# Comparing Deep Learning vs Non Deep Learning approaches to predicting Formula One lap times

Mitchell Freedman
Stevens Institute of Technology
mfreedm1@stevens.edu

Shea Griffin
Stevens Institute of Technology
sgriffin1@stevens.edu

Jesus Rivero
Stevens Institute of Technology
jrivero@stevens.edu

Arfah Ayesha Shahid
Stevens Institute of Technology
ashahid1@stevens.edu

***Abstract*— Formula One represents the pinnacle of motorsport, where highly specialized race cars are engineered to achieve the fastest possible lap times and secure victory reliably and consistently over the many races of the season. The extensive telemetry collected from onboard sensors provides a detailed view of car performance throughout a race. In this study, we investigate whether Formula One lap times can be predicted using a combination of statistical and Deep Learning methods applied to this rich dataset. Our results demonstrate that relatively accurate predictions of future lap times are achievable, particularly through statistical modeling approaches. However, the model remains limited in scope, as it does not account for several critical factors, including environmental conditions, driver behavior, on-track traffic, and aerodynamic advantages such as DRS. Despite these limitations, the model offers a valuable foundation for teams with fewer resources, providing a data driven approach to performance analysis and race strategy development.**

***Keywords—Formula One, F1, Formula1, Lap Prediction, Pace Degradation, LSTM-FCN, XGBoost, Linear Regression, Automotive, Racing, Tire Wear.***

## I. Introduction

For as long as there have been automobiles, there have always been those who chase the thrill of the drive. From backroad runners to casual Sunday drivers, cars are far more than people-movers; they deliver engaging, connected, and genuinely fun experiences, especially on well-paved roads through scenic landscapes. For those who want to push far beyond what is possible on public roads, we built racetracks and proving grounds: dedicated spaces to evaluate the absolute limits of performance. Some "race cars" are simply daily drivers with helmeted owners looking to explore their own boundaries. Others are highly modified machines, factory-built or home-grown, aimed at shattering personal records.

At the summit stands Formula One (F1), the highest form of single-seater racing (whereas most cars have two front seats for driver and passenger) the world has ever seen. These cars are bespoken from the first sketch to the final bolt, designed and constructed by teams of hundreds in the relentless pursuit of faster lap times; chasing a championship that pays in the millions. These cars are hyper efficient, exploiting aerodynamic forces so extreme they can drive upside-down on a ceiling at speeds over 200 mph. The sport's global appeal is unmatched with more than 20+ races across continents and fans spanning every corner of the globe. The bleeding edge of F1 technology constantly redefines what is possible in racing, fusing thermodynamics, aerodynamics, materials science, electronics, data analytics, and human physiology. Pair that with drivers who have trained since childhood, and you get the most competitive, technically sophisticated form of motorsport on Earth.

## II. The Game of Motorsport

Like many elite sports, Formula One revolves around teams, winners, strict rules, penalties, colored flags, and a detailed points system. The now eleven teams, often called constructors, are responsible for hiring two drivers and designing and building two identical cars that must comply with the technical and sporting regulations set by the Federation Internationale de l'Automobile (FIA), Formula One's governing body. These cars are finely tailored to each driver. From suspension geometry and steering feel to brake bias and aerodynamic setup, every detail is adjusted to match the driver's unique style and preferred approach to corners and laps, and even changed for specific tracks and times of day.

Interestingly, in Formula One there are two different championships, namely for the drivers and constructors. The drivers' championship recognizes the best driver on the current field, besting other drivers by scoring the most points during the races. The constructors championship is similarly based on the same point system but is awarded as the aggregate of both drivers points. This detail means the teams needs to focus of the

betterment of both their drivers as well as car reliability, as there are no points awarded to a car that breaks down mid race.

The points are awarded based on finishing position at the end of each Grand Prix. The higher you finish, the more points you score and the season is decided by both the driver and constructor with the most points. Currently of the 22 drivers on the field (as of 2026) the system awards points to only the drivers finishing 10th and above per each race as follows.

| Finish | Points |
|---|---|
| 1st | 25 |
| 2nd | 18 |
| 3rd | 15 |
| 4th | 12 |
| 5th | 10 |

| Finish | Point |
|---|---|
| 6th | 8 |
| 7t | 6 |
| 8th | 4 |
| 9th | 2 |
| 10th | 1 |

Figure 1: Formula One point system

But the drivers alone do not take the victory, for behind every car on the grid stands a massive operation of roughly 1,000 people per constructor, even for teams with more financial backing, since Formula One has enlisted a cost cap on hiring and car development. This includes not just mechanics and race engineers, but aerodynamicists, materials scientists, strategists, data analysts, trainers, physiotherapists, marketers, and the team principal. The Constructors' Championship exists to honor this entire ecosystem. It rewards the reliability and outright speed of the car, the quality of pit stops (where tires are changed during a race), race strategy, and the innovation and organization that happen back at the factory. It is not solely a sport about the drivers' talent on race day. That's why the Constructors' title carries the largest share of the sport's prize money, it recognizes the full team effort that turns raw engineering ambition into on-track dominance.

## III. Strategy

The science of race strategy is far too complex to fully capture in the length of this paper, but we can highlight the core factors that contribute to a driver's success on track. Race position is influenced by a combination of variables, including driver skill, pit stop execution, tire selection & management, slipstreaming slower cars (a feature of aerodynamic gain), and the effective use of novel technologies. Additionally, race craft and measured aggression play a crucial role, as drivers must often fight and defend to gain or hold positions against competitors who are unwilling to yield easily.

Modeling driver aggression for example is far more difficult than observing who has, what is colloquially referred to as, a "lead foot". While telemetry can tell us how many G's the driver is willing to risk on high speed turns and what point they turn in to an apex, we instead account for more measurable factors such as tire degradation and fuel load. Tires in Formula One are highly specialized: a single supplier, Pirelli, works with the FIA to select three race compounds from a range of five for each circuit, and drivers working with their teams are required to use at least two different compounds during a race. These compounds vary in hardness, which directly affects performance and longevity. Softer tires provide greater grip and faster lap times but degrade more quickly, while harder tires last longer but offer less grip and require more time to reach optimal temperature. The FIA and Pirelli determine the available compounds using proprietary models, intentionally ensuring that no single tire can last the entire race distance. thereby introducing strategic variation and forcing teams to balance speed against longevity. These models change for each track and weather as environmental factures such as temperature, altitude, and moisture change how tires interact with the road surfaces.

Fuel is, on the surface, somewhat simpler to model, though it still presents meaningful challenges. In F1, cars are not allowed to refuel during a race, so as fuel is burned, the car becomes lighter, significantly so. The weight difference will affect handling and lap time compared to the start. With roughly 250 pounds of fuel consumed over a race in a car that weighs around 1,700 pounds. This reduction in mass has a substantial impact on performance, aerodynamics, and handling. At different stages of the race, drivers may adjust fuel usage depending on strategy such as pushing harder (burning more fuel) to gain positions or conserving to maintain pace and even ensuring they reach the finish. The cars are not required to start with a full fuel load and often calculate the minimum necessary amount based on predictive models, allowing the car to begin the race as light as possible while still completing the full race distance.

One final concept, included for completeness, is the Drag Reduction System (DRS). In Formula One, DRS was in use from 2019 to 2025 and is no longer present in 2026. Nevertheless, aerodynamics continue to play a critical role in performance. When a car follows closely behind another, it encounters "dirty air" which is the turbulent, unstable airflow left in the wake of the leading car. This reduces downforce (the aerodynamic force that presses the car into the ground to improve traction), which makes handling more difficult and thereby limits overtaking opportunities. DRS is designed to mitigate this effect by allowing the trailing car to open a flap, (Figure 2), on its rear wing, reducing drag and increasing straight-line speed. However, this system is only available under specific conditions set by the FIA, namely the trailing car must be within one second (as measured by telemetry) of the car ahead at a designated detection point, (Figure 4).


Figure 2: Shows the active use of DRS. The left image shows DRS open while the right image shows DRS is closed

Figure 4: DRS zones (light gray) on a circuit set by the FIA where DRS can be activated should a driver be within a second of the leading car

## IV. TIMINGS AND TELEMETRY

Formula One is a sport defined by milliseconds, these cars are nearly 18 feet long and must navigate complex circuits while staying within track limits and crossing the start/finish line to record an official time sanctioned by the FIA. Small errors matter greatly, drivers who exceed track boundaries can have their lap times invalidated, and it is not uncommon for first and second place to be separated by just thousandths of a second, meaning every pit stop, tire choice, and strategic decision is aimed at gaining fractions of a second.

