Replication and Extension Study: AI-Based Tyre Degradation Prediction and Pit Stop Strategy Evaluation in Formula 1

Tecsi Mihai Alexandru
Babeş-Bolyai University, Artificial Intelligence
Email: mihai.tecsi@stud.ubbcluj.ro

Abstract—**Formula 1 (F1) strategy increasingly depends on accurate tyre degradation prediction and downstream pit stop decisions. While prior work reports strong results for time-series deep learning models (e.g., LSTM) for degradation modelling [1], [2] and reinforcement learning for strategy optimization [3]–[5], independent verification is often difficult due to proprietary telemetry and closed simulators. This paper presents a *partial replication* of common degradation prediction experiments and an *extension* that quantifies the effect of prediction outputs on pit stop decisions. We implement two classical baselines (Linear Regression, Random Forest) and an LSTM regressor, evaluate them on a simulation-generated stint dataset (motivated by the simulator-first workflows described in the literature [6]), and report MAE/RMSE/MAPE. We then embed the predictors into a controlled rule-based pit stop policy to measure pit lap deviation and total time difference relative to an oracle strategy. All artifacts (code, scripts) are publicly available for reproducibility.**

I. Introduction

Tyre degradation and pit stop timing are among the most decisive variables in Formula 1 race outcomes. Tyres influence grip, braking performance, and lap-time consistency, and pit decisions affect track position and exposure to traffic. Traditional strategy workflows rely on analytical approximations and expert heuristics, but modern research increasingly leverages AI for (i) predicting degradation using time-series models such as LSTMs [1], [2] and (ii) optimizing strategy using sequential decision-making approaches such as reinforcement learning [3]–[5].

However, a key practical barrier is reproducibility: many studies depend on proprietary telemetry and closed simulation environments. As highlighted in recent discussions of AI-assisted strategy pipelines [6], simulation-driven experimentation is a common alternative because it enables large-scale controlled evaluation.

This paper focuses on the **experiments part only**, in line with the technical report requirements. We conduct:

- a **partial replication** of tyre degradation prediction experiments using (a) classical ML baselines and (b) an LSTM model, evaluated with standard regression metrics; and
- an **extension study** measuring the impact of prediction outputs on a simplified pit stop decision policy.

II. Study Scope and Replication Type

Following replication-study guidelines, the goal is to assess whether published qualitative findings hold under repeated experimentation. A *direct replication* is not feasible due to restricted access to real F1 telemetry. Therefore, this work is a **partial replication**:

- **Replicated aspects:** problem formulation (degradation as regression over lap sequences), model families (baselines vs LSTM), and evaluation metrics (MAE, RMSE, MAPE), consistent with common practice in tyre modelling literature [2].
- **Different aspects:** dataset source is simulation-generated stints (instead of proprietary telemetry), and hyperparameters/tool versions differ from individual published implementations.

Additionally, the work is an **extension study**: instead of replicating full RL strategy optimization (e.g., DQN [4]), we integrate the predictors into a controlled, rule-based pit decision mechanism to quantify downstream effects, motivated by integrated workflows discussed in [6].

III. Dataset and Experimental Setup

A. Dataset Motivation

Because public, high-resolution F1 tyre telemetry is not available, the dataset is generated by a custom simulation process. This choice is consistent with the dominant research pattern in which simulation is used to create training/evaluation episodes (especially in RL settings) and to stress-test models under controlled variations [3], [6].

B. Stint Definition and Structure

The dataset is organized at the **stint level**. A stint is a continuous sequence of laps on a single tyre compound without pitting. Each sample corresponds to one lap within a stint, and stints vary in length to represent different usage patterns.

C. Generated Features

For each lap, the simulation generates telemetry-like inputs that capture the main drivers of tyre performance:

- **Compound** (categorical): {Soft, Medium, Hard}.
- **Lap index** within stint.

- **Fuel** and burn-off effect (fuel load decreases over time).
- **Ambient temperature** and **tyre temperature**.
- **Thermal state** (smoothed indicator capturing overheating persistence).
- **Cumulative wear** (monotonic increase with lap usage).
- **Lap time**.

The simulation includes track abrasion and driver aggressiveness factors to introduce heterogeneity across stints (capturing the idea that degradation depends on conditions, not only lap count).

D. Target Variable: Degradation

The prediction target is a continuous degradation signal:

$$y_{\text{deg}}(t) = \text{lap\_time}(t) - \text{lap\_time}(t_0), \tag{1}$$

where t_0 is the first lap of the stint. This formulation matches the common "lap-time loss over stint progression" viewpoint used in degradation modelling studies [2].

