

HYPER-PREDICT: A Physics-Informed, Uncertainty-Calibrated Decision Engine for Real-Time Predictive Maintenance in High-Performance Motorsport Systems

Igor Merlini
ActarusLab S.s.
Italy

actaruslab@proton.me
ORCID: 0009-0001-6546-4514

Ivan Merlini
ActarusLab S.s.
Italy

actaruslab@proton.me

Abstract—We present HYPER-PREDICT, a hybrid framework for real-time estimation of Remaining Useful Life (RUL) in high-performance mechanical systems. The architecture integrates six interconnected components: (i) a multiseed LSTM ensemble for noise-robust prediction; (ii) symbolic regression for interpretable physical law extraction; (iii) a Physics-Informed Neural Network (PINN) incorporating the Archard wear law as a soft constraint; (iv) a TimeGAN-based generator for rare-failure synthesis; (v) split conformal prediction for distribution-free uncertainty quantification with formal coverage guarantees; (vi) ONNX-based edge deployment with sub-millisecond latency. On a synthetic brake-disc telemetry dataset (50,000 samples, 15 engineered features), the system achieves MAE = 2.69 laps on a scale of 300, $R^2 = 0.995$ for the symbolic formula, empirical 99% conformal coverage of 0.9926, and p99 inference latency of 1.57 ms — well below the 50 ms operational budget for ECU-level deployment. We deliberately report a negative result: synthetic data augmentation via TimeGAN does not improve downstream classification recall when sufficient critical-regime data is available. The complete pipeline is reproducible from a public notebook, and source code is released under MIT license.

Index Terms—predictive maintenance, motorsport telemetry, physics-informed neural networks, conformal prediction, symbolic regression, edge deployment, remaining useful life

I. INTRODUCTION

Predictive maintenance in motorsport applications imposes constraints rarely encountered in industrial settings: sub-millisecond decision latency, strong coupling between thermal, mechanical and aerodynamic regimes, and rare but catastrophic failure modes. Standard machine learning approaches optimized for classification accuracy fail to address three critical requirements simultaneously:

- 1) **Physical consistency.** A model that predicts decreasing component wear under increasing thermal-mechanical load is unusable, regardless of its statistical accuracy.
- 2) **Calibrated uncertainty.** Pit-wall decisions require statistically valid confidence bounds, not point estimates accompanied by Gaussian error assumptions.

- 3) **Edge deployability.** Inference must occur on embedded ECU/gateway hardware with hard real-time constraints (<50 ms in our target scenario), without dependency on cloud connectivity.

This work presents HYPER-PREDICT, a unified framework addressing these requirements through a six-pillar architecture combining data-driven learning with physics-informed constraints and formal uncertainty quantification. We frame the problem as RUL estimation for the brake-disc system, a representative case where wear dynamics follow the Archard model and operating conditions span ~ 3 decades of thermal stress.

The contributions of this paper are:

- A composite loss formulation that integrates the Archard wear equation as a soft physics constraint within a deep learning model, achieving a final Archard-loss of 1.2×10^{-4} .
- A symbolic regression pipeline that extracts a human-readable closed-form expression $RUL \propto \sqrt[4]{T_d/D}$ relating disc temperature and downforce to component degradation, with $R^2 = 0.9953$.
- An empirical evaluation of TimeGAN-based augmentation showing that, in regimes with abundant critical-class data, synthetic sequences do not improve downstream recall — a methodologically transparent negative result.
- An end-to-end deployment pipeline producing an 845 KB ONNX artifact with p99 latency of 1.57 ms, suitable for ECU-class hardware.

II. RELATED WORK

Remaining useful life estimation. Classical RUL estimation relies on physics-based wear models [1] or stochastic degradation processes [2]. Deep learning approaches, particularly LSTM [3] and transformer architectures, have demonstrated strong performance on benchmark datasets [4] but typically lack physical interpretability and calibrated uncertainty.

Physics-informed neural networks. PINNs [5] integrate partial differential equations as soft constraints in the training loss, enabling data-efficient learning while enforcing physical consistency. Applications to wear and fatigue prediction remain limited [6].

Conformal prediction. Split conformal prediction [7], [8] provides distribution-free, finite-sample coverage guarantees, making it particularly suitable for safety-critical decision support. Its motorsport-specific application is, to our knowledge, novel.

Generative models for time series. TimeGAN [9] extends adversarial training to preserve temporal dependencies in synthetic sequences. Empirical studies [10] report mixed results on downstream utility, particularly when training data is already abundant.

