Residual Hybrid Learning for Data-Limited Formula 1 Lap Time Prediction Across Multiple Circuits

Marcos Paulo de Oliveira Pereira[1], Carlos Henrique Gomes Ferreira[2], Alexandre Magno de Sousa[1]

[1] Universidade Federal de Ouro Preto, João Monlevade, Minas Gerais, Brasil
marcos.pop@aluno.ufop.edu.br,alexandre.sousa@ufop.edu.br
[2] Universidade Federal de Ouro Preto, Ouro Preto, Minas Gerais, Brasil
chgferreira@ufop.edu.br

Abstract. Lap time prediction in Formula 1 is a key element of race strategy analysis, as small changes in race pace can affect pit stop timing, tire compound selection, and the viability of tactics such as the undercut or overcut. This work uses publicly available telemetry data extracted with the FastF1 library and examines five Formula 1 circuits from the 2022 to 2025 seasons that fall under a single technical regulation. The methodology uses residual learning in a two-stage deep learning forecasting approach to address data limitations and improve the accuracy of drivers' lap time predictions across multiple seasons. First, a Multiple Linear Regression (MLR) model predicts the aggregated lap time series for all drivers on each circuit. Then, a Long Short-Term Memory (LSTM) network predicts the residuals, filtered by driver and year. In addition, we propose a score that balances performance and model variation, helping identify the most suitable model. We call our approach Hybrid MLR-LSTM. It achieves the best training performance across all circuits. In testing, the hybrid model performs best on Italy and achieves performance comparable to competing models on the other circuits. To support reproducibility, the code repository is publicly available at `https://github.com/emipe09/HybridLSTM-F1`.

CCS Concepts: • **Computing methodologies** → **Machine learning algorithms**.

Keywords: Formula 1, multi-circuit, race pace, lap time prediction, residual learning, LSTM, Hybrid MLR-LSTM.

1. INTRODUCTION

The Formula 1 (F1) racing category is widely recognized as the pinnacle of world motorsport [Stepien 2025; Krzysztoń and Smołka 2026]. This category is not just about sport and entertainment, it is significant for society because its technology benefits passenger vehicles through mechanical advancements, sustainable fuels, hybrid engines, aerodynamic innovations, and enhanced safety [Fédération Internationale de l'Automobile 2025; Stepien 2025]. Beyond these positive impacts and the technical skill of the world's best drivers, F1 also stands out for the massive volume of data generated and analyzed in real time [Bansal et al. 2025; Thomas et al. 2025]. F1 relies on testing, with each practice lap, computer simulation, and main race generating large volumes of data. The technical team uses all data captured by the vehicle telemetry system to make strategic decisions and solve critical daily problems in F1 motorsport [Fatima and Johrendt 2023; Quiroga García 2024]. These problems include pit stops, tire management, aerodynamic development, engine development, and driver feedback [Heilmeier et al. 2018; Heilmeier et al. 2020; Thomas et al. 2025].

For the race, predicting the race pace is considered one of the most important factors influencing the overall impression of the weekend, including free practice, the sprint race, and the main race [Quiroga García 2024]. Race pace is a complex issue influenced by non-linear and stochastic variables, including tire degradation by compound, weather conditions, and subtle differences in handling by the driver or the specific car [Heilmeier et al. 2018; Heilmeier et al. 2020; Fatima and Johrendt 2023]. Because weather conditions vary significantly across countries and affects lap times, to predict a driver's lap times, it is essential to separate the lap time series of each driver by circuit [Zhao 2024]. This requirement reduces the amount of available data and can make the model less accurate [Kara and Küçükmanisa 2025]. In this context, the problem clearly requires careful modeling to answer the following research questions (RQ):

—**RQ1:** *How can a Formula 1 driver's lap time be predicted in scenarios with limited data?*
—**RQ2:** *What strategies can improve the accuracy of models for predicting driver lap times?*
—**RQ3:** *How can the most suitable model be selected while balancing performance and model variance?*

To answer these questions, the study uses real telemetry data from the FastF1 library[1] and covers five distinct circuits from the 2022 to 2025 seasons, encompassing only a single technical regulation. We propose a hybrid deep learning model that incorporates residual learning within a two-stage forecasting approach to improve accuracy. The remainder of this work is organized as follows: Section 2 presents related work, Section 3 describes the methodology, Section 4 presents the results, and Section 5 discusses the conclusions and outlines future directions.

2. RELATED WORK

Formula 1 presents several problems that can be addressed using machine learning and data science techniques. Historically, predicting race scenarios has been closely linked to the development of simulations that forecast specific outcomes based on decisions made during a race [Bekker and Lotz 2009]. However, predicting these scenarios should focus on specific aspects, and the literature addresses some of these aspects of the problem.

