iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.69749

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Optimum Racing: A F1 Strategy Predictor using Reinforcement Learning

Veerasagar S S[1], Reevan Coelho[2], Prof. L Kala Chandrashekar[3]
Department of Computer Science and Engineering, RV College of Engineering, Bangalore, India

Abstract: Formula One race strategy optimization has traditionally relied on predefined heuristics and Monte Carlo simulations, which are computationally expensive and lack adaptability to live race conditions. While prior works have explored reinforcement learning (RL) in other motorsport categories, its application to Formula One strategy remains underdeveloped. This research introduces a reinforcement learning framework aimed at dynamically predicting tire compound choices after the summer break, addressing the gap in adaptive decision-making for in-race strategic planning. The proposed RL model employs a deep recurrent Q-network (DRQN) trained using a Monte Carlo race simulator. The state space incorporates critical race parameters such as tire degradation, gaps to competitors, and race progress, while the action space consists of tire compound selection and pit stop timing. A reward function, balancing immediate lap performance and long-term finishing position, guides the learning process. The model is further enhanced with explainability techniques, including feature importance analysis and decision tree-based surrogate models, to improve transparency and trust in automated strategy recommendations.

I. INTRODUCTION

Formula One (F1) is a premier class of motorsport, often regarded as its pinnacle, with the average annual cost of running a team reaching hundreds of millions of pounds. Teams constantly seek marginal gains in performance by recruiting the best drivers and enhancing their cars through cutting-edge engineering. However, once a race begins, teams cannot alter their cars or drivers, making race strategy a crucial factor in determining finishing positions. A key component of this strategy is tire selection and pit stop timing, which significantly impact race outcomes.

Currently, teams decide on candidate strategies before a race, attempting to account for various live race scenarios such as safety cars, which require cars to slow down due to unsafe track conditions. To evaluate these strategies, teams run Monte Carlo simulations, executing millions of race scenarios to estimate their effectiveness. While this approach provides valuable insights, it is time-consuming, computationally expensive, and requires predefined strategies, making it challenging to adapt dynamically to live race conditions. Moreover, the reasoning behind the selected strategies is not readily interpretable, posing a challenge in trust and decision-making.

The complexity of modern F1 race strategy demands a more efficient, adaptive, and interpretable approach. Traditional methods rely on extensive simulations with limited real-time adaptability, making them suboptimal in rapidly evolving race scenarios. Given the advancements in artificial intelligence, particularly in reinforcement learning (RL), there is potential to enhance race strategy by enabling real-time decision-making based on live data.

II. LITERATURE SURVEY

This chapter provides a comprehensive review of existing literature related to race strategy optimization in motorsports, with a particular focus on Formula 1 (F1) and related racing series. The increasing availability of data, coupled with advancements in computational power and algorithmic techniques, has opened up new avenues for applying data-driven approaches to enhance race performance. This survey explores a range of methodologies, including simulation, machine learning, reinforcement learning, and game theory, used to model, analyze, and optimize various aspects of race strategy. The ultimate goal is to identify effective strategies that minimize race time, maximize finishing position, and provide a competitive advantage. The survey highlights both the successes and limitations of current approaches, setting the stage for further research and development in this rapidly evolving field.

A. *Related Work*

The optimization of race strategies in motorsports, particularly in Formula 1, has been a topic of growing interest in recent years.

Early approaches relied heavily on deterministic models and expert knowledge, but the field has rapidly embraced more sophisticated techniques from data science and artificial intelligence.

One common approach is the use of discrete-event simulation [18]. Bekker and Lotz [18] developed a simulation model to evaluate different pit stop strategies, demonstrating the potential for improved race outcomes through optimized planning. Heilmeier et al. [17] further advanced race simulation by incorporating factors like tire degradation, fuel consumption, and overtaking maneuvers, providing a more realistic representation of race dynamics. Later work by Heilmeier et al. [12] incorporated probabilistic effects, such as accidents and safety cars, using Monte Carlo methods to account for the inherent uncertainty in racing events.