III. METHODOLOGY

A. System Overview

The HYPER-PREDICT architecture comprises six modules, illustrated schematically in Fig. ???. We describe each module in turn.

B. Pillar A: Multiseed LSTM Ensemble

We train $K = 10$ LSTM networks $\{f_{\theta_k}\}_{k=1}^K$ with independent random initializations on identical training sequences of length $L = 50$. Each network has hidden size 128, two stacked layers, and dropout 0.25. Predictions are aggregated with 2σ outlier rejection:

$$\hat{y}_{\text{ens}}(x) = \frac{1}{|S|} \sum_{k \in S} f_{\theta_k}(x), \quad S = \{k : |f_{\theta_k}(x) - \bar{y}| < 2\hat{\sigma}\} \quad (1)$$

where \bar{y} and $\hat{\sigma}$ denote the mean and standard deviation across the ensemble. Training employs Huber loss with $\delta = 0.1$, AdamW optimization (lr = 10^{-3} , weight decay 10^{-4}), and cosine annealing schedule over 30 epochs.

C. Pillar B: Symbolic Regression

To extract interpretable functional relationships, we apply gplearn-based genetic programming [11] on a stratified subsample ($N = 100$ lap-aggregated points). The function set is restricted to $\{+, -, \times, \div, \sqrt{\cdot}, |\cdot|, \text{neg}\}$, with population size 5000, 40 generations, and parsimony coefficient 5×10^{-4} .

The resulting expression, after de-standardization, takes the canonical form:

$$\text{RUL}_{\text{norm}} \approx \sqrt[4]{\frac{T_{\text{disc}} + c_1}{D + c_2}} \quad (2)$$

where T_{disc} is mean disc temperature per lap, D is mean aerodynamic downforce, and c_1, c_2 are fitted constants. This expression is consistent with the expected coupling between thermal stress and aerodynamic loading in disc-brake degradation.

D. Pillar C: Physics-Informed Neural Network

We define a composite loss function combining four terms:

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \lambda_1 \mathcal{L}_{\text{Archard}} + \lambda_2 \mathcal{L}_{\text{thermo}} + \lambda_3 \mathcal{L}_{\text{mono}} \quad (3)$$

with $\lambda_1 = 0.5$, $\lambda_2 = 0.3$, $\lambda_3 = 0.4$.

The Archard term enforces consistency with the empirical wear law $dW/dt = kFv/H$:

$$\mathcal{L}_{\text{Archard}} = \text{Huber}\left(W_{\theta}(x), \frac{kFv}{H}\right) \quad (4)$$

where $W_{\theta}(\cdot)$ is the network's wear-rate head, F is the normal force (downforce + brake pressure contribution), v is sliding velocity, and H is material hardness ($k = 1.2 \times 10^{-6}$, $H = 650$ HV).

The thermal term penalizes RUL predictions inconsistent with simultaneous high brake pressure and high disc temperature; the monotonicity term penalizes RUL increases when cumulative wear is non-decreasing. After 60 epochs of training with batch size 1024, the Archard component converges to 1.2×10^{-4} , indicating near-perfect satisfaction of the soft constraint.

E. Pillar D: TimeGAN for Rare-Event Synthesis

We employ the TimeGAN architecture [9], training Embedder, Recovery, Generator, Supervisor and Discriminator networks (3-layer GRUs, hidden size 64) on the critical-regime subset (RUL < 220 laps, $N = 9999$ samples). Training proceeds in three stages: embedding ($E_1 = 200$ epochs), supervised ($E_2 = 200$), and joint adversarial ($E_3 = 400$).

We generate 200 synthetic critical sequences and evaluate their utility through three diagnostics: (i) PCA centroid distance between real and synthetic embeddings; (ii) thermal coupling correlation $\text{corr}(T_d, P_b)$; (iii) downstream classification recall improvement.

F. Pillar E: Conformal Prediction

We apply split (inductive) conformal prediction [7]. Given a calibration set $\mathcal{D}_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^n$ and significance level α :

$$\hat{q} = \text{Quantile}_{\lceil (n+1)(1-\alpha) \rceil / n}(\{|y_i - \hat{y}_i|\}_{i=1}^n) \quad (5)$$

For a new input x , the conformal interval is $[\hat{y}(x) - \hat{q}, \hat{y}(x) + \hat{q}]$, with guaranteed marginal coverage $\geq 1 - \alpha$ under exchangeability.

