseline once fold-to-fold variance is accounted for. We further show that predictive entropy from a deep ensemble is a well-calibrated, monotonic indicator of classification risk (100% accuracy in the lowest-entropy quintile versus 41-63% in the highest), but that thresholds calibrated on training-set entropy are unusable in deployment, over-flagging 94-98% of unseen samples as "uncertain"; calibrating instead on a held-out validation block reduces this to a deployable 44-60%. We conclude with a practical, evidence-graded architecture recommendation and a risk-stratified decision framework

*Correspondence: Emmanuel Unekuojo Enejor (emmymalism@gmail.com)

Received: 29 August 2026; **Accepted:** 10 September 2026; **Published:** 20 October 2026



Copyright: © The Author(s), 2026. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

for confidence-gated industrial control actions.

Keywords

Predictive Maintenance, Hydraulic Power Units, Condition Monitoring, Uncertainty Quantification, Ensemble Learning, Stacked Neural Networks, Industrial Internet of Things, Explainable AI

1. Introduction

Industrial automation increasingly relies on data-driven condition monitoring to detect component degradation before failure, reducing unplanned downtime and maintenance costs [1-4]. Hydraulic power units (HPU) are a common subsystem in elevators, presses, and material-handling equipment [5]. Their failure modes are cooler fouling, valve wear, internal pump leakage, and accumulator pressure losses. are individually well characterized but frequently co-occur, complicating single-fault diagnostic approaches [6, 7].

A recent stream of work has proposed elaborate stepwise stacked artificial neural network (ESSANN) architectures for industrial automation, in which multiple multilayer perceptron (MLP) stages are stacked so that the output of one stage is propagated as input to the next, with the implicit assumption that additional stages progressively correct the errors of earlier ones [8]. This assumption that architectural depth alone improves predictive performance is intuitive but, to our knowledge, has not been directly tested against (a) leakage-free evaluation protocols appropriate for block-structured industrial time-series data, or (b) simpler ensemble alternatives evaluated under identical conditions. Reported gains for such architectures are typically presented as percentage comparisons against conceptual or non-retrained baselines (e.g., generic blockchain or immune-system-inspired industrial control frameworks), rather than against standard machine learning baselines trained and evaluated on the same data with the same validation protocol.

This paper addresses that gap directly. Using the publicly available UCI Condition Monitoring of Hydraulic Systems dataset [9, 10], *the contribution of this work is as follows:*

A naive random train/test split on the UCI Condition Monitoring of Hydraulic Systems dataset is shown to inflate reported accuracy relative to a block-aware (group-wise) split that keeps each contiguous condition block wholly within one partition. The inflation reaches 26.8 percentage points for individual target/model combinations, because the dataset is organised as long contiguous condition blocks; the same risk is argued to apply to block-structured industrial process data more generally.

The stacking mechanism at the centre of ESSANN-style architectures is reproduced faithfully as a deep-ensemble MLP with feature-concatenation stacking, together with a boosting-style variant that specializes later stages on harder examples. Under leakage-free evaluation, both variants underperform a

single-stage MLP ensemble on the two targets with genuine nonlinear structure — evidence that added depth alone is not a reliable correction mechanism and that later stages are prone to training-data starvation.

Under 4-fold block-aware cross-validation with paired per-fold checks, the same staged-correction idea implemented with tree learners trained on the complete set at every stage (two-stage Random Forest and Gradient Boosting) improves on a single-stage baseline for valve condition in every fold, but not consistently for accumulator pressure, where no tested architecture reliably beats a plain Random Forest. This locates the neural variants' failure in the choice of base learner and training protocol rather than in the staging concept, and shows that a per-target, repeated-CV validation is necessary before any architectural advantage is claimed.

Ensemble predictive entropy is shown to rank-order classification risk well (near-perfect accuracy in the lowest-entropy quintile, 41-63% in the highest), but an escalation threshold taken from training-set entropy over-flags 94-98% of unseen samples and is unusable in deployment; recalibrating the threshold on a held-out validation block reduces flagging to a deployable 44-60%. A confidence-gating procedure decoupled from the point-prediction model follows from this result.

2. Related Works

2.1. Data-Driven Condition Monitoring of Hydraulic Systems

The hydraulic test-rig dataset used in this study was introduced by Helwig et al. [10] and has since been used extensively for fault classification benchmarking. Prior work has applied gradient-boosted trees [6], convolutional neural networks on raw sensor streams [7], and hybrid signal-processing/statistical-learning pipelines combining empirical mode decomposition, PCA, and least-squares support vector machines [5] to this dataset, generally reporting high (>95%) classification accuracy. However, the evaluation protocols in this literature are not always explicit about whether train/test partitioning respects the block-structured, cyclically repeating experimental design of the data; our results in Section 4.1 suggest this is a material methodological concern for this dataset and, plausibly, for other block-structured industrial datasets

more broadly.

2.2. Stacked and Elaborative Neural Architectures in Industrial Automation

Stacked generalisation, combining the outputs of several base learners through a meta-learner, is a well-established ensemble technique [11]. Its application to industrial automation via architectures that stack multiple MLP stages, propagating each stage's output as an input to the next, has been proposed under names including elaborative stepwise stacked artificial neural networks (ESSANN) [8]. This method reports favourable comparisons against conceptual baselines such as unified artificial immune systems and blockchain-based cyber-threat mitigation frameworks evaluated on different metrics (delay, bandwidth, packet loss) rather than head-to-head predictive accuracy on the same task and data. Our study complements this literature by isolating the stacking mechanism itself and testing it, on real sensor data, against machine-learning baselines trained under identical conditions.

2.3. Uncertainty Quantification for Industrial Decision Support

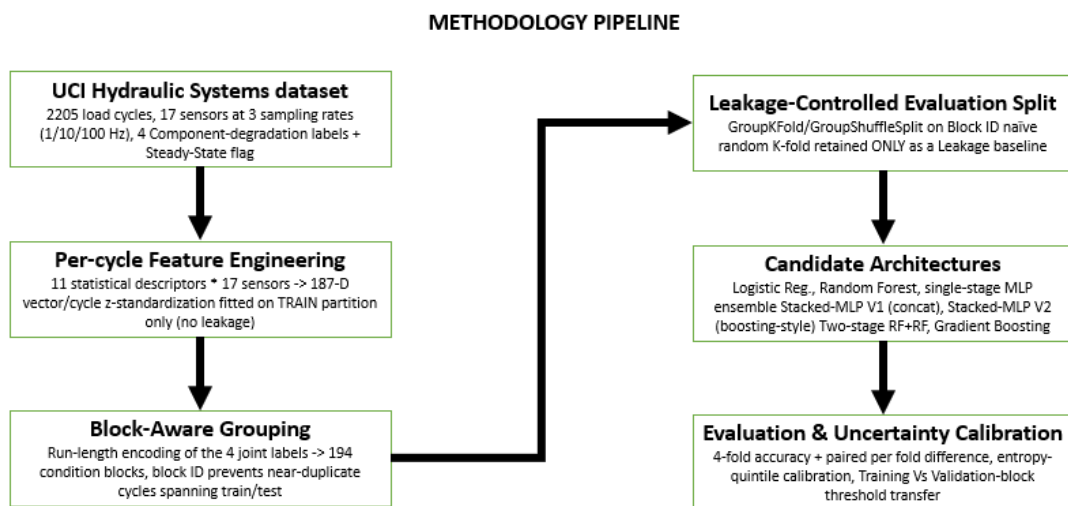
Deep ensembles [12] and related techniques like Monte Carlo dropout [13] provide practical epistemic uncertainty estimates without requiring a fully Bayesian treatment of network weights. In industrial and safety-critical settings, well-calibrated uncertainty is valuable not because it directly improves point-prediction accuracy, but because it enables risk-stratified decision-making like routing low-confidence cases