The rise of machine learning has significantly impacted the field. Fatima and Johrendt [5] introduced Deep-Racing, an embedded deep neural network (EDNN) model, to predict driver rankings and optimal pit stop strategies. Zhao [9] proposed a deep neural network (DNN) for predicting the fastest lap time in qualifying sessions, demonstrating the ability of neural networks to learn complex patterns from historical data. Hojaji et al. [4] used machine learning to predict sim racing performance from telemetry data, identifying key metrics that influence driver performance. More broadly, time series analysis and forecasting techniques have been extensively applied to predict various race-related metrics [7, 15]. Malik et al. [7] provide a comprehensive analysis of these techniques, while Dama and Sinoquet [15] offer a survey focusing on parametric models for forecasting. Han et al. [14] propose a specific time series forecasting model combining deep learning and GARCH modeling for non-stationary series.

Reinforcement learning (RL) has emerged as a powerful tool for optimizing race strategies in dynamic and uncertain environments. Boettinger and Klotz [6] developed an RL-based approach for GT motorsports, specifically for the Nürburgring Nordschleife, demonstrating the potential for automating pit stop and refueling decisions. Thomas et al. [11] introduced RSRL, an explainable reinforcement learning model for F1 race strategy, which outperformed baseline models and incorporated XAI techniques to enhance user understanding. Piccinotti et al. [16] adopted a different RL-based approach, employing online planning algorithms based on Monte Carlo Tree Search.

Game theory provides a framework for modeling strategic interactions between competing drivers. Aguad and Thraves [19] developed a game theory model for optimizing pit stop strategies in a two-driver competitive scenario, considering overtaking and stochastic events. Heine and Thraves [18] employed dynamic programming to optimize pit stop strategies, presenting both deterministic and stochastic models. Paparusso et al. [11, 20] specifically addressed race strategy for hybrid vehicles, taking into accountcompetitors actions and regulations.

Beyond strategy optimization, researchers have also focused on understanding the relative contributions of driver skill and car performance. Van Kesteren and Bergkamp [8] used Bayesian analysis to disentangle driver skill and constructor advantage in F1 race results, finding that the car (constructor) is a significantly larger factor than the driver. Bonomi et al. [10] employed a custom Genetic Algorithm for strategy optimization, demonstrating the strength of this approach. Finally, Peng et al. [14] focuses on the prediction of car rank, highlighting the uncertainty and complexity of this specific task.

B. Literature Summary

This literature survey has highlighted the diverse range of approaches being applied to optimize race strategies in motorsports. Simulation, machine learning, reinforcement learning, and game theory have all proven to be valuable tools, each with its own strengths and limitations. Early work focused on deterministic models and simulation, while more recent research has embraced data-driven techniques, particularly deep learning and reinforcement learning, to handle the complexity and uncertainty inherent in racing. The incorporation of probabilistic effects and the modeling of competitor behavior are becoming increasingly important, reflecting the dynamic and strategic nature of the sport. While significant progress has been made, there remain open challenges, including the need for more robust and explainable models, the integration of real-time data streams, and the development of cooperative strategies for multi-car teams. Future research will likely focus on addressing these challenges and further refining the application of AI and data science to gain a competitive edge in the ever-evolving world of motorsports.

III. METHODOLOGY

This chapter details the design and architecture of the project, which aims to develop a system for optimizing Formula 1 race strategy, specifically focusing on pit stop decisions. The system uses a machine learning technique called Deep Recurrent Q-Networks. We present both the high-level system overview and the detailed design of individual components.