Modern F1 cars are equipped with a wide range of sensors that continuously track the car's position, speed, and performance relative to the circuit. This data allows drivers to monitor their lap time deltas down to milliseconds. With this information, drivers and teams can make real-time adjustments to car settings, sometimes multiple times within a single lap, to optimize performance for each corner. In live race broadcasts, it's common to see drivers frequently adjusting controls on their steering wheels (Figure 5) while maintaining speeds well over 150 mph. They also receive continuous updates on lap time deltas from their race engineers over team radio, helping them refine their driving strategy throughout the race


Figure 5: A modern example on an F1 steering wheel with numerous inputs changed. Drivers update the cars setting multiple times per lap depending on conditions.

## V. INTRODUCING THE PROJECT

Our aim in this project is to predict F1 lap times by modeling race performance using a range of physical, mechanical, and strategic inputs. Specifically, we seek to represent how variables such as tire degradation, fuel load, track characteristics, and race conditions collectively influence lap time across a stint and an entire race.

To achieve this, we propose building and comparing multiple predictive models to evaluate their effectiveness in forecasting lap times under different conditions. In addition to standard modeling approaches, we introduce a pace degradation model to capture how performance evolves over time as tires wear and fuel decreases, generating additional engineered features that reflect the dynamic nature of race pace. We believe this combined offers a more realistic and novel framework for estimating lap times.

## VI. DATA SOURCING & SELECTION

### A. *Our data*

Our data is sourced from the free to use FastF1[1], a python package for accessing and analyzing Formula One results, schedules, timing data and telemetry pulled from Formula One's own API.

FastF1, while it has its limitations, gives us quite a bit of granularity with respect to each driver in a certain track in a certain year. We pull this telemetry data, which consists of high-frequency time-series measurements recorded with a frequency of ~275 milliseconds during a lap, and record them all in a dataset, one telemetry event (i.e. time step) per row. The following are some of the variables the telemetry events contain (Figure 6):

| Variable | Description |
|---|---|
| Date | Event calendar date |
| RPM | Engine revolutions/minute |
| Speed | Speed in km/h |
| nGear | Current gear number |
| Throttle | Throttle % pedal input |
| Brake | Brake %pedal input |
| DRS | Drag reduction status |
| Source | Telemetry data origin |
| Time | Timestamp within lap |
| SessionTime | Session elapsed time |
| Distance | Distance traveled on track |
| LapTime | Total lap duration |
| Compound | Tire compound type |
| LapNumber | Current lap count |
| Stint | Continuous run segment |
| TyreLife | Tire wear duration |

| | |
|---|---|
| StintLap | Laps within stint |
| WarmupLapFlag | Tire warmup indicator |
| NoisyLapFlag | Unreliable lan indicator |
| Driver | Driver identifier name |
| Race | Race track name |
| Year | Season year |

Figure 6: Telemetry captured by the FastF1 data, and their description.

### B. *Data Constraints*

The FIA introduces regulation changes every few years that shape how cars must be designed. Some shifts are more significant than others, and fans often refer to seasons by their defining technical era such as the hybrid era, the V8 era, or the ground-effect era. For the purposes of this project, we will focus on cars within the same regulation window for consistency, specifically selecting every other year from 2019 to 2025 (2019, 2021, 2023, 2025).

Firstly, to reduce variance, scale, and limit anomalies, we focus on just five drivers. These drivers have been in F1 since at least 2019, providing consistency in driving style despite team changes and technological upgrades as they move between constructors in contractual agreements. Below, (Figure 7) shows the 5 drivers we've selected and their respective 2025 teams and driver numbers, though all these drivers are still present in the sport in 2026, albeit under different constructors and racing numbers.

| Sign | Driver | Constructor | Race # | Start year |
|---|---|---|---|---|
| VER | Max Verstappen | Red Bull Racing | 33 | 2015 |
| LEC | Charles Leclerc | Scuderia Ferrari | 16 | 2018 |
| HAM | Lewis Hamilton | Mercedes | 44 | 2007 |
| NOR | Lando Norris | Mclaren | 4 | 2019 |
| GAS | Pierre Gasly | Alpine | 10 | 2017 |

Figure 7: Drivers reviewed in project, team, driving number, and length of time in the sport.

Additionally, due to data limitations, we restrict our analysis to a set of staple circuits that have consistently appeared on the calendar and provide reliable, comparable conditions. Furthermore to ensure consistency across observations, we apply the following filtering criteria to the track selection:

- Tracks must appear on the race calendar for 2019, 2021, 2023, and 2025.
- Races must have been run in dry conditions (no rain), as wet races significantly distort lap times.
- Races must not have been canceled.
- All five drivers under study must have started the race.

We've found 3 circuits that fit this definition, namely the Bahrain Grand Prix (often a season opener), Mexican Grand Prix (typically mid-season), and Abu Dhabi Grand Prix (frequently the season finale). Each of these races covers approximately the same total distance around 305 km (set by the FIA) over a varying number of laps depending on circuit length. The details of the three tracks selected are below (Figure 8).

| Track (GP) | Circuit length | Total Laps | Full distance |
|---|---|---|---|
| Abu Dhabi | 5.281 km | 58 | 306.3 km |
| Mexico | 5.412 km | 57 | 308.2 km |
| Bahrain | 4.304 km | 71 | 305.4 km |

Figure 8: Three different tracks selected and details

## VII. Data Preprocessing

To focus our dataset on only the most relevant observations for modeling lap times, we remove rows that do not reflect representative race pace. This helps ensure that the model learns from "clean" performance data rather than distorted conditions. There are three main scenarios in which drivers are not running at full competitive speed: the start of the race, laps involving pit entry or exit, and laps completed under a Safety Car or virtual Safety Car period. During these phases, (Figure 9, and referenced Figures 10, 11, and 12 and 14), lap times are heavily influenced by external constraints rather than normal race conditions, which would introduce noise and bias into the model.

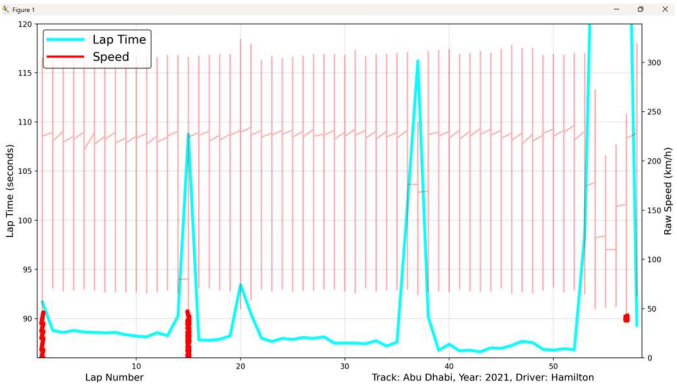


Figure 9: The raw race pace during Lewis Hamilton's Abu Dhabi 2021 race, showing speed and lap time over number of laps. The bolded red lines indicate speeds under 40 km*/h*

### A. *Race Starts*

At a race start drivers must begin from standstill and then accelerate rapidly to full speed while positioning themselves within the pack. This opening phase of the race is often some of the most defining moments as positions can be gained or lost quite quickly. However the first lap is not representative of a normal race pace since drivers cross finish line after launching from zero rather than completing a full "flying" lap where they carry speed into the start/ finish line (Figure 10).