to human review or secondary instrumentation rather than acting on an unreliable automated prediction [14]. This distinction, between uncertainty as an accuracy-improving mechanism and uncertainty as a deployment-risk signal, is central to the reframing of results we present in Section 5.

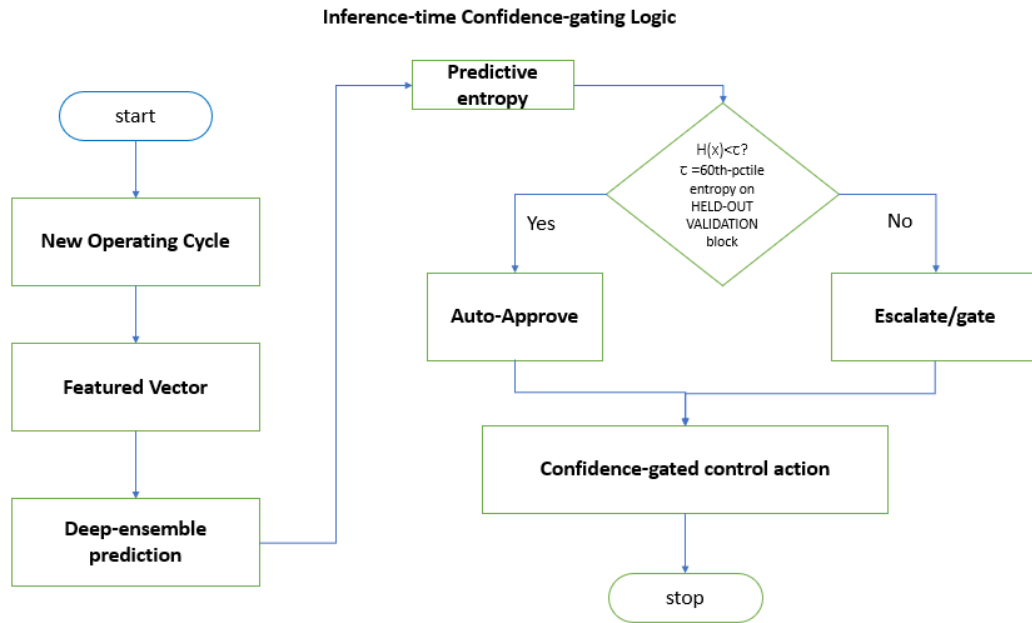
3. Material and Method

The proposed methodology comprises a leakage-controlled pipeline for fault diagnosis in industrial hydraulic power units [15, 16]. The UCI Hydraulic Systems dataset, containing 2,205 load cycles from 17 sensors sampled at three rates, is first transformed through per-cycle feature engineering, producing 187 standardised descriptors from sensor statistics. Block-aware grouping subsequently encodes the four joint labels into condition blocks while preventing duplicate cycles from spanning training and testing partitions. A GroupK-Fold/GroupShuffleSplit strategy with block-level isolation is then applied to obtain a leakage-controlled evaluation set. Candidate architectures, including logistic regression, random forest, single-stage multilayer perceptron, and alternative stacked ensembles, are evaluated using four-fold accuracy, paired performance differences, entropy-quantile calibration, and training-validation threshold transfer.

The resulting framework incorporates inference-time confidence gating based on predictive entropy. Predictions below the calibrated entropy threshold are automatically approved, whereas uncertain predictions are escalated for human review, secondary sensing, or conservative action. This decouples decision routing from the post-prediction model and provides a risk-aware mechanism for deploying automated fault diagnosis under uncertainty.



(a) Proposed methodology block diagram



(b) Proposed confidence-gating flowchart

Figure 1. Proposed Methodology and Confidence-Gating Framework.

3.1. Dataset

We use the Condition Monitoring of Hydraulic Systems dataset (UCI Machine Learning Repository, DOI 10.24432/C5CW21) [9]. The data were collected from a hydraulic test rig comprising a primary working circuit and a secondary cooling-filtration circuit connected via an oil tank. The system executed 2,205 repeated load cycles of 60 s duration, during which the condition of four components (i.e.

cooler, valve, internal pump, and hydraulic accumulator) was quantitatively and independently varied across multiple severity grades. Seventeen sensors were recorded per cycle at three sampling rates (Table 1), yielding raw feature dimensionalities of 6,000 samples/cycle for the six pressure sensors and motor power, 600 for the two flow sensors, and 60 for the four temperature sensors, one vibration sensor, and three virtual efficiency channels.

Table 1. Sensor inventory and sampling characteristics.

Sensor code (s)	Physical quantity	Unit	Sampling rate	Samples/cycle
PS1-PS6	Pressure	bar	100 Hz	6000
EPS1	Motor power	W	100 Hz	6000
FS1-FS2	Volume flow	L/min	10 Hz	600
TS1-TS4	Temperature	°C	1 Hz	60
VS1	Vibration	mm/s	1 Hz	60
CE	Cooling efficiency (virtual)	%	1 Hz	60
CP	Cooling power (virtual)	kW	1 Hz	60
SE	System efficiency factor	%	1 Hz	60

Four target labels are annotated per cycle: cooler condition (3%, 20%, 100% efficiency), valve condition (73%, 80%,

90%, 100% of optimal switching behavior), internal pump leakage (none, weak, severe), and accumulator pressure (90,

100, 115, 130 bar). A fifth binary flag indicates whether steady-state conditions were reached. Class distributions (Table 2) show that cooler and pump-leakage labels are close to

balanced or moderately skewed, while valve condition is dominated by the "optimal" class (51% of cycles).

Table 2. Target label distributions ($n = 2,205$ cycles).