High-Level Design - The overall system architecture follows a typical Reinforcement Learning paradigm. The core components are:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

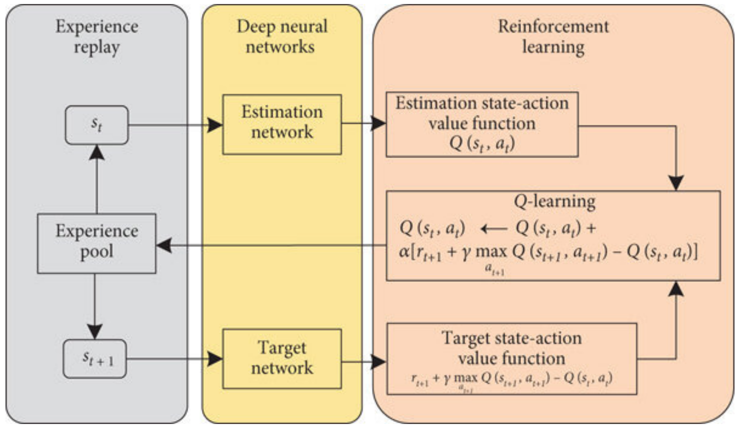


Fig 1: High-Level design of DRQN Network

1) Environment: This component simulates the Formula 1 race. It uses historical race data, recorded lap by lap, to represent the changing conditions of a race. The environment provides the agent with information about the current state of the race and receives actions (whether to make a pit stop) from the agent. It also calculates and returns feedback (rewards) based on how good or bad the lap times are, including any time penalties for pit stops. To help the agent, the environment provides not just the current state, but also a short history of recent states.
2) Agent (Decision-Maker): The agent is the decision-making component. It's a type of neural network specifically designed for handling sequences of data and making decisions in changing environments. The agent uses a specialized memory component (called Long Short-Term Memory) to process the sequence of race states, remembering important events from the past (like how the tires have been worn down). The agent predicts how good each possible action (pitting or not pitting) would be at each point in the race.
3) Experience Replay: This component stores the agent's past experiences. Each experience is a record of what the agent saw (the sequence of states), what it did (the action), what reward it received, and what happened next. This stored experience allows the agent to learn from a wide range of situations, making its learning more stable and effective.
4) Training Loop: The training process involves the agent repeatedly interacting with the environment. It stores its experiences and then learns from those experiences to improve its decision-making. A special "target" network is used to make the training process more stable.
5) Data Preprocessor: This tool is to preprocess and standardize numerical inputs.

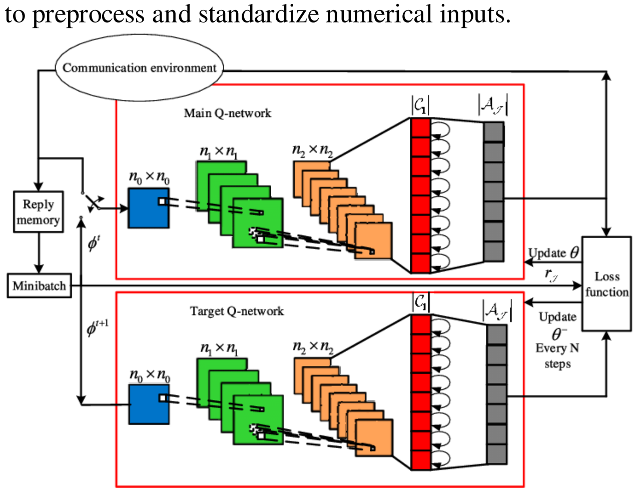


Fig 2: Design of DRQN Network

This implementation centers around training a Deep Recurrent Q-Network (DRQN) for a simulated racing strategy. The code integrates data processing, environment simulation, recurrent neural network architecture, and a reinforcement learning (RL) training loop. Each component is explained in the sections below

A. Data Handling and Preprocessing
The implementation begins with a function that leverages the glob module to gather all CSV file paths from a specified folder, enabling batch processing of multiple datasets, such as those representing different racing seasons. Complementing this, a dictionary named compound mapping converts qualitative tire compound labels like "HARD", "MEDIUM", and "SOFT" into numerical values for subsequent data processing. Central to the design is the Real state class, which transforms individual CSV rows into structured state representations of the racing environment. This class encapsulates a range of attributes, including lap details (lap number and sector timings), tire information (tire life and the numerically mapped compound), race positioning (current position and time gaps to the leader and following cars), environmental conditions (air and track temperatures), and speed metrics. Furthermore, the to_array() method within the Race State class converts these attributes into a consistent NumPy array format, ensuring compatibility with neural network inputs.