First, there is a research strand on race simulation and strategy optimization, particularly concerning pit stop occurrences and tire compound selection [Bekker and Lotz 2009; Heilmeier et al. 2018; Quiroga García 2024]. Another research strand uses classic machine learning algorithms to predict ranking results [Bansal et al. 2025; Todd et al. 2025; El Haber et al. 2025; Krzysztoń and Smołka 2026]. A third research strand, complementary to the previous one, uses deep learning models to capture complex patterns in race data, such as optimal pit stop strategies [Fatima and Johrendt 2023; Sasikumar et al. 2025]. Last but not least, and more closely related to this work, a fourth research strand focuses in lap time predictions. In [Zhao 2024], the authors use a deep neural network to predict lap times in qualifying sessions, splitting the time series by circuit and by driver. The main issue is that this requirement reduces the available data and can make the model less accurate. Similarly, the authors in [Kara and Küçükmanisa 2025] predict lap times using an LSTM network, but they do not split the time series by circuit. In addition, combining data from different circuits into a single time series produces inaccurate models because each circuit has significantly different weather conditions that affect lap times. In sum, in addition to their high computational costs, deep learning models require large volumes of data to produce accurate results.

Despite these studies, and the fact that only a few of them address lap time prediction, a gap remains in the literature: research is needed on how to improve the accuracy of drivers' lap time prediction in data-limited scenarios. This work addresses this gap by using public data from the FastF1 library to model and predict lap times on five different circuits and by proposing a hybrid model to improve lap time prediction performance.

3. METHODOLOGY

This section presents the methodology, with Figure 1 illustrating each step, divided into two parts: (1) Data Preparation, which includes Data Collection, Cleaning and Filtering; and (2) Experiments and Analysis, which consists of Experimental Setup, Results, and Model Analysis.

3.1 Data Collection, Data Preprocessing and Feature Engineering

Data collection was conducted using the FastF1 API. The following tracks were selected to maximize diversity in circuit type, time of day, speed limit, and temperature: (*i*) Jeddah Corniche Circuit in

[1] `https://docs.fastf1.dev/`

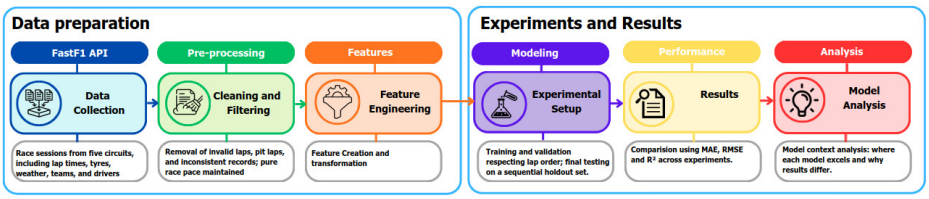


Fig. 1: Experimental methodology pipeline.

Jeddah, Saudi Arabia, a street circuit with a night race; (*ii*) Bahrain International Circuit in Sakhir, Bahrain, a desert circuit with a night race; (*iii*) Circuit of the Americas in Austin, United States, with an afternoon race; (*iv*) Monza National Circuit in Monza, Italy, a high-speed track with an afternoon race; and (*v*) Hungaroring in Mogyoród, Hungary, a low-speed track and afternoon race. To obtain a more homogeneous and representative dataset, records from 2022 to 2025 were selected, which have the same technical regulations. Only records from the main race were included (sprint races, free practice, and qualifying sessions were not considered). The FastF1 API classifies Pirelli tire compounds as soft, medium, and hard, however, Pirelli's tire range actually extends from C1 to C5. A Web scraping was performed on the Pirelli website[2] to identify the correct compounds. Table I shows the dataset's data dictionary, which describes each feature.

Table I: Data dictionary of the dataset.

Variable	Description	Type	Example
`LapTime_sec`	Lap time	Numeric	92.45 sec
`LapTime_prev`	Previous lap time	Numeric	91.80 sec
`TrackTemp`	Track temperature	Numeric	27.80°C
`Humidity`	Air humidity	Numeric	55%
`Pressure`	Atmospheric pressure	Numeric	992.30 mbar
`LapNumber`	Lap number	Numeric	44
`TyreLife`	Tire age	Numeric	12
`Year`	Year of the race	Numeric	2025
`Team`	Team	Categorical	Team_Haas
`Driver`	Driver	Categorical	Driver_GAS
`pirelliCompound`	Tire compound (C1 a C5)	Categorical	pirelliCompound_C3
`WindSpeed`	Wind speed	Numeric	2.30 m/sec

Unlike pure race pace, weather events (e.g. rain), safety car, virtual safety car, and pit entries and exits result in significantly slower laps than normal and considered outliers records. Thus, a filtering process was applied to remove these outliers and telemetry system errors. In addition, the 95th percentile was used as the cutoff threshold, as manual inspections revealed many outliers in the tails of each feature's distribution. So, the number of instances in dataset are: (a) 3,161 in Bahrain; (b) 2,660 in Saudi Arabia; (c) 3,063 in United States; (d) 2,994 in Italy; (e) 4,213 in Hungary.