Target	Classes	Count per class
Cooler condition (%)	3 / 20 / 100	732 / 732 / 741
Valve condition (%)	73 / 80 / 90 / 100	360 / 360 / 360 / 1125
Pump leakage	none / weak/severe	1221 / 492 / 492
Accumulator (bar)	90 / 100 / 115 / 130	808 / 399 / 399 / 599

The dataset is organised as long contiguous blocks of constant condition rather than randomised trials. Across all 2,205 cycles, we identified only 194 contiguous condition blocks (mean block length approximately 11.4 cycles, with several blocks exceeding 700 consecutive cycles for the cooler-condition label). This block structure has immediate implications for correct assessment, as described in Section 3.3.

3.2. Feature Engineering

Rather than operating on raw multi-rate sensor streams directly, we compute eleven per-cycle statistical descriptors for each of the seventeen sensors: mean, standard deviation, minimum, maximum, median, range, skewness, kurtosis, mean of the first 10% of samples, mean of the last 10% of samples, and the difference between the two (an intra-cycle trend indicator). This produces a 187-dimensional feature vector per cycle. For six cycles in which the PS4 pressure sensor recorded a constant value of zero throughout the 60 s window, skewness and kurtosis are undefined (0/0); these values are set to zero, reflecting the absence of within-cycle variation rather than missing data. All features are standardized (zero mean, unit variance) using statistics computed on the training partition only, in every experiment reported below.

3.3. Evaluation Protocol

The dataset is organised in contiguous condition blocks (Section 3.1), thus a randomly shuffled k-fold split places cycles from the same block in both the training and test partitions. Adjacent cycles within a block are physically near-duplicates in operating condition, so this constitutes a form of temporal/condition leakage that inflates apparent generalization performance. We therefore report results under two protocols for comparison: (i) a naive random k-fold split ($k = 4$, shuffled), representative of protocols not explicitly designed for this block structure; and (ii) a block-aware (group) split, in which entire contiguous condition blocks are assigned wholly

to either the training or test partition, implemented via scikit-learn's GroupKFold and GroupShuffleSplit using a block identifier derived from run-length encoding of the four target labels jointly [17]. All architecture-comparison and calibration experiments (Sections 4.2-4.4) use the block-aware protocol exclusively.

3.4. Proposed Ensemble Architectures

We compare six model classes, all trained on the standardized 187-dimensional feature vector:

Logistic Regression, as a linear reference model.

Random Forest (RF), 150-300 trees, as a strong non-neural nonlinear baseline [18].

Single-stage deep-ensemble MLP: five independently bootstrap-sampled and randomly initialized MLPs (two hidden layers, 64 and 32 units), whose class-probability outputs are averaged; predictive entropy of the averaged distribution serves as the uncertainty estimate.

Stacked MLP, feature-concatenation variant (V1): a faithful reproduction of the ESSANN stacking mechanism. Each subsequent ensemble stage is trained on the full training set augmented with the class-probability outputs of the previous stage; at inference, a sample is finalized at the first stage whose predictive entropy falls below a percentile-based threshold, or at the final stage otherwise.

Stacked MLP, boosting-style variant (V2): as V1, but each subsequent stage is trained only on the subset of training examples flagged as uncertain by the previous stage, in the manner of staged boosting.

Two-stage Random Forest (RF+RF): as V1's architecture, but with Random Forest as the base learner at both stages and the second stage always trained on the complete training set.

Gradient Boosting (GB): 200 estimators, max depth 3, learning rate 0.05, as a properly staged reference in which every boosting round is fit on the full training set [19-21].

The primary architecture-comparison results (Section 4.2,

Table 4) use 4-fold block-aware cross-validation (GroupK-Fold on the block identifier defined in Section 3.3) with paired per-fold robustness reporting; for tractability across six architectures and four folds, ensemble size is set to three members per stage with 400 MLP training iterations, applied uniformly across all ensemble-based architectures. An initial single-split screening comparison (75/25 block-aware partition, five ensemble members per stage, 600 MLP training iterations) is used only for the calibration analyses in Sections 4.3-4.4, together with an additional held-out validation partition, as described in Section 3.5.

3.5. Uncertainty Quantification and Calibration

For deep-ensemble models, predictive uncertainty for an input x is the Shannon entropy of the ensemble-averaged class-probability vector. $H(x) = -\sum_i p_i(x) \log p_i(x)$. For Random Forest, the analogous quantity is computed from the forest's internal tree-averaged class-probability output. We evaluate calibration in two ways: (i) binning test-set predictions into entropy quintiles and reporting empirical accuracy per bin, to assess whether entropy rank-orders correctness; and (ii) comparing an escalation threshold (the 60th percentile of entropy) computed on training-set predictions against the same percentile computed on a held-out validation block (20%

of the training partition, disjoint from both training and test blocks), to assess whether a threshold calibrated on already-seen data transfers to unseen operating blocks [4].

4. Results

4.1. Evaluation Leakage Under Random Splitting

Table 3 compares 4-fold cross-validated accuracy under random versus block-aware splitting for Logistic Regression, Random Forest, and the single-stage MLP ensemble, across all four fault targets, using the full 187-feature representation.

The gap is largest for the valve and accumulator targets (up to 26.8 percentage points for the MLP ensemble on valve condition) and negligible for the near-saturated pump-leakage target. Cooler and pump-leakage classification are effectively solved by all three model classes regardless of split protocol and are therefore treated as sanity-check targets rather than primary evaluation targets in the remainder of this study; valve condition and accumulator pressure, which combine genuine nonlinear structure with meaningful leakage sensitivity, are used as the primary targets for architecture comparison.

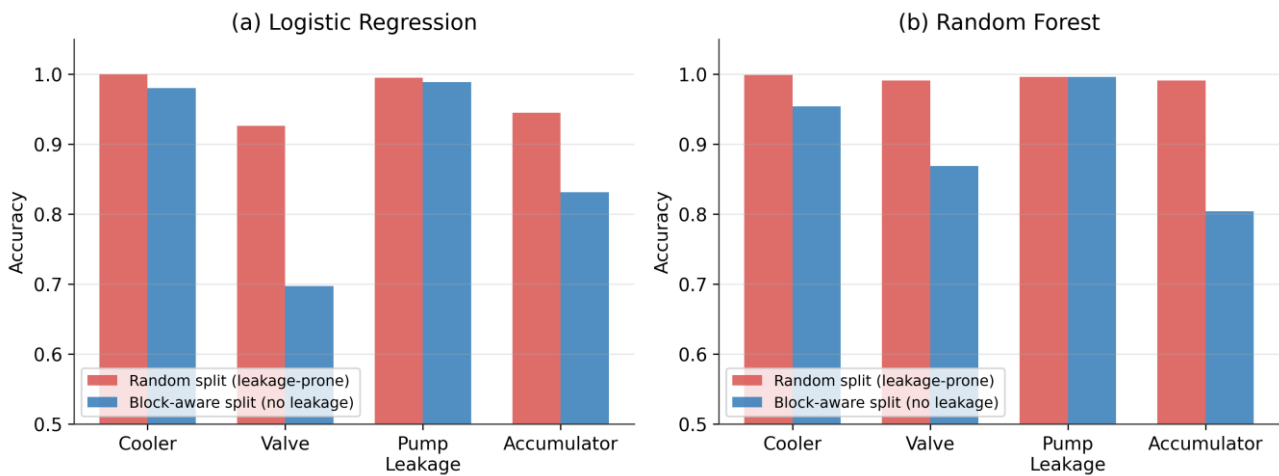


Figure 2. Random-split versus block-aware-split accuracy for (a) Logistic Regression and (b) Random Forest, across all four fault targets.