B. Action Definition
The RaceAction class defines a simplified action space for the simulation by encapsulating two distinct behaviors: NO_PIT, which represents the decision to continue racing without stopping, and PIT_STOP, which indicates the choice to make a pit stop—acknowledging the additional time penalty incurred by such a decision. To facilitate the decision-making process, the class includes a static method, get_action_space(), which returns a list of these valid actions. This design ensures that outputs from decision-making algorithms or models are directly mappable to concrete, pre-defined actions in the racing simulation environment.

C. Environment Definitions
The Race Environment serves as the foundational setup for simulating a racing session, where each lap is represented by a row in a CSV file. The reset() method initializes the environment by setting the lap index to 0 and constructing the initial RaceState from the first row, effectively preparing the simulation for a new race. During each simulation step, the step(action) method is invoked to retrieve the current lap's data—this includes both the lap time and any additional pit stop time. An "effective lap time" is computed, which incorporates a penalty if a PIT_STOP action is chosen, thereby directly affecting the reward calculation; shorter effective lap times yield higher rewards as the reward function is designed as an inverse of the lap time. Following these computations, the environment advances by incrementing the lap index and updating the state accordingly, and it signals the end of the episode when there are no more laps to process.

D. DRQN Model and Training Components
The DRQN model is built using Keras' Sequential API through the function build_drqn_model(seq_length, feature_size, action_space_size). The model starts with an LSTM layer comprising 64 units to process the sequential state inputs, where the input shape is defined by the sequence length and feature size. Following the LSTM, a hidden Dense layer with 64 neurons and ReLU activation captures further complexities in the data. The network concludes with an output Dense layer that has as many neurons as there are actions in the action space, employing a linear activation function to yield Q-values for each possible action. Finally, the model is compiled using the Adam optimizer paired with a mean squared error (MSE) loss function, making it well-suited for regression tasks in Q-learning.

E. Training Loop
The train_drqn function orchestrates the training of the DRQN model across multiple episodes in a sequence-wrapped environment. It starts by resetting the environment to produce an initial state sequence and sets up two models: the main DRQN and a target network, the latter being periodically updated to match the main model for improved stability. Training proceeds using an epsilon-greedy strategy where, based on the exploration rate epsilon, the function either selects a random action or chooses the action with the highest predicted Q-value. As actions are executed, resulting experiences—comprising the next state, the reward, and a termination signal—are stored in a replay buffer. Once enough experiences accumulate, a random batch is sampled and used to compute target Q-values: terminal states receive the immediate reward as their target, while non-terminal states include the discounted maximum Q-value predicted by the target network. The main model is then updated using this batch, with hyperparameters such as the discount factor gamma, batch size, and epsilon decay (which gradually reduces the exploration rate down to a minimum value) guiding the training process. Additionally, the target network is synchronized with the main network at regular intervals defined by target_update_freq, ensuring that learning remains robust and stable throughout the training period.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

F. Training Execution and Model Saving

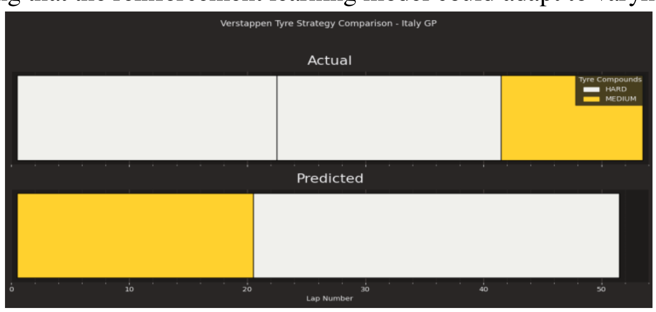
The implementation iterates over multiple folders, each representing a different racing season (e.g., 2021–2024), to expose the model to diverse racing conditions. For every CSV file found within these folders, the environment is reinitialized with the new dataset, and the training function is invoked to further train the DRQN on this fresh data.

