F1 TELEMETRY GAME DATA ANALYSIS

Submitted In Partial Fulfilment of Requirements

For the Degree Of

Honours in Data Science and Analytics

(Offered by Department of Computer Engineering)

By

Esha Shelar

Roll No: 16010120136

Husain Bengali

Roll No: 16010120140

Bharat Mahesh

Roll No: 16010120141

Yashraj Deshmukh

Roll No: 16010120142

Guide

Prof. Bharathi HN

Somaiya Vidyavihar UniversityVidyavihar, Mumabi - 400 0772020-24

Somaiya Vidyavihar University

K. J. Somaiya College of Engineering

Certificate

This is to certify that the dissertation report entitled F1 Telemetry Game Data Analysis submitted by Esha Shelar, Husain Bengali, Bharat Mahesh, Yashraj Deshmukh at the end of semester VIII of LY B. Tech is a bona fide record for partial fulfilment of requirements for the degree Honours in Data Science and Analytics (**Offered by Department of Computer Engineering**) of Somaiya Vidyavihar University

_____ _____

Guide Head of the Department

_____

Examiner

Date: 28-02-2024 Place: Mumbai-77

Abstract:

Table of Content:

List of Figures:

List of Tables:

Nomenclatures:

Chapter 1: Introduction

- 1.1 Background/Motivation
- 1.2 Problem Statement
- 1.3 Scope
- 1.4 Objectives
- 1.5 Hardware and software requirements for development
- 1.6 Hardware and software requirements for deployment
- 1.7 Dataset Dictionary

Chapter 2: Literature Survey

Chapter 3: Project Design

- 3.1 Proposed System model/ Architecture
- 3.2 Software Project Management Plan
- 3.3 Software Design Document (All applicable diagrams)

Chapter 4: Implementation and experimentation of one of the issues related to project work/ topic

- 4.1 Proposed system model implementation
- 4.2 Inclusion of Any additional details as suggested by Project Guide/during progress seminar
- 4.3 Software Testing (Software testing reports at various levels)
- 4.4 Experimental results and its analysis

Chapter 5: Conclusion and future work

- 5.1 Conclusion and discussion

5.2 Scope for future work

Bibliography

Acknowledgement

Chapter 1: Introduction

This comprehensive introduction sets the stage for the development of a sophisticated web application aimed at enhancing the gaming experience and performance analysis capabilities of F1-2021 players.

1.1 Background/Motivation

In recent years, motorsports, including Formula 1, have embraced a data-centric approach to gain a competitive advantage. This trend extends to virtual platforms like F1-2021, the official video game for the 2021 F1 season, which boasts a thriving ESports community with substantial prize pools and widespread recognition. To stay ahead in this competitive landscape, players require precise live data analysis tools that can help them improve their performance and provide a more immersive gaming experience.

1.2 Problem Statement

The objective is to develop a web application capable of visualising live telemetry data from races in F1-2021. This application will empower players to analyse their performance in real-time, enabling them to identify areas for improvement and gain a competitive edge over their rivals while enhancing the realism of their F1 gaming experience.

1.3 Scope

The scope of this project encompasses the extraction of live data from F1-2021, its processing, and the creation of an intuitive interface for visualising various performance parameters. While the primary focus is on enhancing racecraft through data analysis, this project does not delve into predictive analytics or alternative strategy formulation.

1.4 Objectives

The key objectives of this project include:

- Extracting live data from F1-2021.
- Processing the live data to derive meaningful insights.
- Creating an intuitive interface that visualises different performance parameters, enabling players to identify weaknesses and refine their skills.

- Harness the power of historical lap time data to deliver precise and reliable predictions, providing invaluable insights for optimizing race strategies and performance analysis in the highly competitive world of motorsport.

1.5 Hardware and Software Requirements for Development

Hardware:

- F1-2021 (the game)
- PC with sufficient processing power to run F1-2021

Software:

- Apache Kafka/Kinesis data streams for real-time data processing

1.6 Hardware and Software Requirements for Deployment

Hardware: Any standard PC capable of running F1-2021

Software: F1-2021 (the game)

1.7 Dataset Dictionary

Telemetry Data Parameters:

- Throttle: The degree to which the throttle pedal is pressed, represented as a value between 0 and 1.
- Speed: The current speed of the car in kilometres per hour (km/h).
- Distance: The distance travelled by the car, likely in metres.
- Brake: Indicates whether the brake pedal is being pressed (1 for pressed, 0 for not pressed).
- Gear: The current gear engaged by the car.
- Lap: Indicates the lap number or segment of the race.

Chapter 2: Literature Survey

These references cover various aspects of telemetry data analysis in Formula One racing, including data processing techniques, machine learning applications, race strategy optimization, visualisation tools, and security considerations.

Sr No.	Name	Author and Publication	Year	Conclusion from the paper
1	A Telemetry-driven Approach to Simulate Data-intensive Manufacturing Processes	Gianfranco E. Modoni a, Marco Saccob, Walter Terkaj b ElseVier	2016	This paper aims to assess the effectiveness of using telemetry data in manufacturing, enabling realistic simulations of production processes. By harnessing telemetry, a virtual model of the factory can be created to predict failures, troubleshoot issues, and minimize downtime. Drawing from Formula One racing techniques, this study explores how to adapt these methods for factory environments. Additionally, it involves

				developing a software application to support telemetry integration, beginning with defining functional requirements.
2	**Challenges in designing measurement systems for Formula One cars**	**Przemysław Wojciechowski, Konrad Wojtowicz**, ***2023*** **IEEE International Workshop on Metrology for Automotive**	2023	The challenges and recent developments in developing measurement systems for Formula One (F1) cars are examined in this research. It talks about how teams' reluctance to share information makes research difficult to acquire and suggests utilizing lab tests to measure sensor endurance under strains caused by race. Suggestions for future study directions are included in the paper's conclusion.
3	**Metrology and Formula One Car**	**Cocco and P. Daponte IEEE Instrumentation and Measurement Technology Conference, Victoria, BC, Canada, 2008**	2008	This paper reviews the challenges and recent advancements in designing measurement systems for Formula One (F1)