Table 3. Random-split vs. block-aware-split accuracy (4-fold CV, mean $\pm$ SD).

Target	Model	Random-split acc.	Block-aware acc.	Gap (pts)
Cooler	Logistic	1.000 $\pm$ 0.001	0.980 $\pm$ 0.023	2.0
Cooler	Random Forest	0.999 $\pm$ 0.002	0.954 $\pm$ 0.075	4.5
Cooler	MLP ensemble	0.998 $\pm$ 0.002	0.966 $\pm$ 0.049	3.2
Valve	Logistic	0.926 $\pm$ 0.005	0.697 $\pm$ 0.161	22.9
Valve	Random Forest	0.991 $\pm$ 0.005	0.869 $\pm$ 0.067	12.2

Target	Model	Random-split acc.	Block-aware acc.	Gap (pts)
Valve	MLP ensemble	0.917 ± 0.006	0.649 ± 0.145	26.8
Pump leakage	Logistic	0.995 ± 0.004	0.989 ± 0.012	0.6
Pump leakage	Random Forest	0.996 ± 0.002	0.996 ± 0.003	0.0
Pump leakage	MLP ensemble	0.993 ± 0.006	0.983 ± 0.012	1.0
Accumulator	Logistic	0.945 ± 0.003	0.831 ± 0.060	11.4
Accumulator	Random Forest	0.991 ± 0.003	0.804 ± 0.119	18.7
Accumulator	MLP ensemble	0.963 ± 0.006	0.807 ± 0.056	15.6

4.2. Architecture Comparison Under 4-Fold Block-Aware Cross-Validation

An initial single-split comparison (75/25 block-aware partition) suggested that both tree-staged architectures (RF+RF, Gradient Boosting) improved on the Random Forest baseline for both primary targets. Because a single split can overstate or understate true architectural differences, particularly on a

dataset with only 194 condition blocks of highly variable length. We repeated the full six-architecture comparison under 4-fold block-aware cross-validation (GroupKFold, groups defined identically to Section 3.3). For computational tractability across six architectures and four folds, ensemble size was reduced to three members per stage (from five) and MLP training iterations to 400 (from 600); this reduction was applied uniformly to every ensemble-based architecture, preserving the fairness of the comparison. Table 4 and Figure 3 report the resulting per-fold mean and standard deviation.

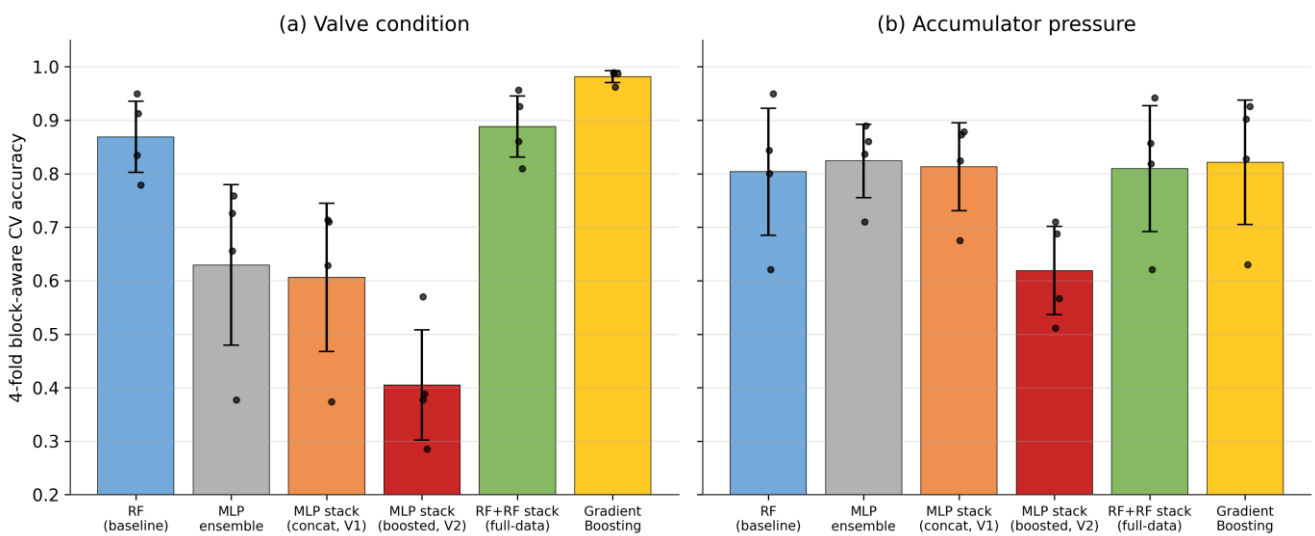


Figure 3. 4-fold block-aware CV accuracy by architecture on (a) valve condition and (b) accumulator pressure. Bars show mean $\pm$ SD across folds; points show individual fold accuracies.

Table 4. Architecture comparison, 4-fold block-aware cross-validation (mean $\pm$ SD across folds).

Architecture	Valve accuracy	Accumulator accuracy	Mechanism
Random Forest (baseline)	0.869 ± 0.067	0.804 ± 0.119	Single-stage
MLP ensemble (single-stage)	0.629 ± 0.150	0.824 ± 0.069	Single-stage
Stacked MLP, concatenation (V1)	0.606 ± 0.138	0.813 ± 0.082	Full-data, 3 stages

Architecture	Valve accuracy	Accumulator accuracy	Mechanism
Stacked MLP, boosting-style (V2)	0.405 ± 0.103	0.619 ± 0.083	Shrinking-subset, 3 stages
Two-stage Random Forest (RF+RF)	0.888 ± 0.057	0.810 ± 0.118	Full-data, 2 stages
Gradient Boosting	0.981 ± 0.011	0.821 ± 0.116	Full-data, staged residual correction.