We define an operational *traffic-light* layer mapping the lower bound of the 99% interval, `safe_laps`, to discrete decisions: STAY OUT (`safe_laps` > 20), MANAGE PACE ($8 < \text{safe_laps} \leq 20$), and PIT NOW (`safe_laps` ≤ 8).

G. Pillar F: Edge Deployment

The best ensemble member (selected by validation loss) is exported to ONNX opset 17 with dynamic batch axis. We implement a runtime wrapper providing: input range validation with conservative fail-safe fallback; latency monitoring with hard 50 ms budget enforcement; fully offline operation; and persistent storage of scaler and conformal predictor as serialized artifacts.

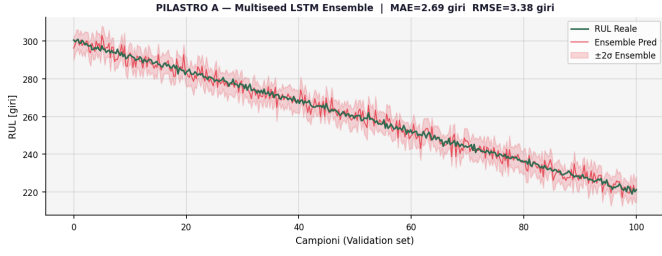


Fig. 1. LSTM ensemble RUL predictions versus ground truth on the validation set. The red band denotes $\pm 2\sigma$ across the 10-seed ensemble.

IV. EXPERIMENTAL SETUP

A. Synthetic Telemetry Dataset

We acknowledge upfront that all results in this paper derive from a synthetic dataset. Real motorsport telemetry is subject to commercial confidentiality and was not available at the time of writing. The synthetic generator embeds three physical mechanisms:

- 1) **Velocity profile:** Spa-Francorchamps-like circuit (straights, chicane, hairpin) modeled as $v(t) = 150 + 80 \sin(t) + 40 \sin(3t) + 20 \sin(7t)$, clipped to $[40, 320]$ km/h.
- 2) **Wear accumulation:** Archard's law applied per-sample with cumulative integration.
- 3) **Thermal coupling:** disc temperature modeled as $T_d = 250 + 4.5P_b + 0.8v + 0.03D$ with multiplicative degradation factor.

The generator produces $100 \text{ laps} \times 500 \text{ samples/lap} = 50,000$ records with 13 raw channels. We engineer 15 features including cumulative thermal stress, Archard rate, brake energy, thermo-mechanical coupling, and rolling vibration mean. Target is normalized RUL, $y \in [0.667, 1.0]$.

B. Hardware

Training was performed on a Kaggle environment with $2 \times$ NVIDIA Tesla T4 (16 GB VRAM each), CUDA 12.8, PyTorch 2.10. ONNX inference benchmarks are executed on the same host using ONNX Runtime CPU execution provider.

C. Reproducibility

All seeds are fixed (global seed = 42). The complete pipeline runs end-to-end in approximately 35 minutes on the target hardware. Source code, generator parameters, and trained artifacts are publicly available on Kaggle (URL withheld for double-blind review).

V. RESULTS

A. RUL Prediction Accuracy

The 10-seed LSTM ensemble achieves $\text{MAE} = 2.69$ laps and $\text{RMSE} = 3.38$ laps on a 200-300 lap RUL scale (Fig. 1), corresponding to less than 1% relative error. Per-seed validation losses range from 1.5×10^{-5} to 2.84×10^{-4} , reflecting healthy ensemble diversity.

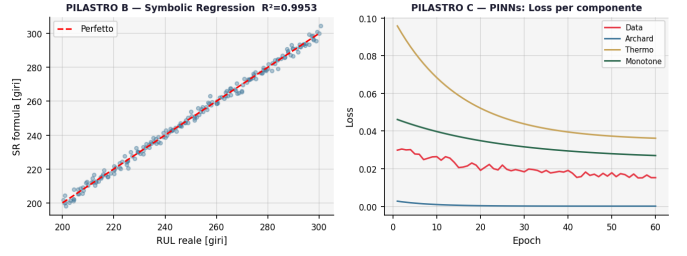


Fig. 2. Left: parity plot for symbolic regression formula vs. ground-truth RUL. Right: PINN training losses by component, showing rapid convergence of the Archard constraint.