				cars. It discusses the difficulties in accessing research due to teams' reluctance to share knowledge. Additionally, it introduces the concept of laboratory tests to evaluate sensor resistance to race-induced stresses. The paper concludes with suggestions for future research directions.
4	**Embedded computing and Formula One racing**	**Waldo, IEEE Pervasive Computing**	2005	This paper explores the intersection of embedded computing and F1 racing, highlighting how embedded systems are integral to various aspects of race operations, from vehicle control and telemetry to data analysis and strategy optimization. It examines the challenges and innovations in embedded computing within the context of F1 racing, showcasing how advancements in

				technology continually push the boundaries of performance and competitiveness in the sport.
5	**Motorsport Data Acquisition System and Live Telemetry using FPGA based CAN controller**	**Banerjee et al, IOP Publishing Ltd**	2022	This paper presents a motorsport data acquisition, logging, live telemetry, and display system developed using the Controller Area Network (CAN) communication protocol. The system utilises the myRIO controller programmed with LabVIEW as its main controller. Designed specifically for Formula One cars, the system incorporates over a hundred sensors to provide real-time data during testing, validate design assumptions, and ensure vehicle performance. The FPGA-based CAN controller aims to improve upon existing systems, with custom-made choices for controllers,

				sensors, and formatting tailored to the team's requirements. The data acquired by the system plays a crucial role in optimising the car's performance and achieving design goals.
6	**Data Visualization: Bringing Data to Life in an Introductory Statistics Course**	Lynette M Hudiburgh, Diana Garbinsky, Journal of Statistics Education	2020	This paper highlights the importance of data visualization in teaching statistics, especially in the age of big data. It proposes a semester-long group project where students learn to analyze and create data visualizations, improving their statistical thinking, collaboration, and communication skills. The project starts with learning visualization principles, followed by dataset selection, project planning, and visualization development. Students receive ongoing feedback

				and present their final projects to the class.
7	**Formula 1 Race Car Performance Improvement by Optimization of The Aerodynamic Relationship Between The Front and Rear Wings**	**Unmukt Rajeev Bhatnagar, The Pennsylvania State University**	2014	This F1 thesis tackles optimizing wing designs for faster lap times on a specific track. It explores how aerodynamics have shaped F1 cars, emphasizing their role in speed and safety. The study uses an optimization algorithm (CMA-ES) and a racing simulator (AeroLap) to fine-tune front and rear wing aerodynamics for Sepang circuit. It balances downforce for grip against drag for speed, leading to substantial lap time improvements compared to real races.
8	**Aerodynamics of a 2017 Formula 1 car: Numerical Analysis of a Baseline Vehicle and Design Improvements in Freestream and Wake Flows**	Umberto Ravelli, Università degli Studi di Bergamo Department of Engineering and Applied Sciences	2019	This F1 thesis dives deep into race car aerodynamics using OpenFOAM® software. It starts with basic simulations to grasp ground effect, then zooms

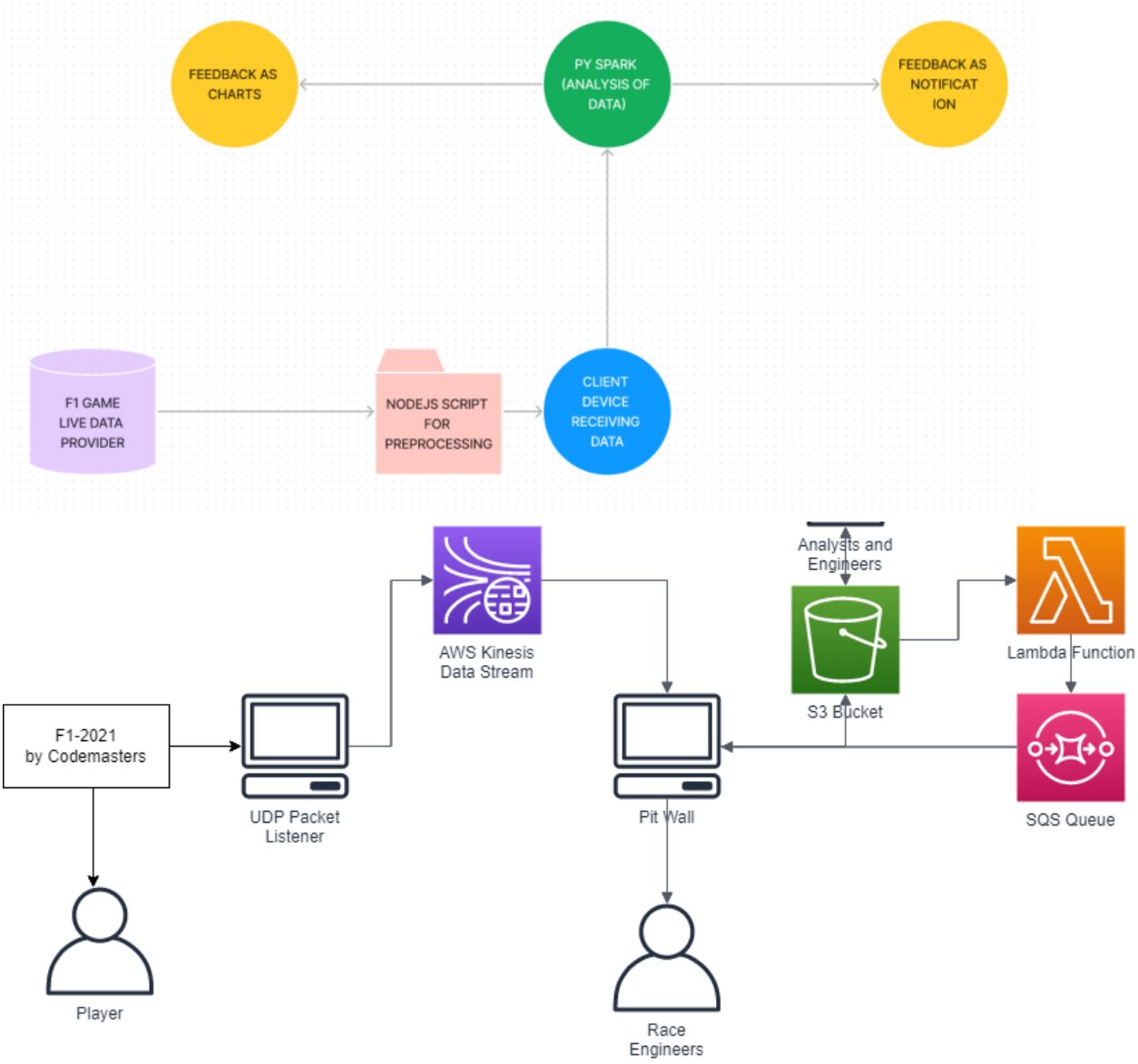
				in on a 2017 F1 car. Validating drag, downforce, and efficiency, the study analyzes airflow around the car. It explores how ride height affects handling and evaluates performance in dirty air – a key factor in racing. New aerodynamic designs are proposed to boost performance in close racing, aiming for higher downforce and efficiency while enabling safer overtaking. The thesis concludes by showing how much faster these new designs could make the car, staying relevant to modern F1.
9	**A Data-Driven Analysis of Formula 1 Car Races Outcome**	Ankur Patil, Nishtha Jain, Rahul Agrahari, Murhaf Hossari, Fabrizio Orlandi, Soumyabrata Dev Springer	2023	The paper proposes a data-driven approach to identifying the factors that contribute to the total points scored by each driver in a Formula 1 (F1) season. The authors perform a correlation analysis and