Cross-validation confirms the single-split finding for valve condition but substantially revises it for accumulator pressure. On valve condition, the ranking is stable and well-separated: Gradient Boosting (0.981 ± 0.011) and RF+RF (0.888 ± 0.057) both clearly exceed the Random Forest baseline (0.869 ± 0.067), while both neural stacking variants fall well below the single-stage MLP ensemble (0.629 ± 0.150), with the boosting-style variant (V2) collapsing to 0.405 ± 0.103 . On accumulator pressure, however, the four best-performing architectures (RF, MLP ensemble, V1, RF+RF, GB) all fall within a

narrow, heavily overlapping band of 0.804-0.824, with standard deviations (0.069-0.119) far exceeding the differences between their means; only the boosting-style variant (V2, 0.619 ± 0.083) is distinguishable from the rest.

To assess whether the apparent single-split advantages for RF+RF and Gradient Boosting are genuine rather than an artifact of fold-to-fold variance, we compute paired per-fold accuracy differences against the Random Forest baseline (Table 5).

Table 5. Paired per-fold accuracy differences vs. baseline (robustness check).

Target	Comparison	Per-fold differences	Mean diff.	Sign consistent across all folds?
Valve	RF+RF – RF	+0.007, +0.031, +0.013, +0.025	+0.019	Yes (all positive)
Valve	GB – RF	+0.038, +0.209, +0.049, +0.154	+0.113	Yes (all positive)
Valve	V1 – MLP single	–0.027, –0.004, –0.049, –0.013	–0.023	Yes (all negative)
Valve	V2 – MLP single	–0.279, –0.093, –0.370, –0.156	–0.224	Yes (all negative)
Accumulator	RF+RF – RF	+0.018, –0.007, 0.000, +0.013	+0.006	No (mixed sign)
Accumulator	GB – RF	+0.027, –0.024, +0.009, +0.058	+0.018	No (mixed sign)
Accumulator	V1 – MLP single	–0.013, +0.018, –0.016, –0.034	–0.011	No (mixed sign)
Accumulator	V2 – MLP single	–0.127, –0.294, –0.201, –0.198	–0.205	Yes (all negative)

On valve condition, every comparison is sign-consistent across all four folds: RF+RF and Gradient Boosting improve on the Random Forest baseline in every single fold, and both neural stacking variants degrade relative to the single-stage MLP ensemble in every fold. This is strong evidence that these differences reflect genuine architectural properties rather than sampling variance. On accumulator pressure, only the boosting-style variant's degradation (V2 vs. MLP single) is sign-consistent; the RF+RF and Gradient Boosting improvements over the Random Forest baseline are not -- each is positive in some folds and negative in others, with a mean difference small relative to the fold-to-fold standard deviation. We therefore do not claim that RF+RF or Gradient Boosting reliably outperform a plain Random Forest on the accumulator target; the honest conclusion is that no tested architecture reliably beats the Random Forest baseline on this target, and that ac-

cumulator pressure classification is substantially more sensitive to which operating blocks happen to fall in the training versus test partition than valve condition is. This instability is itself consistent with the entropy-calibration and threshold-transfer findings in Sections 4.3-4.4, which independently identify accumulator pressure as the less stable of the two primary targets under block-wise distribution shift.

The feature-concatenation and boosting-style neural stacking mechanisms fail for the same underlying reasons on both targets. For V1, concatenating a previous stage's probability outputs as additional input features to a freshly initialized network does not constitute a genuine correction mechanism; the new network must relearn the underlying decision boundary from a randomly initialized state, and the added features provide limited additional signal. For V2, restricting each subsequent stage's training set to a shrinking pool of hard examples induces severe data starvation in later stages, which is the

dominant cause of its more pronounced and consistently negative degradation on both targets. In contrast, the two-stage Random Forest and Gradient Boosting architectures stage learners on the complete training set at every round; this is the more defensible design, and it is the design that produces a robust improvement -- but only on the target (valve condition) where the underlying classification problem itself is stable across operating blocks. This isolates the failure of the neural variants to the specific combination of base learner and training protocol used in ESSANN-style stacking, while also showing that even a well-designed staged architecture cannot

manufacture a reliable improvement on a target whose difficulty is dominated by block-to-block distribution shift rather than by within-block nonlinearity.

4.3. Calibration of Predictive Uncertainty

Independent of whether escalation to a subsequent stage improves point predictions, we evaluate whether ensemble predictive entropy is a reliable indicator of classification risk. Test-set predictions from the single-stage MLP ensemble are sorted by entropy and partitioned into quintiles (Table 6, Figure 4).

Table 6. Predictive-entropy calibration, single-stage MLP ensemble (test-set quintiles).

Entropy quintile	Valve: mean entropy	Valve: accuracy	Accumulator: mean entropy	Accumulator: accuracy
0-20% (most confident)	0.002	0.989	0.002	1.000
20-40%	0.050	0.908	0.034	0.943
40-60%	0.298	0.724	0.179	0.862
60-80%	0.633	0.506	0.463	0.736
80-100% (least confident)	0.993	0.409	0.740	0.625

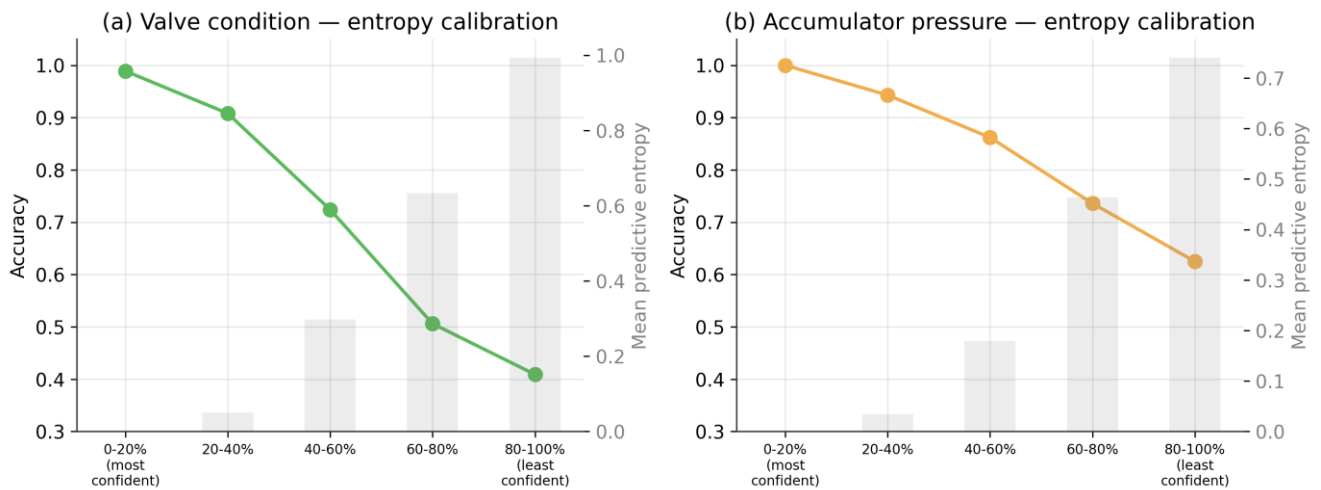


Figure 4. Entropy-binned accuracy for (a) valve condition and (b) accumulator pressure. Bars indicate mean predictive entropy per bin (right axis).