Missing values in numerical features were imputed using the median. In addition, the original data collection includes nominal categorical features: driver identifiers (`Driver`), team identifiers (`Team`), tire compound identifiers (`pirelliCompound`). The One-hot Encoding technique converts these features to numerical data. Yet, the subtle relationship between air temperature (`AirTemp`) and track temperature (`TrackTemp`) led to the creation of the variable `TempDelta`, defined as $\Delta T = T_{TrackTemp} - T_{AirTemp}$, where $T_{TrackTemp}$ represents the track temperature and $T_{AirTemp}$ represents the air temperature. According to physics, $T_{TrackTemp}$ is the primary predictor of mechanical grip and thermal degradation of the tire, while ΔT indicates the energy balance at the surface [Adwan et al. 2021]. A high ΔT indicates strong radiation and active asphalt heating, while a low ΔT suggests reduced radiation by cloud cover or cooling. Correlation analysis was performed using Pearson's correlation coefficient, that is, ρ. To avoid multicollinearity, as a feature selection method, features with strong correlations ($\rho > 0.85$) were selected based on their impact through Principal Component Analysis (PCA) loading analysis[3]. For this purpose, we used the first and second principal components, which together explain 57.8% of variance for Bahrain, 63.8% for Hungary, 51.1% for Italy, 61.8% for

[2] `https://press.pirelli.com/`
[3] Details are not provided due to space constraints.

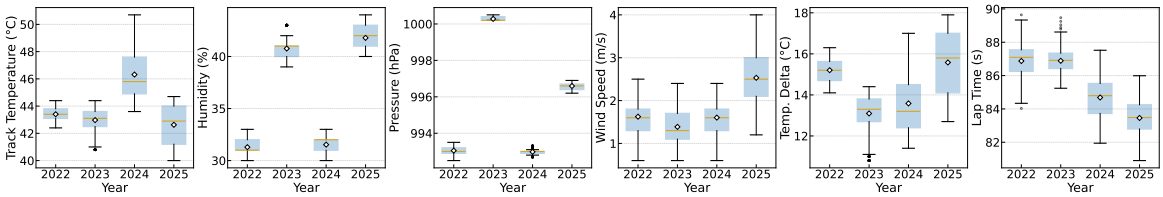
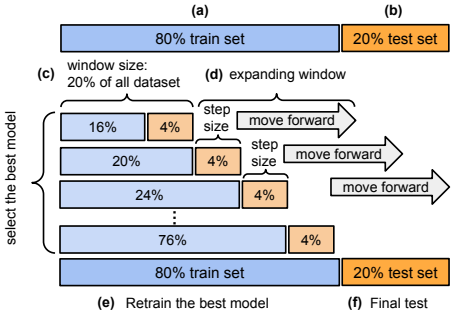


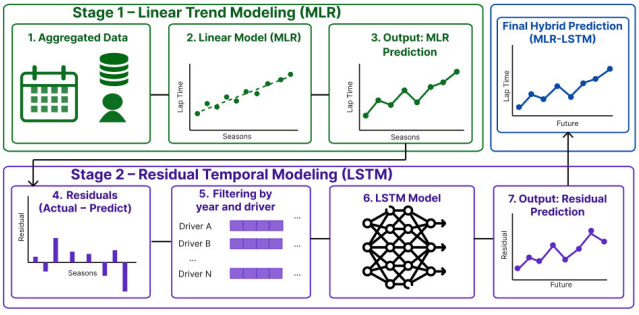
Fig. 2: Distribution of weather condition features and lap times over the years at the Monza National Circuit.

Saudi Arabia, and 69.2% for the USA. As a result of this feature selection, `Year` and `TempDelta` were removed for the United States, `TrackTemp` was removed for Saudi Arabia, and `Humidity` was removed for Hungary. Finally, all features were retained for the remaining tracks.

To emphasize the specific characteristics of circuits based on their location, Figure 2 shows the distribution of weather-condition features and the lap-time dependent variable. This figure illustrates that these variables differ substantially across years. The complete exploratory data analysis and statistical summaries are omitted due to space constraints. After verifying the feature distributions, it was found that although the climatic variables are not right- or left-skewed, they exhibit a multi-modal distribution for all tracks (i.e., `Humidity`, `Pressure`, `TrackTemp`, `WindSpeed`, and `TempDelta`). Therefore, for these features, the Gaussian Radial Basis Function (RBF) kernel transformation was applied, using a similarity measure relative to the respective median, which serves as the reference point [Géron 2019]. The data were normalized to have a mean of zero and a unit standard deviation. The target variable, `LapTime_sec`, was kept in seconds without transformation or scaling.



(a) Expanding window validation example.



(b) Two-stages forecasting approach for Hybrid MLR-LSTM.

Fig. 3: Experimental setup details.

3.2 Experimental Setup

To address data limitations in predicting drivers' lap times, all lap times for every driver from the 2022 to 2025 seasons are arranged sequentially by circuit to form a single time series, while also accounting for the weather conditions in each country. Next, a validation strategy based on rolling forward forecasting for time series was used, specifically *expanding window validation* (EW), which maintains the chronological order of the race laps [Hyndman and Athanasopoulos 2021; Peixeiro 2022]. This approach prevents future laps from influencing the training data and avoids data leakage that could occur if traditional cross-validation were used. Figure 3(a) illustrates this strategy, the dataset, ordered by lap and year for each circuit, was split into 80% for training and validation (Fig. 3(a)-a), and 20% for testing (Fig. 3(a)-b). Initially, for example, a window size of 20% of all instances from the dataset (Fig. 3(a)-c) is divided into 80% for training and 20% for testing. The window expands with each fold by a forward step size of 4% (Fig. 3(a)-d), increasing the training set as the process continues. To determine the window size, tests are conducted with sizes ranging from 5% to 50% of

all dataset instances for each model, using 80% of the window size for training and 20% for testing. The window size that provides the best performance with the least variation is selected.

3.3 Baseline Algorithms and Two-Stage Deep Learning Approach

For the experiments, Multiple Linear Regression (MLR) and the Extreme Gradient Boosting (XGBoost) ensemble learning algorithm were used as baselines. XGBoost was selected for its efficiency in capturing nonlinear relationships and complex interactions between variables. For the XGBoost algorithm, hyperparameter tuning was performed using the Optuna library, a hyperparameter optimization framework. The hyperparameters included the learning rate, maximum tree depth, minimum sum of instance weights in a child leaf, subsample ratio of training instances, column subsample ratio, minimum loss reduction (gamma), and L1 and L2 regularization parameters[4]. Initially, to configure the search space, some hyperparameter values were proposed and executed on 80% of training data and on 20% of test data. Based on the results of this initial configuration, a customized search space was created for each circuit. To explore different combinations of model hyperparameters within this search space, 100 trials were defined. For each trial, the model is trained with early stopping on the training laps and tested on the validation laps for each circuit. The final model is trained on 80% of the training set. Hyperparameters are selected using the median of the best hyperparameters from each window. The model is then tested on 20% of the test set.

Deep learning models require large volumes of data for high-performance applications. Because of this, and following [James Fong et al. 2020], we use hybrid models to improve accuracy and provide better predictions with small datasets. Our strategy is to combine traditional linear regression, which produces linear predictions, with a Long Short-Term Memory (LSTM) network that captures nonlinear temporal dependencies in sequential data. We call this hybrid MLR-LSTM, which uses a residual learning approach with two-stage forecasting approach [Vatsa et al. 2024; Mahmud et al. 2025]. This strategy is presented in Figure 3(b). First, the MLR component models the pace race's long-term trends across multiple seasons using aggregated lap times for all drivers on a given circuit. In turn, the differences between the actual and predicted values, i.e., the residuals, are then extracted and treated as a new time series filtered by year for each driver. Thus, the LSTM models the random variations in these MLR residuals as a multivariate time series, using the feature's time series as its input. At the end, the prediction of the hybrid model is obtained by adding the MLR predictions to the residuals predicted by the LSTM. Details of the LSTM experiments are described below.

For high-cardinality categorical variables such as Driver and Team, embeddings were used to map each driver and team to a dense, low-dimensional vector, which were learned during training. LSTM hyperparameter tuning was performed using the Optuna library which includes: the number of units in the recurrent layer, the number of units in the dense layer, dropout rates and recurrent dropout, learning rate, batch size, the delta parameter for the Huber loss function, and the L2 regularization[4]. To explore different combinations within the search space, 50 trials were run per circuit. The search space was defined based on initial baseline runs and was kept consistent across all circuits. The 80% training set is sequentially split into training and validation subsets. The number of epochs is tuned using early stopping based on the validation loss, combined with adaptive learning-rate reduction. The final model is retrained on the full 80% training set and evaluated on the remaining 20% test set.

3.4 Balanced Score Approach to Select Models

To assess the balance between performance and model variance, grounded on [Michelucci and Venturini 2025], the Composite Overfitting Score (COS) indicator is defined by

$$\mathrm{COS} = \alpha \frac{\mathrm{Performance}_{\mathrm{test}}}{\mathrm{Performance}_{\mathrm{train}}} + \beta \frac{\sigma_{\mathrm{test}}}{\sigma_{\mathrm{train}}}, \tag{1}$$

[4]Details about the ranges of hyperparameter search spaces are available in the code repository for this work.

Table II: Performance comparison between models.

Circuit	Model	RMSE	(95%C.I.)	MAE	(95%C.I.)	R^2	(95%C.I.)	#driver	m	m/driver
Bahrain	MLR	0.69	(0.48,0.90)	0.61	(0.41,0.82)	-1.40	(-3.46, 0.65)	29	3,161	109.00
	XGBoost	0.53	(0.43,0.63)	0.45	(0.36,0.54)	-0.11	(-0.56,0.34)			
	LSTM	1.26	(0.95,1.57)	1.15	(0.84,1.46)	-5.01	(-7.79,-2.23)			
Saudi Arabia	MLR	0.95	(0.43,1.46)	0.86	(0.35,1.36)	-13.95	(-34.98,7.09)	28	2,660	95.00
	XGBoost	0.69	(0.55,0.83)	0.60	(0.47,0.72)	-3.45	(-6.75,-0.16)			
	LSTM	0.85	(0.68,1.02)	0.75	(0.60,0.90)	-6.28	(-10.76,-1.80)			
USA	MLR	2.04	(0.74,3.34)	1.92	(0.63, 3.22)	-156.48	(-365.34,52.38)	29	3,063	105.60
	XGBoost	0.64	(0.47,0.81)	0.54	(0.38,0.69)	-1.13	(-2.15,-0.12)			
	LSTM	1.41	(1.17,1.65)	1.31	(1.08,1.55)	-10.79	(-16.55,-5.03)			
Italy	MLR	0.65	(0.45,0.84)	0.57	(0.39,0.75)	-4.97	(-9.89,-0.05)	30	2,994	99.80
	XGBoost	0.80	(0.63,0.96)	0.69	(0.54,0.84)	-4.38	(-7.13,-1.64)			
	LSTM	1.96	(1.58,2.34)	1.91	(1.53,2.29)	-36.81	(-49.24,-24.39)			
Hungary	MLR	0.63	(0.44,0.81)	0.54	(0.36,0.72)	-0.10	(-0.72,0.53)	29	4,213	145.30
	XGBoost	0.84	(0.68,1.01)	0.70	(0.55,0.85)	-0.70	(-1.50,0.09)			
	LSTM	1.55	(1.32,1.77)	1.41	(1.20,1.63)	-5.60	(-8.05,-3.15)			

here, $\text{Performance}_{\text{train}}$ and $\text{Performance}_{\text{test}}$ denote the RMSE, MAE or R^2 for training and testing, while σ_{train} and σ_{test} denote the standard deviation of predictions for training and testing, and α and β weight the importance of average performance (first term) and model variance (second term). In this work, $\alpha = 0.5$ and $\beta = 0.5$. Thus, $COS \approx 1$ indicates an ideal model in which training and testing errors coincide, and the variance between folds during training is consistent.

4. RESULTS

To address **RQ1**, we then organize lap-time series by circuit for each driver. The data are split into 80% for the training and validation sets and 20% for the test set. Table II presents performance metrics such as RMSE, MAE, and R^2 for MLR, XGBoost, and LSTM. In addition, this table shows the number of drivers, total of instances (i.e. m) and the number of instances per driver. In terms of RMSE and MAE, LSTM yields the worst results for the Bahrain, Italy, and Hungary circuits. XGBoost performs worst for Italy and Hungary. The effect of limited data is more pronounced for LSTM, as it is a deep neural network. Surprisingly, XGBoost delivers the best performance for Bahrain, Saudi Arabia, and the United States. Overall, none of the models achieve a good fit with $R^2 < 0$. Yet, the number of instances per driver is one order of magnitude lower than m.

In this context, to evaluate model performance with our solution by aggregating lap times from seasons for all drivers, MLR, XGBoost and LSTM were directly compared across the five analyzed circuits. The results for model performance on the training and test sets are shown in Table III, where the optimal window size is presented in parentheses before the model name. This table presents the RMSE, MAE, and R^2 metrics for each dataset listed in the "Set" column (e.g., Train or Test). The performance metrics are reported with their respective 95% confidence intervals (C.I.). Thus, the models in Table III perform better than those in Table II, where the time series are filtered by driver, reducing the number of instances. For example, in Table II, for the USA circuit, the MLR and Hybrid MLR-LSTM models obtained a MAE of 1.92 seconds and 1.31 seconds, respectively. While in Table III for the same circuit, MLR get a MAE of 0.26 seconds and MLR-LSTM get a MAE of 0.27 seconds, which confidence interval differ significantly from results in Table II.

