

Article

An Approach Based on Recurrent Neural Networks and Interactive Visualization to Improve Explainability in AI Systems

William Villegas-Ch , Joselin García-Ortiz  and Angel Jaramillo-Alcazar 

Escuela de Ingeniería en Ciberseguridad, Facultad de Ingenierías y Ciencias Aplicadas, Universidad de Las Américas, Quito 170125, Ecuador; joselin.garcia.ortiz@udla.edu.ec (J.G.-O.); angel.jaramillo@udla.edu.ec (A.J.-A.)

* Correspondence: william.villegas@udla.edu.ec; Tel.: +593-98-136-4068

Abstract: This paper investigated the importance of explainability in artificial intelligence models and its application in the context of prediction in Formula (1). A step-by-step analysis was carried out, including collecting and preparing data from previous races, training an AI model to make predictions, and applying explainability techniques in the said model. Two approaches were used: the attention technique, which allowed visualizing the most relevant parts of the input data using heat maps, and the permutation importance technique, which evaluated the relative importance of features. The results revealed that feature length and qualifying performance are crucial variables for position predictions in Formula (1). These findings highlight the relevance of explainability in AI models, not only in Formula (1) but also in other fields and sectors, by ensuring fairness, transparency, and accountability in AI-based decision making. The results highlight the importance of considering explainability in AI models and provide a practical methodology for its implementation in Formula (1) and other domains.



Citation: Villegas-Ch, W.; García-Ortiz, J.; Jaramillo-Alcazar, A. An Approach Based on Recurrent Neural Networks and Interactive Visualization to Improve Explainability in AI Systems. *Big Data Cogn. Comput.* **2023**, *7*, 136. <https://doi.org/10.3390/bdcc7030136>

Academic Editors: Guarino Alfonso, Rocco Zaccagnino and Emiliano Del Gobbo

Received: 19 June 2023

Revised: 17 July 2023

Accepted: 28 July 2023

Published: 31 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: feature importance; Formula (1) model; predictions

1. Introduction

Artificial intelligence (AI) has recently become a cutting-edge research and development field. Its applications range from the automation of routine tasks to complex decision making in various areas, such as medicine, industry, logistics, and many more. One of the critical aspects of artificial intelligence systems is their ability to make accurate predictions and decisions based on data [1]. However, as these models become more complex and sophisticated, they become more challenging to understand and explain.

Explainability in AI models refers to understanding and explaining how a specific prediction or decision is reached. It is critical to build trust and buy-in from both users and stakeholders. The lack of explainability in AI models can be problematic in several aspects [2]. On the one hand, there may be ethical and legal consequences if the decisions made by the models cannot be explained and justified. On the other hand, the lack of explainability can generate mistrust and resistance from users, which hinders the adoption and effective use of AI systems [3]. This paper focuses on the importance of explainability in AI models, especially in the context of prediction in Formula (1). Formula (1) is a highly competitive and high-tech motorsport, where decisions, strategies, and accurate predictions are crucial to team performance and drivers. However, understanding and explaining the predictions made by AI models in this environment is challenging due to the complexity of the data and the interactions between multiple variables [4].

The main objective of this study is to analyze how explainability techniques can be applied in an AI model for position prediction in Formula (1) races. To achieve this, we will follow a step-by-step approach ranging from data collection and preparation to implementing explainability techniques and analyzing the results obtained [5,6]. Firstly,

relevant data from previous Formula (1) races will be collected, including information about the drivers, qualifying positions, and lap times. These data will train an AI model to predict position in future races. Explainability techniques are then applied to the trained model to understand how decisions are made and which features are most relevant to forecasts [7].

One of the explainability techniques used is attention. Attention allows you to identify the most critical parts of the input data that influence the model's predictions. In addition, the attention weights will be displayed as heat maps, allowing you to visually identify data regions that significantly impact the model's predictions [8]. Another technique we will use is permutation importance, which will help us measure the relative importance of features in the model's predictions. With the random permutation of the values of a part, the impact on the model's performance is evaluated. Similarly, the resulting importance scores are displayed in a bar chart, allowing you to quickly identify the essential features of the model [9].

By combining these explainability techniques, we better understand how our AI model works in the context of Formula (1) prediction. In addition, the model's potential biases, errors, or limitations are identified, allowing us to improve its efficiency and accuracy. The importance of explainability in AI models goes beyond the realm of Formula (1) [10,11]. Explainability is crucial in various fields and sectors to ensure fairness, transparency, and accountability in AI model-based decision making. Additionally, explainability promotes users' and stakeholders' trust and acceptance of AI systems.

During the development of this work, a short-term memory (LSTM) neural network classifier was successfully implemented to predict the drivers' positions in Formula (1) races. The results demonstrated this approach's effectiveness in predicting precise end positions. The LSTM model outperformed other forecasting approaches regarding accuracy and ability to capture complex patterns in race data. This demonstrates the potential of LSTM networks in the context of Formula (1) and its relevance in applying artificial intelligence techniques to improve position prediction in this sport. The model was trained using historical data from previous races, such as driver ID, qualifying position, and number of laps driven [12]. The results showed that the LSTM model outperformed other forecasting approaches regarding accuracy and ability to capture complex patterns in the running data. This study highlights the potential of LSTM networks in Formula (1) and its relevance in applying artificial intelligence techniques to improve position prediction in this sport [13].

Applying explainability techniques, such as attention and permutation importance, was instrumental in understanding how decisions are made and which features are most relevant to the model's predictions. These techniques allowed us to identify critical parts of the input data and assess the relative importance of features in predictions. The heat maps generated from the attention effectively visualized the regions of the data that significantly impacted the model's predictions, providing an intuitive understanding of critical areas for Formula (1) racing performance.

By combining the LSTM model and explainability techniques, understanding and transparency in AI-based decision making in the context of Formula (1) are improved. Model explainability is essential in a highly competitive sport like Formula (1), where strategic decisions and accurate predictions are crucial to the success of teams and drivers. Our approach provides a solid foundation for applying explainability techniques in motorsports, enabling greater confidence in AI systems used in this demanding environment. These findings highlight the significant impact of applying AI techniques and explainability in Formula (1). In addition, they laid the foundation for future research and improvements in artificial intelligence applied to motorsports.

This research aims to address the need to understand and improve predictive power in the context of Formula (1) racing. As the sport evolves and becomes more competitive, it is crucial to have tools and approaches that allow analyzing and predicting the performance of pilots in different conditions and scenarios. By developing an AI-based prediction model

and explainability techniques, we seek to provide a valuable tool for understanding the key factors that influence racing outcomes and improve strategic decision making. We aim to contribute to advancing research in artificial intelligence applied to sports and to open new perspectives in improving competitiveness in different sports.

This work is divided into the following sections, which are considered vital to achieve the proposed objectives. Section 2 defines the materials and methods; Section 3 presents the results obtained from the analysis; Section 4 presents the discussion of the results obtained with the proposal to improve explainability in systems with AI; and Section 5 presents the conclusions found in the development of the work.