Accuracy declines monotonically with entropy quintile for both targets, from near-perfect (98.9-100%) in the lowest quintile to 40.9-62.5% in the highest. This confirms that ensemble predictive entropy rank-orders classification risk reliably, independent of the stacking-mechanism findings in Section 4.2 -- i.e., the model can correctly identify which of its own predictions are untrustworthy even when it cannot correct them through additional stacked stages.

4.4. Calibration Threshold Transfer

A practical escalation or confidence-gating mechanism requires selecting an entropy threshold above which a prediction is flagged for secondary review [4]. We compare a threshold computed from the 60th percentile of training-set entropy (available without any held-out data) against the same percentile computed on a held-out validation block, disjoint from

both the training and test partitions (Table 7, Figure 5).

Table 7. Escalation threshold value and resulting flagging rate on the test set.

Target	Threshold source	Threshold value	% of test set flagged “uncertain”
Valve	Training-set entropy (60th percentile)	0.138	98.4%
Valve	Validation-block entropy (60th percentile)	0.734	59.9%
Accumulator	Training-set entropy (60th percentile)	0.111	94.3%
Accumulator	Validation-block entropy (60th percentile)	0.712	44.3%

Effect of calibration source on the proportion of test samples flagged for review

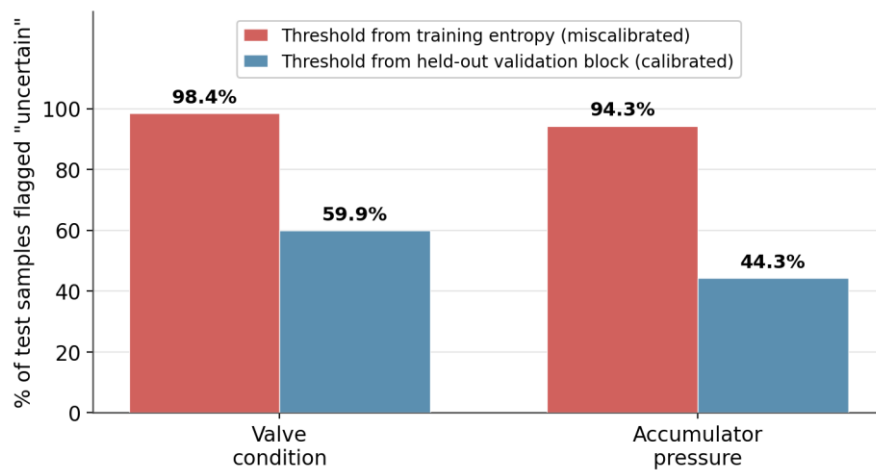


Figure 5. Effect of calibration source on the proportion of test samples flagged for secondary review.

Thresholds computed from training-set entropy are systematically too low, because the model is more confident on data resembling its own training distribution than on genuinely unseen operating blocks; this results in 94-98% of test samples being flagged “uncertain,” rendering the confidence-gating mechanism practically useless (almost nothing would be auto-approved). Calibrating the threshold on a held-out validation block -- itself unseen during training, structurally identical to the test set -- reduces the flagging rate to 44.3-59.9%, a proportion consistent with the entropy distribution actually observed on unseen operating blocks and therefore usable as an operating point for a real deployment. We note that, in a separate single-split RF+RF experiment used specifically to isolate this threshold-transfer effect, final classification accuracy was similar under both thresholds, because the second-stage model performed comparably on both the genuinely hard and the merely training-miscalibrated portions of the flagged set; the calibration correction is nonetheless necessary for the flagging decision itself to be actionable and interpretable by a control-room operator, independent of its effect on point-prediction accuracy.

5. Discussion

These results carry four implications for the design of AI-enabled instrumentation and control systems for industrial condition monitoring.

First, the evaluation protocol is not a secondary methodological detail on block-structured industrial datasets -- it determines whether reported accuracy reflects genuine generalisation to new operating conditions or merely interpolation within an already-observed condition block. A gap of up to 26.8 percentage points between random and block-aware splitting (Section 4.1) is large enough to change a reported result from state-of-the-art to mediocre. We recommend that future work on this and structurally similar datasets (e.g., other cyclically repeated industrial process datasets) report block-aware or group-aware validation explicitly, and that reviewers treat near-ceiling accuracy on such datasets with appropriate scrutiny absent an explicit accounting for block structure. This recommendation is not new in principle: leakage from naive

random splitting is a documented driver of inflated, non-reproducible results across machine-learning-based science generally [15], and group-aware cross-validation for data with temporal, spatial, or hierarchical dependence has been an established methodology in ecology and spatial statistics for over a decade [16]. What this study adds is a concrete demonstration that the same failure mode, at a magnitude large enough to overturn a state-of-the-art claim, is present in a widely used industrial condition-monitoring benchmark whose block structure has not, to our knowledge, been explicitly flagged in prior applications of this dataset (Section 2.1).

Second, architectural depth in stacked neural networks is not, by itself, a correction mechanism. The original motivation for elaborative stepwise stacking -- that each additional MLP stage progressively refines the prediction of the previous stage -- requires either (a) that later stages receive genuinely new information not available to earlier stages, or (b) that later stages are trained in a manner that specifically targets the earlier stage's errors without starving the training process of data. This requirement echoes Wolpert's original formulation of stacked generalisation [11], in which the meta-learner is deliberately trained on out-of-fold predictions so that it learns to correct genuine generalisation error rather than simply memorising the base learner's in-sample outputs -- a safeguard that neither the feature-concatenation nor the boosting-style variant tested here implements, since both train each subsequent stage on in-sample outputs from the preceding stage. Neither condition was met by the feature-concatenation or boosting-style variants tested here, and both consequently degraded accuracy relative to the single-stage ensemble consistently across every cross-validation fold on both classification targets (Table 5). Notably, the same staging principle succeeded when implemented with Random Forest [18] or Gradient Boosting [19, 20], both of which stage learners on the complete dataset every round and use base learners well suited to structured, engineered tabular features -- but, as discussed next, this success was itself target-dependent. Practitioners designing staged/stacked architectures for industrial tabular sensor data should therefore treat tree-based staging as the safer default choice over neural stacking, reserving the latter for settings where the input modality (raw waveforms, images, or genuinely sequential/temporal structure not captured by per-cycle statistical features) favours representation learning that trees cannot perform.