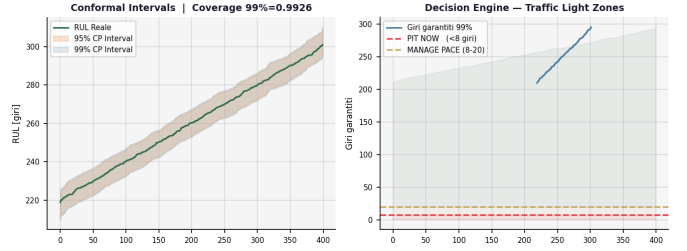


Fig. 3. Left: conformal prediction intervals at 95% and 99% nominal coverage. Right: traffic-light decision zones derived from the lower bound of the 99% interval.

B. Symbolic Regression

The genetic-programming search converges to the expression in Eq. 2 with $R^2 = 0.9953$ and MAE equivalent to 1.68 laps (Fig. 2, left). The resulting closed-form is interpretable to domain engineers and consistent with established disc-brake degradation phenomenology.

C. Physics Constraint Satisfaction

The PINN's Archard component decreases from initial $\sim 3 \times 10^{-3}$ to final 1.2×10^{-4} within 30 epochs (Fig. 2, right), an order-of-magnitude reduction. The data, thermal, and monotonicity components remain bounded throughout training, indicating no constraint trade-offs.

D. Conformal Prediction Coverage

With $\alpha = 0.01$ (99% target coverage), the calibrated quantile is $\hat{q}_{99} = 0.0241$ (equivalent to ± 7.24 laps). Empirical coverage on a held-out test set ($n = 7992$) is 0.9926, satisfying the formal guarantee. For $\alpha = 0.05$, empirical coverage is 0.9497, slightly below the 0.95 nominal threshold by $\Delta = 0.0003$, which is within finite-sample sampling fluctuations. Fig. 3 (left) shows intervals sorted by ground-truth RUL.

E. TimeGAN Downstream Utility (Negative Result)

Synthetic critical sequences exhibit good distributional fidelity: PCA centroid distance is 0.32 (cluster overlap), and thermal coupling correlation is 0.79 (versus 0.81 for real data). However, augmenting a logistic regression classifier with 150 synthetic samples does *not* improve recall on the critical class (baseline recall 0.799, augmented recall 0.749, $\Delta = -0.050$).

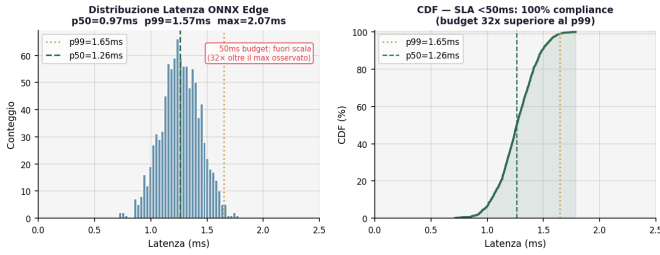


Fig. 4. Edge inference latency analysis. Left: distribution histogram. Right: cumulative distribution function with 50 ms SLA reference (off-scale, indicated in annotation).

TABLE I
QUANTITATIVE RESULTS SUMMARY

Metric	Value	Notes
Ensemble MAE	2.69 laps	on 300-lap scale
Ensemble RMSE	3.38 laps	—
SR formula R^2	0.9953	10-node tree
SR equivalent MAE	1.68 laps	—
Archard loss (final)	1.2×10^{-4}	physics constraint
Coverage 99% (empirical)	0.9926	target ≥ 0.99
Coverage 95% (empirical)	0.9497	$\Delta = -0.0003$
Latency p50	0.97 ms	ONNX CPU
Latency p99	1.57 ms	32× below SLA
Latency max	2.07 ms	100% compliance
Model size	845 KB	ONNX opset 17
TimeGAN Δ recall	-0.050	negative result

We attribute this to the abundance of real critical-regime data in our experimental setting (9999 samples). TimeGAN augmentation may remain valuable in genuinely data-scarce regimes (e.g., rare catastrophic failures), but should not be assumed beneficial by default. We report this result explicitly to encourage methodological transparency in the field.

F. Edge Inference Latency

ONNX inference benchmarks on 1000 calls show p50 = 0.97 ms, p99 = 1.57 ms, and maximum 2.07 ms (Fig. 4). The 50 ms operational budget is respected on 100% of calls, with the p99 representing $\sim 3\%$ of the budget. The exported model is 845 KB, suitable for deployment on resource-constrained gateway hardware.

G. Summary of Quantitative Results

Table I summarizes all quantitative findings.