Addressing **RQ2**, we propose a strategy to improve lap time prediction accuracy using the hybrid MLR–LSTM model. Table III highlights in red all circuits for which our solution achieves better performance, with the lowest mean for all training results. This scenario changes when we analyze the results for the test sets, where better performance is highlighted in green. However, there is a tie among all models because there are no significantly different results (i.e. C.I. overlap). The exception is Italy, where the hybrid model shows the best performance, with results for all metrics significantly different from those of the other models (its C.I. are in red). When observing RMSE and MAE, the MLR-LSTM is at least 11% and 27% more accurate regarding to MLR and XGBoost, respectively. The tie observed for the other circuits may be related to the model's generalization capability on the test set, which consists of records from 2025. Manual inspection reveals that these data distributions

Table III: Performance comparison between models.

Circuit	Set	Model	σ	RMSE	(95%C.I.)	COS_{RMSE}	MAE	(95%C.I.)	COS_{MAE}	R^2	(95%C.I.)	COS_{R^2}
Bahrain	Train	MLR (EW 5%)	0.3059	0.36	(0.25,0.47)	–	0.30	(0.20,0.39)	–	0.64	(0.37,0.91)	–
		XGBoost (EW 30%)	0.2821	0.29	(0.24,0.33)	–	0.22	(0.19,0.26)	–	0.88	(0.86,0.91)	–
		Hybrid MLR-LSTM	0.3168	**0.25**	(0.23,0.27)	–	**0.19**	(0.18,0.21)	–	**0.91**	(0.90,0.93)	–
	Test	MLR	0.3140	**0.31**	(0.30,0.33)	0.95	0.25	(0.23,0.26)	0.93	**0.88**	(0.86,0.90)	0.88
		XGBoost (EW 30%)	0.3122	0.34	(0.32,0.36)	1.15	0.27	(0.26,0.29)	1.16	0.86	(0.83,0.88)	1.07
		Hybrid MLR-LSTM	0.3164	0.32	(0.30,0.34)	1.25	**0.24**	(0.23,0.26)	1.26	0.87	(0.85,0.89)	1.15
Arabia Saudi	Train	MLR (EW 10%)	0.3406	0.37	(0.34,0.39)	–	0.29	(0.27,0.32)	–	0.80	(0.76,0.84)	–
		XGBoost (EW 50%)	0.3168	0.32	(0.27,0.36)	–	0.24	(0.20,0.29)	–	0.87	(0.81,0.94)	–
		Hybrid MLR-LSTM	0.3088	**0.30**	(0.28,0.33)	–	**0.24**	(0.22,0.26)	–	**0.94**	(0.93,0.95)	–
	Test	MLR	0.3185	0.34	(0.31,0.36)	0.92	0.26	(0.24,0.28)	0.91	0.91	(0.90,0.93)	0.91
		XGBoost (EW 50%)	0.3175	**0.33**	(0.30,0.35)	1.02	**0.25**	(0.23,0.27)	1.01	**0.92**	(0.90,0.93)	0.98
		Hybrid MLR-LSTM	0.3186	**0.33**	(0.31,0.35)	1.06	**0.25**	(0.24,0.27)	1.05	**0.92**	(0.90,0.93)	1.03
USA	Train	MLR (EW 45%)	0.3518	0.36	(0.34,0.39)	–	0.28	(0.26,0.31)	–	0.83	(0.80,0.86)	–
		XGBoost (EW 5%)	0.3117	**0.32**	(0.30,0.33)	–	0.26	(0.24,0.27)	–	0.83	(0.81,0.86)	–
		Hybrid MLR-LSTM	0.3186	**0.32**	(0.30,0.34)	–	**0.25**	(0.23,0.27)	–	**0.91**	(0.89,0.92)	–
	Test	MLR	0.3330	**0.33**	(0.32,0.35)	0.93	**0.26**	(0.25,0.28)	0.93	**0.87**	(0.85,0.88)	0.95
		XGBoost (EW 5%)	0.3378	0.36	(0.34,0.38)	1.11	0.28	(0.27,0.30)	1.10	0.84	(0.82,0.86)	1.04
		Hybrid MLR-LSTM	0.3301	0.34	(0.32,0.36)	1.05	0.27	(0.25,0.28)	1.05	0.86	(0.84,0.88)	1.04
Italy	Train	MLR (EW 5%)	0.2930	0.36	(0.30,0.41)	–	0.29	(0.24,0.35)	–	0.61	(0.42,0.79)	–
		XGBoost (EW 50%)	0.3309	0.36	(0.20,0.51)	–	0.28	(0.16,0.40)	–	0.80	(0.72,0.88)	–
		Hybrid MLR-LSTM	0.2675	**0.27**	(0.25,0.29)	–	**0.21**	(0.19,0.22)	–	**0.90**	(0.88,0.92)	–
	Test	MLR	0.2715	0.33	(0.31,0.35)	0.92	0.27	(0.25,0.28)	0.92	0.84	(0.82,0.87)	0.82
		XGBoost (EW 50%)	0.3413	0.43	(0.40,0.46)	1.12	0.33	(0.30,0.35)	1.10	0.74	(0.71,0.76)	1.06
		Hybrid MLR-LSTM	0.2657	**0.29**	(0.28,0.31)	1.05	**0.24**	(0.22,0.25)	1.07	**0.87**	(0.85,0.89)	1.01
Hungary	Train	MLR (EW 45%)	0.3238	0.33	(0.30,0.36)	–	0.26	(0.23,0.28)	–	0.88	(0.84,0.91)	–
		XGBoost (EW 40%)	0.3201	0.33	(0.30,0.37)	–	0.26	(0.23,0.29)	–	0.85	(0.81,0.90)	–
		Hybrid MLR-LSTM	0.3110	**0.31**	(0.29,0.33)	–	**0.23**	(0.22,0.25)	–	**0.90**	(0.89,0.92)	–
	Test	MLR	0.3412	**0.34**	(0.33,0.36)	1.05	**0.27**	(0.25,0.28)	1.05	**0.91**	(0.90,0.92)	1.01
		XGBoost (EW 40%)	0.5466	0.59	(0.56,0.62)	1.74	0.44	(0.41,0.46)	1.71	0.73	(0.71,0.75)	1.44
		Hybrid MLR-LSTM	0.3462	0.35	(0.33,0.37)	1.12	**0.27**	(0.26,0.29)	1.14	0.90	(0.89,0.91)	1.06