Third, an architecture's advantage on one fault target does not automatically transfer to another target within the same dataset and sensor suite, and single-split evaluation can obscure this. Gradient Boosting's superiority on valve condition is robust: it wins in every cross-validation fold by a wide margin (mean +11.3 points over Random Forest, minimum +3.8 points). On accumulator pressure, the same architecture's apparent advantage in the initial single-split experiment (+3.9 points) was not reproduced consistently under 4-fold cross-validation (mean +1.8 points, sign-inconsistent across folds) -

- the honest conclusion is that no architecture tested here reliably improves on a plain Random Forest for this target. This asymmetry is diagnostically informative rather than merely disappointing: it indicates that accumulator-pressure misclassification is driven substantially by block-to-block distribution shift (different operating blocks produce systematically different sensor signatures for the same nominal pressure class) rather than by within-block nonlinear structure that a more expressive or better-staged model could exploit. Comparable instability has been reported for other industrial multiclass fault-classification problems characterised by high-dimensional, nonlinear, and nonstationary sensor signals, where hybrid signal-decomposition and shrinkage-based pipelines have been proposed specifically to counter this kind of variability [5], suggesting that accumulator-pressure-like targets may be a recurring rather than idiosyncratic challenge in industrial condition monitoring rather than an artefact particular to this dataset. Architecture selection for industrial condition monitoring should therefore be validated per fault target, not assumed to transfer from one component's diagnostic model to another's, even within the same physical system and feature set. We regard this as a broadly applicable methodological caution: single train/test splits, even when block-aware, are insufficient to establish that an architectural improvement is genuine, and repeated or k-fold evaluation with explicit paired-difference reporting (as in Table 5) should be treated as a minimum standard for architecture-comparison claims in industrial informatics, consistent with broader recommendations for cross-validation under structured dependence [16].

Fourth, predictive uncertainty and predictive correction are distinct capabilities that should not be conflated in system design. Ensemble entropy is a reliable indicator of when a model's prediction should not be trusted (Section 4.3), independent of whether the model architecture can act on that signal to improve the prediction itself (Section 4.2), and independent of whether that architecture's advantage is robust across folds (this section, above). For safety- and cost-relevant industrial control decisions, this distinction motivates a two-part deployment architecture: a calibrated confidence estimate used purely as a routing/gating signal (auto-approve high-confidence predictions; route low-confidence predictions to human review, a secondary sensor, or a more conservative default control action), decoupled from whichever point-prediction model achieves the best accuracy. Critically, the threshold defining that routing boundary must be calibrated on data disjoint from training (Section 4.4); calibrating on training data alone can render the mechanism non-functional in deployment even though the underlying uncertainty ranking remains valid. This training-versus-deployment miscalibration gap echoes a broader, well-documented finding that modern neural networks tend to be poorly calibrated even when highly accurate, motivating dedicated post-hoc calibration procedures in the wider machine-learning literature [17]. Deep ensembles [12] and Monte Carlo dropout [13] are both commonly presented as practical remedies for this miscalibration, and our results

support that framing at the level of uncertainty ranking; however, they also show that good ranking alone does not guarantee a usable operating threshold without deployment-representative calibration data, a distinction not always made explicit in industrial applications of these techniques [14].

6. Limitations and Future Work

This study has four principal limitations. The architecture comparison rests on 4-fold block-aware cross-validation, but the calibration analyses still use a single block-aware train/validation/test partition, and the cross-validation configuration was deliberately reduced (three ensemble members, 400 iterations) for tractability; larger ensembles and a higher fold count — for example leave-one-block-out or repeated k-fold — would tighten the accuracy and calibration estimates, most importantly for the accumulator target, where no architecture reliably beats a Random Forest given the observed fold-to-fold variance. Feature engineering is also limited to per-cycle statistical descriptors: frequency-domain features (FFT, wavelet) that prior work found informative for this dataset were not evaluated and could shift the gaps between architectures.

Two limitations bound scope rather than confidence. Only two of the four fault targets (valve condition, accumulator pressure) were used for architectural comparison, because cooler condition and pump leakage are near-saturated across all models; a multi-task formulation predicting all four labels jointly — exploiting the label correlations noted in the preliminary analysis — is a natural extension. Finally, the paper offers an architectural guideline for confidence-gated control, not a validated closed-loop system; a supervisory-control simulation that drives maintenance-alarm or setpoint logic from confidence-gated forecasts, reporting false-alarm and missed-detection rates against a fixed-threshold baseline, is the clear next step.

7. Conclusion

This paper conducted a rigorous, leakage-aware, cross-validated empirical evaluation of stacked and ensemble machine learning architectures for fault diagnosis in industrial hydraulic power units. We found that naive random-split evaluation substantially inflates reported accuracy on this block-structured dataset (by up to 26.8 points). We also found that a faithfully reproduced neural stacking mechanism from the recent ESSANN literature consistently degrades accuracy relative to a single-stage ensemble across every cross-validation fold on both primary fault targets, once evaluated under a leakage-free protocol. The same staging concept succeeded when implemented with tree-based learners staged on the complete training set, but only robustly so for one of the two targets studied (valve condition): under 4-fold cross-validation with paired per-fold robustness checks, Gradient Boosting improved on a

plain Random Forest baseline in every fold for valve condition, while no architecture tested reliably improved on the Random Forest baseline for accumulator pressure once fold-to-fold variance was properly accounted for. This target-dependent result underscores that architectural comparisons in industrial condition monitoring must be validated per fault target and under repeated, block-aware cross-validation with explicit paired-difference reporting. A single train/test split, even a block-aware one, is not sufficient evidence of a genuine architectural advantage. We further showed that deep-ensemble predictive uncertainty is well calibrated in rank on both targets but requires validation-block calibration -- not training-data calibration -- to be usable as a deployment-ready confidence-gating signal. We recommend Gradient Boosting for targets exhibiting robust nonlinear structure (exemplified here by valve condition). A plain, well-tuned Random Forest as a safe default where architectural gains do not replicate across folds (exemplified here by accumulator pressure). A validation-calibrated ensemble uncertainty for confidence-gated routing of low-confidence predictions to secondary review in both cases, as a practical and evidence-supported basis for predictive condition monitoring of industrial hydraulic systems. This paper encourages the broader industrial-informatics community to adopt block-aware, repeated cross-validation with paired robustness reporting as standard practice for cyclically structured process data [15, 16].

Abbreviations

HPU	Hydraulic Power Unit
ESSANN	Elaborative Stepwise Stacked Artificial Neural Network
MLP	Multilayer Perceptron
RF	Random Forest
V1	Stacked MLP, Feature-concatenation Variant
V2	Stacked MLP, Boosting-style Variant
RF+RF	Two-stage Random Forest
GB	Gradient Boosting
PS1-PS6	Pressure Sensors 1-6
EPS1	Motor Power Sensor
FS1-FS2	Volume Flow Sensors
TS1-TS4	Temperature Sensors
VS1	Vibration Sensor
CE	Cooling Efficiency
CP	Cooling Power
SE	System Efficiency Factor
UCI	University of California, Irvine
PCA	Principal Component Analysis
DOI	Digital Object Identifier
CV	Cross-Validation
SD	Standard Deviation
AI	Artificial Intelligence
FFT	Fast Fourier Transform

Acknowledgments

We sincerely appreciate and acknowledge the efforts and contributions of Engr. Abdullateef Ogundipe, Electrical and Electronic Department, University of Ibadan, for his wealth of knowledge and encouragement.