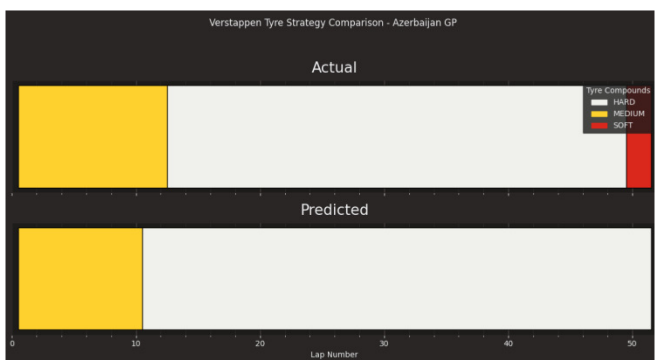
This iterative process allows the model to gradually generalize its strategy across various datasets. Once the training on all available datasets is complete, the final DRQN model is saved as "drqn_race_strategy_model.h5", making it available for future inference or additional training.

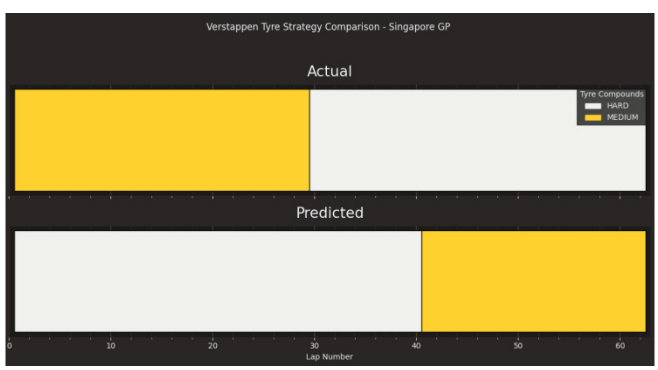
IV. EVALUATIONS AND RESULTS

The evaluation of the reinforcement learning model was conducted to determine which tire compound the model suggests and at which lap based on historical training data spanning from the 2021 season to the mid-season break of 2024. The model was trained using race data, including tire degradation rates, weather conditions, track-specific factors, and race incidents such as safety car deployments. Once the model was trained, it was tested against unseen race conditions to verify its decision-making accuracy. The primary objective was to compare its predicted pit stop strategies with those executed by professional F1 strategists.

For later seasons, the model's suggested strategies were evaluated against real-world strategies used by teams in live races. This comparison helped assess the alignment of the model's decision-making process with industry-standard race strategies and identify areas where AI-driven approaches could offer potential improvements. Monte Carlo simulations were employed to validate the reliability of the predictions, ensuring that the reinforcement learning model could adapt to varying race conditions.