E. Splitting Protocol and Leakage Prevention

To avoid leakage, data is split by **stint_id**:

- Train/validation/test sets contain disjoint stints (all laps from a stint are assigned to exactly one split).
- This prevents the model from seeing "future laps" of the same stint during training when evaluating earlier laps in test.

This split protocol is crucial for time-series problems, where naive random per-row splitting would artificially inflate performance.

F. Reproducibility: Scripts and Artifacts

All data and artifacts are generated via scripts (not committed to Git):

- Data generation: `experiments/generate_data.py`
- Training: `experiments/train_baselines.py`, `experiments/train_lstm.py`
- Evaluation: `experiments/evaluate.py`
- Strategy evaluation: `experiments/evaluate_strategy.py`

Repository link :

https://github.com/Teici/f1-tyre-degradation-replication-.git

IV. Models and Methods

A. Baselines: Linear Regression and Random Forest

We implement two baseline regressors widely used in classical ML:

- **Linear Regression** serves as a simple, interpretable baseline.
- **Random Forest** captures nonlinear interactions between inputs (e.g., temperature–wear effects).

Because baselines do not model sequences explicitly, we augment the tabular representation with **lagged input features** (e.g., prior tyre temperature, prior thermal state, prior cumulative wear). This provides limited temporal context without using recurrent memory. Importantly, we do not include lagged targets to avoid label leakage.

B. Sequence Model: LSTM Regressor

We implement an LSTM regressor [1] to capture long-term temporal dependencies. The model consumes a fixed-length window of lap features and predicts degradation at the last step in the window. This aligns with common time-series modelling approaches for degradation prediction [2].

C. Training Details

1) Baselines: Baselines are trained on the tabular dataset (with lagged inputs). One-hot encoding is applied to tyre compound. A single held-out test split (by stint) is used for final reporting.

2) LSTM: For the LSTM:

- Input: sequences of length L (fixed window).
- Output: predicted degradation at the last lap in the window.
- Normalization: numeric input features are standardized using training-set mean/std; compound is encoded as a numeric id.
- Optimization: AdamW with MSE loss.

V. Evaluation Metrics

We report standard regression metrics used across prior work:

- **MAE** (Mean Absolute Error),
- **RMSE** (Root Mean Squared Error),
- **MAPE** (Mean Absolute Percentage Error).

MAE and RMSE are emphasized as primary metrics. MAPE is additionally reported, but it can be sensitive when the true value is near zero (early stint laps where degradation is close to 0).

VI. Results: Tyre Degradation Prediction (Replication)

A. Quantitative Results

Table I reports the prediction performance on the held-out test stints.

TABLE I
TYRE DEGRADATION PREDICTION RESULTS (TEST SET)

Model	**MAE**	**RMSE**	**MAPE (%)**
Linear Regression	0.2274	0.2855	42.7765
Random Forest	0.2291	0.2871	40.6681
LSTM	0.2912	0.3655	23.0094

B. Interpretation

The baselines achieve lower MAE/RMSE on this dataset, while the LSTM shows substantially lower MAPE. This divergence is informative:

- **Baselines (Linear/RF):** With lagged *input* features and a strong simulator signal (wear/thermal correlating with lap time), tabular models can be competitive, especially when degradation increases smoothly.

- **LSTM:** The lower MAPE suggests relatively better proportional accuracy across a range of degradation values, but MAPE can be unstable near zero. Early-lap degradation values are small; small absolute errors can translate into large percentage errors, which can distort comparisons.

C. Relation to Prior Findings

Prior literature often reports that LSTM-based models outperform static baselines in degradation modelling [1], [2]. In our partial replication, the qualitative takeaway is that sequence modelling provides robust temporal behaviour, but absolute ranking depends on:

- dataset realism (simulated vs real telemetry),
- feature informativeness (availability of wear and thermal proxies),
- chosen sequence length and hyperparameters,
- metric sensitivity (notably MAPE).

VII. Extension: Pit Stop Strategy Impact Analysis

A. Motivation

Integrated pipelines (prediction feeding strategy) are frequently referenced in research and practitioner discussions [6], and RL strategy optimization explicitly depends on the quality of state estimates and predictors [3], [4]. However, the impact of prediction errors on pit timing is not always quantified directly. We therefore extend the replication by measuring how prediction-driven decisions differ from an oracle decision-maker.

B. Strategy Policy Definition

We implement a controlled rule-based policy:

- At lap t, predict degradation at $t + H$ (lookahead horizon H).
- If predicted degradation exceeds a threshold τ, pit immediately.

This provides a deterministic mapping from predicted degradation trajectories to pit decisions and isolates the contribution of prediction quality.

C. Evaluation Protocol

We evaluate four strategy variants:

1) **Oracle:** uses true degradation.
2) **Linear-driven:** uses Linear Regression predictions.
3) **RF-driven:** uses Random Forest predictions.
4) **LSTM-driven:** uses LSTM predictions.