				principal components analysis (PCA) on 21 input variables related to F1 car characteristics, aiming to reduce the dataset into a lower-dimensional subspace that explains most of the variance. The dataset covers five years (2015-2019) of F1 race data obtained from a publicly available website.

Chapter 3: Project Design

This chapter aims to provide an understanding of the background, logical thinking, and fundamental concepts relevant to the chosen thesis work. It delves into the theoretical foundations necessary for the development and implementation of the project, drawing upon existing literature, tables, figures, and remarks to elucidate key concepts.

3.1 Proposed Architecture



The game transmits its data to a designated device, identified by its IP address within the game settings. Upon receiving the real-time data, the device executes a script to decode the data packets and conduct necessary preprocessing tasks. Subsequently, the processed data is transmitted to Kinesis. To generate graphs and conduct analysis, data is extracted from Kinesis. Real-time charts displaying various parameters are then generated. Users are prompted to select the track for their race, with the track's length ultimately serving as the x-axis for the generated graphs.

3.2 Software Project Management Plan

1. Project Overview:
Extract live data from the game . Process the live data. Create an interface to visualise the data with different parameters allowing players to identify weak spots and correct themselves.

2. Requirements:
F1 2021 game, laptop with good internet connection and Amazon AWS account.

3. Design
1. Initialise variables and data structures as needed.
2. Start the data collection process by connecting to the F1 Racing Game API.
3. While the game is running, collect data from the game API and preprocess it:
a. Retrieve the raw data from the API.
b. Clean and filter the raw data to obtain relevant information.
c. Store the preprocessed data temporarily in a data structure.
4. Analyse the preprocessed data to provide feedback to the player:
a. Calculate relevant statistics based on the preprocessed data.
b. Determine appropriate feedback based on the calculated statistics.
c. Generate visualisations to provide feedback to the player.
5. Provide the feedback to the player through an interface:
6. Repeat steps 3-5 until the game is over.
7. End the data collection process by disconnecting from the F1 Racing Game API.
8. Output any relevant statistics or data for future analysis, if necessary.

4. Implementation
We are a group of 4 students. Bharat and Husain contributed in writing the NodeJS code to extract data. Esha and yashraj contributed by writing the python code for extracting data from kinesis and plotting chart along with analysis.

5. Contingency Plan
The project requires the game to be played for real time data and data is uploaded and extracted afterwards from kinesis, both of which need a good internet connection.
If data is to be received on a device other than the one on which the game is played, then the Ip connection is required to be checked.

6. Budget Management
No expenditure incurred during the project as it is not built currently for large scale professional application. The cost includes buying the game and kinesis service fees.

3.3 Software Design Document (All applicable diagrams)

1. Frontend Interface

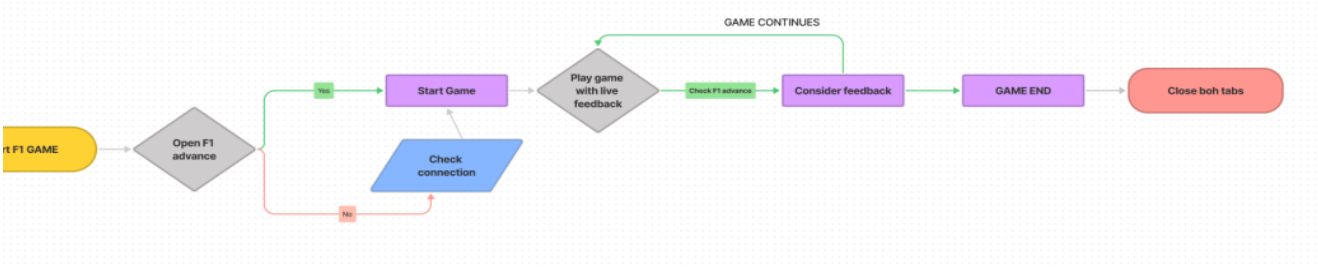


Figure 1 Flowchart Frontend

2. Backend Design

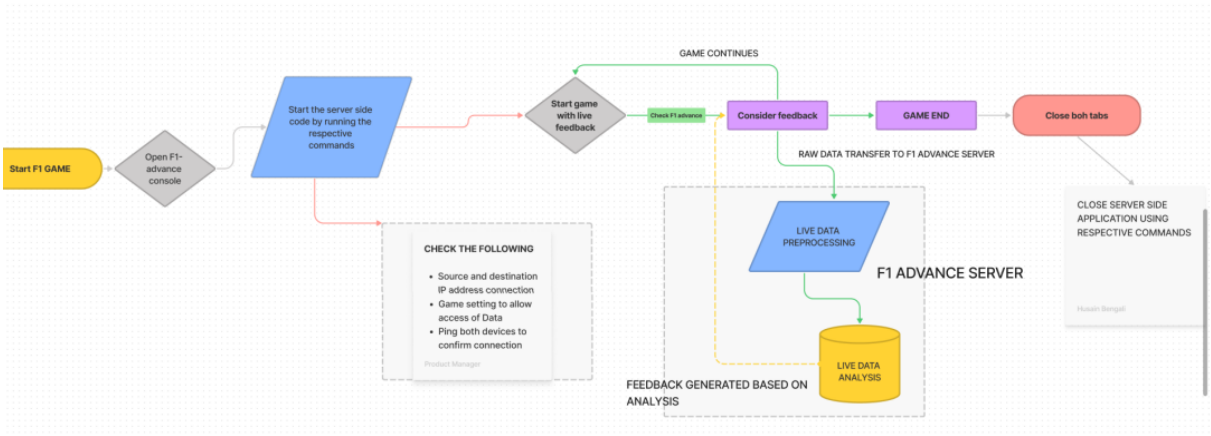


Figure 2 Flowchart backend

3. **UML Diagram**

Figure 3 Project UML Diagram

Chapter 4: Project Implementation

Chapter 4 outlines the project's implementation, covering steps such as real-time data extraction, preprocessing, and graph plotting using JavaScript and Python. Additional features include dynamically adjusting the graph's X-axis based on the selected track and implementing a machine learning model for predictions. Software testing ensures accurate real-time data display and performance under load, achieving the project's objective of providing accurate telemetry data visualisation for Formula One racing.

4.1: Proposed System Model Implementation

STEPS FOLLOWED:

- Extract real time data from the game into the local terminal.
- Send data from local machine to Kinesis
- Pre-process the data to get the required fields
- Plot different types of graphs
- Future implementation

CODE CONSISTS OF TWO PARTS:

1) Code to extract and pre-process data received from the game in real time and then sending the data to amazon kinesis. Code is written in JavaScript.

Creating Kinesis Client with required parameters

```
const kinesis = new KinesisClient({
region: "ap-south-1",
});
```

This code sets up a connection to the F1 game, listens for updates on the participants, and finds the index of the player's teammate in the received data.

```
const streamName = 'grande-strategia';
const client = new F1TelemetryClient({binaryButtonFlags: false});
client.start();
client.on('participants',function(data) {
teammate=((data.m_participants.findIndex(x=>
x.m_teamId==data.m_participants[19].m_teamId)));
})
```

This code sets up an event listener for the 'carTelemetry' event emitted by the `F1TelemetryClient` instance.When this event is triggered, it creates a `record_myCar` object containing telemetry data for the player's own car, and then sends this data to an Amazon Kinesis stream specified by the `streamName` variable.

```
client.on('carTelemetry', function(data) {
// console.log(data.m_carTelemetryData[teammate])
const record_myCar = {
Data:
Buffer.from(JSON.stringify(data.m_carTelemetryData[data.m_header.m_playerC
arIndex])),
PartitionKey: 'Car-1'
};
const params_myCar = {
Data: record_myCar.Data,
StreamName: streamName,
PartitionKey: record_myCar.PartitionKey
};
```

2) Python code to plot graphs and perform analysis.
This code initializes a Kinesis client using the `boto3` library and sets the `stream_name` variable to the name of the Kinesis stream to be accessed. It then retrieves the latest shard iterator for the specified shard in the stream using the `get_shard_iterator()` method of the Kinesis client, and stores the result in the `shard_iterator` variable.

```
client = boto3.client('kinesis')
stream_name = 'grande-strategia'
# get the latest shard iterator
shard_iterator = client.get_shard_iterator(
StreamName=stream_name,
ShardId='shardId-000000000000',
ShardIteratorType='LATEST'
)['ShardIterator']
```

This snippet allows user to select a track from the available tracks provided with their respective lengths. The length of the chosen track becomes the value on the X- axis of the graph which will be plotted

```
ith open('tracks.json', 'r') as f:
data = json.load(f)
print("Select a track number:")
for i, t in enumerate(data['tracks']):
print(f"{i+1}. {t['name']}")
track_number = int(input())
if 1 <= track_number <= len(data['tracks']):
track = data['tracks'][track_number-1]
track_name = track['name']
track_length = track['length']
print(f"The length of {track_name} is {track_length} meters.")
else:
print(f"Invalid track number. Please select a number between 1 and
{len(data['tracks'])}.")
```

This code enters an infinite loop where it repeatedly retrieves the latest records from a Kinesis stream using the `get_records()` method of the Kinesis client, and updates the `shard_iterator` variable with the next shard iterator from the response. It then extracts telemetry data from each

record, including the car speed, throttle, and brake values, as well as lap number and lap distance information if the record corresponds to a lap data message. The extracted data is appended to `speeds`, `throttle`, `brake`, and `distance` arrays.

```
while True:
# get the latest records
response = client.get_records(
ShardIterator=shard_iterator,
Limit=100
)
# update the shard iterator
shard_iterator = response['NextShardIterator']
# extract the speed data and lap count from the records
for record in response['Records']:
if record["PartitionKey"]=='Lapdata':
data_str2 = record["Data"].decode('utf-8')
data_obj2 = json.loads(data_str2)
m_lapNumber=data_obj2['m_currentLapNum']
m_lapdistance=data_obj2['m_lapDistance']
else:
data_str=record["Data"].decode("utf8")
data_obj = json.loads(data_str)
m_speed = data_obj['m_speed']
m_throttle=data_obj['m_throttle']
m_brake=data_obj['m_brake']
if m_speed !=0:
speeds.append(m_speed)
distance.append(m_lapdistance)
throttle.append(m_throttle)
brake.append(m_brake)
```

4.2 Inclusion of Any additional details as suggested by Project Guide/during progress seminar

Implementation of analytics:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
```

```
import tensorflow as tf

data = pd.read_csv("lap_times.csv")

# Normalize the lap times
scaler = MinMaxScaler(feature_range=(0, 1))
data["lap_times_normalized"] = scaler.fit_transform(
    data["lap_time"].values.reshape(-1, 1)
)

# Split the data into input (X) and target (y) variables
X = data["lap_times_normalized"].values
y = np.roll(X, -1)  # Shift the lap times by one to predict the next lap time
y[-1] = 0  # Set the last target to 0 (since we shifted the array)

# Reshape the data for LSTM input (samples, time steps, features)
X = X.reshape(X.shape[0], 1, 1)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Define the LSTM model

model = tf.keras.Sequential(
    [
        tf.keras.layers.LSTM(50, input_shape=(X_train.shape[1],
X_train.shape[2])),
        tf.keras.layers.Dense(1),
    ]
)

# Compile the model
model.compile(optimizer="adam", loss="mean_squared_error")

history = model.fit(
    X_train,
    y_train,
    epochs=50,
    batch_size=32,
    validation_data=(X_test, y_test),
    verbose=1,
)
```

```
# Evaluate the model
loss = model.evaluate(X_test, y_test, verbose=0)
print("Test Loss:", loss)

with open("lap_times.csv", "r", encoding="utf-8", errors="ignore") as scraped:
    final_line = scraped.readlines()[-1]

final_line_arr = final_line.split(",")
last_lap_count = int(final_line_arr[1])

new_lap_number = (
    last_lap_count + 1
)
# For example, to predict the lap time for lap number 50

# Normalize the new lap number
new_lap_number_normalized = scaler.transform(np.array([[new_lap_number]]))

# Reshape the data for LSTM input (samples, time steps, features)
new_lap_number_input = new_lap_number_normalized.reshape(1, 1, 1)

# Predict the next lap time
predicted_lap_time_normalized = model.predict(new_lap_number_input)

# Inverse transform to get the actual lap time
predicted_lap_time = scaler.inverse_transform(predicted_lap_time_normalized)

print(
    "Predicted lap time for lap number", new_lap_number, ":",
predicted_lap_time[0][0]
)
```

This model harnesses the power of historical lap time data to deliver precise and reliable predictions, providing invaluable insights for optimizing race strategies and performance analysis in the highly competitive world of motorsport.

4.3 Software Testing (Software testing reports at various levels)

The purpose of this software testing report is to provide an overview of the testing conducted for our project. The project involves the real-time visualisation of telemetry data from Formula One racing cars. The testing was conducted to ensure that the application functions properly and provides accurate real time data visualisations.

The testing objectives were to ensure that the application:
*Displays real-time telemetry data accurately
*Provides accurate visualisations of the data
*Performs well under heavy load and stress
*is understandable to users

Real-time telemetry data visualisation:
The real-time telemetry data was displayed accurately and in real-time. All data points were displayed on the dashboard as intended. No issues were found during the testing.

Performance and load testing:
The application was tested under heavy load. It performed well and was able to handle the load without any issues. No errors or crashes were reported. As we increased more parameter the graphs still were plotted properly

Accuracy of data visualisations:
All data visualisations were accurate and matched the expected values. The data was displayed in an easy-to-understand format, making it easy for users to interpret the telemetry data.

The respective real time graph is plotted.

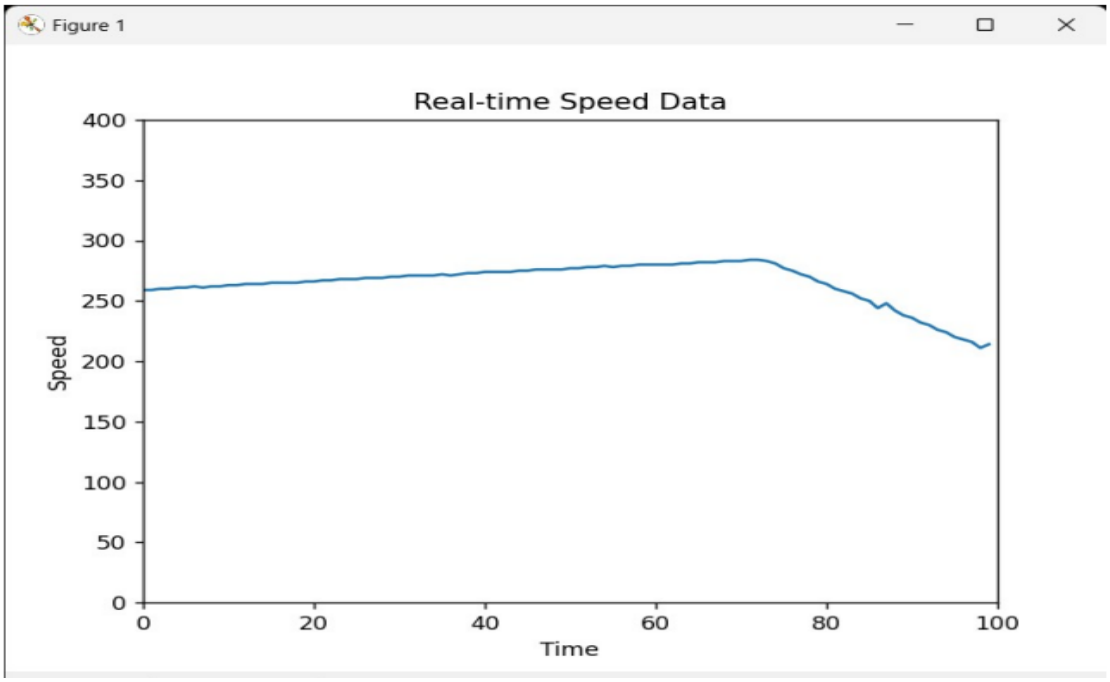


Figure 4 Real time Speed vs Time Data in Graph

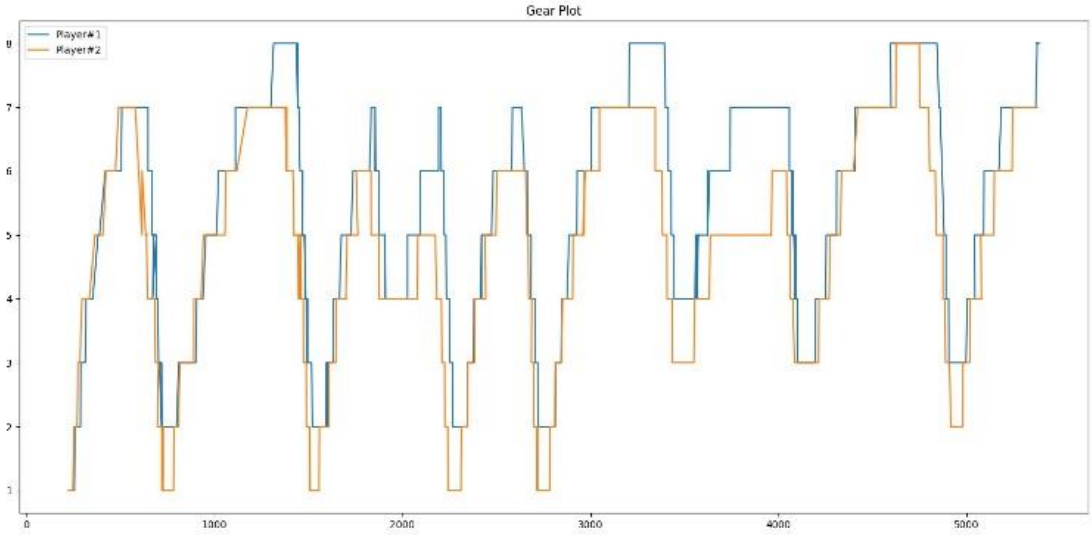


Figure 5 gear Plot - Gears vs Distance of 2 players in a lap.

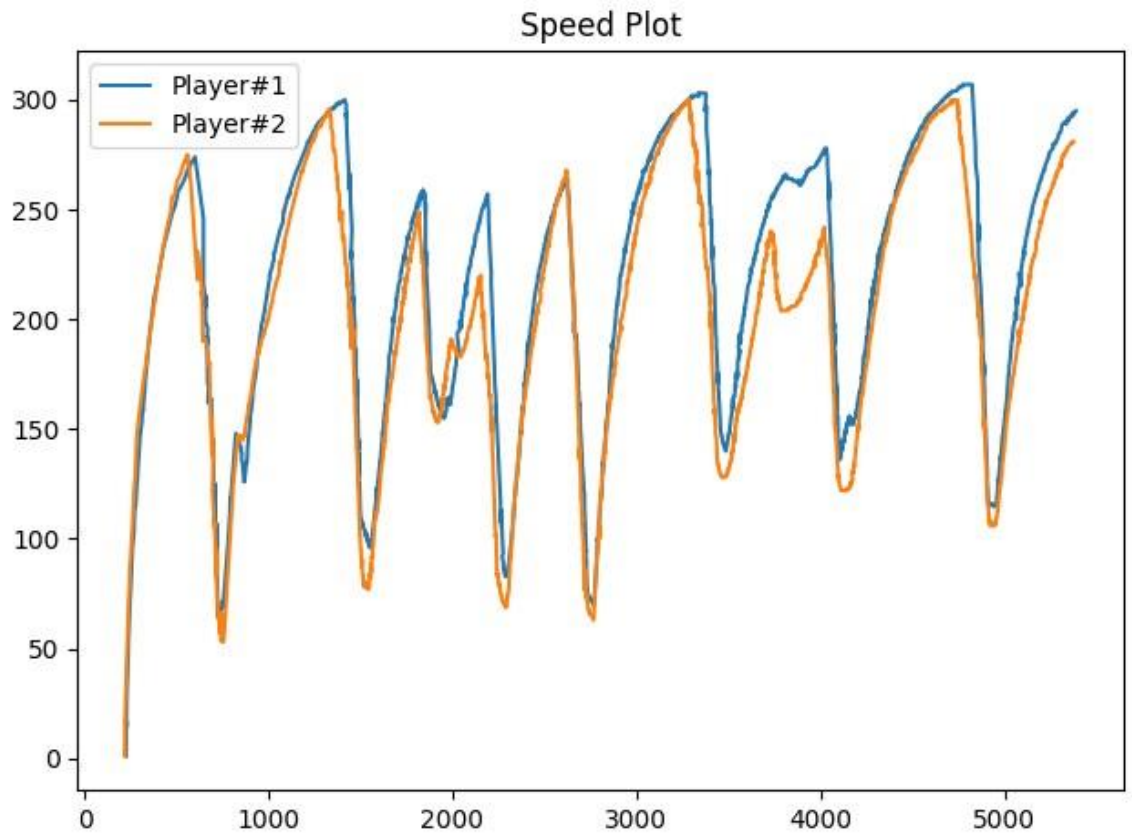


Figure 6 Speed Plot - Speed vs Distance of 2 players in a lap

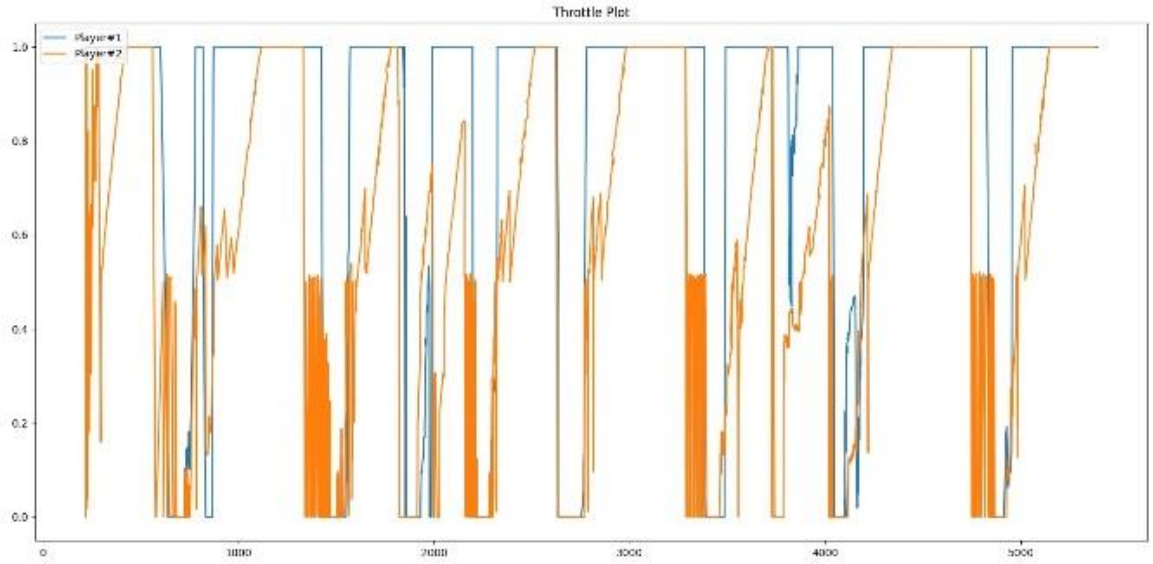


Figure 7 Throttle plot - Throttle vs Distance of 2 players in lap

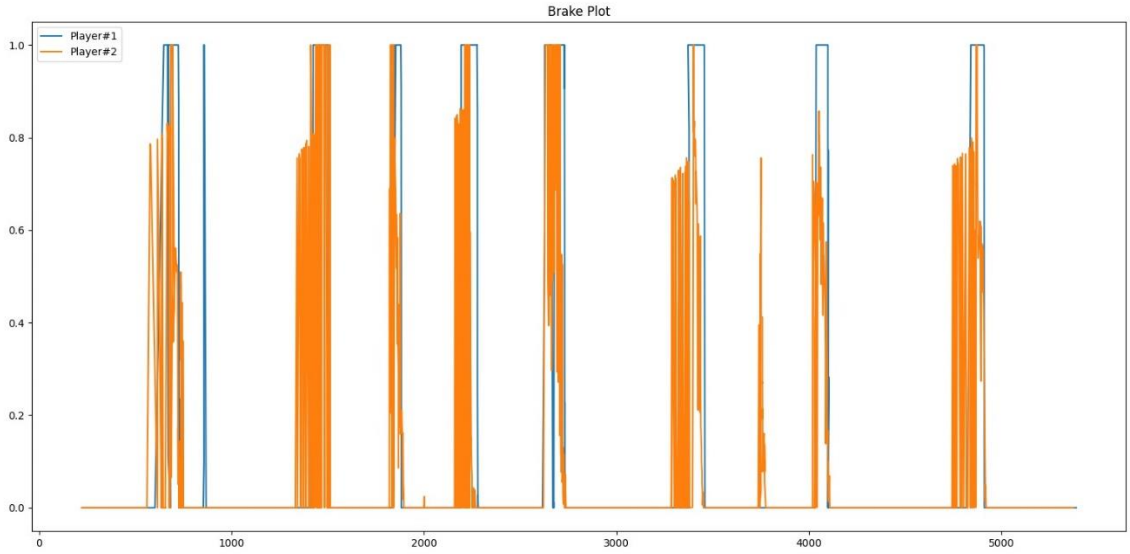


Figure 8 Brake plot - Brake vs Distance plot of 2 players in a lap

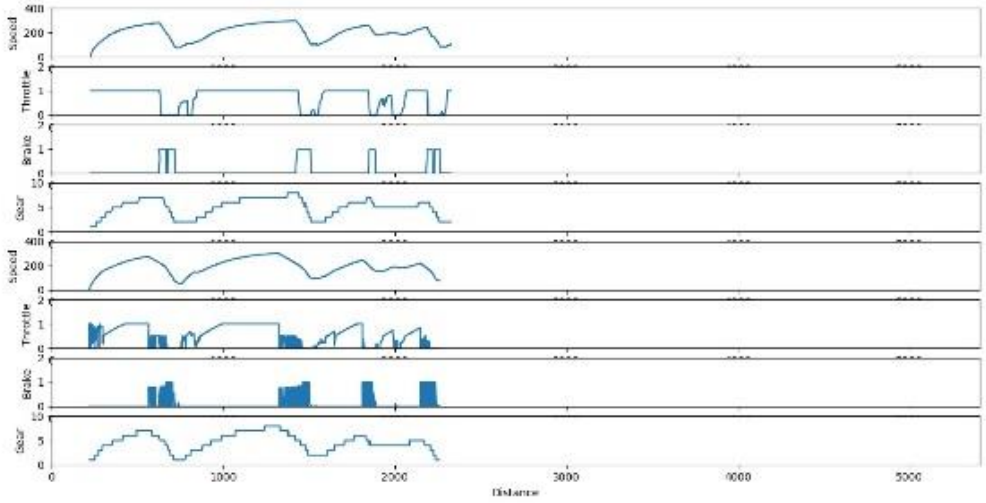


Figure 9 All plots vs Distance in 1 lap

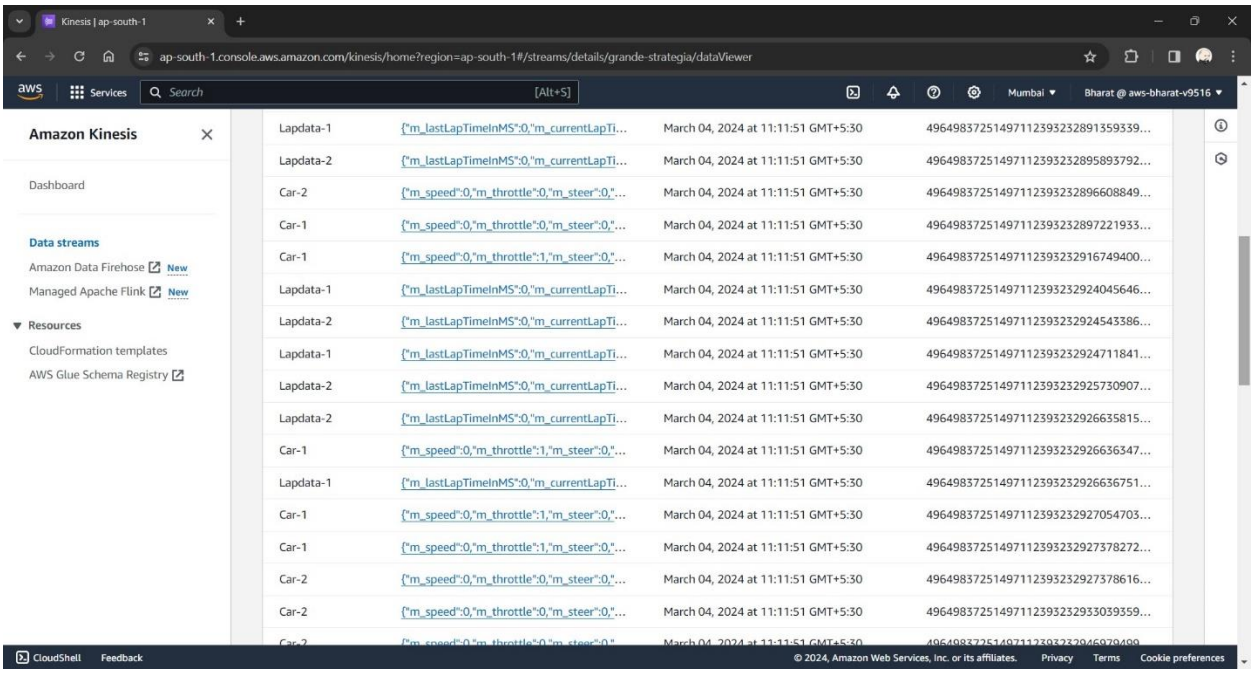


Figure 10 Sample Data packet interface in Kinesis.

Chapter 5 : Conclusion and Future Work

It has been clearly established that telemetry and data analysis are an important part of modern F1, and accurate data analysis done in real time can be the difference between winning and losing. The same can be said for the video game equivalent, as the full potential of telemetry is not harnessed by the player. Thus, our application can help bridge the gap between the player and the telemetry data to make a more immersive and interactive experience for the player.

F1 telemetry data can be massive, with hundreds or even thousands of variables recorded for each lap of each race. We understood working with large datasets,Working with real-time data helped us develop skills in data visualization, real-time analysis, and decision-making. Using cloud services to extract and store data was another valuable learning experience , how we stored the data in a S3 bucket how we utilize Kinesis for live data streaming, etc.

Future Work:

- Use machine learning algorithms to predict when car components are likely to fail, allowing teams to perform proactive maintenance and avoid costly breakdowns during races and tyre wear prediction.

- Develop driver assistance systems that use real-time telemetry data to provide drivers with feedback and suggestions for improving their lap times and driving performance.
- Apply machine learning algorithms and to visualize the track and show in real time where the car is compared to other players.
- Live Timing Screen to keep track of lap time and sector time done by all players
- Make a UI and web application so that it can be better understood by players

Bibliography

1] C. Wyawahare, “Formula 1 Grand Prix Analysis,” Medium, Feb. 02, 2021. https://towardsdatascience.com/formula-1-grand-prix-analysis-d05d73b1e79c?gi=9150f4fdcf6e (accessed Mar. 05, 2024).

[2] A. Struthers, “Formula One Telemetry Analysis,” Symposium Of University Research and Creative Expression (SOURCE), May 2022, Accessed: Mar. 05, 2024. [Online]. Available: https://digitalcommons.cwu.edu/source/2022/COTS/99/

[3] G. E. Modoni, M. Sacco, and W. Terkaj, “A Telemetry-driven Approach to Simulate Data-intensive Manufacturing Processes,” Procedia CIRP, vol. 57, pp. 281–285, 2016, doi: https://doi.org/10.1016/j.procir.2016.11.049.

[4]

“Metrology and Formula One Car | IEEE Conference Publication | IEEE Xplore,” ieeexplore.ieee.org. https://ieeexplore.ieee.org/abstract/document/4547138 (accessed Mar. 05, 2024).

Acknowledgement

We would sincerely like to thank our mentor for this project, Professor Bharathi H.N. for her constant guide and support. We would also like to thank our co-ordinator Prof. Kavita Kelkar and all the faculty members of the Honors department for giving us this opportunity which led to us learning and implementing this project.