Legend: *red cells* and *green cells* indicate the best results on the training set and the test set, respectively. The lowest COS values are shown in *blue*, and statistically significant C.I. values are shown in *red*.

differ from those in the training set.

To answer **RQ3**, we also report in Table III the standard deviation of the residuals for the training and test sets to illustrate the model's variability, σ_{test} and σ_{train} are used to compute COS. Table III reports COS_{RMSE}, COS_{MAE} and COS_{R^2}, respectively. These scores are used to balance average performance with model variability, helping identify the most suitable model for each circuit. In this table, the lower COS scores are highlighted in blue. According to the COS, the best model for predicting lap times on the Italy circuit is the hybrid MLR-LSTM, with the lowest COS scores. Furthermore, only its RMSE and MAE differ significantly from those of the other models. On the USA circuit, the hybrid model also has lower COS scores, at about 1.05 across all performance metrics. In contrast, the worst COS for MLR-LSTM occurred on the Bahrain and Hungary circuits, where it showed a poor balance between average performance and model variability. In these circuits with statistical dead heats, we use COS as the tiebreaker, so MLR has the best COS. In Saudi Arabia, XGBoost achieves the best COS, followed closely by MLR-LSTM with comparable values. In contrast, the lowest COS score for XGBoost was observed in Hungary, with very poor balance.

5. DISCUSSION AND FUTURE WORK

In this work, we propose a hybrid MLR–LSTM model with two-stage forecasting approach to improve the accuracy of predicting a driver's lap time across multiple F1 race circuits. Specifically, we use an LSTM network to predict the MLR model's residuals relative to the driver's target lap time. The final prediction is obtained by adding the baseline prediction to the residuals predicted by the LSTM network. Although direct comparisons are not possible, our results show a clear advancement over current lap time prediction methods in the literature. We obtain MAE and RMSE values on the order of tenths of a second, whereas previous work has reported errors exceeding 5 seconds on certain tracks [Zhao 2024] and a MAE of 1.404 seconds [Kara and Küçükmanisa 2025], representing a substantial average difference that can decide the winner. In addressing three research questions, our work makes the following contributions: (1) we propose aggregating all drivers' lap times for a given circuit into

a single combined time series spanning multiple seasons to overcome data limitations; (2) we propose a hybrid model for predicting lap times that uses a deep learning model to correct the baseline's predictions; at the end, (3) we introduce a score that balances performance and model variance to help select the most suitable model. As future work, we plan to explore in depth which factors affect the model's predictions and to explain the performance of the hybrid MLR-LSTM model, with the goal of improving its accuracy and incorporating explainability into the results to extract insights into race pacing. To this end, we intend to use Local Interpretable Model-Agnostic Explanations (LIME) and to incorporate an attention layer [Kiser et al. 2024].