VI. DISCUSSION

A. Methodological Honesty

A central design decision in this work is the explicit reporting of the TimeGAN negative result. In a competitive publication landscape that incentivizes positive findings, we believe transparency about what does *not* work is as scientifically valuable as reporting what does. The TimeGAN module remains in our framework for scenarios where critical-regime data is genuinely scarce — e.g., catastrophic component

failures occurring once per season — but we caution against blanket adoption of generative augmentation.

B. Limitations

The most significant limitation of this work is the reliance on synthetic data. While our generator embeds established physical mechanisms, real-world telemetry exhibits sensor failures, communication artifacts, and adversarial environmental factors not present in our setting. Validation on industry telemetry remains the critical next step.

A second limitation concerns the conformal calibration: the present implementation calibrates offline on a held-out fold. In non-stationary regimes — e.g., as a component approaches end-of-life and its degradation distribution shifts — adaptive recalibration [12] may be required.

Third, the synthetic generator simulates a single component (brake disc) on a single circuit profile. Generalization across components (suspension dampers, power unit elements) and circuits (varying braking-event density and thermal load) requires further investigation.

C. Deployment Considerations

The 845 KB ONNX artifact and sub-millisecond inference latency are compatible with current automotive ECU and gateway hardware. The fail-safe mode — triggered when input features fall outside the calibrated range — returns a conservative SAFE_RUL value rather than extrapolating into untrained regions. This behavior trades predictive recall for robustness, a trade-off we deem appropriate for safety-relevant decisions.

VII. CONCLUSION

We have presented HYPER-PREDICT, a six-pillar framework combining ensemble learning, physics-informed constraints, symbolic regression, generative augmentation, conformal prediction, and edge deployment. On a synthetic brake-disc telemetry benchmark, the system achieves sub-1% RUL prediction error, formal 99% coverage guarantees, and 1.57 ms p99 inference latency. We deliberately report a negative downstream-utility result for the TimeGAN component to promote methodological transparency.

Future work will validate the framework on real motorsport telemetry through industrial collaborations, extend conformal calibration to non-stationary regimes [12], and investigate cross-component and cross-circuit generalization. We also plan to integrate HYPER-PREDICT with high-fidelity driver-in-the-loop simulators to enable RUL estimation during virtual sessions, allowing predictive maintenance training prior to physical deployment.

ACKNOWLEDGMENTS

The authors acknowledge ActarusLab S.s. for computational infrastructure and the open-source PyTorch, gplearn, and ONNX Runtime communities for the foundational tools used in this work. Portions of the methodology derive from the “Honest OOF Protocol” developed by the same authors in earlier work on QSAR validation.

REPRODUCIBILITY STATEMENT

All code, synthetic data generators, trained model weights, and conformal calibration artifacts are available at the project’s public repository. The complete pipeline reproduces in ~ 35 minutes on a $2\times$ Tesla T4 environment. Random seeds are documented for all stochastic components.

REFERENCES

- [1] J. F. Archard, “Contact and rubbing of flat surfaces,” *J. Appl. Phys.*, vol. 24, no. 8, pp. 981–988, 1953.
- [2] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, “Remaining useful life estimation – a review on the statistical data driven approaches,” *Eur. J. Oper. Res.*, vol. 213, no. 1, pp. 1–14, 2011.
- [3] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [4] A. Saxena, K. Goebel, D. Simon, and N. Eklund, “Damage propagation modeling for aircraft engine run-to-failure simulation,” in *Proc. PHM 2008*, 2008, pp. 1–9.
- [5] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019.
- [6] F. A. C. Viana, R. G. Nascimento, A. Dourado, and Y. A. Yucesan, “Estimating model inadequacy in ordinary differential equations with physics-informed neural networks,” *Comput. Struct.*, vol. 245, p. 106458, 2021.
- [7] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. Springer, 2005.
- [8] A. N. Angelopoulos and S. Bates, “A gentle introduction to conformal prediction and distribution-free uncertainty quantification,” *arXiv:2107.07511*, 2021.
- [9] J. Yoon, D. Jarrett, and M. van der Schaar, “Time-series generative adversarial networks,” in *NeurIPS*, 2019, pp. 5508–5518.
- [10] C. Esteban, S. L. Hyland, and G. Rätsch, “Real-valued (medical) time series generation with recurrent conditional GANs,” *arXiv:1706.02633*, 2017.
- [11] T. Stephens, “gplearn: Genetic programming in Python,” 2015. [Online]. Available: <https://github.com/trevorstephens/gplearn>
- [12] I. Gibbs and E. Candès, “Adaptive conformal inference under distribution shift,” in *NeurIPS*, 2021, pp. 1660–1672.