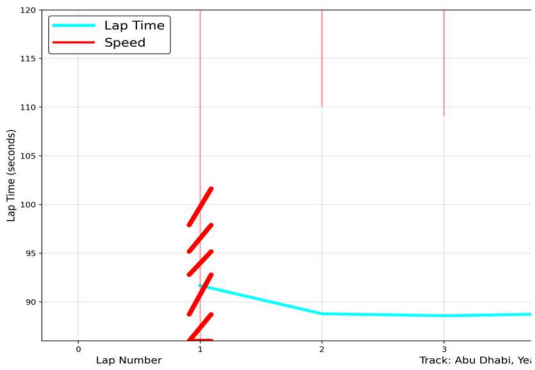

Figure 10: A zoomed in view of Figure 9, shows speeds starting from 0 at race start

## B. *Pit lane*

When drivers enter the pit lane (the area in which cars change heir tires) they are required by the FIA to hit a certain cruising speed of 80 kilometers an hour, this is done as a safety requirement to keep all drivers near active workers, TV crew, and those who are in the pit boxes safe. Cars that enter the pit lane over the 80 mile km/h regulation speed, even for a millisecond, will receive a penalty. Drivers often maximize this by slowing down at the last possible second to reach the pit lane max speed and then engage cruise control once entering the pit lane. Since the pit box (the specific marked area in the pit lane assigned to each team where their car stops for service) often overlaps the position of the start/finish line, means laps entered and laps exited from a pit lane do not represent race pace either. See (Figure 11) below where we can indicate that because the car comes to a complete stop means it is in the pitbox.

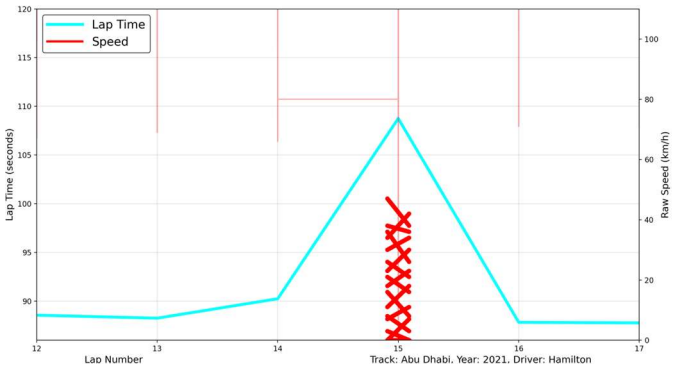

Figure 11: From a zoomed in view of Figure 9, shows the car slowing down and speeds fall to 0 as the car comes into the pits to change tires, then releasing back on to the field.

## C. *Defending or Attacking Position*

As the race progresses, drivers can interact with back markers (drivers who have been effectively lapped around the track) or are in competition with active drivers fighting for the current spot. When doing so, the focus of an ideal driving line isn't relevant anymore since there is now a competitor in front of you and that means lap times will most likely degrade as you battle it out amongst competitors. Perhaps you need to break early to make a turn faster than normal such that you can gain position, although this would be slower for achieving faster lap times, it's better for getting in front of your competitor. As shown in (Figure 12) the lap time slows down significantly on this lap, yet the overall speed drops only slightly. This pattern reveals an on-track traffic situation, specifically, Lewis Hamilton battling wheel-to-wheel with Sergio Perez.

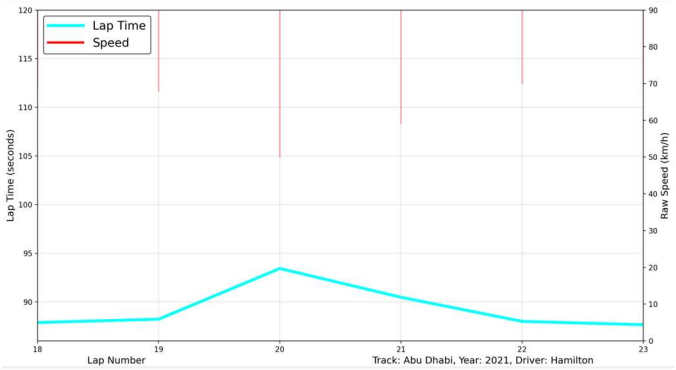

Figure 12: A zoomed in view of Figure 9, shows lap time decreases along with speed but only to a certain point meaning that there is an indication that the cars are battling or weaving through traffic.

## D. *Safety Cars*

As races do happen in a live setting, there are cases where accidents or crashes happen during the session and as such a Safety Car, or a car that leads the drivers away from danger, is deployed. Safety Cars are sent out during the race when there is an incident or accident on track. A driver in a designated Safety Car will leave the pit lane and head to the front of the pack of drivers and slow down the pace of the cars as they move around the track (Figure 13). This has done such that clean-up crews and marshals can brush debris off the parts of the race track where the cars are not at that moment.

The Safety Car is intended to have all the cars huddled up such that the FIA knows the position of all the cars on track and it is safe to get any and all work done quickly. Virtual Safety Cars are identical in that regard but they do not require an actual Safety Card to be on track, rather they tell all drivers to slow down to a certain speed, maintaining position, giving the marshals optimal amount of time to prep the track to racing again.. Now under our Safety Car or under a virtual Safety Car the drivers are limited in speed, it's not exactly a science as to what speed that is so long as they maintain behind the Safety Car. However that speed is slower than that which they would do without a Safety Car. In (Figure 14) you can see 2 Safety Cars are deployed at two independent race events. Lap times continue to be counted as the laps count down under a Safety Car. In fact, races have finished under the Safety Car on several occasions. This is typically frowned upon by fans, as overtaking is not permitted under Safety Car conditions. As a result, the race order is effectively locked, depriving viewers of any last-lap drama. However, Safety Cars can also boost excitement as they act as a mid-race restarts, bunching up the field and wiping out the pace advantage that leaders have built up.


Figure 13: Shows a Safety Car leading the race under an incident until the FIA determines it is safe to race again.

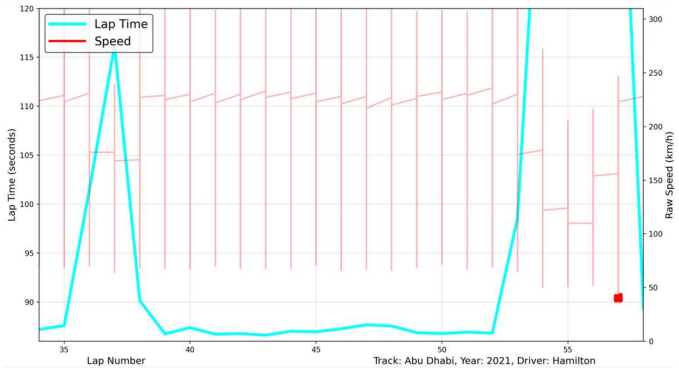


Figure 14: From another zoomed in view of Figure 9, indicates two instances where Safety Cars were deployed for two race events on track accidents between other drivers. See how the speed decreases and the lap time increases significantly over these times.

## VIII. DATA CLEANUP

In order to minimize the chances bad data could affect the accuracy of our models, we implemented a set of rules to clean the acquired data sets: (a) we removes telemetry events from laps 1 and 2, since these laps tend to be noisy due to the close proximity of racing cars off the starting line and the possibility of crashes and off-track excursions; (b) removed laps during and immediately after pitstops, since these add unpredictability due to the pit lane speed limits and the duration of the pit stop; (c) removed laps with times that were above 20% of the mean, which are related to on-track events such as yellow flags and Safety Cars which artificially increase the time a car takes to cross the lap marker; and (d) finally removed telemetry events from full laps where target feature "Lap Time" was missing. or additional evidence of our cleaning process see Figure 16.

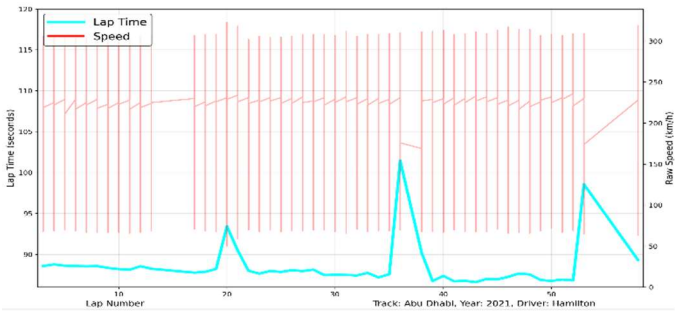


Figure 15: Shows the cleaned version of Figure 9.