REFERENCES

Adwan, I., Milad, A., Memon, Z. A., Widyatmoko, I., Zanuri, N. A., Memon, N. A., and Yusoff, N. I. M. Asphalt pavement temperature prediction models: A review. *Applied Sciences* 11 (9): 3794, 2021.

Bansal, A., Arora, A., Bhati, L., Sethia, K., Verma, I., and Kapoor, N. Advanced machine learning approaches for formula 1 race performance prediction: A comprehensive analysis of championship point forecasting, 2025. Preprint available at ResearchGate.

Bekker, J. and Lotz, W. Planning formula one race strategies using discrete-event simulation. *Journal of the Operational Research Society* 60 (7): 952–961, 2009.

El Haber, E., Sawaya, E., Attieh, M., Tannous, A., Ghazaly, W., and Owayjan, M. Formula 1 Race Winner Prediction Using Random Forest and SHAP Analysis. In *IC2AI*. IEEE, Beirut, France, pp. 1270–1274, 2025.

Fatima, S. S. W. and Johrendt, J. Deep-racing: An embedded deep neural network (ednn) model to predict the winning strategy in formula one racing. *International Journal of Machine Learning* 13 (3), 2023.

Fédération Internationale de l'Automobile. F1's new era. `https://www.fia.com/news/f1s-new-era-everything-you-need-know-about-how-fia-making-formula-1-more-competitive-more`, 2025. Accessed: 2026-05-03.

Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Sebastopol, CA, 2019.

Heilmeier, A., Graf, M., Betz, J., and Lienkamp, M. Application of monte carlo methods to consider probabilistic effects in a race simulation for circuit motorsport. *Applied Sciences* 10 (12): 4229, 2020.

Heilmeier, A., Graf, M., and Lienkamp, M. A race simulation for strategy decisions in circuit motorsports. In *Proc. of the IEEE 21st ITSC*. IEEE, pp. 2986–2993, 2018.

Hyndman, R. J. and Athanasopoulos, G. *Forecasting: Principles and Practice*. OTexts, Melb., Australia, 2021.

James Fong, S., Li, G., Dey, N., González Crespo, R., and Herrera Viedma, E. Finding an accurate early forecasting model from small dataset: A case of 2019-ncov novel coronavirus outbreak. *International Journal of Interactive Multimedia and Artificial Intelligence* 6 (1): 132–140, Mar., 2020.

Kara, M. F. and Küçükmanisa, A. Predicting lap times in formula 1: A deep learning approach using lstm networks. In *2025 IEEE 21st ICCP*. pp. 1–5, 2025.

Kiser, A. C., Shi, J., and Bucher, B. T. An explainable long short-term memory network for surgical site infection identification. *Surgery* vol. 176, pp. 24–31, 2024.

Krzysztoń, S. A. and Smołka, J. Application of machine learning for predicting formula 1 race results. *Journal of Computer Sciences Institute*, 2026.

Mahmud, A., Noor, S., Musa, K., Hamzah, F., Yudin, Z., Kamaruddin, N., Madawana, A., and Awang Nawi, M. Hybrid arima-lstm for covid-19 forecasting: a comparative ai modeling study. *PeerJ Comp. Sci.* vol. 11, pp. e3195, Sep., 2025.

Michelucci, U. and Venturini, F. Best practices for machine learning experimentation in scientific applications, 2025.

Peixeiro, M. *Time Series Forecasting in Python*. Manning Publications, 2022.

Quiroga García, M. P. Optimization of pit stop strategies in formula 1 racing: A data-driven approach, 2024.

Sasikumar, A., Leema, A. A., and Balakrishnan, P. Data-driven pit stop decision support for formula 1 using deep learning models. *Frontiers in Artificial Intelligence* vol. Volume 8, 2025.

Stepien, Z. The pro-ecological evolution of powertrains and fuels in formula 1. *Energies* 18 (22): 6013, 2025.

Thomas, D., Jiang, J., Kori, A., Russo, A., Winkler, S., Sale, S., McMillan, J., Belardinelli, F., and Rago, A. Explainable reinforcement learning for formula one race strategy. *arXiv preprint arXiv:2501.04068*, 2025.

Todd, J., Jiang, J., Russo, A., Winkler, S., Sale, S., McMillan, J., and Rago, A. Explainable time series prediction of tyre energy in formula one race strategy. In *Proc. 40th SIGAPP*. SAC '25. ACM, pp. 1082–1089, 2025.

Vatsa, A., Hati, A. S., Kumar, P., Margala, M., and Chakrabarti, P. Residual lstm-based short duration forecasting of polarization current for effective assessment of transformers insulation. *Sci. Rep.* 14 (1369), 2024.

Zhao, Z. Deep neural network-based lap time forecasting of formula 1 racing. *Applied and Computational Engineering* vol. 47, pp. 61–66, 03, 2024.