International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

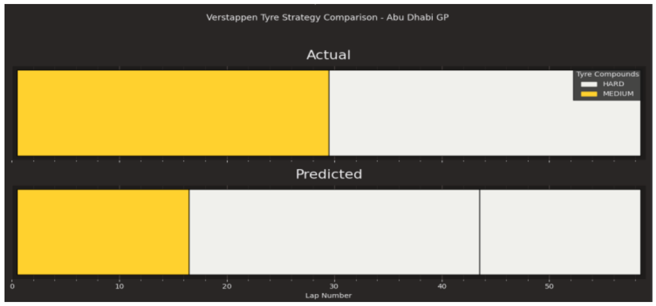


Figure 3. Strategy Prediction outputs

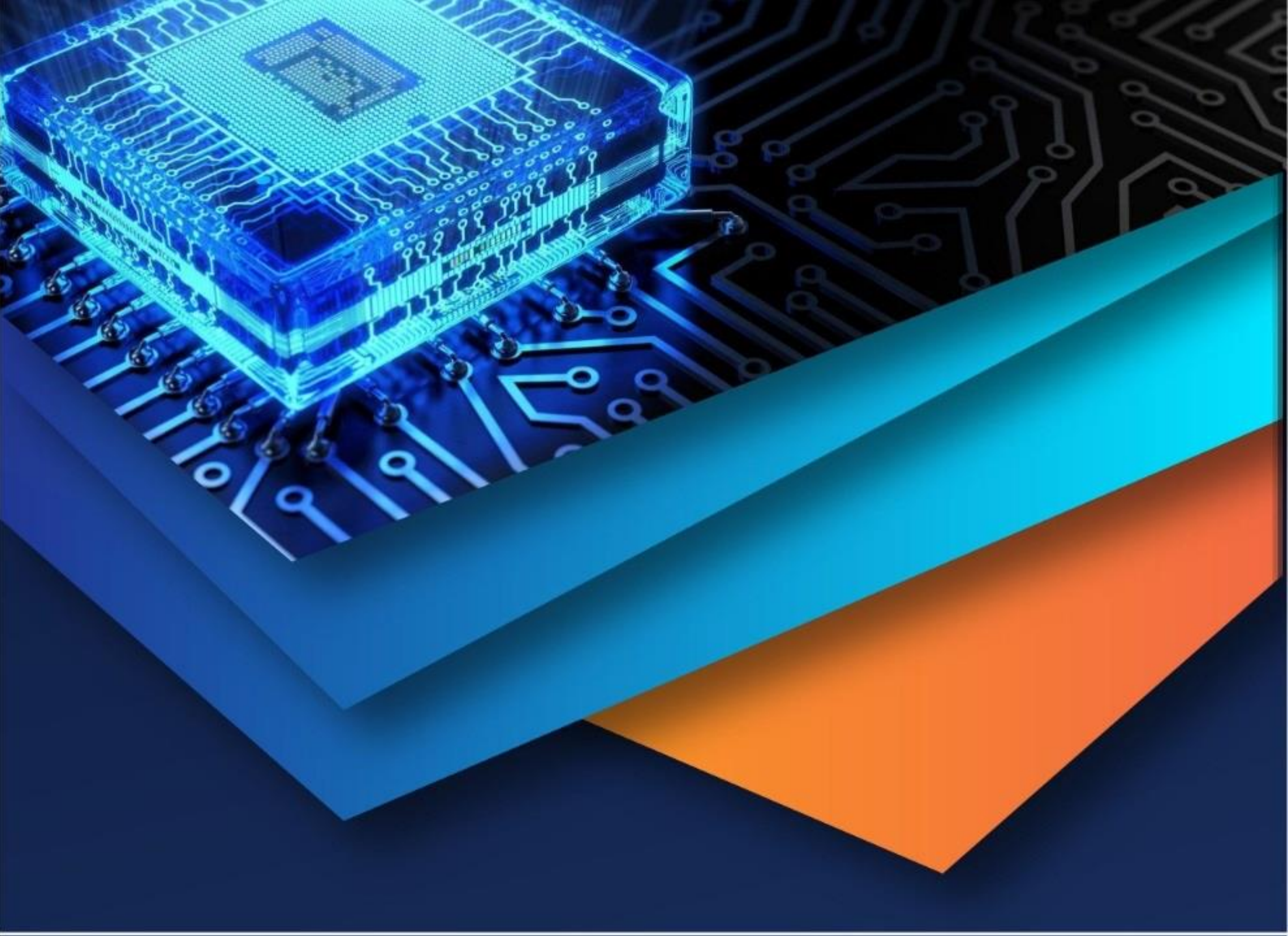
V. CONCLUSION AND FUTURE DIRECTIONS

Early deterministic models provided a foundational understanding of race dynamics, yet their inability to adapt to real-world uncertainties necessitated the evolution toward data-driven techniques. Machine learning, particularly deep learning, has enabled the extraction of complex patterns from vast amounts of telemetry and sensor data, thereby enhancing predictive accuracy and performance analysis.

Deep reinforcement learning (DRL) builds on these advancements by introducing an adaptive, interactive element to strategy development. DRL models—such as Deep Q-Networks (DQN) and Deep Recurrent Q-Networks (DRQN)—demonstrate significant promise in managing the inherent uncertainty and dynamic conditions of racing. By leveraging trial-and-error learning and incorporating probabilistic effects, these models can anticipate competitor behavior and adapt strategies in real time, offering a distinct competitive edge. It provides a structured framework to analyze both cooperative and adversarial behaviors, enabling teams to develop strategies that account for the actions and responses of opponents.