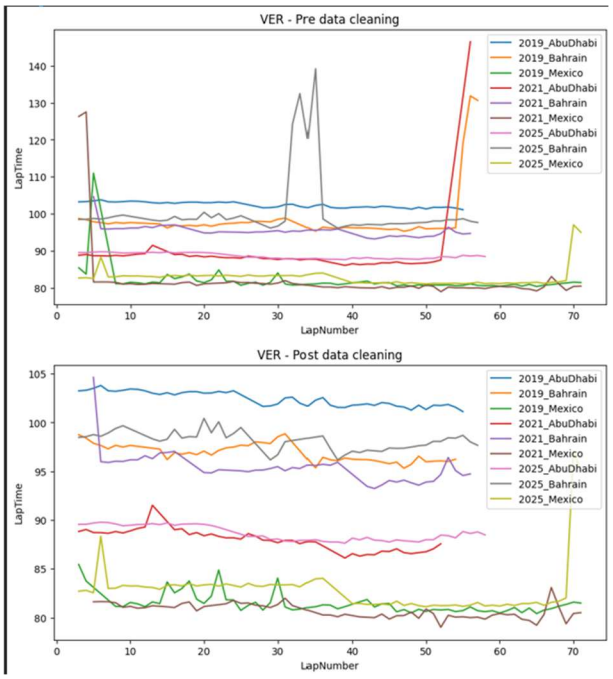


Figure 16: (top) Shows telemetry for VER (Max Verstappend) pre-clean up, where outliers and missing data can be spotted. (bottom) Post data cleanup graph, showing that extreme outliers and missing data were removed, while leaving telemetry for lap times within 20% of the mean time across all laps for a given track.

### *A. Preparing the Data to Train and Evaluate the Models.*

The data extracted from FastF1 was organized into CSV files per Driver, per Year, per Race, yielding a total of 60 files: 4 years {2019, 2021, 2023, 2025} x 3 races {Abu Dhabi, Bahrain, Mexico} x 5 drivers {VER, HAM, LEC, NOR, GAS} = 60 files. We then proceeded to data clean up as mentioned above.

Before training the LSTM-FCN driver model, we also performed additional data preparation by one-hot encoding of some features like "Brake state" and scaled other features using StandardScalar from the scikit-learn Python library, in particular ones like Speed, Throttle and Distance.

For the XGBoost driver model, the "Brake state" column was also encoded from True/False to 1/0. Normalization of numerical features was not needed since XGBoost is a tree-based method.

## IX. Proposed Models in this Research

In this section, we describe the purpose and detailed architecture of the three models developed for this research. The modeling pipeline consists of (1) a pace degradation linear regression model that generates lap time predictions based on driver-agnostic trends with the intent of supplementing the driver-specific models, and (2) two complementary lap-time prediction models: a Deep Learning sequence model (LSTM-FCN) and a gradient-boosted tree model (XGBoost). The pace degradation model is used upstream to enrich the feature set for both predictors, while the LSTM-FCN and XGBoost models serve as competing approaches to forecast lap times, allowing us to compare the strengths of sequential Deep Learning versus tree-based ensemble methods on the same task.

## X. *Model for Pace Degradation*

The *pace* of a race car refers to its ability to produce fast lap times relative to competitors. Some factors affecting pace, such as power unit performance and driver skill, remain stable throughout a race, while others like fuel level and tire condition evolve over the course of a race. As fuel is burned throughout a race, the car becomes lighter and faster. Conversely, tire rubber wears down with every lap, causing the car to lose pace. A driver can put on fresh tires mid-race (and must do so at least once per race) but at the cost of a 20-25 second pit stop.

Drivers choose from three dry-weather compounds: soft, medium, and hard. As mentioned above, soft tires are faster but degrade more easily, hard tires are slower but last longer, and medium tires lay somewhere in the middle. Optimizing tire strategy is central to race strategy, but isolating tire effects on pace from fuel effects remains difficult, even with data analysis and machine learning.

### A. *Initial Analysis*

Analyzing average lap times over the course of a tire stint (period a driver runs on the same set of tires before pitting for a new set) shows a downward trend in lap time. Without additional context, the negative correlation between tire age and lap time would suggest cars get faster as tires wear down. However, since we know tire wear should slow the car down, this indicates the trend is dominated by fuel burn-off.

Fuel burn rate, along with its consequential effect on pace, is found to be essentially linear throughout a race, as discussed by Boettinger & Klotz [2] and adapted by Heilmeier et al. [3]. While tire degradation is more dynamic than fuel burn, Heilmeier et al. [3] also found that a linear model was robust in consistently capturing tire degradation rates.

To capture fuel burn, we treated "*total laps completed*" as a fuel factor, since refueling is not permitted and fuel weight decreases consistently. To capture tire effect, we treated "*laps completed on current tire*" as our tire degradation factor. In order to create a driver-agnostic lap time target, we defined driver-race-normalized lap time as:

$$l_N = l_C - med(L_{DR}) \quad (1)$$

where $l_C$ is the current lap time and $L_{DR}$ is that driver's laps in the race.

We then split data by track and tire compound and fit the two features to normalized lap time using linear regression, since our research found both effects to be linearly representable. Prior to modeling, we removed outlier lap times, pit laps, first laps, laps under Safety Car, and laps from unusually short stints.

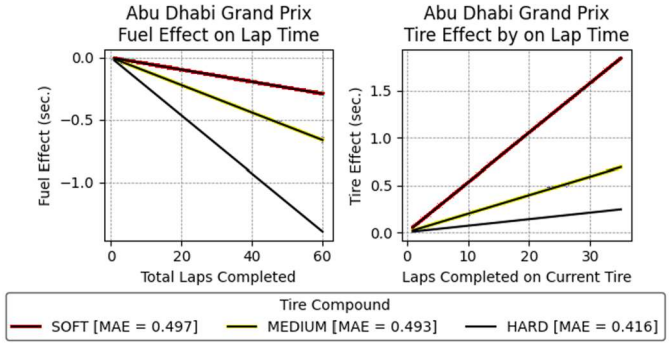


Figure 17: Fuel and Tire Effects throughout the Abu Dhabi Grand Prix.

For any linear fit on a specific track and tire compound, we get the following predictive equation for normalized lap time:

$$l_N = b - (E_F * L_T) + (E_T * L_C) \quad (2)$$

where b is the bias, $E_F$ is the fuel effect, $L_T$ is total laps completed, $E_F$ is tire effect, and $L_C$ is total laps on current tire. In racing terms, this means that a driver is $E_F$ seconds faster each lap from fuel burn, but $E_T$ seconds slower for each lap on the same tires due to wear.

Our initial regression model results are very promising. Across soft, medium, and hard tires at Abu Dhabi, Mexico, and Bahrain, Mean Absolute Error (MAE) ranged from *0.336* to *0.774* seconds.

### B. *Model Construction*

Modeling normalized lap time was great for analytical purposes. It helped us understand the roles of tire wear and fuel burn in pace progression throughout a race. However, for the sake of predicting a driver's next lap time, normalized lap time as a target poses a data leakage issue. Normalized lap time is relative to average lap time of a session, which is not known until the end of the race. If we seek to build a model that predicts the next lap time during a race, we'll need a different target.

Predicting raw lap times would eliminate the data leakage issue but would also nullify the driver-agnosticism of the Pace Degradation model, as every car and driver has a different

baseline lap time. The most logical choice of target for our model is delta (Δ), or change in lap time:

$$\Delta_n = LapTime_n - LapTime_{n-1} \quad (3)$$

Lap-to-lap changes in delta are more consistent across drivers than absolute lap times. Predicting delta is also very interoperable with lap time predictions. To predict the next lap time, we simply add our predicted delta to the most recent lap time. However, as we shift targets, we will have to adjust our model accordingly. Previously, "total laps" was used as a feature to capture fuel effect. Now that we're modeling for delta, the fuel effect, along with any other pace factors with a constant rate of influence, will be captured by the bias of a linear model. This justifies the elimination of total laps from the feature set, leaving us with a singular feature: StintLap (laps on current tire). Our Pace Degradation model looks like so:

$$\hat{\Delta} = B + (W * StintLap) \quad (4)$$

Our models' bias (B) should capture pace effect from fuel and any other constant factors, while weight (W) should capture the effect of tire wear on pace. To train our Pace Degradation models, we pulled all F1 lap times from our selected tracks, drivers and years, along with each lap's associated Track, Compound, and StintLap. Using each lap's current and previous lap time, we calculated delta. We then split the data by track and tire compound. Prior to training any models, we filtered out any laps with excess noise. Once the noise was removed, we trained a linear regression model for each track/tire pair, reserving 20% of data from each subset to test our models.