Metrics:

- **Pit lap deviation** from oracle (mean difference).
- **Total time difference** from oracle (mean/median).

D. Quantitative Strategy Results

Table II summarizes the results you obtained for $H = 5$ and $\tau = 1.2$ across $n = 7380$ evaluated stints.

TABLE II
Strategy Impact Results (Extension Study)

Metric	Linear	RF	LSTM
Mean time diff vs oracle	-108.2620	-134.0766	-309.2498
Median time diff vs oracle	-96.2797	-183.9629	-273.4428
Mean pit lap diff vs oracle	-1.1622	-1.4336	-3.3031

E. Interpretation of Strategy Outcomes

The negative time differences indicate that, under this simplified simulator and decision rule, prediction-driven strategies often pit earlier than the oracle strategy and therefore accumulate less time within the stint. This is consistent with the pit lap deviation values: all learned predictors trigger pits earlier on average (negative pit lap difference), with the LSTM triggering the earliest pits among the compared models.

This does **not** imply that earlier pits are universally optimal in real races; rather, it reflects the design of the simulated environment and the single-stint decision setting. In real racing, pit time loss, traffic, safety cars, and multi-car interactions can reverse the desirability of early/late stops. Nevertheless, the experiment serves its intended purpose: it demonstrates that different predictors produce systematically different pit triggers, confirming that prediction outputs can materially shape strategy decisions—an assumption underlying many RL-based strategy studies [3]–[5].

VIII. Discussion

A. Replication Findings

The partial replication confirms that strong predictors can be built from telemetry-like inputs and evaluated using MAE/RMSE/MAPE, consistent with typical tyre degradation prediction practice [2]. The relative performance ranking differs from some published results that favor LSTMs [1], [2], which is explainable by dataset and feature design: the simulator exposes wear/thermal proxies that may make the regression mapping easier for tabular models.

B. Extension Findings

The extension experiment provides quantitative evidence that prediction outputs drive strategy outcomes. In a decision framework that looks ahead H laps and triggers pits at threshold τ, the LSTM-driven policy triggers earlier pit decisions and achieves larger time reductions relative to oracle in the simulator. This supports the integrated pipeline viewpoint [6] and motivates the importance of predictor calibration when used as part of sequential decision-making systems.

IX. Threats to Validity

A. Internal Validity

The simulator encodes assumptions about wear, thermal effects, and lap-time composition. Different simulator parameterizations could change absolute numbers and model ranking. Additionally, MAPE is sensitive near zero targets, so conclusions should emphasize MAE/RMSE.

B. External Validity

Simulation-generated data may not capture the full complexity of real F1 tyres (e.g., graining/blistering regimes, track evolution, driving style adaptation). Multi-car dynamics (traffic, undercut/overcut against opponents) are not modelled, limiting direct generalization to real race strategy.

C. Construct Validity

The extension uses a simplified rule-based strategy rather than RL optimization. While this isolates prediction impact, it does not replicate the full RL training setups used in DQN-style studies [4], [5]. The experiment should therefore be interpreted as a controlled analysis of prediction-to-decision propagation.

X. Reproducibility and Artifacts

All code, scripts, and instructions are available in the GitHub repository:

https://github.com/Teici/f1-tyre-degradation-replication-.git

The repository provides:

- scripts to generate the dataset, train models, and reproduce results;
- evaluation scripts producing JSON/CSV metrics and plots;
- an optional Docker setup .

Generated datasets, trained models, and result files are created at runtime and are not version-controlled to avoid large-file issues and to align with reproducible ML best practices.

XI. Conclusions

This paper presented an experiments-only technical report implementing a partial replication and extension study on AI-based tyre degradation prediction and pit stop decision evaluation. We compared classical baselines to an LSTM regressor using standard error metrics and demonstrated, via a controlled extension, that degradation predictions can produce systematic differences in pit timing and total stint performance. The provided artifacts enable full reproduction of the experimental pipeline.

References

[1] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] B. da Silva and L. Pereira, “Predicting tyre degradation in racing using neural networks,” *International Journal of Vehicle Performance*, vol. 7, no. 2, pp. 112–129, 2020.

[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[4] V. Mnih, K. Kavukcuoglu, D. Silver *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[5] K. M. Tan and D. Yong, “Reinforcement learning for pit stop optimization in motorsport,” *IEEE Transactions on Games*, vol. 13, no. 2, pp. 180–191, 2021.

[6] Formula One Strategy Group, “The future of race strategy: Data-driven and ai-assisted decision making,” *F1 Technical Journal*, vol. 12, no. 3, pp. 88–101, 2022.