Author Contributions

Emmanuel Unekuojo Enejor: Conceptualization, Formal Analysis, Methodology, Validation, Writing – original draft, Writing – review & editing

Nelson Oluwole Ibigbami: Data curation, Investigation, Validation, Visualization

Ismail Olusegun Lawal: Formal Analysis, Validation, Writing – review & editing

Temidayo Adeoti: Data curation, Methodology, Software, Validation

Ademola Adesoye: Data curation, Investigation, Software

Samson Oluwasina Ajibesin: Data curation, Investigation, Software

Adekunle Francis Adeniyi: Data curation, Software, Validation

Kolawole Bamidele Jayeola: Software, Validation, Visualization

Oludare Michael Shotade: Data curation, Software, Validation

Data Availability Statement

The data supporting the outcome of this research work have been reported in this manuscript.

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] Kamatchi Sundari, V.; Nithyashri, J.; Kuzhaloli, S.; Subburaj, J.; Vijayakumar, P.; Subha Hency Jose, P. Comparison Analysis of IoT Based Industrial Automation and Improvement of Different Processes - Review. *Mater. Today Proc.* 2021, *45*, 2595-2598. <https://doi.org/10.1016/j.matpr.2020.11.338>
- [2] Gundall, M.; Glas, C.; Schotten, H. D. Feasibility Study on Virtual Process Controllers as Basis for Future Industrial Automation Systems. In *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*; 2021; Vol. 1, pp 1080-1087. <https://doi.org/10.1109/ICIT46573.2021.9453651>
- [3] Garcia, J.; Rios-Colque, L.; Peña, A.; Rojas, L. Condition Monitoring and Predictive Maintenance in Industrial Equipment: An NLP-Assisted Review of Signal Processing, Hybrid Models, and Implementation Challenges. *Appl. Sci.* 2025, *15* (10), 5465. <https://doi.org/10.3390/app15105465>
- [4] Carvalho, T. P.; Soares, F. A. A. M. N.; Vita, R.; Francisco, R. da P.; Basto, J. P.; Alcalá, S. G. S. A Systematic Literature Review of Machine Learning Methods Applied to Predictive Maintenance. *Comput. Ind. Eng.* 2019, *137*, 106024. <https://doi.org/10.1016/j.cie.2019.106024>
- [5] Buabeng, A.; Simons, A.; Frempong, N. K.; Ziggah, Y. Y. Hybrid Intelligent Predictive Maintenance Model for Multiclass Fault Classification. *Soft Comput.* 2024, *28* (15), 8749-8770. <https://doi.org/10.1007/s00500-023-08993-1>
- [6] Asuquo, A. Enhancing Hydraulic System Reliability via XGBoost-Based Predictive Maintenance. *ChemRxiv* 2025 (0908). <https://doi.org/10.26434/chemrxiv-2025-4b395>
- [7] Kokate, M.; Kumar, T.; Kankar, P. K.; Miglani, A. Condition Monitoring of Hydraulic Systems Using Multi-Output Classification Convolutional Neural Networks. In *Proceedings of the Unified Conference of DAMAS, InCoME VIII and TEPEN Conferences*; Singh, M., Soni, G., Sinha, J., Ball, A. D., Gu, F., Ouyang, H., Featherston, C., Eds.; Springer Nature Switzerland: Cham, 2026; pp 371-384. https://doi.org/10.1007/978-3-031-95963-9_27
- [8] Al Shahrani, A. M.; Alomar, M. A.; Alqahtani, K. N.; Basingab, M. S.; Sharma, B.; Rizwan, A. Machine Learning-Enabled Smart Industrial Automation Systems Using Internet of Things. *Sensors* 2023, *23* (1), 324. <https://doi.org/10.3390/s23010324>
- [9] Nikolai Helwig, E. P. Condition Monitoring of Hydraulic Systems, 2015. <https://doi.org/10.24432/C5CW21>
- [10] Helwig, N.; Klein, S.; Schütze, A. Identification and Quantification of Hydraulic System Faults Based on Multivariate Statistics Using Spectral Vibration Features. *Procedia Eng.* 2015, *120*, 1225-1228. <https://doi.org/10.1016/j.proeng.2015.08.835>
- [11] Wolpert, D. H. Stacked Generalization. *Neural Netw.* 1992, *5* (2), 241-259. [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1)
- [12] Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2017; Vol. 30.
- [13] Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on Machine Learning*; PMLR, 2016; pp 1050-1059.
- [14] Maschler, B.; Weyrich, M. Deep Transfer Learning for Industrial Automation: A Review and Discussion of New Techniques for Data-Driven Machine Learning. *IEEE Ind. Electron. Mag.* 2021, *15* (2), 65-75. <https://doi.org/10.1109/MIE.2020.3034884>
- [15] Kapoor, S.; Narayanan, A. Leakage and the Reproducibility Crisis in Machine-Learning-Based Science. *Patterns* 2023, *4* (9). <https://doi.org/10.1016/j.patter.2023.100804>
- [16] Roberts, D. R.; Bahn, V.; Ciuti, S.; Boyce, M. S.; Elith, J.; Guiller-Arroita, G.; Hauenstein, S.; Lahoz-Monfort, J. J.; Schröder, B.; Thuiller, W.; Warton, D. I.; Wintle, B. A.; Hartig, F.; Dormann, C. F. Cross-Validation Strategies for Data with Temporal, Spatial, Hierarchical, or Phylogenetic Structure. *Ecography* 2017, *40* (8), 913-929. <https://doi.org/10.1111/ecog.02881>

- [17] Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K. Q. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*; PMLR, 2017; pp 1321-1330.
- [18] Breiman, L. Random Forests. *Mach. Learn.* 2001, 45 (1), 5-32. <https://doi.org/10.1023/A:1010933404324>
- [19] Friedman, J. H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* 2001, 29 (5), 1189-1232.
- [20] Fouladvand, S.; Noshad, M.; Goldstein, M. K.; Periyakoil, V.; Chen, J. H. Mild Cognitive Impairment: Data-Driven Prediction, Risk Factors, and Workup. *AMIA Summits Transl. Sci. Proc.* 2023, 2023, 167-175.
- [21] Chawathe, S. S. Condition Monitoring of Hydraulic Systems by Classifying Sensor Data Streams. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*; 2019; pp 0898-0904. <https://doi.org/10.1109/CCWC.2019.8666564>