REFERENCES

[1] AD. Thomas, et al., "Explainable Reinforcement Learning for Formula One Race Strategy," arXiv preprint arXiv:2501.04068, 2025.
[2] F. Hojaji, A. J. Toth, J. M. Joyce, and M. J. Campbell, "AI-enabled prediction of sim racing performance using telemetry data," Computers in Human Behavior Reports, vol. 14, p. 100414, 2024.
[3] Z. Zhao, "Deep Neural Network-based lap time forecasting of Formula 1 Racing," Applied and Computational Engineering, vol. 47, pp. 61-66, 2024.
[4] H. Han, Z. Liu, M. Barrios, J. Li, Z. Zeng, N. Sarhan, and E. Awwad, "Time series forecasting model for non-stationary series pattern extraction using deep learning and GARCH modeling," Journal of Cloud Computing, vol. 13, no. 1, pp. 1-22, 2024.
[5] S. S. W. Fatima and J. Johrendt, "Deep-Racing: An Embedded Deep Neural Network (EDNN) Model to Predict the Winning Strategy in Formula One Racing," International Journal of Machine Learning and Computing, Vol. 13, No. 3, 2023.
[6] M. Boettinger and D. Klotz, "Mastering Nordschleife--A comprehensive race simulation for AI strategy decision-making in motorsports," arXiv preprint arXiv:2306.16088, 2023.
[7] P. Malik, A. Dangi, A. Singh, T. Asst, A. Pratap, S. Parihar, U. Sharma, and L. Mishra, "An Analysis of Time Series Analysis and Forecasting Techniques," IJARCCE, vol. 9, 2023.
[8] E.-J. van Kesteren and T. Bergkamp, "Bayesian analysis of Formula One race results: disentangling driver skill and constructor advantage," Journal of quantitative analysis in sports, vol. 19, no. 4, pp. 273-293, 2023.
[9] A. Bonomi, E. Turri, and G. Iacca, "Evolutionary F1 Race Strategy," In Proceedings of the Companion Conference on Genetic and Evolutionary Computation (GECCO '23 Companion), pp. 1925–1932, 2023.
[10] O. F. C. Heine and C. Thraves, "On the optimization of pit stop strategies via dynamic programming," Central European Journal of Operations Research, vol. 31, no. 1, pp. 239-268, 2023.
[11] L. Paparusso, M. Riani, F. Ruggeri, and F. Braghin, "Competitors-Aware Stochastic Lap Strategy Optimisation for Race Hybrid Vehicles," in IEEE Transactions on Vehicular Technology, vol. 72, no. 3, pp. 3074-3089, March 2023.
[12] A. Heilmeier, M. Graf, J. Betz, and M. Lienkamp, "Application of Monte Carlo Methods to Consider Probabilistic Effects in a Race Simulation for Circuit Motorsport," Applied Sciences, vol. 10, no. 12, p. 4229, 2020.
[13] A. Heilmeier, A. Thomaser, M. Graf, and J. Betz, "Virtual Strategy Engineer: Using Artificial Neural Networks for Making Race Strategy Decisions in Circuit Motorsport," Applied Sciences, vol. 10, no. 21, p. 7805, 2020.
[14] B. Peng, et al., "Rank position forecasting in car racing," In 2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 1057-1066, 2021.
[15] F. Dama and C. Sinoquet, "Time Series Analysis and Modeling to Forecast: a Survey," arXiv preprint arXiv:2104.00164, 2021.
[16] D. I. E. G. O. Piccinotti, et al., "Online Planning for F1 Race Strategy Identification," In International Conference on Automated Planning and Scheduling (ICAPS), 2021.
[17] A. Heilmeier, M. Graf, and M. Lienkamp, "A Race Simulation for Strategy Decisions in Circuit Motorsports," In 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2986-2993, 2018.
[18] J. Bekker and W. Lotz, "Planning Formula One race strategies using discrete-event simulation," Journal of the Operational Research Society, vol. 60, no. 1, pp. 26-32, 2009.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY