Transformer Health Index Prediction Using Static and Temporal Learning Models

Omar Ahmed Mohammed Hamood Al-Shaikh[1], Arfah Ahmad[2]*,
Ahmad Jazlan Haja Mohideen[3], Muhammad Sharil Yahaya[4]
Faculty of Electrical Technology and Engineering,
Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia[1, 2, 4]
Kulliyyah of Engineering, International Islamic University Malaysia (IIUM), Kuala Lumpur, Malaysia[3]

***Abstract*—Transformer health index (THI) prediction supports condition-based maintenance by mapping dissolved-gas, oil-quality, and furan indicators to an interpretable asset-condition score. This study develops a supervised benchmarking framework using 3,392 transformer oil diagnostic records from 510 transformers. Thirteen diagnostic features from dissolved gas analysis, oil quality analysis, and furan analysis are evaluated across three modelling groups: static machine learning, static deep learning, and variable-length temporal deep learning. Transformer-level splitting is applied to reduce data leakage from repeated transformer histories. Static gradient-boosted tree models achieved the strongest results. XGBoost obtained the lowest test root mean square error of 3.4254 with a coefficient of determination of 0.9780 and health-index class accuracy of 88.20 percent. The best temporal model, the liquid time-constant neural network (LTC-LNN), achieved a test root mean square error of 4.8691 and a coefficient of determination of 0.9567. Permutation feature importance identified 2FAL, C2H2, and dielectric breakdown as the most influential predictors. For the present dataset, static gradient-boosted tree models produced the strongest observed point-estimate performance, while LTC-LNN remained a promising temporal-learning alternative for repeated diagnostic histories. However, the uncertainty analysis indicates that small numerical differences among the leading static models should not be interpreted as definitive evidence of model superiority.**

Keywords*—Transformer health index; machine learning; deep learning; liquid neural networks; oil diagnostic data; condition-based maintenance; smart grid asset management

I. Introduction

Oil-immersed power and distribution transformers are among the most critical assets in electrical networks. Their failure can cause supply interruption, long replacement lead times, safety hazards, and high operational costs. For this reason, transformer condition assessment has shifted from age-based replacement to condition-based and risk-informed asset management. Oil diagnostic testing is central to this transition because dissolved gas analysis (DGA), oil quality analysis (OQA), and furan analysis provide non-invasive evidence of thermal faults, electrical faults, oil degradation, and paper-insulation ageing [1-7].

The transformer health index (THI) is widely used to consolidate multiple diagnostic indicators into a single condition score. Weighted-score-sum approaches, including Kinectrics-type and other utility-oriented health-index methods, remain attractive because they are transparent and interpretable for engineers [8-15]. However, these methods usually apply fixed scoring thresholds and fixed weights. As a result, they may not fully capture nonlinear dependencies among DGA gases, oil-quality parameters, and furan compounds, especially when the same transformer has repeated oil-test histories.

Machine learning (ML) and deep learning (DL) provide a data-driven route to learn the relationship between diagnostic measurements and a supervised THI target. Ensemble tree models such as random forest, gradient boosting, XGBoost, LightGBM, and CatBoost are especially relevant for tabular engineering data because they can model nonlinear feature interactions and threshold-like behaviour [16-23]. Neural models provide another route for static and sequential learning, including multilayer perceptrons and other deep feed-forward architectures [24-26], recurrent models such as long short-term memory (LSTM) and gated recurrent units (GRUs) [27], [28], attention-based Transformers [29], temporal convolutional networks (TCNs) [30], and continuous-time liquid neural networks [31-35].

Despite the growing interest in intelligent transformer diagnostics, two methodological gaps remain. First, many studies evaluate either classical ML or a single neural model, without distinguishing between static tabular learning and temporal history learning. Second, repeated oil-test records from the same transformer are often treated as independent samples, which can lead to leakage if samples from the same transformer appear in both the training and testing sets. This study addresses these issues by using transformer-level data splitting and by separating the evaluation into three modelling groups: static ML, static DL, and variable-length sequence DL.

The main contributions of this study are as follows: 1) a leakage-aware benchmarking framework for THI prediction using 3,392 oil diagnostic records; 2) a three-level experimental comparison between static ML, static DL, and variable-length sequence DL; 3) a liquid time-constant neural network (LTC-LNN) implementation for repeated transformer oil-test histories; and 4) a feature-importance analysis identifying the diagnostic variables most responsible for THI prediction.

* Corresponding author.

This research was funded by MOHE Malaysia through FRGS under Grant Codes FRGS/1/2024/TK07/UTEM/02/24 and FRGS/1/2024/FTKE/F00579.

II. Related Work

DGA is one of the most established diagnostic tools for mineral-oil-filled transformers. IEEE and IEC guides define gas interpretation procedures and support the identification of thermal, partial discharge, and arcing faults [1], [2]. OQA parameters such as breakdown voltage, moisture, neutralization value (NV), interfacial tension, and colour provide complementary information about insulation-fluid degradation [3], [4]. Furan analysis, especially 2-furfural, is commonly associated with paper-insulation ageing [5].

Health-index methods integrate these diagnostic channels into an asset-level condition indicator. Early approaches combined condition factors, scores, and weights derived from expert knowledge and operational practice [8-13]. These methods are useful because they preserve physical interpretability. However, their rule-based scoring structure can become restrictive when diagnostic indicators interact nonlinearly or when different assets exhibit different ageing trajectories.

ML methods have been increasingly used for asset condition assessment because they can learn nonlinear functions from data. Random forests and gradient boosting provide robust baselines for tabular prediction [16], [17] while AdaBoost, support vector regression, and k-nearest-neighbour models provide complementary comparisons [18], [19], [23]. Modern boosting frameworks such as XGBoost, LightGBM, and CatBoost are particularly suitable for mixed-scale engineering diagnostics because they handle nonlinear feature interactions efficiently [20-22]. DL methods can model more complex functions, but they typically require careful optimization and regularization [24-26], [36-38]. Sequence models such as LSTM, GRU, Transformers, and TCNs are designed for ordered data, although their usefulness depends on sufficiently long and informative temporal histories [27-30].

Liquid time-constant networks (LTCs) are continuous-time recurrent neural models that adapt their hidden-state dynamics through input-dependent time constants [31], [32]. Neural circuit policies and closed-form continuous-time models further demonstrate the relevance of liquid and continuous-time neural models for dynamic systems [33], [34]. For transformer condition monitoring, liquid neural networks based on liquid time-constant dynamics (LTC-LNNs) are attractive because oil diagnostic records are irregularly sampled and may contain short histories. This motivates the sequence-learning experiment in this study.

III. Materials and Methods

The study used a model-ready transformer oil diagnostic dataset containing 3,392 actual records, excluding the Excel header row. The diagnostic inputs consisted of 13 parameters: H2, CH4, CO, CO2, C2H4, C2H6, C2H2, dielectric breakdown, moisture, neutralization value (NV), interfacial tension (IFT), colour, and 2FAL. The supervised output was Real_HI%, the reference health index (HI), calculated using a conventional weighted health-index procedure. For maintenance-oriented evaluation, continuous HI values were mapped into five condition classes: Good ($80 < HI \leq 100$), Acceptable ($60 < HI \leq 80$), Need Caution ($40 < HI \leq 60$), Poor ($20 < HI \leq 40$), and Critical ($0 \leq HI \leq 20$). Identifier fields were not used as prediction features; Serial No. was used only to define transformer-level splits and repeated histories.

Three experiments were implemented. Experiment 1 evaluated static ML models using single oil-test records. Experiment 2 evaluated static DL models using the same single-record structure. Experiment 3 evaluated variable-length sequence DL models using transformers with at least two repeated oil-test samples. For a transformer with n chronological records, $n-1$ sequences were generated, beginning with the second record. Each sequence used all records from the first sample to the current sample and predicted the current Real_HI%. The time interval between consecutive oil-test records was included as an additional temporal feature for sequence modelling.

All experiments used transformer-level splitting so that records from the same transformer were not shared across training, validation, and test partitions. This design is important because repeated samples from one transformer are statistically dependent and may otherwise inflate the apparent predictive performance. Feature scaling was fitted on the training set only. Missing diagnostic feature values in the prepared dataset were handled using median imputation where required. The overall experimental workflow is illustrated in Fig. 1. The diagnostic variables and their roles are summarized in Table I, while the dataset composition and transformer-level splits are presented in Table II. The supervised THI target distribution and the distribution of repeated oil-test records per transformer are shown in Fig. 2 and 3, respectively.

TABLE I. Diagnostic Variables used in the Study

Group	Variables	Purpose
DGA gases	H2, CH4, CO, CO2, C2H4, C2H6, C2H2	Internal electrical and thermal fault indicators
Oil quality	Dielectric breakdown, moisture, neutralization value, interfacial tension, colour	Insulating-oil degradation and dielectric condition
Paper insulation	2FAL	Cellulose ageing indicator
Target	Real_HI%	Supervised transformer health index output
Identifier for split	Serial No. / Transformer_ID	Used only for leakage-safe grouping; not used as a feature

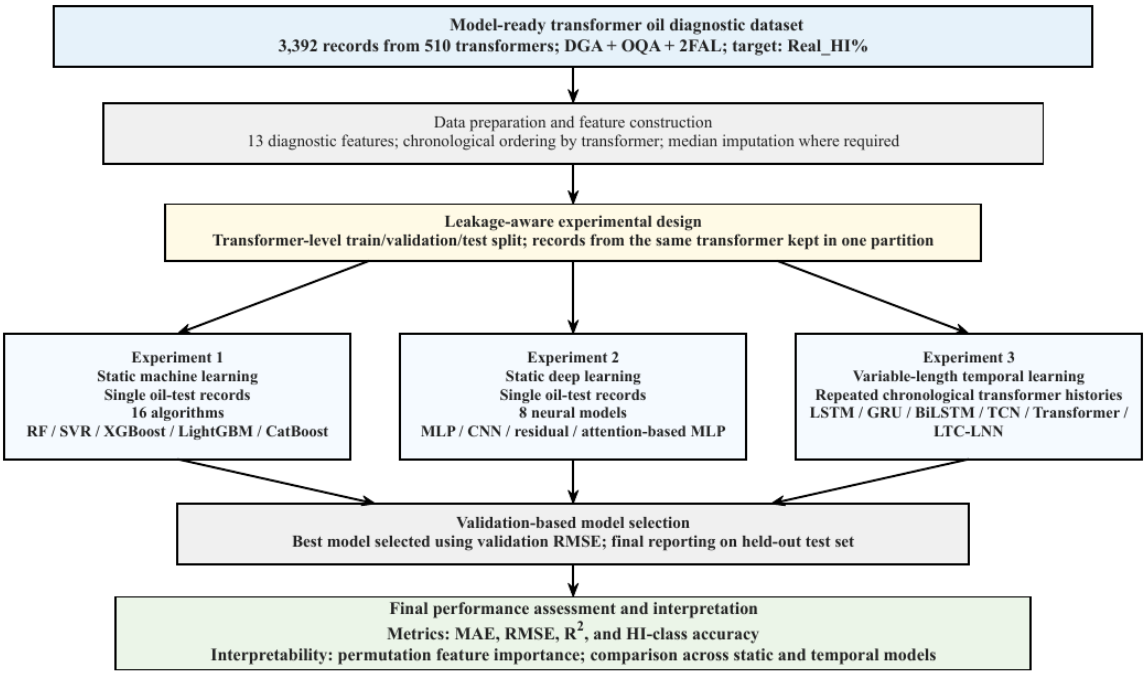

Fig. 1. Overall framework of the three-level benchmarking design.

TABLE II. DATASET AND SPLIT SUMMARY

Component	Unit	Value	Description
Static ML and Static DL	Records	3392	510 transformers; 13 diagnostic features; 3,392 records retained
Static train split	Records/transformers	2352 / 357	Transformer-level split
Static validation split	Records/transformers	489 / 76	Transformer-level split
Static test split	Records/transformers	551 / 77	Transformer-level split
Sequence repeated subset	Records/transformers	3305 / 425	Transformers with at least two records
Sequence train sequences	Sequences/transformers	2019 / 297	Variable-length histories
Sequence validation sequences	Sequences/transformers	432 / 64	Variable-length histories
Sequence test sequences	Sequences/transformers	429 / 64	Variable-length histories

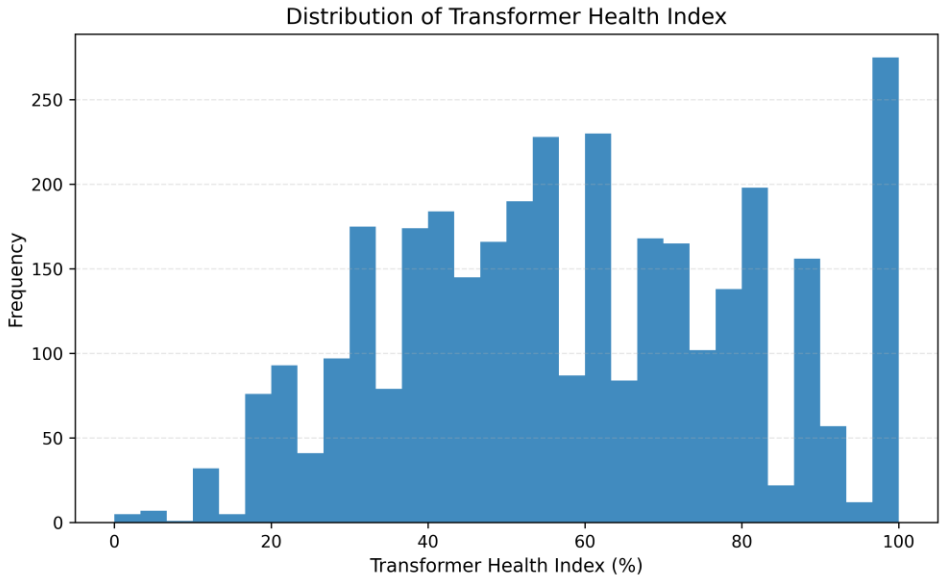

Fig. 2. Distribution of the supervised transformer health index target.

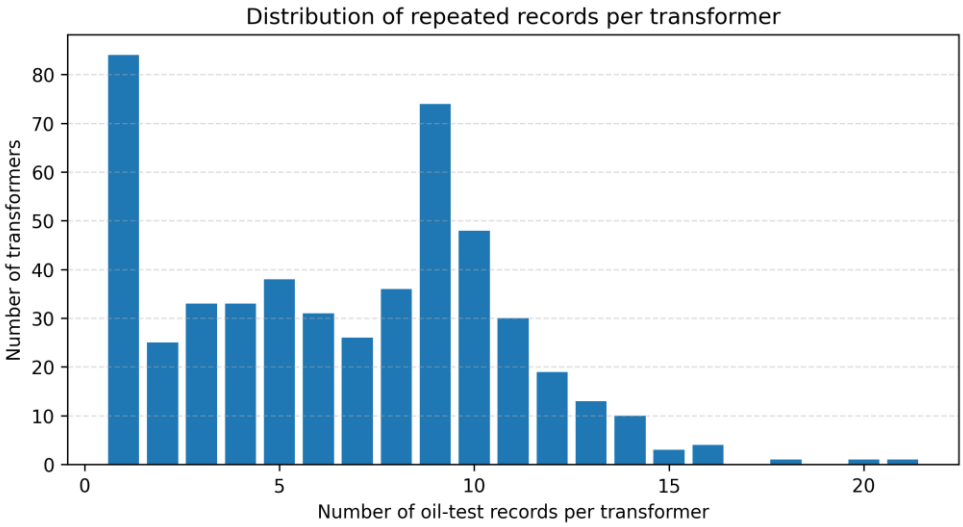

Fig. 3. Distribution of repeated oil-test records per transformer.

IV. Modeling Framework

Experiment 1 trained 16 static ML regressors: dummy mean baseline, linear regression, ridge, lasso, elastic net, k-nearest neighbours, support vector regression, decision tree, random forest, extra trees, gradient boosting, AdaBoost, histogram gradient boosting, XGBoost, LightGBM, and CatBoost. The model set was designed to cover classical linear baselines, kernel learning, tree ensembles, and modern gradient-boosting frameworks [16-23]. The best model was selected using validation RMSE, and all models were also evaluated on the held-out test set.

Experiment 2 trained eight static DL models: MLP-ANN, deep MLP, regularized MLP, residual MLP, wide-deep MLP, feature-attention MLP, 1D-CNN, and CNN-MLP hybrid. The feed-forward architectures were motivated by standard back-propagation and deep-learning principles [24-26], while the residual and regularized designs used skip connections, dropout, and batch normalization to improve trainability and reduce overfitting [39], [36], [38]. Inputs were scaled using statistics from the training set only, the target was scaled to [0, 1], and Adam-based optimization with early stopping and learning-rate reduction was used for neural training [36].

Experiment 3 trained variable-length sequence models on repeated transformer histories: LSTM, GRU, BiLSTM, Transformer encoder, temporal convolutional network (TCN), and liquid time-constant neural network (LTC-LNN) variants U32 and U64. LSTM and GRU were included as recurrent baselines for ordered diagnostic histories [27], [28], the Transformer encoder represented attention-based sequence modelling [29], and TCN represented convolutional temporal modelling [30]. The time gap between consecutive samples was added as an additional sequence feature, increasing the sequence input size from 13 to 14. The LTC-LNN models used the ncps.torch implementation and were motivated by liquid time-constant, neural circuit policy, closed-form continuous-time, and neural ordinary differential equation concepts [31-35, 40].

A. Deep Learning Hyperparameter Configuration

To improve reproducibility, the architectural and training configurations used in the static and sequence deep learning experiments are summarized below. For the static deep learning experiment, the input features were scaled using statistics fitted on the training set only, the target was scaled to [0, 1], and Adam optimization with early stopping and learning-rate reduction was used. The principal architecture and training settings are summarized in Table III.

TABLE III. Static Deep Learning Model Configurations

Model	Architecture	Dropout	Initial LR
MLP-ANN	Dense 64-32-16-1	None	0.001
Deep MLP	Dense 128-64-32-16-1	None	0.001
Regularized MLP	Dense 128-64-32-1 + BatchNorm	0.25	0.001
Residual MLP	64-unit residual blocks -> Dense 32-1	None	0.001
Wide-Deep MLP	Deep 128-64-32 + linear wide branch	0.20	0.001
Feature-Attention MLP	13-unit attention gate -> 128-64-32-1	0.20	0.001
1D-CNN	Conv1D 32 -> Conv1D 64 -> GAP -> Dense 32-1	None	0.001
CNN-MLP Hybrid	Conv1D 32 -> Conv1D 64 -> Dense 64-32-1	0.20	0.001

All static deep learning architectures used ReLU activation in the hidden layers and a linear output layer for regression. The saved training histories confirm an initial learning rate of 0.001 for all eight static DL models, with adaptive learning-rate reduction according to validation performance.

For the variable-length sequence experiment, each time step contained 14 input features consisting of the 13 diagnostic variables and Time_Gap_Days. The principal sequence architectures, dropout rates, initial learning rates, and validation-selected training epochs are summarized in Table IV.

TABLE IV. SEQUENCE DEEP LEARNING MODEL CONFIGURATIONS

Model	Architecture	Dropout	Initial LR	Best epoch
LSTM	1-layer LSTM, hidden 64 -> Dense 32-1	0.10	0.001	78
GRU	1-layer GRU, hidden 64 -> Dense 32-1	0.10	0.001	117
BiLSTM	1-layer BiLSTM, 32/dir -> Dense 32-1	0.10	0.001	106
Transformer	2 layers; d_model 64; 4 heads; FFN 128 -> Dense 32-1	0.10	0.001	114
TCN	2 x 64 channels; k=3; dilations 1,2 -> Dense 32-1	0.10	0.001	52
LTC-LNN U32	32 liquid -> Dense 32-1; 6 ODE unfolds	0.10	0.001	276
LTC-LNN U64	64 liquid -> Dense 32-1; 6 ODE unfolds	0.10	0.0005	282

The recurrent baselines used one recurrent layer followed by a 32-unit dense regression head. The Transformer Encoder used two encoder layers (d_model = 64, four attention heads, and a feed-forward dimension of 128), whereas the TCN used two 64-channel temporal blocks with kernel size 3 and dilations of 1 and 2. All sequence configurations used a dropout rate of 0.10. The LTC-LNN U32 and U64 used six ODE unfolding steps. Model selection was based on the lowest validation RMSE, and the corresponding best validation checkpoint was retained for final evaluation.

B. Additional Evaluation Protocols

To quantify uncertainty in the comparison between the leading static ML models, a paired transformer-level bootstrap analysis with 10,000 resamples was performed on the held-out test set. Transformers, rather than individual records, were resampled with replacement to preserve the dependence among repeated measurements from the same transformer. The bootstrap distributions were used to derive 95% confidence intervals (CIs) for the RMSE values of XGBoost and CatBoost and for their paired RMSE difference. Statistical distinguishability was assessed at the 0.05 significance level based on whether the 95% CI of the paired RMSE difference excluded zero.

Computational efficiency was also evaluated for four representative models: XGBoost, CatBoost, MLP-ANN, and LTC-LNN U32. XGBoost represented the static ML model with the lowest test RMSE, CatBoost represented the validation-selected static ML model, MLP-ANN represented the best static DL model, and LTC-LNN U32 represented the best variable-length sequence DL model. All runtime measurements were obtained in the same CPU-only environment using an Intel Xeon processor at 2.20 GHz with 12.67 GB RAM. Wall-clock training time, average inference latency, peak additional resident memory above the process baseline, and serialized model-plus-preprocessing storage size were recorded. Neural-network trainable parameters were counted directly from the implemented architectures. A fixed FLOP count was reported only where it was technically meaningful; for tree ensembles, computation depends on tree traversal, while for LTC-LNN it depends on variable sequence length and the number of ODE unfolding steps. Consequently, structural complexity is reported for these models instead of a potentially misleading fixed FLOP estimate.

Residual analysis was also performed for representative models from the three experimental groups. Residuals were defined as the actual HI minus the predicted HI. For each model, the mean residual was used to assess systematic prediction bias, while the residual standard deviation and variance were used to characterize error dispersion. Tail and worst-case behavior were quantified using the 95th percentile of the absolute error and the maximum absolute error. Because the evaluated models generate deterministic point predictions at inference, a per-sample predictive variance is not intrinsically available; therefore, residual variance across held-out observations was used as an empirical measure of prediction variability. Condition-class-specific errors were also examined to identify whether performance degraded across different HI operating ranges.

To examine dependence among the diagnostic predictors, Pearson correlation analysis and variance inflation factor (VIF) diagnostics were performed on the 13 model inputs. Pairwise correlation coefficients were used to identify strongly related diagnostic variables, while VIF quantified the extent to which each predictor could be linearly explained by the remaining inputs. VIF values above 5 were treated as indicating elevated multicollinearity and values above 10 as indicating severe multicollinearity. Because the benchmarking framework intentionally retained the complete set of physically meaningful DGA, oil-quality, and furan indicators, correlated variables were not removed solely based on these diagnostics; instead, the resulting dependence was considered when interpreting feature-importance rankings.

To assess prediction confidence for the validation-selected static ML model, a post-hoc prediction-interval calibration analysis was performed using the validation and test partitions. Absolute prediction errors from the 489-record validation set were used to determine finite-sample residual quantiles for nominal 90% and 95% prediction intervals. For a test prediction, the corresponding calibrated residual quantile was added to and subtracted from the point prediction, with interval limits restricted to the valid HI range of 0-100%. The resulting intervals were evaluated on the held-out 551-record test set by comparing their nominal coverage levels with the empirical proportion of true HI values contained within the intervals. Because the validation partition was also used for model selection, these intervals are

interpreted as an empirical post-hoc uncertainty assessment rather than a formal deployment guarantee.

V. RESULTS

The test performance of the leading static ML models is summarized in Table V.

TABLE V. TOP STATIC ML MODELS RANKED BY TEST RMSE

Model	MAE	RMSE	R^2	HI acc.
XGBoost	2.5759	3.4254	0.9780	88.20%
LightGBM	2.5813	3.5213	0.9767	87.48%
CatBoost	2.6569	3.6017	0.9757	86.39%
Hist. Gradient Boosting	2.7084	3.6870	0.9745	86.21%
Gradient Boosting	3.1522	4.1146	0.9683	84.21%
Random Forest	3.6697	5.0080	0.9530	82.58%
Extra Trees	4.1695	5.6000	0.9412	80.04%
SVR-RBF	5.1564	6.8787	0.9113	75.50%
Decision Tree	5.0136	7.9868	0.8804	78.22%
AdaBoost	6.6450	8.0613	0.8781	71.14%

The static ML experiment produced the strongest observed point-estimate results. CatBoost was selected by validation RMSE, achieving validation RMSE of 3.7092 and validation R^2 of 0.9706. On the held-out test set, the validation-selected CatBoost model achieved MAE of 2.6569, RMSE of 3.6017, R^2 of 0.9757, and HI-class accuracy of 86.39%. XGBoost achieved the lowest observed test RMSE among all ML models at 3.4254, with R^2 of 0.9780 and HI-class accuracy of 88.20%.

To determine whether the observed difference between XGBoost and CatBoost reflected a statistically meaningful performance improvement, a paired transformer-level bootstrap analysis with 10,000 resamples was conducted on the held-out test set. XGBoost achieved a test RMSE of 3.4254 with a 95% bootstrap CI of 3.1391-3.7030, whereas CatBoost achieved an RMSE of 3.6017 with a 95% CI of 3.1828-4.0522. The paired RMSE difference (XGBoost - CatBoost) was -0.1763, with a 95% CI of -0.5723 to 0.1423 and a two-sided bootstrap p-value of 0.3816. Because the confidence interval for the paired difference included zero and $p > 0.05$, the lower point-estimate RMSE obtained by XGBoost was not statistically distinguishable from that of CatBoost at the 95% confidence level. Therefore, XGBoost should be interpreted as having the lowest observed test RMSE rather than demonstrating statistically significant superiority over CatBoost. The corresponding confidence intervals are summarized in Table X.

The strong performance of gradient-boosted tree models is consistent with the structure of the problem. The supervised THI target was derived from diagnostic factors with nonlinear thresholds, weighted combinations, and class boundaries. Tree boosting can represent such piecewise relationships efficiently, which explains why CatBoost, XGBoost, and LightGBM achieved lower errors than linear models, SVR, KNN, and standalone neural models. The static ML test RMSE comparison is shown in Fig. 4, while the agreement between the actual and predicted THI values for the validation-selected static ML model is illustrated in Fig. 5.

The test performance of the evaluated static DL models is presented in Table VI.

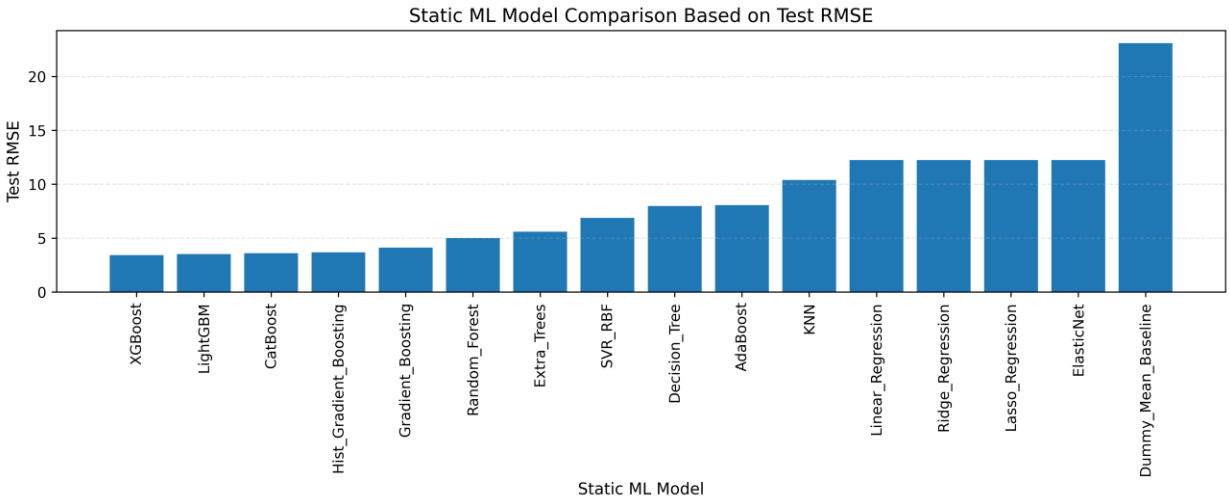

Fig. 4. Static ML model comparison based on test RMSE.

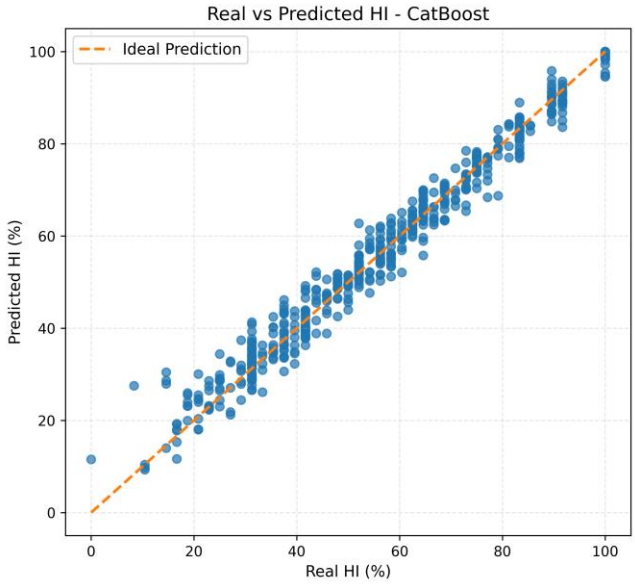

Fig. 5. Real versus predicted THI for the validation-selected static ML model.

TABLE VI. STATIC DL MODELS RANKED BY TEST RMSE

Model	MAE	RMSE	R^2	HI acc.
MLP-ANN	4.7676	6.4480	0.9220	79.49%
Deep MLP	8.7559	10.9349	0.7758	59.35%
Feature-Attention MLP	8.4871	11.0802	0.7698	60.62%
Residual MLP	11.5511	15.1814	0.5678	50.82%
1D-CNN	12.6159	15.7276	0.5362	43.01%
Wide-Deep MLP	13.3638	17.3844	0.4333	46.28%
Regularized MLP	33.6164	39.2058	-1.8824	12.34%
CNN-MLP Hybrid	47.0373	51.3838	-3.9511	3.81%

The best static DL model was MLP-ANN, with a test MAE of 4.7676, RMSE of 6.4480, R^2 of 0.9220, and HI-class accuracy of 79.49%. Although this result confirms that neural networks can learn the THI mapping, static DL did not outperform the boosted tree models. In contrast, the Regularized MLP and CNN-MLP Hybrid produced negative R^2 values of -1.8824 and -3.9511, respectively, indicating that these architectures performed worse than a mean-target baseline on the held-out test set and failed to generalize effectively. This suggests that the additional architectural complexity and regularization were not beneficial for the present tabular THI prediction task. These models were retained for benchmarking completeness but were not considered suitable candidates for final model selection. Overall, deeper and more complex neural structures degraded performance, indicating that the available tabular dataset favors regularized, shallow nonlinear models over high-capacity neural architectures. The static DL test RMSE comparison is shown in Fig. 6, and the training and validation loss of the best static DL model is presented in Fig. 7.

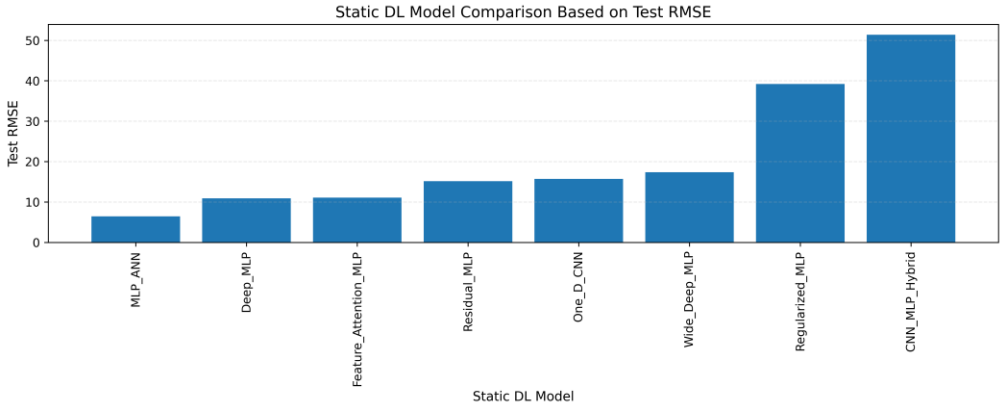

Fig. 6. Static DL model comparison based on test RMSE.

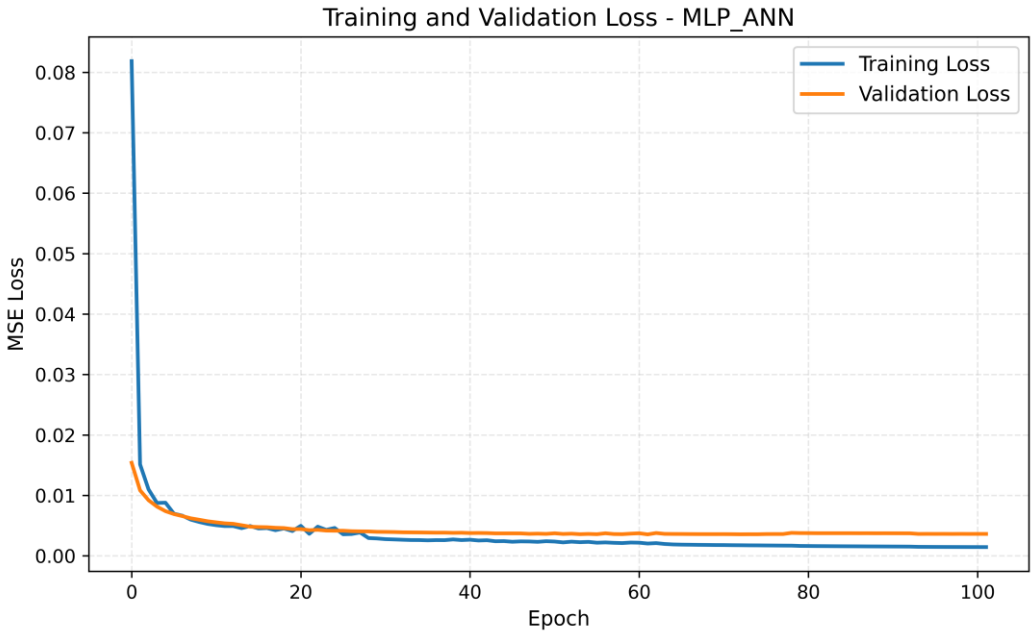

Fig. 7. Training and validation loss of the best static DL model.

The validation performance and best training epoch of each variable-length sequence model are summarized in Table VII.

Among the variable-length sequence DL models, LTC-LNN U32 achieved the best validation performance, with validation MAE of 3.8029, RMSE of 4.8595, R^2 of 0.9519, and HI-class accuracy of 82.18%. On the test set, the same model achieved MAE of 3.7766, RMSE of 4.8691, R^2 of 0.9567, and HI-class accuracy of 79.49%. These results show that LTC-LNN outperformed the LSTM, GRU, BiLSTM, TCN, and Transformer sequence baselines on the repeated-history subset.

TABLE VII. VARIABLE-LENGTH SEQUENCE DL MODELS RANKED BY VALIDATION RMSE

Model	Val. MAE	Val. RMSE	Val. R^2	Val. HI acc.	Epoch
LTC-LNN U32	3.8029	4.8595	0.9519	82.18%	276
LTC-LNN U64	4.0200	5.0791	0.9475	79.63%	282
Transformer	4.4164	5.5786	0.9367	74.77%	114
BiLSTM	5.3077	6.7855	0.9063	71.53%	106
GRU	5.1330	6.7932	0.9061	73.15%	117
TCN	5.3005	6.8272	0.9051	72.69%	52
LSTM	5.1759	6.8807	0.9036	73.15%	78

The sequence models were trained on a repeated-history subset containing 425 of the 510 transformers used in the static experiments, corresponding to a 16.7% reduction in the number of transformers. This restriction was necessary because temporal sequence construction requires at least two diagnostic records per transformer. Nevertheless, the repeated-history subset retained 3,305 of the original 3,392 records (97.4%). Thus, although 85 transformers were excluded from the sequence experiment, they accounted for only 87 records (2.6% of the complete dataset). The effect of this subset restriction on predictive performance cannot be isolated from the present experiments because the static and sequence models use different sample structures; a matched evaluation of all model groups on the same 425-transformer subset would be required to quantify this effect directly. Because the static and sequence experiments use different sample structures and held-out test sets, their performance metrics should be interpreted as descriptive cross-group comparisons rather than strictly paired comparisons. The sequence experiment therefore primarily evaluates the relative performance of temporal models under the same repeated-history setting. The validation RMSE comparison of the variable-length sequence models is shown in Fig. 8, while the actual and predicted THI values for the best sequence model are illustrated in Fig. 9. The overall comparison across the three modelling groups is summarized in Table VIII.

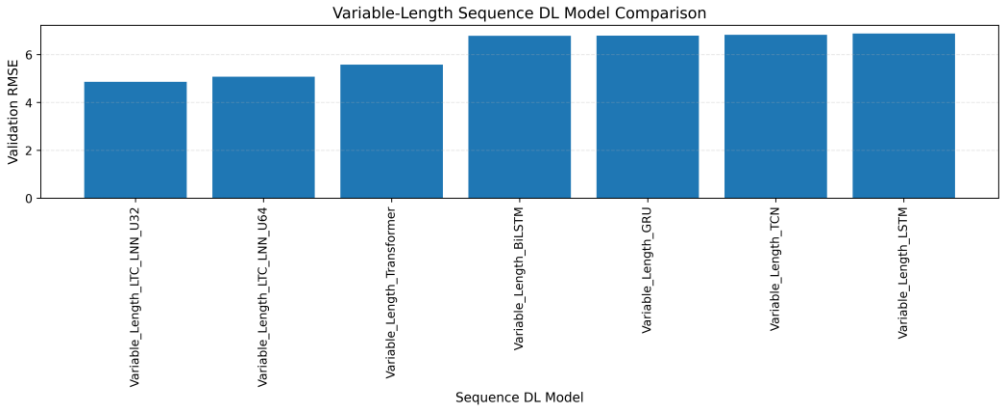

Fig. 8. Variable-length sequence DL model comparison based on validation RMSE.

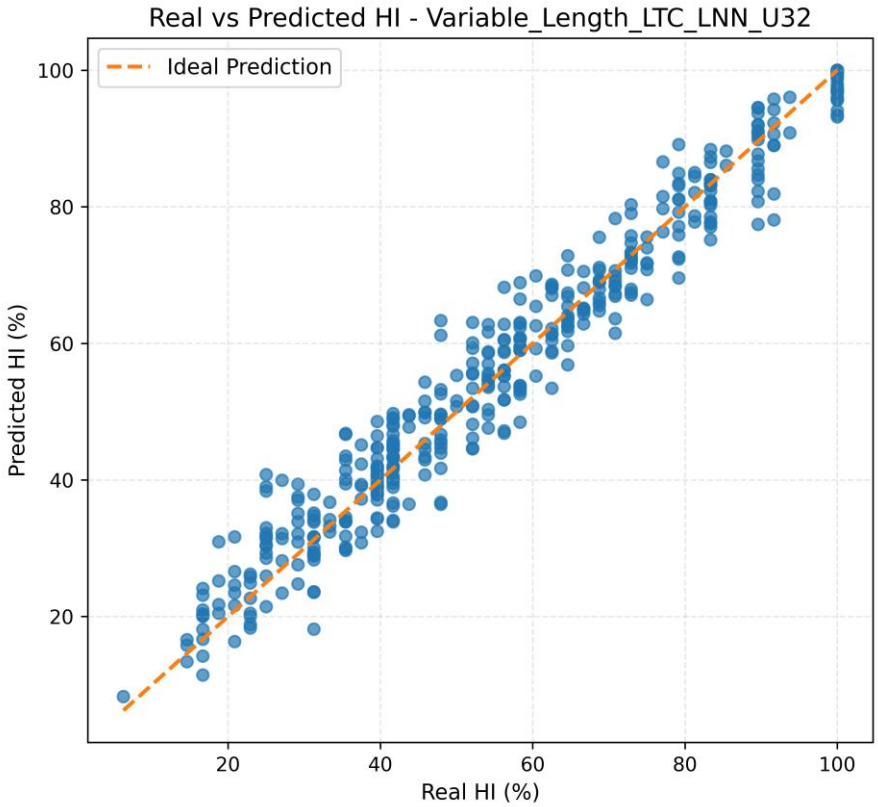

Fig. 9. Real versus predicted THI for the best variable-length sequence model.

TABLE VIII. OVERALL COMPARISON ACROSS MODELLING GROUPS

Model group	Test RMSE	Test R^2	Test HI acc.
Static ML (CatBoost, validation-selected)	3.6017	0.9757	86.39%
Static ML (XGBoost, lowest test RMSE)	3.4254	0.9780	88.20%
Static DL (MLP-ANN)	6.4480	0.9220	79.49%
Sequence DL (LTC-LNN U32)	4.8691	0.9567	79.49%

The test RMSE values of the representative models from each experimental group are compared in Fig. 10. The ten most influential diagnostic variables identified by permutation feature importance are summarized in Table IX, and the corresponding feature-importance ranking is illustrated in Fig. 11.

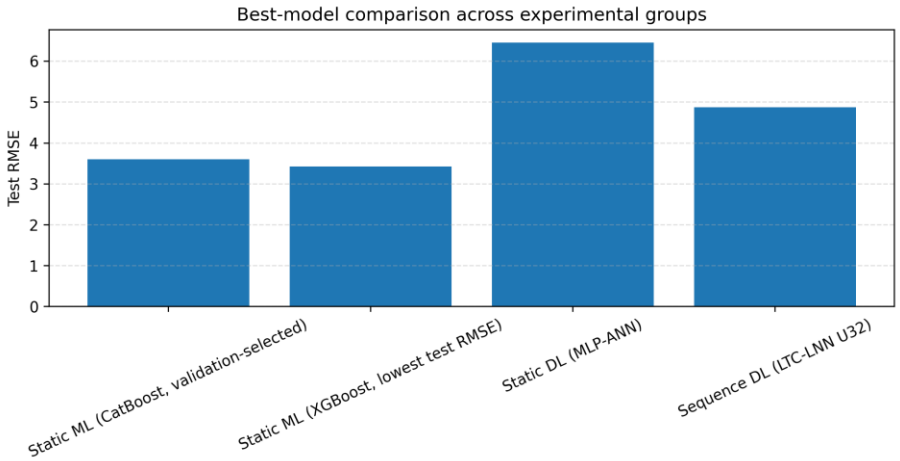

Fig. 10. Test RMSE comparison across best modelling groups.

TABLE IX. TOP 10 PERMUTATION FEATURE IMPORTANCE RESULTS FOR THE VALIDATION-SELECTED STATIC ML MODEL

Feature	Importance Mean	Importance SD
2FAL	0.2620	0.0111
C2H2	0.1681	0.0096
Dielectric Breakdown	0.0961	0.0040
C2H6	0.0676	0.0043
C2H4	0.0597	0.0038
Colour	0.0464	0.0037
Moisture	0.0378	0.0029
CO2	0.0355	0.0022
CH4	0.0305	0.0023
IFT	0.0246	0.0019

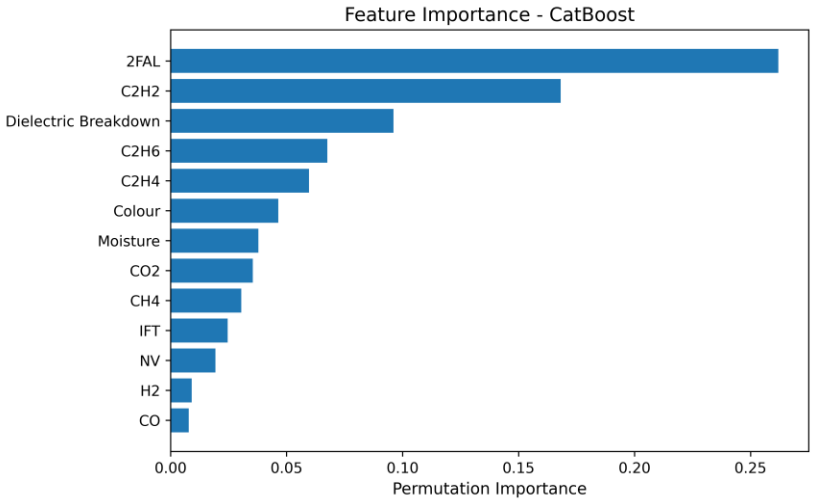

Fig. 11. Permutation feature importance for the validation-selected static ML model.

TABLE X. STATISTICAL UNCERTAINTY OF THE LEADING STATIC ML MODELS

Model	Test RMSE	95% CI
XGBoost	3.4254	3.1391-3.7030
CatBoost	3.6017	3.1828-4.0522

The paired RMSE difference (XGBoost - CatBoost) was -0.1763 (95% CI: -0.5723 to 0.1423; p = 0.3816).

The computational benchmark showed substantial differences among the representative models. XGBoost required 2.8939 s for training and 0.0352 ms per static record for inference, whereas CatBoost required 4.6119 s for training but

achieved the lowest inference latency at 0.0104 ms per record. MLP-ANN required 36.7382 s for training and 0.0126 ms per record for inference, with 3,521 trainable parameters and an approximate dense-layer cost of 6,816 FLOPs per input. In contrast, LTC-LNN U32 required 3625.3234 s for training and 1.6859 ms per variable-length sequence for inference, reflecting the higher computational cost of continuous-time sequence modelling with repeated ODE unfolding. Peak additional training RAM was 1.9297 MB for XGBoost, 3.1367 MB for CatBoost, 28.6406 MB for MLP-ANN, and 94.9414 MB for LTC-LNN U32. The corresponding serialized model-plus-preprocessing sizes were 0.9734, 0.6465, 0.0733, and 0.0423 MB, respectively. Thus, the static models provided substantially lower runtime requirements, while LTC-LNN incurred greater computational cost to model variable-length temporal histories. The computational-efficiency results are summarized in Table XI.

TABLE XI. COMPUTATIONAL EFFICIENCY OF REPRESENTATIVE MODELS

Model	Train time (s)	Inference (ms/input)	Peak add. train RAM (MB)	Model + prep. size (MB)	Complexity
XGBoost	2.8939	0.0352	1.9297	0.9734	600 trees; max depth 4
CatBoost	4.6119	0.0104	3.1367	0.6465	600 trees; depth 6
MLP-ANN	36.7382	0.0126	28.6406	0.0733	3,521 params; ~6,816 FLOPs/input
LTC-LNN U32	3625.3234	1.6859	94.9414	0.0423	7,165 params; 6 ODE unfolds; FLOPs sequence-dependent

Note: For XGBoost, CatBoost, and MLP-ANN, one input denotes one static diagnostic record; for LTC-LNN U32, one input denotes one variable-length diagnostic sequence. Peak additional inference RAM was 0.0039 MB for XGBoost and CatBoost and 0.0078 MB for MLP-ANN and LTC-LNN U32. Serialized storage values are framework-specific and should be interpreted as approximate deployment footprints. The MLP FLOP estimate covers dense-layer multiply-add operations; a single fixed FLOP value is not reported for the tree ensembles or variable-length LTC-LNN.

The residual analysis provided additional information beyond MAE, RMSE, and R^2. XGBoost showed the smallest overall systematic bias, with a mean residual of -0.1040, together with the lowest residual standard deviation (3.4269) and residual variance (11.7439) among the representative models. Its 95th-percentile absolute error was 7.3013, and its maximum absolute error was 11.6427. CatBoost also exhibited low average bias (-0.2885) and a residual standard deviation of 3.5933, although its maximum absolute error increased to 19.1718. MLP-ANN showed substantially greater error dispersion, with a residual standard deviation of 6.4523, residual variance of 41.6316, and maximum absolute error of 33.3333. LTC-LNN U32 produced an intermediate error spread, with a residual standard deviation of 4.8554 and maximum absolute error of 15.7975 on the repeated-history test subset. The leading gradient-boosted models had lower average errors, lower residual dispersion, and smaller extreme errors than the representative static neural model. The residual robustness metrics are summarized in Table XII.

TABLE XII. RESIDUAL ROBUSTNESS ANALYSIS OF REPRESENTATIVE MODELS

Model	Test n	Mean residual (bias)	Residual SD	Residual variance	95th pct. abs. error	Maximum abs. error
XGBoost	551	-0.1040	3.4269	11.7439	7.3013	11.6427
CatBoost	551	-0.2885	3.5933	12.9122	7.1036	19.1718
MLP-ANN	551	0.1455	6.4523	41.6316	11.8886	33.3333
LTC-LNN U32	429	-0.4344	4.8554	23.5745	9.7010	15.7975

Note: Residual = actual HI - predicted HI. A negative mean residual denotes a tendency to overestimate HI, whereas a positive mean residual denotes a tendency to underestimate HI. The sequence model was evaluated on the repeated-history test subset and therefore has a different test-sample count from the static models.

TABLE XIII. MULTICOLLINEARITY DIAGNOSTICS FOR THE 13 PREDICTOR VARIABLES

Predictor	VIF
C2H2	9.350
H2	9.049
NV	2.414
C2H4	2.301
Colour	2.176
CH4	1.958
IFT	1.823
CO2	1.635
Moisture	1.535
CO	1.516
Dielectric Breakdown	1.439
C2H6	1.396
2FAL	1.237

The correlation analysis identified several related diagnostic variables. The strongest Pearson correlation occurred between H2 and C2H2 (r = 0.9402). Other notable relationships included NV and Colour (r = 0.6711), CH4 and C2H4 (r = 0.6466), NV and IFT (r = -0.6072), IFT and Colour (r = -0.5414), CO and CO2 (r = 0.5150), and dielectric breakdown and moisture (r = -0.5048). The VIF analysis further showed elevated values for C2H2 (9.350) and H2 (9.049), whereas the remaining 11 predictors had VIF values between 1.237 and 2.414. These findings indicate that multicollinearity is concentrated mainly in the H2-C2H2 relationship rather than being widespread across the complete diagnostic feature set. No predictor exceeded a VIF of 10. The VIF results are summarized in Table XIII, while the corresponding Pearson correlation heatmap is shown in Fig. 12.

Prediction-confidence analysis was performed for the validation-selected CatBoost model using residual quantiles obtained from the validation set. The nominal 90% interval used a calibrated half-width of 6.2764 HI points and achieved an empirical test coverage of 92.92%, with a mean interval width of

12.2081 HI points. The nominal 95% interval used a half-width of 7.4494 HI points and achieved 95.83% empirical coverage, with a mean width of 14.4744 HI points. The observed test-set coverage was close to the nominal levels, showing that the residual-based intervals reasonably represented prediction uncertainty for the selected static model. However, condition-specific inspection showed that the 95% interval covered 76.19% of the 21 Critical-class test records, compared with 93.44-99.14% coverage across the other condition classes. This lower coverage in the small Critical-class subset indicates that prediction confidence is less uniform in the most severe condition region and should be considered when using model outputs for maintenance decisions. The prediction-interval calibration results are summarized in Table XIV.

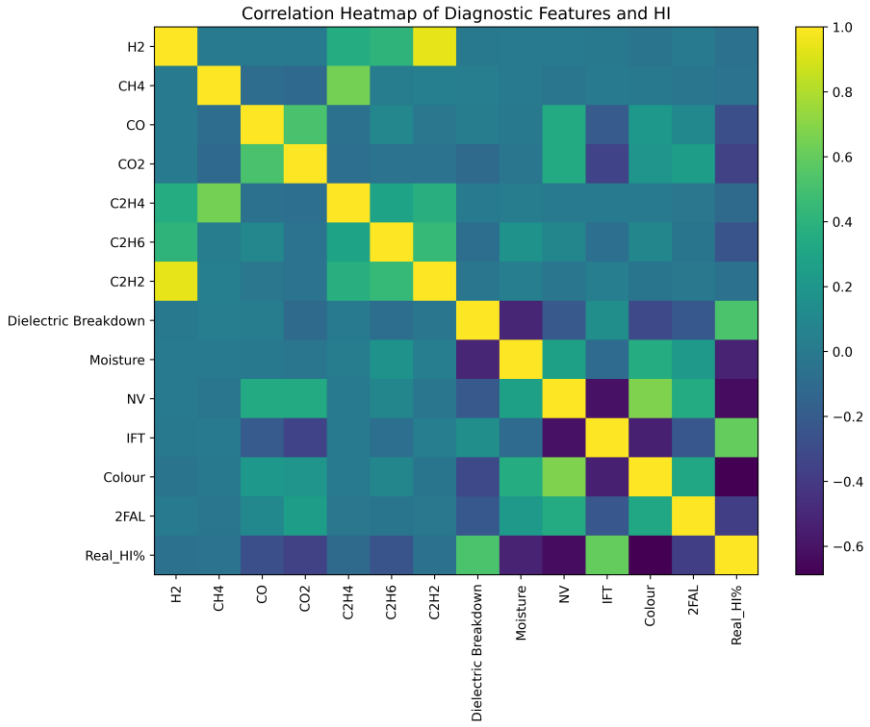

Fig. 12. Pearson correlation heatmap of the diagnostic features and transformer health index.

TABLE XIV. PREDICTION-INTERVAL CALIBRATION OF THE VALIDATION-SELECTED CATBOOST MODEL

Nominal interval	Calibrated half-width (HI points)	Empirical test coverage	Mean interval width (HI points)
90%	±6.2764	92.92%	12.2081
95%	±7.4494	95.83%	14.4744

VI. DISCUSSION

The main result of this study is that static gradient-boosting models provided the highest observed predictive performance for the available THI dataset. This does not mean that neural models are unsuitable for transformer health assessment; rather, it indicates that the target and input structure are mainly tabular and threshold-driven. The Real_HI% target is derived from conventional diagnostic scoring, and boosting models are well matched to this type of nonlinear but structured mapping. These findings identify gradient-boosted tree models as strong candidates for static THI prediction on the present dataset; however, broader deployment suitability should also consider statistical uncertainty, prediction confidence, computational requirements, and external validation.

Although XGBoost produced the lowest point-estimate test RMSE, the bootstrap analysis showed that its performance was not statistically distinguishable from that of CatBoost. This finding indicates that small differences among the leading gradient-boosted models should not be interpreted as definitive evidence of model superiority. The results instead support the broader conclusion that gradient-boosted tree models constitute a strong-performing model family for the present THI dataset, while the relative ranking of XGBoost and CatBoost remains subject to sampling uncertainty.

The negative R^2 values obtained by the Regularized MLP and CNN-MLP Hybrid indicate that these models failed to generalize effectively on the held-out test set. Their additional architectural complexity and regularization did not provide an advantage for the relatively low-dimensional tabular THI inputs and likely contributed to poorer generalization compared with the simpler MLP-ANN architecture. Consequently, both models performed worse than the mean-target baseline and were retained only for benchmarking completeness rather than final model selection. Although these results indicate architecture-specific generalization limitations, a dedicated ablation analysis would be required to isolate the individual contribution of each architectural component.

The feature-importance results are physically meaningful. 2FAL was the strongest predictor, reflecting the importance of paper-insulation ageing in the health-index target and the diagnostic role of furan compounds. C2H2 was also highly influential, which is consistent with the role of acetylene in severe electrical fault interpretation using gas-in-oil analysis. Dielectric breakdown ranked third, showing that insulation-oil dielectric strength remains a major driver of the overall THI. The appearance of moisture, colour, interfacial tension (IFT), and DGA gases among the influential predictors confirms that the model did not rely on a single diagnostic channel.

The correlation and VIF analyses show that the predictor set contains localized dependence, particularly between H2 and C2H2. This dependence does not invalidate the predictive comparison because the leading models are nonlinear tree-based ensembles rather than coefficient-based linear regressors. However, correlated predictors can share or redistribute model-based importance, and the importance assigned to an individual variable should therefore not be interpreted as an independent causal contribution. The complete diagnostic set was retained because the correlated variables represent distinct physical fault indicators and because most predictors exhibited low VIF values. Future studies could additionally evaluate grouped feature importance, dimensionality reduction, or correlation-aware feature selection when larger multi-utility datasets become available.

The variable-length liquid time-constant neural network (LTC-LNN) result is important because it demonstrates a temporal-learning route for repeated oil-test histories. Unlike fixed-window modelling, the variable-length design allows each transformer to use its available historical depth. For transformers with sparse histories, this design avoids discarding samples or over-padding long sequences. The strong validation and test performance of LTC-LNN suggests that continuous-time recurrent modelling is promising for irregularly sampled asset-condition data.

The computational results also highlight differences in the practical requirements of the models. XGBoost and CatBoost required only a few seconds of CPU training time and sub-0.04-ms inference per static record, indicating a favorable computational profile for single-record THI estimation. MLP-ANN also provided low inference latency and a compact parameterization, although its predictive accuracy was lower than that of the boosted tree models. LTC-LNN U32 had a small serialized model footprint but required substantially longer training time, higher peak training memory, and higher per-sequence inference latency because it processes variable-length histories through continuous-time recurrent dynamics and repeated ODE unfolding. Considering both predictive accuracy and computational cost, the boosted tree models therefore provide the strongest observed accuracy-efficiency trade-off for static THI prediction in the present CPU benchmark, whereas LTC-LNN remains a temporal alternative when repeated histories are available, and the additional computational cost is acceptable. Absolute timing values remain hardware- and software-dependent and should therefore be re-evaluated on the intended deployment platform.

Model performance also varied across the HI condition classes. The Critical class was particularly challenging for the static models: the mean residual was -2.7919 for XGBoost, -4.8594 for CatBoost, and -7.5747 for MLP-ANN, indicating a tendency to predict a healthier condition than the observed HI in this class. The corresponding Critical-class MAEs were 3.7153, 5.6887, and 8.9841, respectively. LTC-LNN U32 showed a Critical-class mean residual of -2.7867 and an MAE of 3.7690. This class-dependent behavior is important for maintenance applications because optimistic errors in severely degraded transformers may be more consequential than errors in healthier classes. Therefore, aggregate metrics should be interpreted together with residual and condition-specific analyses, particularly when considering safety-critical deployment.

The prediction intervals provide additional information about the uncertainty of the point predictions. For the validation-selected CatBoost model, the empirical coverage of the 90% and 95% intervals was 92.92% and 95.83%, respectively, suggesting that the calibrated residual bands broadly reflected the observed test uncertainty. Nevertheless, interval coverage was not uniform across health-condition classes. In particular, the Critical class showed lower 95% coverage, although this estimate was based on only 21 test records. This finding is important for safety-critical use because the most severe transformer conditions are precisely those for which overconfident predictions should be avoided. Accordingly, the reported prediction intervals should be treated as decision-support information rather than as a substitute for diagnostic testing, expert assessment, or utility-specific maintenance rules. Future work should evaluate dedicated calibration sets, class-conditional or transformer-level conformal methods, and external calibration on independent utility datasets.

A limitation of the present sequence experiment is that repeated histories were still relatively short for many transformers, and only transformers with at least two records could be used. Consequently, the sequence subset is smaller than the full static dataset. Future work should evaluate the framework on longer longitudinal histories, include load and maintenance information, and conduct external validation across different utilities or voltage classes.

Another methodological issue is that MAPE was not used as a primary discussion metric because some HI values were close to or equal to zero, making percentage errors unstable. Therefore, RMSE, MAE, R^2, and HI-class accuracy were emphasized as the main evaluation metrics. This is more appropriate for the present target distribution. A further limitation is that all experiments were conducted using a proprietary dataset from a single Malaysian utility. Consequently, the reported performance may reflect utility-specific transformer populations, operating practices, diagnostic procedures, and voltage classes. External validation using independent datasets from other utilities and transformer populations is therefore required before the findings can be generalized more broadly.

The uncertainty analysis also shows that the numerical ranking of the models should be interpreted with caution. Although XGBoost achieved the lowest point-estimate test RMSE, its paired RMSE difference relative to the validation-selected CatBoost model was -0.1763, with a 95% bootstrap confidence interval of -0.5723 to 0.1423 and a two-sided bootstrap p-value of 0.3816. Because the confidence interval included zero and $p > 0.05$, the observed difference was not statistically

distinguishable at the 95% confidence level. In addition, prediction-interval analysis for CatBoost produced overall empirical coverages of 92.92% and 95.83% for the nominal 90% and 95% intervals, respectively, while coverage for the Critical condition subgroup was lower. These findings indicate that aggregate accuracy alone is insufficient to justify a definitive deployment ranking and that uncertainty, condition-specific reliability, computational cost, and external validation should be considered together when selecting models for operational use.

VII. Conclusion

This study presented a comprehensive benchmarking framework for transformer health index prediction using dissolved gas analysis, oil-quality, and furan diagnostic records. Three experimental groups were evaluated: static machine learning, static deep learning, and variable-length sequence deep learning. For the present single-record tabular dataset, gradient-boosted tree models achieved the strongest observed predictive performance among the evaluated model families. XGBoost obtained the lowest point-estimate test RMSE of 3.4254, while CatBoost was selected based on validation RMSE. However, the paired transformer-level bootstrap analysis showed that the XGBoost-CatBoost RMSE difference was not statistically significant (difference = -0.1763, 95% CI: -0.5723 to 0.1423; p = 0.3816). Therefore, the small numerical difference between these models should not be interpreted as definitive evidence that one is universally superior to the other.

Among the static deep learning models, MLP-ANN achieved the strongest performance, whereas LTC-LNN U32 provided the best result among the variable-length sequence models and demonstrated the potential of continuous-time learning for irregularly sampled repeated transformer histories. Prediction-confidence analysis of the validation-selected CatBoost model further showed empirical test coverages of 92.92% and 95.83% for nominal 90% and 95% prediction intervals, respectively. Nevertheless, lower coverage in the Critical condition subgroup highlights the need for caution when model outputs are used in high-consequence maintenance decisions.

Overall, the results support gradient-boosted tree models as strong candidates for static THI prediction within the evaluated dataset rather than establishing a universally preferred deployment model. Model selection for operational use should consider predictive accuracy together with statistical uncertainty, prediction confidence, computational requirements, and the consequences of errors in critical condition classes. External validation using independent utilities, transformer populations, and voltage classes, together with richer longitudinal histories and additional operating and maintenance information, is required before broader deployment recommendations can be made.

Acknowledgment

The authors gratefully acknowledge the utility company in Malaysia for providing the diagnostic dataset and Universiti Teknikal Malaysia Melaka (UTeM) for its support. This research was funded by the Ministry of Higher Education (MOHE), Malaysia, through the Fundamental Research Grant Scheme (FRGS), FRGS/1/2024/TK07/UTEM/02/24 (MOHE grant code) and FRGS/1/2024/FTKE/F00579 (UTeM grant code).

Declarations

A. Funding

This work was supported by the Ministry of Higher Education (MOHE), Malaysia, under the Fundamental Research Grant Scheme (FRGS), with Grant No. FRGS/1/2024/TK07/UTEM/02/24.

B. Declaration on Generative AI and AI-Assisted Technologies

During manuscript preparation, AI-assisted tools were used only for language refinement and formatting support. The authors reviewed and approved the final content and remain responsible for the scientific accuracy of the work.

C. Author Contributions

Omar Ahmed Mohammed Hamood Al-Shaikh: Conceptualization, methodology, software, validation, formal analysis, data curation, writing—original draft, and visualization. Arfah Ahmad: Supervision, conceptualization, methodology, project administration, writing—review and editing. Ahmad Jazlan Haja Mohideen: Supervision, methodology, validation, writing—review and editing. Muhammad Sharil Yahaya: Supervision, validation, writing—review and editing.

D. Conflict of Interests

The authors declare no conflict of interest.

E. Data Availability

Data are available upon reasonable request and subject to permission from the data-owning utility company.

References

[1] I. Standard, "IEEE guide for the interpretation of gases generated in mineral oil‐immersed transformers," ed, 2019, pp. 1‐98.

[2] I. Commission, "Mineral oil-filled electrical equipment in service–Guidance on the interpretation of dissolved and free gases analysis," IEC, vol. 60599, p. 2015, 2015.

[3] I. C. 106-, "IEEE guide for acceptance and maintenance of insulating mineral oil in electrical equipment," ed: Institution of Electrical and Electronics Engineers Piscataway, 2015.

[4] B. EN, "60422: 2013 Mineral insulating oils in electrical equipment. Supervision and maintenance guidance," International Electrotechnical Commission, Geneva, Switzerland, 2013.

[5] M. I. Oils, "Methods for the Determination of 2-Furfural and Related Compounds," Standard IEC, vol. 61198, 1993.

[6] M. Duval, "A review of faults detectable by gas-in-oil analysis in transformers," IEEE electrical Insulation magazine, vol. 18, no. 3, pp. 8–17, 2002.

[7] M. Duval and J. Dukarm, "Improving the reliability of transformer gas-in-oil diagnosis," IEEE Electrical Insulation Magazine, vol. 21, no. 4, pp. 21–27, 2005.

[8] A. Naderian, S. Cress, R. Piercy, F. Wang, and J. Service, "An approach to determine the health index of power transformers," in Conference Record of the 2008 IEEE International Symposium on Electrical Insulation, 2008: IEEE, pp. 192–196.

[9] A. Jahromi, R. Piercy, S. Cress, J. Service, and W. Fan, "An approach to power transformer asset management using health index," IEEE Electrical Insulation Magazine, vol. 25, no. 2, pp. 20–34, 2009.

[10] A. E. Abu-Elanien, M. Salama, and M. Ibrahim, "Calculation of a health index for oil-immersed transformers rated under 69 kV using fuzzy logic," IEEE transactions on power delivery, vol. 27, no. 4, pp. 2029–2036, 2012.

[11] J. Haema and R. Phadungthin, "Condition assessment of the health index for power transformer," in 2012 power engineering and automation conference, 2012: IEEE, pp. 1–4.

[12] I. N. S. Hernanda, A. Mulyana, D. Asfani, I. Y. Negara, and D. Fahmi, "Application of health index method for transformer condition assessment," in Tencon 2014-2014 Ieee Region 10 Conference, 2014: IEEE, pp. 1–6.

[13] A. D. Ashkezari, H. Ma, T. K. Saha, and C. Ekanayake, "Application of fuzzy support vector machine for determining the health index of the insulation system of in-service power transformers," IEEE Transactions on Dielectrics and Electrical Insulation, vol. 20, no. 3, pp. 965–973, 2013.

[14] W. H. Tang and Q. H. Wu, Condition monitoring and assessment of power transformers using computational intelligence. Springer Science & Business Media, 2011.

[15] S. Li, X. Li, Y. Cui, and H. Li, "Review of transformer health index from the perspective of survivability and condition assessment," Electronics, vol. 12, no. 11, p. 2407, 2023.

[16] L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5–32, 2001.

[17] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," Annals of statistics, pp. 1189–1232, 2001.

[18] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of computer and system sciences, vol. 55, no. 1, pp. 119–139, 1997.

[19] V. Vapnik, "Support-vector networks," Machine learning, vol. 20, pp. 273–297, 1995.

[20] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[21] G. Ke et al., "Lightgbm: A highly efficient gradient boosting decision tree," Advances in neural information processing systems, vol. 30, 2017.

[22] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," Advances in neural information processing systems, vol. 31, 2018.

[23] F. Pedregosa et al., "Scikit-learn: Machine learning in Python: the Journal of machine Learning research, v. 12," 2011.

[24] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, Deep learning (no. 2). MIT press Cambridge, 2016.

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, pp. 436–444, 2015.

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," nature, vol. 323, no. 6088, pp. 533–536, 1986.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1724–1734.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008, 2017.

[30] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," arXiv preprint arXiv:1803.01271, 2018.

[31] R. Hasani, M. Lechner, A. Amini, D. Rus, and R. Grosu, "Liquid time-constant networks," in Proceedings of the AAAI conference on artificial intelligence, 2021, vol. 35, no. 9, pp. 7657–7666.

[32] R. M. Hasani, M. Lechner, A. Amini, D. Rus, and R. Grosu, "Liquid time-constant recurrent neural networks as universal approximators," arXiv preprint arXiv:1811.00321, 2018.

[33] M. Lechner, R. Hasani, A. Amini, T. A. Henzinger, D. Rus, and R. Grosu, "Neural circuit policies enabling auditable autonomy," Nature Machine Intelligence, vol. 2, no. 10, pp. 642–652, 2020.

[34] R. Hasani et al., "Closed-form continuous-time neural networks," Nature Machine Intelligence, vol. 4, no. 11, pp. 992–1003, 2022.

[35] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," Advances in neural information processing systems, vol. 31, 2018.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.

[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International conference on machine learning, 2015: pmlr, pp. 448–456.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[40] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," Advances in neural information processing systems, vol. 32, 2019.