### C. *Model Results*

The linear Pace Degradation models performed rather well, with test set MAE ranging from 0.299 to 0.395 seconds across all models. All models pertaining to Abu Dhabi and Mexico City behaved as expected; a negative bias that models constant pace gain due to fuel loss, and a positive coefficient that models progressive pace loss due to tire wear. The Bahrain models behaved differently. The soft, medium, and hard tire models for Bahrain resulted in respective biases of +0.026, +0.092, +0.098 seconds and respective coefficients of -0.001, -0.003, -0.003 seconds. This modeling anomaly is likely driven by Bahrain's unique environmental conditions. Bahrain has an extremely hot climate, resulting in excessive track temperatures that decrease tire grip quickly. While Bahrain is hot, the race itself takes place at sun set, meaning the high track temperatures progressively cool down into night, potentially increasing tire grip later in the race. This anomaly, while genuinely reflective of the track's nature, still poses a limitation of the model. For unusually long tire stints, the Bahrain models' predictions tell the story "extreme tire wear = increase in pace" which we know is not the case.

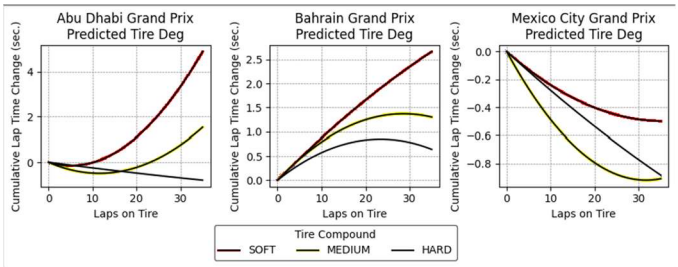


Figure 18: Predicted lap time progression throughout a tire stint according to the Pace Degradation model. While the delta-predicting model is linear, the cumulative lap time changes appear parabolic due to the integration of lap-to-lap deltas.

To test a more realistic application of the Pace Degradation model, we also used the model to predict next lap times for all clean laps in 2024 for our chosen drivers and tracks. We predicted next lap time by applying the following formulas:

$$\widehat{LapTime}_{n+1} = LapTime_n + \widehat{\Delta_{n+1}} \quad (5)$$

Where $\widehat{\Delta_{n+1}} = B_{(Tr,Ti)} + [W_{(Tr,Ti)} * (StintLap + 1)]$

And $B_{(Tr,Ti)}$, $W_{(Tr,Ti)}$ = bias and weight of Pace Degradation model corresponding to current track and tire compound

The results were very encouraging. The predictions resulted in race-wide MAEs ranging from 0.144 to 0.440 seconds and a grand MAE of 0.284 seconds. The model, while very simple, is relatively effective and robust on its own and could potentially aid driver-specific models with its driver-agnostic methodology.

## XI. *LSTM-FCN DEEP LEARNING MODEL*

Faruk and Küçükmanisa[4] in 2025, proposed an LSTM network with three layers of 128, 64, and 32 units respectively, followed by two fully connected (i.e. dense) layers of 128 and 64 units, and an output of 1 unit, which represented the lap time prediction. The network was trained by using a fixed-length window of multivariate previous laps to predict the target. This model was trained with the Adam optimizer, mini batch of 32 and a learning rate 0.001 for 200 epochs, showing a MAE of 1.4 seconds and proving that the model "can reliably track lap time trends and deliver accurate estimates under stable race conditions".

Other previous attempts at this problem can be found in Brusik [5] in 2024, who attempted to compare different Deep Learning models, including an LSTM-FCN, for univariate and multivariate time series to predict lap times. Brusik found that LSTM-FCN were suitable for the task, providing a MAE of 3479 (or 3.479 seconds since Brusik reported their numbers in milliseconds) but training the networks for under 20 epochs.

In this work, we propose the use of a variation of the LSTM network used by Faruk and Küçükmanisa, a Long-Short Term Memory + Fully Convolutional Network (or LSTM-FCN) initially proposed by Karim, Majumdar, Darabi and Chen [4] in 2017 to attempt to improve the lap time predictions of the previous work.

According to the work by Karim, et.al., LSTM-FCNs have shown strong performance in capturing complex temporal dynamics within time series datasets because they combine LSTMs and their ability to store temporal relationships from distant time steps, and the convolutional network ability to extract local patterns from closely related time steps.

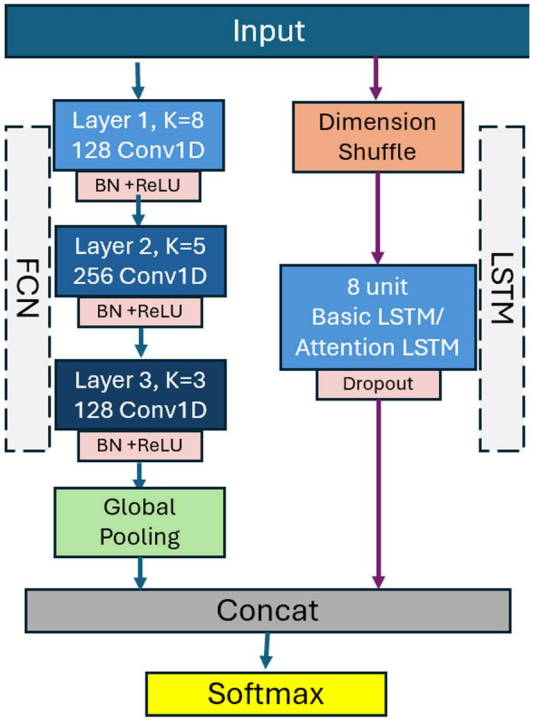


Figure 19: Architecture of the LSTM-FCN model.

### A. Model Architecture

An LSTM-FCN is an architecture that combines 2 networks in one, a Temporal Convolutional Network and a Long-Short Term Memory Recurrent Neural Network, or LSTM RNN.

### B. Fully Convolutional Network (FCN)

The Temporal Convolutional Network is composed of 3 blocks of 1D Convolutions (Conv1D), each one followed by a *Batch Normalization* layer and an activation unit, which can be a Rectified Linear Unit (*ReLU*). At the end of these 3 convolutional blocks, there is a Global Pooling layer, which is used to reduce the dimensionality of the convolutions.

Each Conv1D block consists of a series of filters which are applied in a sliding window of size *kernel_size* over the input feature vector. Each filter in the block generally learns, or extracts, patterns from the data across and within each input feature:

$$A_{i,t}^{(l)} = ReLU\left(b_i^{(l)} + \sum_{t'=1}^{d} \left\langle W_{i,t'}^{(l)}, A_{t+d-t'}^{(l-1)} \right\rangle\right) \quad (6)$$

The output of a layer $l$ in a Conv1D is given by the activations of the previous layers and previous timesteps, where b is the bias vector for layer $l$, $d$ is the filter size (kernel) and $t'$ is the timestep based on the filter size (effect of sliding a window of size kernel over the length of features in the current timestep), then ReLU is applied for the final activation.

### C. Long-Short Term Memory (LSTM)

An LSTM [6] network is a kind of Recurrent Neural Network (RNN) which introduces additional complexity to minimize some drawbacks from the RNNs, such as vanishing gradients. RNNs use hidden states to store information about previous timesteps to enhance predictions for current timesteps, from layer to layer. The LSTM enhances this mechanism by introducing *memory cells* which are composed of similar hidden states but controlled by *gates: input, output and forget.* The *forget* gate was later introduced by Gers, Schmidhuber and Cummins [7] in 2000.

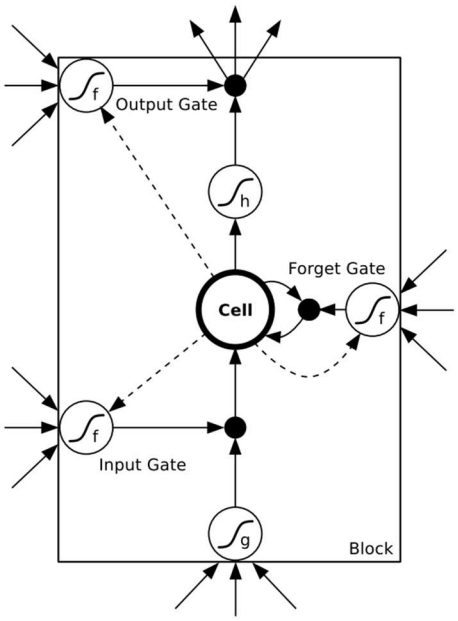


Figure 20: Diagram of an LSTM memory cell[8].

Each gate in a memory cell is a multiplicative operation between the input (input gate) and outputs (output gate) of the cell, and the previous state of the cell (forget gate). The $f$ functions are logistic sigmoids (between 0 and 1) which determine if the gate is open or closed, or in other words, if the gate influences the input or outputs to the unit for the given timestep. Notice how each gate in the memory unit contains three inputs, these correspond to the input features, output of the cell for the previous timestep and the previous state of the cell. The $g$ and $h$ functions could be a logistic sigmoid or a tanh function. Equation 7 shows the interaction between inputs are gates in a memory cell.

$$
\begin{aligned}
g^u &= \sigma(W^u h_{t-1} + I^u x_t) \\
g^f &= \sigma(W^f h_{t-1} + I^f x_t) \\
g^o &= \sigma(W^o h_{t-1} + I^o x_t) \\
g^c &= \sigma(W^c h_{t-1} + I^c x_t) \\
m_t &= g^f \bigodot m_{t-1} + g^u \bigodot g^c \\
h_t &= \tanh\left(g^o \bigodot m_t\right)
\end{aligned}
\tag{7}
$$

In the above equation, subscripts u, f, o and c correspond to the calculations of gates in the memory cell for input (u), forget (f), output (o), intra cell (c), while $m_t$ represent the internal memory vector of the cell and $h_t$ is the final hidden state of the cell.

### D. Model Construction

For this work, we implemented an LSTM-FCN network composed of an LSTM layer (with 8 memory units), followed by a Dropout (with 0.2 rate) layer. The Dropout layer is used for training, to avoid overfitting.

For the Fully Convolutional Netowrk, we used three convolutional layers (Conv1D) with the following filter and kernel sizes (f, k) of (128, 8), (256, 5) and (128, 3) respectively, followed by a ReLU activation function. Additionally, and only for training, we added a batch normalization step with momentum of 0.99 and epsilon of 0.001. As a final step, the output of the convolutional blocks are reduced with a Global Pooling layer.

The output of both the FCN and the LSTM networks are concatenated and the result is then forwarded to a Dense layer with one output. In the original paper [4], Karim, *et al.* used the network for classification tasks (hence the Softmax layer at the output in Figure 19). In our case, we want to predict a continuous output, so we are just using a Dense layer with one output unit at the end.

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| input_layer_12 (InputLayer) | (None, 1, 102) | 0 | – |
| sequential_14 (Sequential) | (None, 8) | 3,552 | input_layer_12[0… |
| sequential_13 (Sequential) | (None, 128) | 265,728 | input_layer_12[0… |
| concatenate_2 (Concatenate) | (None, 136) | 0 | sequential_14[0]… sequential_13[0]… |
| dense_2 (Dense) | (None, 1) | 137 | concatenate_2[0]… |

Figure 21: Sample Keras summary of a LSTM-FCN model, with the pace degradation feature and a aggregation window of 20 events (101 input features). The total number of parameters to the network is 806,205.

We also implemented a variation of the windowing in the data preparation used in Karim, *et al.* in which instead of windowing previous lap times as inputs to the LSTM-FCN, we windowed telemetry events for the same lap, providing the network with more data to project predictions of the lap time. Recall from section VII (Data Cleanup), the data we collected for this project (for a given driver, race and year) consisted in *n* telemetry events per lap, taken at roughly ~275 milliseconds. We preprocessed this data and transformed each input sample at timestep *T* into an augmented sample consisting of $W_T$ =20 consecutive telemetry events, representing a short segment of a lap (~275milliseconds * WT(20) = 5.5 seconds), making the augmented sample a combination of windowed telemetry-level events (such as speed, throttle, break on/off, etc) and lap-level events up to time T (such as distance since start of lap, pace degradation projections, lap number and, most importantly, the target lap time).

### E. Train/Test/Validation Split

Once the data was prepared, we split it into Train/Test/Validation datasets with a proportion of 0.8, 0.15, 0.05 respectively. This split was done on full laps per driver per race per year, as such was the organization of the data sets. For example, the file containing the data for VER/Abu Dhabi/2021 contains 16,700 telemetry events for 52 laps after data clean up, which corresponds to ~320 telemetry events per lap. This dataset yielded 41 laps for training, 8 laps for testing and 3 laps for validation.

### F. LSTM-FCN Results

For the LSTM-FCN model we ended up defining three different experiments:

Experiment #1**:** Training the LSTM-FCN model with all telemetry events {Distance, Speed, RPM, TimeSec, LapNumber, Throttle, Brake_True, Brake_False} plus the prediction from the Pace Degradation model, as PaceDeg.

Experiment #2: All the features mentioned in Experiment #1 except we removed the PaceDeg feature. We wanted to explore the impact of the feature in the accuracy of the prediction of the lap time.

Experiment #3**:** In order to establish a baseline, we designed an experiment with minimal features, with only Speed and distance as the relevant feature: {Distance, Speed, TimeSec, LapNumber}. If this experiment performed closely to the other two, then this would present a hypothesis where accurately predicting a lap time in Formula One might be a matter of only a few engineered features.

| | Experiment #1 | Experiment #2 | Experiment #3 |
|---|---|---|---|
| **Min** | 0.5990 | 0.6020 | 0.7652 |
| **Max** | 7.9378 | 7.1013 | 8.4558 |
| **Avg** | 1.8537 | 1.8130 | 2.9919 |

Figure 22: min, max and average of the mean absolute error across experiments in all races for all drivers.

As noted in (Figure 22), using Pace Degradation as an input feature (Experiment #1) provided a slight advantage to not using it (Experiment #2) but the differences are extremely close (in all categories). Experiment #3 resulted in the largest MAE across all experiments, calling it that using more input features from the telemetry might be given a more complete story.

The numbers from Experiments #1 and #2, were close to those found in Faruk and Küçükmanisa [4], which proves the models can get close to an accurate prediction. However, the standard deviation of the MAE for each experiment is quite large, as can be seen in Figure 23.

| | Experiment #1 | Experiment #2 | Experiment #3 |
|---|---|---|---|
| **AVG** | 1.860 | 1.838 | 2.973 |
| **STDDEV** | 1.587 | 1.423 | 2.3363 |

Figure 23: Average MAE and standard deviation for each LSTM experiment.

Other examples can be seen in (Figure 24), where we show the MAE for Verstappen (VER) in Abu Dhabi for Experiment #1 (20260422_PaceDeg_2). The error is not stable throughout the race, exhibiting some spikes between laps 30 and 45 which are not in the original data. Compared with Figure 25, we show Experiment #3 which exhibits similar patterns but shows higher and more pronounced and longer spikes in the plotted error.

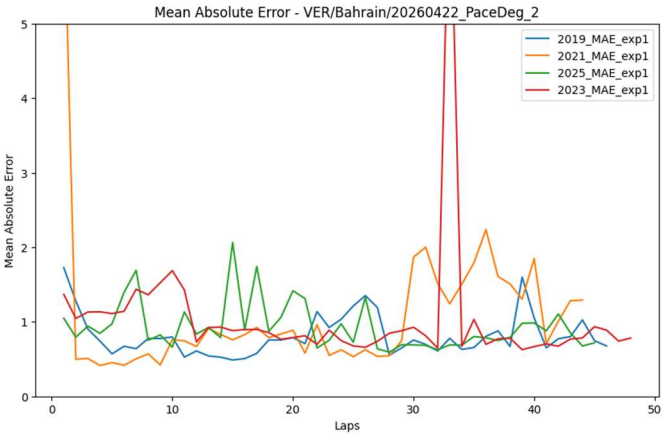


Figure 24: Mean absolute error for VER in Abu Dhabi for Experiment #1.

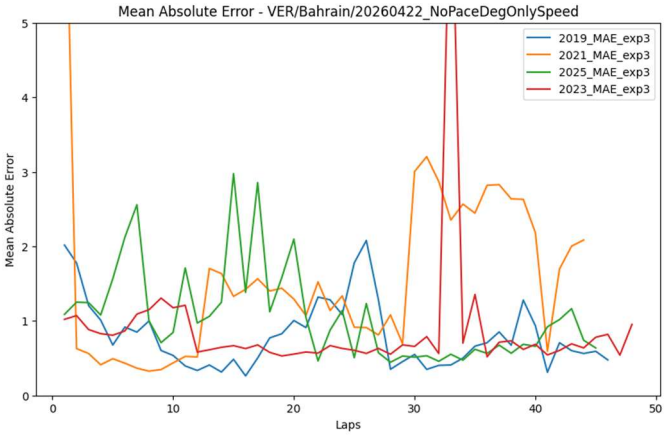


Figure 25: Same plot but for Experiment #3 (only using Speed as the most relevant feature).

More work is needed to understand how to improve the LSTM-FCN models for this prediction task. Training these models is time consuming, taking at least 4 hours to train one model for a given driver when using 200 epochs. This is prohibitively expensive for the scope of this project and limited the amount of hyperparameter tuning we could perform. At the beginning of the project, during the initial exploratory phase, we found that the mean absolute error remained too high (at around 7) when training the models for less than 200 epochs.

We are confident that with more time to perform stricter feature engineering and hyperparameter tuning we could produce much better results.

## XII. *PREDICTING LAP TIMES WITH XGBOOST.*

XGBoost is a tree-based ensemble learning method. The model consists of an additive sequence of decision trees, where the first tree predicts the target and every tree following corrects the residual errors of all prior trees.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \quad (8)$$

Where K = number of trees, $x_i$ = feature values.

Each subsequent tree seeks to minimize a loss term, along with a complexity penalty if regularization terms are added. XGBoost differs from traditional gradient-boosting models by using a loss term that accounts for the first-order and second-order gradient information (i.e. Gradient and Hessian) [11].

$$L_t = \sum_i \left[ g_i f_t(x_i) + \frac{1}{2} h_i {f_t}^2(x_i) \right] + \Omega(f_t) \quad (9)$$

Where $g_i$ and $h_i$ are first and second-order gradients, respectively, and $\Omega$ is the regularization term.

XGBoost is highly popular among data scientists due to its high performance and ability to capture complex relationships, which is why we chose this method to build our statistical driver model. Telemetry data such as braking patterns, gear shifts, and

RPM progression may have complex relationships with resulting lap times, and a powerful model like XGBoost is likely needed to extract those patterns.

### A. Model Construction

Each driver and track pair gets their own XGBoost model, as both every driver and every track possesses unique performance patterns. In this case, given our five drivers and three tracks, we will have 15 XGBoost models. For each model, we gathered all telemetry from 2019, 2021, 2023, and 2025, removed all noise/outliers, and aggregated all telemetry events by lap to get the following descriptive metrics of each lap:

- Avg. RPM
- Max RPM
- Avg. Speed
- Max Speed
- Min. Speed
- Speed StDev.
- Avg. Throttle
- High Gear %
- Throttle StDev.
- Full Throttle %
- Braking %
- Lap Delta

Each of these metrics are used as a feature in our model. To give our model some performance insights on recent laps as well as the current lap, we also added "lagged features" for each lap, which provide each record with data from the previous two laps. For lap delta, we added an additional "Lag2Delta" column that represents the difference between current lap time and two laps ago. For top speed, full throttle %, and brake %, we added "Lag1", "Lag2", "Lag1Delta", and "Lag2Delta" columns which represent the selected metrics from the previous two laps and their differences from the current lap's same metric. We only included lagged features for records where the previous corresponding laps were clean laps. This results in records with null values, but XGBoost is notably robust in handling them. Our target variable is next lap's delta, which we add to our most recent lap time to predict the next lap time. This approach is similar to the Pace Degradation model, which maximizes the efficiency of comparing and ensembling the two models.

We hyperparameter-tuned each of the 15 driver models to maximize performance. For every model, we held out 20% of laps for testing and ran a 4-fold cross-validated randomized search with 25 iterations across the hyperparameter space.

| Hyperparameter | Definition | Search Space |
|---|---|---|
| max_depth | Max. depth of each tree | 3, 5, 7 |
| min_child_weight | Min. instances required in leaf node | 1, 3, 5 |
| subsample | % of records per tree | 0.6, 0.8, 1.0 |
| colsample_bytree | % of features per tree | 0.6, 0.8, 1.0 |
| learning_rate | Step size for updates | .01, .05, .10 |
| n_estimators | Total boosting rounds | 100,300,500 |
| gamma | Min. loss reduction for splits | 0.0, 0.1, 0.3 |
| reg_alpha | L1 regularization term | 0.0, 0.1, 1.0 |
| reg_lambda | L2 regularization | 1, 5, 10 |

Figure 26: Table of XGBoost model hyperparameters.

### B. Results

The resulting 15 driver models achieved test set MAEs ranging from 0.220 to 0.487 seconds. When examining average feature importance across all models, our lap time delta and our delta from two laps ago were the two most prominent features, which was expected. The third, fourth, and fifth most important features were not as expected. The third and fourth most important features were RPM max and RPM mean. While we did not expect RPM to be more influential than metrics like speed, braking, and throttle, the result is reasonable in hindsight, as RPM can provide insights on how aggressively the car is being driven. The fifth most important feature, on average, was speed standard deviation. We expected average speed and top speed to have more influence on the model but understand how speed standard deviation could bear the most weight, since the variation in speed helps quantify the driver's overall efficiency.

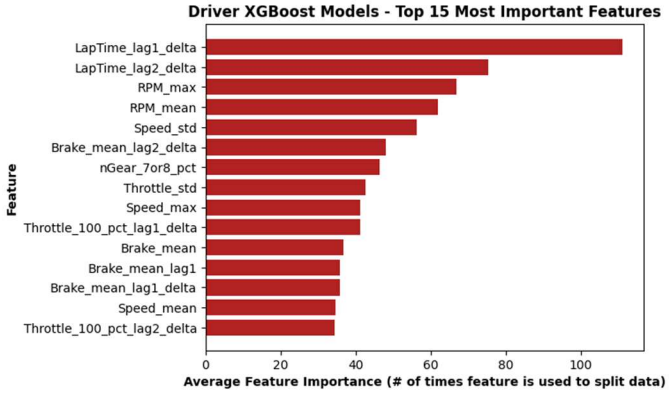


Figure 27: Bar chart of 15 most important features, on average, across all 15 XGBoost driver models.

Similarly to the Pace Degradation model, we tested the XGBoost model on clean laps from 2024, to allow for direct comparison to Pace Degradation model performance and to evaluate the model's ability to predict on race years that had no presence in the training data. The predictions resulted in race-wide MAEs ranging from 0.179 to 0.389 seconds and a grand MAE of 0.289 seconds. While we believe there is still opportunity for improvement by experimenting with different feature sets and hyperparameter spaces, these results are very encouraging.

### C. Ensembling with Pace Degradation

Two efforts were made to ensemble the XGBoost model with the Pace Degradation model. The first attempt was by adding the Pace Degradation predictions to the XGBoost feature set. This methodology was tested on a subset of 3 driver-track combinations (Verstappen - Abu Dhabi, Gasly - Bahrain, Leclerc - Mexico City). Two of the ensembled models resulted in a 0.001 second increase in MAE, while the third resulted in

a 0.007 second decrease, suggesting virtually no impact on performance. The second attempt was by fitting a linear regression meta-learner to assign optimal weights to each model's prediction. The resulting equation:

$$\hat{y}_{Meta} = 1.044\hat{y}_{XGBoost} - 0.044\hat{y}_{PaceDeg} \quad (10)$$

Despite both models performing very similarly, the meta learner gave all weight to the XGBoost models and then some. The lack of impact from ensembling the two models suggest that, despite having completely different architectures, the XGBoost models and Pace Degradation models are capturing the same signals. The Pace Degradation models are learning the same performance trends from tire age that the XGBoost models are learning from telemetry indicators.

## XIII. *Results*

In this work, we set out to compare Statistical (non-Deep Learning) vs Deep Leaning models for the task of predicting Formula One Lap times for a subset of drivers, races and years. The statistical models used in this work, and described in previous sections were: XGBoost and the Pace Degradation model. The Deep Learning model was an LSTM-FCN trained in 3 different configurations: With Pace Degradation (LSTM-PaceDeg/Experiment #1), without Pace Degradation (LSTM / Experiment #2) and the last one, trained with minimal features (LSTM-Base / Experiment #3). The results are shown in Figure 28.

| Driver | Race | XG-BOOST | PACE DEG | LSTM + PaceDeg | LSTM-FCN | LSTM-FCN-Base |
|---|---|---|---|---|---|---|
| HAM | Abu Dhabi | 0.298 | 0.288 | 2.503 | 3.021 | 3.838 |
| | Bahrain | 0.308 | 0.310 | 1.915 | 1.796 | 7.184 |
| | Mexico City | 0.387 | 0.405 | 2.041 | 2.028 | 2.218 |
| VER | Abu Dhabi | 0.278 | 0.266 | 2.428 | 2.623 | 2.332 |
| | Bahrain | 0.269 | 0.271 | 1.046 | 0.778 | 0.889 |
| | Mexico City | 0.221 | 0.248 | 1.352 | 1.233 | 1.037 |
| ALL* | MEAN | 0.289 | 0.284 | 1.854 | 1.813 | 2.992 |
| | MIN | 0.179 | 0.144 | 0.599 | 0.602 | 0.765 |
| | MAX | 0.389 | 0.440 | 7.9378 | 7.1013 | 8.4558 |

Figure 28: Experimental results (mean absolute error) across all models trained, for a subset of drivers and races. The table also shows the min, max, and average across all models. It is worth noting that the numbers in the table are from a year were none of the models were trained on (2024). *Summary statistics also account for sessions not included in the table

The statistical models demonstrated stronger performance in the task of predicting Formula One lap times. The statistical model presented a more compact error distribution (min, max, avg) than the LSTM models, which struggled with outliers and large prediction errors in some instances. This is a surprising result, given the large complexity of the LSTMs which should be able to capture more nuanced patterns and relationships between features. Future work should include narrowing down to a smaller set of drivers and races and a more detailed data preparation process to see if the source of the discrepancy is surmountable.

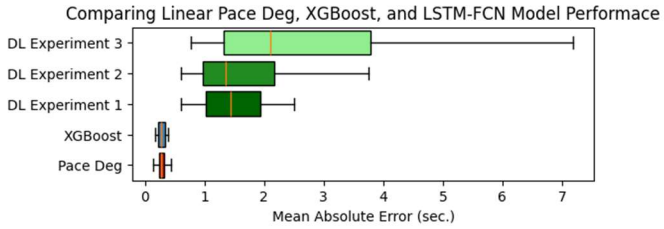


Figure 29: Box-and-whisker plot displaying different model architectures' MAE distributions across different sessions

The two statistical-based models performed very similarly across all 2024 sessions. The overall MAE for XGBoost (0.289) was within 0.005 seconds of the Pace Degradation model (0.284), indicating nearly identical performance. At the session level, the models also tracked closely. XGBoost and PaceDeg MAEs were within 0.03 seconds in 12 of 15 sessions, and within 0.01 seconds in 7 of 15 sessions. The consistency is further supported by a strong Pearson correlation of 0.912 between the models' session-level MAEs, suggesting both model architectures' performance are influenced by similar conditions. This finding reinforces the conclusion we reached when we saw no impacts from ensembling PaceDeg and XGBoost; both models are capturing the same signals, despite having very different methodologies.

## XIV. *Novel Ideas*

The main objective of this work was to directly compare the performance of statistical models (non-DL) vs Deep Learning models in the task of predicting Formula One lap times. We present in this work several new factors, such as the introduction of a Pace Degradation model which predicts the lap time deltas from one lap to the next using only tire compound and laps-on-tire. While the Pace Degradation model was not as effective in supporting our driver models as we had hoped, the linear model's performance on its own was very promising. The Pace Degradation model's performance shows us that, while telemetry is extremely useful, it isn't completely necessary to predict changes in lap time. This finding can be very inspiring for low-level racing teams outside of F1 that lack funding for sophisticated telemetry systems, or for racing teams that are trying to forecast their opponents' performance without access to their telemetry to gain a strategic edge.

We also tested the inclusion of the per-lap deltas in the LSTM-FCN as an input feature to the model, with the hopes that feature would boost performance of the predictions. Another novel idea was to use the full per-lap telemetry data

aggregated in a sliding window, to provide the LSTM-FCN model with more data during training.

## XV. *Future Work and Limitations*

There are several ideas that came up during the experiments that would be interesting to test. For both the XGBoost and LSTM-FCN models, more feature engineering and hyperparameter tuning would be ideal to decrease errors and make predictions more stable. For XGBoost specifically, experimenting with different combinations of lagged features could potentially improve model performance.

In terms of data, Formula One lap times are extremely susceptible to on-track events that can add wild swings to the final timings. Events such as crashes, yellow flags, Safety Cars (virtual or otherwise) and off-track excursions, can have large effects in the time a given car crosses the lap threshold. It would be quite interesting to explore the possibility of using this information in a model ensemble to attempt to incorporate it into the prediction pipeline and still produce accurate timing predictions. Incorporating variable weather and expanding the race and driver catalog, would be fine additions too.

The LSTM-FCN models were hard to train. The base LSTM-FCN model for any given driver, using 3 years to train data, took around 4 hours to produce. Training a total of 18 models (3 experiments, 2 drivers, 3 years) took around 72 hours in total, which doesn't leave much room for subtle bugs or tweaking hyper-parameters. Maybe with more compute power and more parallelization we could have found better outcomes.

In addition to continued optimization of our lap time prediction models, another way we could expand our project is by moving beyond the goal of predicting the next lap time in a race. By recursively applying our models, we could make rest-of-race lap time predictions (the Pace Degradation model would be very helpful for this purpose, as no recent telemetry is needed for its predictions). We could then grow beyond rest-of-race predictions by simulating different pit strategies (i.e. which lap to complete a pit stop and which type of tires to switch to) and predicting the optimal strategy. Building a model that predicts the next lap time won't win a Formula One driver or constructor a championship on its own, but it serves as a potential foundation for far greater, more impactful projects.

## *XVI. References*


1. Nilsson, O. (2024). FastF1 (Version 3.4.0) [Software]. https://github.com/theOehrly/Fast-F1.

2. Boettinger, M., & Klotz, D. (2023). Mastering Nordschleife - A comprehensive race simulation for AI strategy decision-making in motorsports. arXiv. https://doi.org/10.48550/arXiv.2306.16088https://www.google.com/search?q=https://doi.org/10.48550/arXiv.2306.16088

3. Heilmeier, A., Wischnewski, A., Hermansdorfer, L., Betz, J., & Lienkamp, M. (2020). Virtual strategy engineer: Using artificial neural networks for making race strategy decisions in circuit motorsport. Semantic Scholar. https://www.semanticscholar.org/paper/Virtual-Strategy-Engineer%3A-Using-Artificial-Neural-Heilmeier-Wischnewski/52396bd331ce9b73cd7bc0139f9c679a0df76a7fhttps://www.google.com/search?q=https://www.semanticscholar.org/paper/Virtual-Strategy-Engineer%253A-Using-Artificial-Neural-Heilmeier-Wischnewski/52396bd331ce9b73cd7bc0139f9c679a0df76a7f

4. Kara, M. F., & Küçükmanisa, A. (2024). Predicting lap times in Formula 1: A deep learning approach using LSTM networks. In 2024 International Conference on Computer Science and Engineering (ICCP). IEEE Xplore.

5. Brusik, F. (2024). Deep neural networks for pit stop strategy in Formula 1 [Master's thesis, Tilburg University]. Tilburg University Research Portal. https://arno.uvt.nl/show.cgi?fid=180319https://arno.uvt.nl/show.cgi?fid=180319

6. Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2017). LSTM fully convolutional networks for time series classification. IEEE Access, 6, 1662–1669. https://doi.org/10.1109/ACCESS.2017.2779939https://doi.org/10.1109/ACCESS.2017.2779939

7. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735https://doi.org/10.1162/neco.1997.9.8.1735

8. Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural Computation, 12(10), 2451–2471. https://doi.org/10.1162/089976600300015015https://doi.org/10.1162/089976600300015015

9. Graves, A. (2012). Supervised sequence labelling. In Supervised sequence labelling with recurrent neural networks (pp. 5–13). Springer. https://doi.org/10.1007/978-3-642-24797-2_2https://doi.org/10.1007/978-3-642-24797-2_2

10. Maragkos, N., & Refanidis, I. (2025). A comparative evaluation of time-series forecasting models for energy datasets. Computers, 14(7), 246. https://doi.org/10.3390/computers14070246https://doi.org/10.3390/computers14070246

11. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. arXiv. https://doi.org/10.48550/arXiv.1603.02754