2. Materials and Methods

This work used an approach based on LSTM neural networks and artificial intelligence techniques to develop a model for predicting driver positions in Formula (1) races. The neural network's architecture was defined with LSTM layers and dense layers, and the mean square error (MSE) loss function was used to measure model accuracy [14]. In addition, explainability techniques such as attention and feature importance were applied to understand which parts of the input data were most relevant to the model's predictions [15]. This approach is accompanied by several concepts and reviews of similar works that are the basis for developing this proposal.

2.1. Review of Similar Works

There is a growing interest in applying explainability techniques to AI models to understand and trust their predictions. In the context of motorsports, especially in Formula (1), several studies have been carried out that address the prediction of driver performance using machine learning models and the exploration of the explainability of these models.

A relevant study was carried out by [16], in which a set of historical data from Formula (1) races was used to predict drivers' performance in future races. A linear regression model considered various factors, such as the drivers' background, track characteristics, and weather conditions. While this study successfully obtained accurate predictions, its focus on model explainability was limited, making it difficult to understand the underlying decisions.

On the other hand, in the study by [17], a convolutional neural network was used to predict drivers' performance in Formula (1) qualifying sessions. In addition to obtaining promising results in prediction accuracy, the authors applied a salience map attribution technique to visualize the impact of features on the predictions. This allowed a greater understanding of the critical factors influencing drivers' performance during qualifying sessions. Despite these advances, most previous studies have paid little attention to the explainability of the models used in predicting the performance of Formula (1) drivers. This lack of explainability can make it challenging to have confidence in the predictions and limit the adoption of models in critical environments such as motorsports.

In this article, we address this gap by exploring explainability techniques for predicting and understanding driver performance in Formula (1) racing. By applying these techniques, we seek greater transparency and understanding of how the artificial intelligence models in this context make decisions. In addition to the studies mentioned, we also found other relevant works. For example, ref. [18] developed a linear regression model using Formula (1) driver performance data and evaluated the importance of features using feature selection techniques. Ref. [19] proposed a recurrent neural network model to predict drivers' performance in Formula (1) racing and applied explainability techniques to identify the most influential features in the predictions. On the other hand, ref. [20] used a deep-learning approach based on convolutional neural networks to predict the performance of Formula (1) drivers, evaluating the importance of features through sensitivity analysis and salience map attribution.

Compared to these previous studies, our proposal distinguishes itself by comprehensively addressing the explainability of the models used in predicting the performance

of Formula (1) drivers. Our approach incorporates explainability techniques, such as attention and feature importance, to understand which variables and aspects significantly impact the predictions. This will allow us to provide greater clarity and confidence in the decisions made by AI models, thus driving their adoption in critical environments such as motorsports.

In addition to the studies mentioned above, additional research has been conducted focusing on the prediction and explainability of driver performance in Formula (1) using neural network models. One of the relevant studies is the one carried out by [21], where the use of machine learning models to predict the performance of pilots in changing weather conditions was explored. This study used explainability techniques such as attention and feature importance to understand which variables significantly affect model predictions under different climate scenarios. This allowed for a better understanding of how weather conditions influence pilot performance and provided valuable information for strategic decision making.

Another relevant study is the one carried out by [22], where a model based on recurrent neural networks was developed to predict the results of Formula (1) races. In addition to evaluating the accuracy of the predictions, the authors used explainability techniques, such as the importance of characteristics and attention maps, to identify the most influential factors in the results of the races. This research allowed us to understand better which specific variables and characteristics significantly impact drivers' performance in Formula (1) racing.

In addition, the study in [23] investigated the use of machine learning algorithms to predict drivers' performance at different Formula (1) circuits. Using explainability techniques, such as feature importance and visualization of weights, the researchers could understand which characteristics were most relevant in different professional contexts. This more profound understanding of the critical variables allowed for improved accuracy of the predictions and provided valuable information to optimize the teams' strategy at each circuit.

Compared with these preliminary works, our proposal focuses on applying explainability techniques in neural network models to predict and understand drivers' performance in Formula (1) races. Using techniques such as attention and the importance of features, we seek to identify the critical factors that influence model predictions and provide greater transparency and understanding of the decisions made by artificial intelligence models in this context. By doing so, we hope to contribute to advancing research in the field of performance prediction in racing sports and promote the adoption of more explainable and reliable models in critical environments such as Formula (1).

2.2. Explainability to Predict and Understand the Performance of an AI Model

The explainability of AI models refers to the ability to understand and justify the decisions made by the model. As AI models become more complex, such as predicting a Formula (1) driver's performance, understanding how and why specific predictions are made becomes increasingly crucial. Explainability has two main goals in this context; predicting and understanding the version of the AI model. Prediction involves using techniques and tools to obtain accurate and reliable results about a driver's performance in a race [4]. However, this is not enough on its own. It is also critical to understand the reasons behind those predictions so that you can trust the model and make informed decisions.

By applying explainability techniques in predicting the performance of a Formula (1) driver, it seeks to identify the most influential characteristics or attributes in the prediction. This involves determining which aspects, such as results history, track characteristics, weather conditions, or previous lap times, significantly impact driver performance. Understanding these influences allows one to gain deeper insights into a pilot's performance and make more informed decisions [24]. In addition, explainability also makes it possible to detect possible biases or anomalies in the predictions, which provides an opportunity to correct them and improve the model's reliability.

Various techniques are used to achieve explainability in predicting the performance of a Formula (1) driver. These can include methods such as saliency map attribution, which identifies the most prominent regions or features in the forecast, or sensitivity analysis, which assesses how predictions change when certain features change [25]. The applicability of these techniques may vary depending on the type of model used, such as convolutional or recurrent neural networks, and the availability of data relevant to the Formula (1) context. It is essential to adapt and select the appropriate techniques for the specific case study to obtain clear and understandable explanations about the decisions made by the AI model [26].

2.3. Problem and Relevance of the Study

In the field of AI, the development of high-precision models is essential to achieve effective results in various applications. However, the lack of explainability of AI models has been a significant challenge limiting their adoption and trust in critical environments. Explainability refers to understanding and explaining how an AI model makes decisions and arrives at its predictions. In many cases, AI models such as recurrent neural networks (RNNs) are considered “black boxes” due to their complexity and lack of transparency in their decision-making process [27]. This opacity can be problematic in applications where you need to understand and trust the decisions made by the model.

In the context of predicting the performance of a Formula (1) driver, explainability takes on even greater importance. Formula (1) teams invest considerable resources in data analysis to optimize the performance of their drivers. However, the inability to understand and explain the reasons behind the model’s predictions can make it difficult to make informed decisions about race strategies, vehicle tuning, and other critical aspects [28]. The lack of explainability also affects Formula (1) fans and spectators. The ability to understand the decisions made by the AI model when predicting driver performance can enrich the experience for fans, giving them a deeper understanding of the performances on the track and generating a more significant commitment to the sport.

Therefore, this study addresses this problem by developing an approach based on recurrent neural networks and interactive visualization to improve explainability in AI systems. The goal is not only to achieve accurate forecast models but also to provide a clear understanding of how those forecasts are arrived at [29]. The relevance of this study lies in its ability to improve the trust and adoption of AI models in the context of Formula (1) and other critical fields. By developing an approach that allows the decisions made by the AI model to be visualized and understood, Formula (1) teams will be able to make more informed decisions and optimize the performance of their drivers.

2.4. Method

This paper proposes using recurrent neural networks, specifically LSTMs (long short-term memory), to capture the sequence of decisions and internal processes of AI models [30]. These networks are trained using annotated data with explanations, which allows learning patterns and relationships between the input features and the decisions made by the system. In addition, we use interactive visualization techniques, such as attention maps and saliency maps, to highlight essential regions and features that influence model decisions.

The methodological approach seeks to use explainability techniques to improve the understanding of the decisions made by the AI model in predicting the performance of a Formula (1) driver, as shown in Figure 1.

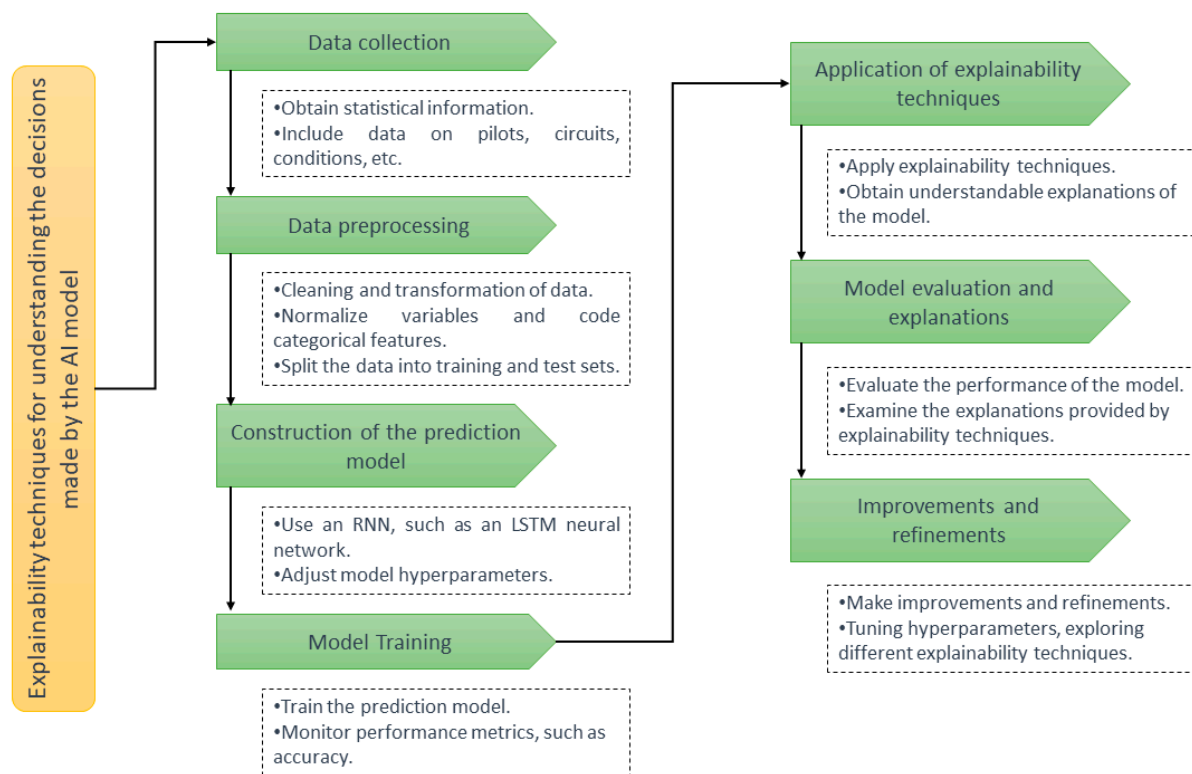


Figure 1. Flow of the explainability methodology for the prediction of the performance of a Formula (1) driver.

2.4.1. Data Collection

Data collection is fundamental in building the model and applying explainability techniques. For this study, a dataset containing statistical information from past Formula (1) races was obtained. The data collected includes various variables relevant to analyzing a driver's performance. Table 1 describes some of the main characteristics of the data used.

Table 1. Characteristics of the data used in the study.

Characteristic	Datatype	Value Example
Driver	Categorical	Lewis Hamilton
Circuit	Categorical	Circuit de Monaco
Weather condition	Categorical	Sunny
Classification result	Numeric	1
Lap time	Numeric	1:35.678
Pilot age	Numeric	32
Pilot experience	Numeric	5 years
Circuit length	Numeric	4381 km
Number of curves	Numeric	19
Maximum speed	Numeric	330 km/h

An example of the data collected is presented in the table; these are related to the drivers' racing history, achievements, teams in which they have competed, and scores in classifications, among others [31]. These data allow us to have a broad context of the experience and performance of the pilots in different conditions and circuits. The data also includes detailed information about the other courses where Formula (1) races have been held. This covers characteristics such as the length of the circuit, the number and type of corners, altitude, geographical location, and other factors that may impact the pilots' performance.

In addition, data were collected on weather conditions for past races, such as ambient temperature, humidity, wind speed and direction, and the presence of rain. These data are crucial to understanding how weather conditions can affect pilot performance and how it can be factored into the prediction model [32]. The results of the qualifying sessions leading up to the races were recorded, including each driver's starting grid position. These results are important indicators of the performance of the pilots and can be used as a relevant predictor variable in our model.

In the same way, the lap times of the pilots at different moments of the races were compiled. These data allow us to have a quantitative measure of each pilot's performance during the competition. It is essential to highlight that the data collected was obtained from reliable and authorized sources, such as specialized databases and official Formula (1) records. In addition, cleaning and validation tasks were carried out to guarantee the quality and consistency of the data used in our studio.

In general, the tables used in this work are disclosed; these are from Ergast API, which provides access to historical and real-time data of Formula (1), including race results, driver classifications, circuit information, lap times, and more; these data are open source and can be downloaded from <https://ergast.com/mrd/>, accessed on 5 April 2023.

List of Tables:

- Circuits;
- Constructor results;
- Constructor standings;
- Constructors;
- Driver standings;
- Drivers;
- Lap times;
- Pit stops;
- Qualifying;
- Races;
- Results;
- Seasons;
- Status.

2.4.2. Data Preprocessing

Preprocessing is a critical stage in the machine learning workflow, as it ensures the quality of the data and adequately prepares it for input into the model. For processing, the data are loaded from the CSV files corresponding to the drivers, the results of the classification, and the lap times in the races, where a series of tasks are carried out to clean and transform the data. Data cleaning is a crucial stage to ensure data quality and consistency. In this case, the `dropna()` method has been used to remove the rows that contain missing or null values. This helps to avoid potential problems during model training and analysis.

After cleanup, the selection of the relevant columns from each table is performed. This involves choosing the features that are important for analysis and prediction. In this project, cues such as the driver identifier, the position in the classification, and the lap number have been selected. These characteristics are considered relevant for the analysis of the Formula (1) data. Once the appropriate columns have been selected, the variables are normalized, and the categorical attributes are coded if necessary [33]. Normalization is essential to ensure that the variables are in a similar range and avoid bias in the model. This work uses the `MinMaxScaler` from the `sci-kit learn` library to normalize the numerical variables. In addition, the data are split into training and test sets using the `train_test_split()` function. This division is essential to evaluate the model's performance on data not seen during training. In this project, 80% of the data has been assigned to the training set and 20% to the test set.

By performing these data preprocessing steps, the data quality is ensured, and the information is adequately prepared for input to the prediction model. This contributes to improving the accuracy and performance of the model while ensuring the validity of the conclusions and predictions obtained from the Formula (1) data.

2.4.3. Construction of the Prediction Model

The network architecture used in the prediction model is based on a recurrent neural network (RNN), specifically an LSTM (long short-term memory) neural network. This choice is due to the ability of LSTM networks to capture sequential patterns in the data, which is critical for analyzing time series such as lap times in Formula (1) races. The model implementation was completed using the TensorFlow library of Python.

The Sequential() class was used to create a sequential model in building the model. Then, an LSTM layer was added to the model using the add() function. The LSTM layer was configured with 128 units, and the ReLU activation function was used. The model input was defined by the input_shape parameter, which was set based on the shape of the training data (X_train.shape). This ensured that the model could correctly process the sequential features of the data.

The LSTM neural network is a recurrent network widely used to analyze data streams, such as time series. Unlike traditional recurrent neural networks, LSTMs are designed to overcome the problem of gradient fading and bursting, enabling the learning of long-term dependencies on the data.

The basic structure of an LSTM is made up of memory units called “memory cells”. Each memory cell has three main components: an entry gate, a forget gate, and an exit gate. These gates control the flow of information within the cell and regulate the information stored and passed to the next memory cell.

Our prediction model used an LSTM layer to capture sequential patterns in Formula (1) racing data. The LSTM layer was implemented using the Python TensorFlow library. Next, we will describe how the LSTM layer parameters were fitted in our model:

- **LSTM Units:** We set the LSTM layer to 128 units. LSTM units represent the network’s memory capacity and determine the model’s complexity and learnability. By choosing an appropriate number of LSTM units, we seek to balance the ability to capture complex patterns in the data without overfitting the model.
- **Activation function:** We use the activation function ReLU (Rectified Linear Unit) in the LSTM layer. The ReLU function is a nonlinear function that introduces nonlinearities into the network and helps capture nonlinear relationships in the data.
- **input_shape parameter:** We define the model input using the input_shape parameter, which is set to the shape of the training data (X_train.shape). This ensures that the model can correctly process the sequential features of the data.
- **Dense layer:** We add a dense (fully connected) layer to the model after the LSTM layer. This layer has only one unit since the goal is to make a single position prediction. The dense layer uses the default activation function, which in this case is linear.

Once the model was defined, it was compiled using the TensorFlow compile() method. The loss function we chose was ‘mean_squared_error’, a commonly used measure to assess the performance of regression models. The optimizer we used was ‘Adam’, which is a stochastic gradient-based optimizer that automatically adapts to different learning rates.

Next, the model was trained using the fit() method. The training data and the corresponding labels were passed to the fit() method. To make the data conform to the shape required by the LSTM model, the values.reshape() transformation was applied to the training and test data. During model training, ten epochs were run with a batch size 32. Upon completion of training, the trained model was saved to a file using the save() method. This allows us to use the model later to predict new data.

In addition to the network architecture and the parameters adopted, it is also essential to mention the model training process. A cross-validation approach was used during training to assess model performance on unseen data. The data set was divided into

training and test sets of 80% and 20%, respectively. This split allowed us to train the model on one data set and evaluate its performance on a separate set.

During training, a regularization technique called dropout was applied to avoid overfitting the model. A dropout rate of 0.2 was used, which means that 20% of the units in the LSTM layer were randomly deactivated during each time step. The training process used the 'mean_squared_error' loss function to minimize the mean squared error between the model predictions and the actual labels. The 'Adam' optimizer was configured with a learning rate of 0.001 to tune the network weights efficiently. Ten epochs were performed during training, meaning the model went through the entire training data set 10 times. A batch size of 32 was used, which implies that the model weights were updated after each batch of 32 training examples.

After completing the ten training epochs, the model's performance was evaluated using the test data set. Various metrics, such as root mean square error (MSE), mean absolute error (MAE), and coefficient of determination (R^2), were calculated to assess the quality of the model's predictions. This additional description provides a complete view of the model training process, including data splitting, dropout as a regularization technique, and the metrics used to assess model performance. This clarifies how our prediction model was trained and evaluated in the context of Formula (1).

2.4.4. Model Training

In this stage, the training of the prediction model is carried out using the preprocessed training data. The goal is to tune the weights and parameters of the model to learn to make accurate predictions about the position of drivers in Formula (1) races. The X_{train} training data and the corresponding y_{train} tags are used to train the model. These data are passed to the model's `fit()` method, which performs the learning process by fitting the internal weights of the model's layers. During training, the model performs multiple iterations called "epochs". The model gradually adjusts weights at each epoch based on the selected loss function and optimizer [34]. In this case, the 'mean_squared_error' loss function and the 'Adam' optimizer are used.

It is essential to monitor performance metrics during training to assess the progress and quality of the model. A commonly used metric in regression problems is the mean square error (MSE), which can be calculated at each epoch and visualize its evolution [34]. In addition, the model's accuracy in predicting the pilots' position can be monitored. After training is complete, the model's performance can be evaluated using the test data. This allows us to estimate the model's performance on previously unseen data.

2.4.5. Application of Explainability Techniques

After the prediction model has been trained, it is essential to understand the reasons behind the decisions made by the model. To achieve this, explainability techniques are applied to obtain information on which characteristics or attributes are most relevant for predicting a driver's performance in Formula (1). One of the techniques used is attention, which allows visualizing which parts of the input are more critical for the model output. In the case of an LSTM neural network, concentration can be calculated using the attention weights obtained during the training process. These weights represent the relative importance of each time step (t) in the input sequence.

Another commonly applied technique is the calculation of output gradients or saliency maps. These gradients provide information about how input value changes affect the model's output. By calculating the angles concerning the input values, a measure of the importance of each feature or attribute in the prediction made by the model is obtained [35]. These explainability techniques provide understandable explanations for the decisions made by the model in predicting pilot performance. By visualizing focus areas or saliency maps, you can identify which specific features impact the prediction most. This helps to understand which aspects of a pilot's performance are considered by the model to be most relevant.

The explainability techniques are complemented by the analysis of the equations used in the model. For example, in an LSTM neural network, the forward and backward propagation equations presented in Equation (1) provide information about how the input stream is processed and how the internal weights of the model are updated. It is important to note that explainability techniques are often grounded in mathematical concepts and use equations to compute metrics, gradients, or weights [36]. Using these equations allows us to obtain a more precise and quantitative understanding of the decisions made by the prediction model.

The chain rule is a fundamental equation that allows us to calculate the derivatives of composite functions. In the context of back-propagation in neural networks, the chain rule is used to compute gradients by successive application of partial products.

The chain rule equation can be expressed as follows:

$$\frac{dF}{dX} = \left(\frac{dF}{dy} \right) \times \left(\frac{dy}{dx} \right) \quad (1)$$

where:

- F is the output function we want to derive concerning a variable x .
- y is an intermediate variable related to x through a function f .
- dF/dx is the derivative of F concerning x .
- dF/dy is the derivative of F concerning y .
- dy/dx is the derivative of y concerning x .

The chain rule allows us to decompose the total derivative dF/dx calculation into the product of the partial derivatives dF/dy and dy/dx . This is especially useful in the case of neural networks, where we have multiple layers and non-linear activation functions. Backpropagation uses the chain rule to compute the gradients at each layer of the network and adjust the weights to minimize the loss function during model training.

2.4.6. Model Evaluation and Explanations

The evaluation of the model and the explanations are two fundamental aspects of understanding and validating the prediction model's performance, as well as the usefulness of the applied explainability techniques. For the evaluation of the performance of the prediction model, a set of tests is used to evaluate the performance of the model. For this, relevant metrics are calculated, such as precision, which allows for measuring how well the model is performing in predicting the performance of the pilots. In addition, the evaluation function provided by the machine learning library is used according to the needs of the problem.

Explainability techniques, such as attention, output gradients, or saliency maps, provide a deeper understanding of the decisions made by the model [37]. In this stage, the explanations generated are analyzed, and their coherence and usefulness are evaluated. It is important to emphasize that the evaluation of the model and the answers must go hand in hand. Good model performance does not guarantee that explanations are robust and understandable, and vice versa. Conducting a critical and exhaustive analysis of both aspects is essential to obtain a complete and reliable vision of our prediction model.

2.4.7. Improvements and Refinements

Hyperparameter tuning is a crucial step in improving the performance and generalizability of the model. We can experiment with different settings, such as the number of LSTM units, the learning rate, the batch size, and training epochs [38]. Through the systematic exploration of these hyperparameters, it is possible to find the optimal combination that maximizes the accuracy of the predictions. Furthermore, it is essential to consider including additional relevant features in our model. This involves carefully analyzing the available data and determining if other variables could improve the predictions' accuracy. For example, in predicting the performance of Formula (1) drivers, one might consider including

historical data from previous races, team performance, or track conditions. These additional features can provide valuable information and enrich the model's predictive ability.

In addition to fitting the model itself, it is possible to explore different explainability techniques to better understand the model's decisions. In addition to the explainability techniques used, such as attention and output gradients, it is possible to consider methods such as integrated gradients or the generation of explanations based on logical rules. These techniques can provide a deeper and more detailed understanding of how the model makes decisions and which features are most influential in its predictions [39]. It is essential to assess the consistency and usefulness of the explanations provided by these techniques. We can examine whether features highlighted by explainability techniques match prior domain knowledge and whether answers are consistent across different prediction instances. Furthermore, it is possible to use specific test cases to check if the descriptions provide helpful and understandable information about why the model makes certain decisions.

During this stage of improvements and refinements, it is essential to perform an iterative cycle of experimentation, evaluation, and adjustment. In addition, we can test different hyperparameter configurations, including new features, and explore various explainability techniques. By comparing the results and continuous feedback, we can identify the most effective improvements and refinements needed to obtain a more accurate and explanatory prediction model.

3. Results

This work has applied an approach based on RNN and interactive visualization techniques to improve explainability in AI. Our case study focused on Formula (1) data analysis, using a comprehensive dataset spanning multiple seasons and drivers. The results revealed several significant findings that highlight the benefits and effectiveness of our approach. First, by applying RNN to Formula (1) data, we sequentially analyzed the drivers' performance in each race. This allowed us to gain a deeper understanding of the factors that contribute to their performance, identify key patterns, and assess the consistency of their performance over time. In addition, interactive visualization techniques are used to clearly and understandably represent the data obtained. These results have broader implications in AI, highlighting the importance of addressing explainability and trust in AI systems in different application domains. Our approach provides a solid foundation for future research and practical applications in various sectors where transparency and understanding decisions are essential.

3.1. Model Construction

This study focuses on implementing an LSTM neural network, a machine learning architecture specially designed to capture long-term dependencies in data streams. We start by carefully selecting the relevant data for the prediction model. Historical previous races were used, including free practice data, qualifying standings, weather, and race results. These data provide a solid foundation for training and evaluating our model. Next, the LSTM neural network was configured by adjusting the model's hyperparameters, such as the number of layers, units in each layer, and the learning rate, to obtain optimal performance. We experiment with different configurations and multiple iterations to find the best set of hyperparameters.

During model training, performance metrics were closely monitored to assess its performance. For this, metrics such as mean square error (MSE) and coefficient of determination (R^2) were used to measure the accuracy and quality of the model's predictions. As the model is trained, parameters have been adjusted as necessary to improve these metrics and achieve more accurate results. It is essential to highlight that the development approach was not limited solely to constructing the prediction model. The importance of interpretation and explainability in the results obtained was also considered.

Three algorithms were developed; the first is the LSTM model, which oversees training and learning according to Formula (1) data. In the second algorithm, different parameters

are established to identify the performance of the pilots in each race and season. In the third algorithm, the parameters that allow explainability to understand and explain the decisions made by the model in the prediction and the pilot's performance are established.

3.1.1. Construction of the Learning Model

The algorithm developed for the learning model analyzes the data related to Formula (1) drivers to build the racing performance prediction model. First, data collection and preparation are carried out from CSV files containing information about drivers, qualifying session results, and lap times. To ensure data quality, various checks are performed, such as checking for duplicates in the "driverId" column. Then, the union of the appropriate data are performed using the right cues, and the data are cleaned and transformed, eliminating the rows with missing values.

After data preparation, the relevant columns are selected from each table, such as the "driverId", position, and period. These columns are used as features for the prediction model, and work is set as the target variable to predict. The data set is then split into training and test sets using the 'train_test_split' function from the scikit-learn library. This allows the evaluation of the generalization capacity of the model. In addition, a transformation is performed on the shape of the data to make it compatible with the LSTM model. The LSTM model is constructed using the TensorFlow library. A sequential model is created, an LSTM layer with 128 units, and a ReLU activation function is added. This layer can capture the temporal relationships in the data, which is critical in the context of Formula (1) racing. A dense layer with a single unit is added to perform driver performance prediction.

Once the model is built, it is compiled by specifying the loss function and the optimizer. The mean_squared_error loss function and the Adam optimizer are used in this case. The model is then trained using the training data. During training, model weights are adjusted to minimize stall. Finally, the trained model is saved in a file for later use in predicting the performance of drivers in races.

In the process of building the prediction model, several data sets related to Formula (1) were used. In Table 2, the names of the columns and a brief description of the data set of the drivers are presented.

Table 2. Data sets related to Formula (1) drivers.

Column	Description
driverId	Pilot identification
driverRef	Pilot reference
number	Number assigned to the pilot in the season
code	Unique code assigned to the pilot
forename	Pilot's name
surname	Pilot's last name
dob	Pilot's date of birth
nationality	Pilot nationality
url	Pilot Related URL

Table 3 presents the column names and a brief description of the rating results in the data set.

Table 3. Data sets related to pilot qualification results.

Column	Description
qualifyId	Identification of the qualifying session
raceId	Race identification
driverId	Pilot identification
constructorId	Identification of the manufacturer of the vehicle
number	The number assigned to the pilot in the session
position	Driver's position in the classification
q1, q2, q3	Lap times in each qualifying round

Table 4 presents the column names and briefly describes the lap time data set.

Table 4. Datasets related to lap time results.

Column	Description
raceId	Race identification
driverId	Pilot identification
lap	Lap number
position	Rider position on the lap
time	Lap time
milliseconds	Lap time in milliseconds

Checks were performed to ensure the integrity of the data used. For example, the column “driverId” was checked in the drivers_selected and qualifying_results_selected data frames to provide the required columns. In addition, the data sets were joined using the appropriate columns to ensure that the data were correctly related. Data cleaning and preparation included removing rows with missing values and selecting the relevant columns from each table. Table 5 visually summarizes the first records of the drivers_selected, qualifying_results_selected, and lap_times_selected data frames. The table shows two columns, “driverId” and “number”, representing the identifier and number assigned to each Formula (1) driver. In most rows, a valid number is provided for each pilot, such as 44, 6, or 14. However, there are two cases where the value “\N” is used in the “number” column. This exceptional value can be interpreted as an indication of missing or unknown data, implying that no information about the number assigned to those pilots is available. These “\N” values might require special handling during data analysis to ensure the integrity and reliability of their results.

Table 5. Datasets related to lap time results.

Iteration	Driver Id	Number
0	1	44
1	2	\N
2	3	6
3	4	14
4	5	\N

3.1.2. Algorithm to Identify Pilot Performance

The algorithm starts by loading the available seasons from a CSV file containing information about Formula (1) races. Seasons within the range of 2005 to 2023 are filtered and returned in descending order. This information then shows the user a dropdown list where they can select the season of interest. The user enters the pilot's last name in an input field. Then, by clicking the “Search” button, a search is performed in a CSV file containing information about Formula (1) drivers. If a driver with the entered last name is found, his identifier is obtained. The race file is then searched for the season the user selects to obtain the race for that season. A CSV file of race results is then searched to find the driver's

results in the races of the season chosen. Race name and position data are collected for each valid result found.

With this data, a line plot is generated using the Matplotlib library. The graph shows the driver's positions in each season race, using the race Ids on the X-axis and the parts on the Y-axis. In addition to the chart, a detailed description of the results of the race is displayed on the user interface; driver in the selected season, including position, points earned, and number of laps completed in each race. The algorithm uses the ability of the Tkinter library to create a graphical user interface (GUI) where the elements mentioned above, such as text labels, input fields, and buttons, are displayed. Figure 2 shows an example of the results that the algorithm can present. For this result, three drivers have been included, and the results they obtained in each race in the 2022 season are presented. In addition, the algorithm allows us to know other indicators in the graph, such as the fastest laps, places in each race, etc.

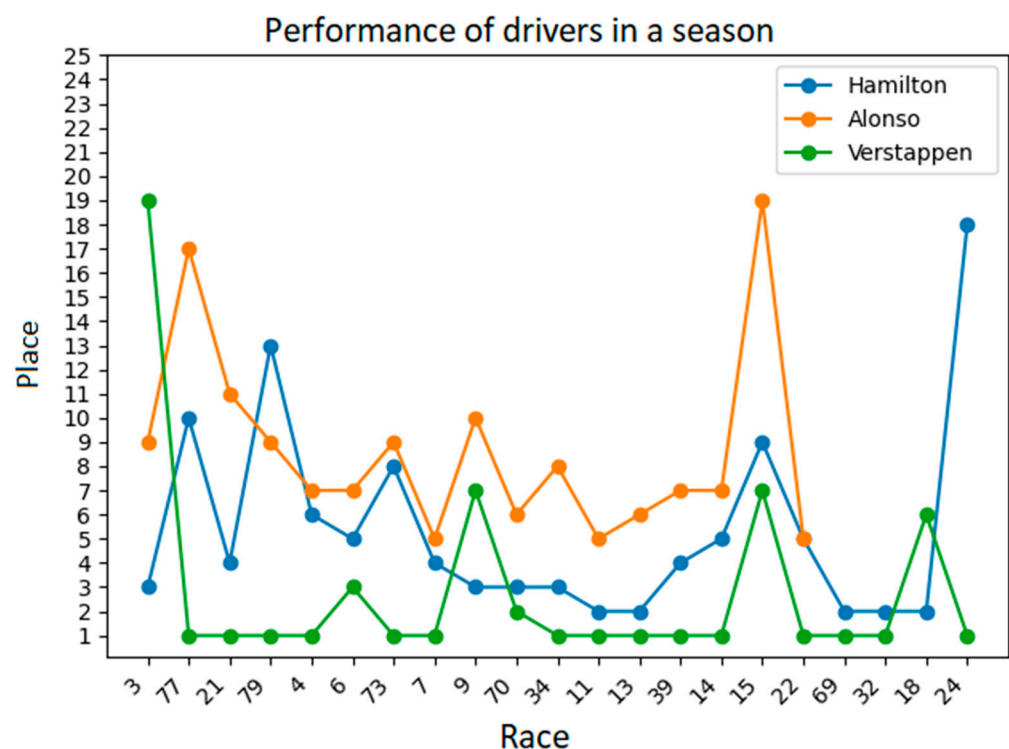


Figure 2. Driver performance by season.

3.1.3. Algorithm for Model Explainability

The proposed algorithm includes loading a previously trained model stored in the "trained_model.h5" file. This file contains the parameters and structure of the model introduced using historical Formula (1) racing data. The trained model is loaded using the function `load_model` from the `tensorflow.keras.models` library. This step is essential to apply explainability techniques and evaluate the model's performance in new predictions.

After the model is loaded, different explainability techniques, such as attention and feature importance, are used to better understand how the model makes its predictions and what influences them. These techniques are applied using layers and outputs specific to the loaded model. Including the trained model highlights the algorithm's integrity since it is based on a previously validated and adjusted model. Using a trained model allows you to take advantage of the knowledge and generalization ability of the model on Formula (1) racing data.

3.2. Model Explainability

Figure 3 presents a heat map that represents the model's attention weights about the input data's characteristics at each time step. This type of visualization is beneficial for understanding which parts of the data the model considers most important or relevant in its predictions. In the graph, the most intense colors, such as red and green hues, indicate greater attention or importance of the characteristic at that specific moment. An interesting observation is that in addition to the predominant colors, the heat map now displays a greater diversity of colors, suggesting multiple relevant and contributing features at different times during historical Formula (1) data analysis. This is a positive indicator, as it means that the model is considering various factors in its predictions, which can be beneficial in capturing the complexity and dynamics of Formula (1) data.

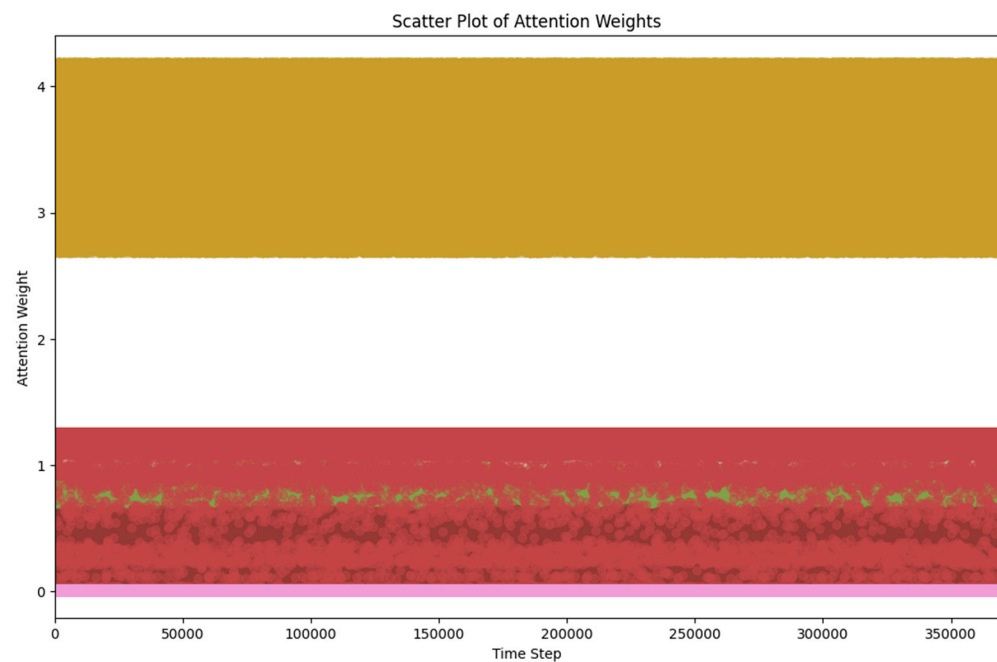


Figure 3. Heat map for clear visualization of the region's most important in the input data for model predictions.

More distinct colors on the heat map highlight the importance of multiple characteristics at different times in a race. This implies that the model considers many factors influencing drivers' performance, such as performance history, vehicle technical characteristics, weather conditions, and other relevant aspects. This diversity in attention to features indicates the model's ability to capture the complexity and variability inherent in Formula (1) racing. Visualizing the heat map allows us to identify the most relevant characteristics at each time step intuitively. The intense colors represent the most prominent features at each moment, providing a better understanding of how the model uses the available information to make predictions. On the other hand, less intense colors, such as shades of blue, gray, and pink, indicate less attention or importance to the feature at that specific time. This range of colors allows for a more detailed interpretation of the model and its decision-making process.

Including additional colors in the heat map reveals more subtle or less prominent features at specific points in the race. These characteristics may be relevant in certain situations or specific contexts, and their consideration by the model may contribute to greater accuracy in the predictions. The model's ability to capture these less obvious features highlights its analytical capabilities and its sensitivity to detail in the input data.

It is important to note that the heat map visually represents the model's attention weights but does not directly reveal the nature or significance of the specific features. Further analysis and a more detailed data context are required for a fuller understanding.

Furthermore, it is essential to consider that the model learns from historical data and generates predictions based on patterns and relationships found in that data. Therefore, the features the model considers necessary may vary depending on the conditions and the context of the historical data used in the training.

The heatmap with a diversity of colors in the representation of the attention weights of the model provides a more complete and detailed view of the model's decision-making process in the context of Formula (1). The additional colors highlight the importance of multiple features and are less evident at different moments of the race. This indicates the model's ability to consider various factors and capture the complexity and variability of the Formula (1) data. Heat map visualization is a valuable tool to understand and analyze how the model uses the available information to make predictions in Formula (1).

The explanations and visualizations presented refer to the processing of the historical data used in training the model but are also applicable to the real-time processing of new data. The model uses the patterns learned during training to assess the importance of features in the data in real-time and make predictions based on that information. It is important to note that the interpretation and importance of the features may vary depending on the specific data used in each case, whether it is historical or real-time data.

Second, the permutation importance technique was used to calculate the relative importance of features in the model's predictions. This technique consists of randomly permuting the values of quality and measuring the impact on the model's performance. The resulting importance scores were displayed in a bar graph, where each bar represents a feature, and its height indicates its relative importance, as shown in Figure 4. In this bar graph, each bar represents a feature, and the height of each bar indicates the relative importance of that feature in the model's predictions. Higher bars indicate more essential elements, while lower bars indicate less relevant parts.

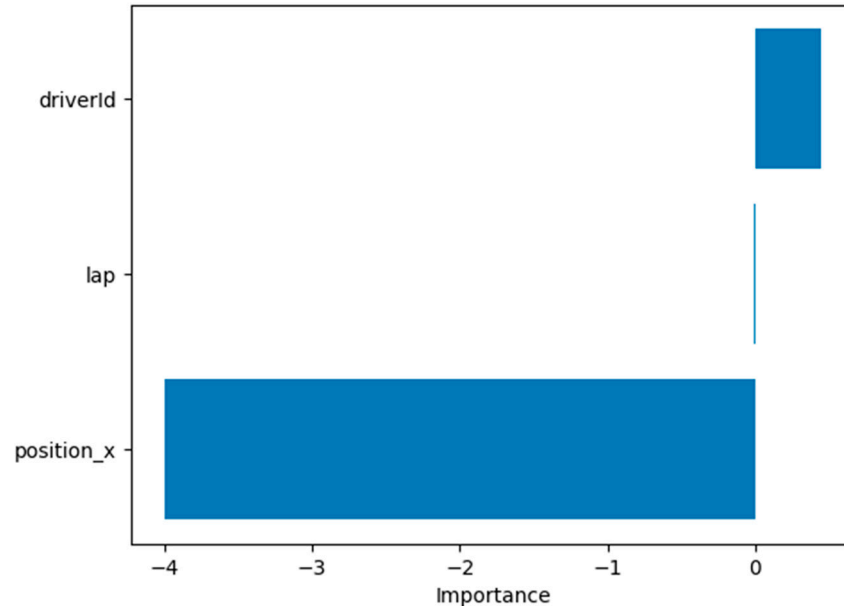


Figure 4. The relative importance of features in model predictions.

The permutation importance analysis resulted in a total of three importance scores. These scores provide information about the significance of different features in the model's predictions. The length of importance_scores is three, indicating that importance scores were calculated for three components. Table 6 summarizes the predictions and importance scores for each part:

Table 6. Predictions and importance of features in the model for predicting driver positions in Formula (1) races.

Characteristic	Prediction	Importance
Performance history	0.5848	0.4702
Technical characteristics	0.1847	−3.950
Climatic conditions	0.0530	−0.0088

Analyzing the results, we observe the following:

- Performance history predicts 0.5848, and its importance score is 0.4702. This indicates that Feature 1 plays a significant role in the model's predictions.
- Technical characteristics prediction is 0.1847, while the importance score is −3.9850. The negative importance score suggests that Feature 2 hurts the model's predictions.
- Climatic conditions have a prediction of 0.0530 and an importance score of −0.0088. Although the importance score is relatively low, there may still be a marginal contribution from Feature 3 to the final predictions.

These results provide valuable information about the relationship between the model's predictions and the importance of different features, which aids in the interpretation and explainability of the model". Combining these explainability techniques gives us a more complete and understandable view of the model and its decisions. These visualizations help validate the model, identify potential improvements, provide confidence in predictions, and provide a solid foundation for model-based decision making.

4. Discussion

In this work, an exhaustive analysis of the explainability of a prediction model developed for data classification in the context of Formula (1) has been carried out. We have obtained valuable information about the most relevant features for the model predictions by applying explainability techniques, such as attention and importance by permutation. The attention analysis has allowed us to visualize the attention weights as heat maps, which has been especially useful for identifying the regions in the input data that have the most significant impact on the model's predictions [37,40]. These heat maps reveal the most relevant pieces of data at each time step, which is critical to understanding how the model makes its predictions. In addition, by providing a visual representation, heat maps make it easy to interpret and communicate the results.

On the other hand, the technique of importance by permutation has made it possible to calculate the relative importance of each characteristic in the model's predictions. By randomly permuting the values of each feature and measuring their impact on model performance, we have obtained importance scores that reflect the degree of contribution of each part to the final predictions [41]. This has allowed us to quickly identify the most influential features and those that may hurt model performance.

We have observed exciting patterns by comparing the results of the permutation importance technique with the model predictions. Some features have significantly influenced forecasts, while others have hurt model performance [42]. These findings are consistent with the literature in similar works, where the importance of specific features in classifying data related to Formula (1) has been highlighted. It is essential to emphasize that the explainability approach provides information about the most critical features and contributes to a deeper understanding of the model [4,43]. By analyzing the results obtained, it is possible to gain insight into how the model uses features to make its predictions and how certain factors can affect its performance. This knowledge can benefit researchers interested in developing prediction models in other fields.

Previous studies have used explainability techniques to understand and analyze models in this domain. When comparing our results with similar research, we found similarities in the importance of specific features in the model's predictions. For example, previous studies have highlighted the relevance of driver performance history, vehicle technical

characteristics, and weather conditions in Formula (1) qualifying and race results. Our findings support these observations, as we have identified similar features influencing the model predictions [19,44]. This reinforces the consistency of the results and the importance of considering these characteristics when developing prediction models in Formula (1).

An overview of explainability methods in AI is provided in [45]. Our work differs by focusing specifically on applying these techniques in the context of Formula (1). While the article by Angelov et al. spans a wide range of applications and explicability approaches, we focus on performance predictions in motorsport. This allows us to analyze and understand the predictions in a specific domain, providing valuable insights for the players involved in Formula (1).

On the other hand, ref. [46] deals with explainability in black box models. Our work shares the goal of making models more understandable but differs in the application context and the techniques used. While the approach of Díaz et al. focuses on general explainability methods, we apply specific techniques, such as attention and permutation importance, in the context of Formula (1). This allows us to identify the most relevant features and understand how they affect model predictions, which is crucial in a highly competitive environment like Formula (1).

In [47], the author focuses on explainability in the medical field. Although our work focuses on Formula (1) and not the medical field, we share the common goal of providing understandable and reliable explanations in artificial intelligence models. In addition, our focus on Formula (1) allows us to identify specific characteristics related to driver performance and the factors that influence racing. These findings may be valuable for developing predictive models in other domains.

Finally, ref. [48] focuses on the explainability of decision tree models. Although our work does not focus exclusively on tree models, we share the concern to understand and explain the decision-making process of artificial intelligence models. Our approach combines different explainability techniques, such as attention and permutation importance, to better understand how decisions are made in the context of Formula (1). This allows us to identify the most influential features and improve interpretation and confidence in the model results.

Our work stands out by applying explainability techniques in the specific context of Formula (1), allowing us to understand and explain predictions in a highly competitive and complex environment. We provide valuable information about the relevant features and their influence on the model predictions using techniques such as attention and permutation importance. Key players in motorsport can use these insights to improve strategic decision making and optimize driver performance in Formula (1) races. Furthermore, by comparing our results with related work, we find similarities and differences that strengthen the originality and contribution of our study in the field of explainability in Formula (1).

Despite the similarities, it is essential to note that this study makes a unique contribution by using multiple explainability techniques, such as attention and permutation importance, to gain a complete understanding of the model. This combination of approaches has allowed us to obtain detailed information about the relevant features and their influence on predictions, which enriches the field of explainability in Formula (1). These results provide a solid foundation for future studies in which further investigations can be made.

5. Conclusions

This work has addressed the explicability of a prediction model in the context of Formula (1). By using explicability techniques such as attention and importance by permutation, it has been possible to obtain valuable information about the relevant characteristics and their influence on the model predictions. The results have shown the importance of considering factors such as the drivers' performance history, the vehicles' technical characteristics, and the weather conditions when predicting the outcomes of the races.

Explainability techniques in the sports domain, specifically in Formula (1), have several implications. Firstly, it allows teams and analysts to better understand the key factors influencing driver and racing performance, which can help in strategic and tactical decision making. Furthermore, explainability can bring transparency and confidence to the prediction models used in this highly competitive sport.

Although this study has provided a greater understanding of the model and its explainability, there are areas for improvement and opportunities for future research. For example, different explainability approaches, such as decision-tree-based model interpretation or ensemble model-based feature importance analysis, could be explored. Furthermore, it would be interesting to assess how teams and decision makers can use the results of explainability in other settings.

Including explainability techniques in AI models improves the understanding of the results, increases trust in the systems, and ensures fairness, transparency, and accountability in their application. Explainability is critical to overcoming artificial intelligence's ethical, legal, and societal challenges. It promotes a more humane and people-centered approach to developing and implementing these models. In the future, explainability research and application will continue to play a crucial role in advancing artificial intelligence and its responsible and ethical adoption in various fields and industries.

Author Contributions: Conceptualization, W.V.-C. and J.G.-O.; methodology, W.V.-C.; software, A.J.-A.; validation, J.G.-O.; formal analysis, W.V.-C.; investigation, J.G.-O.; data curation, W.V.-C. and A.J.-A.; writing—original draft preparation, J.G.-O.; writing—review and editing, W.V.-C.; visualization, J.G.-O.; supervision, A.J.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data and code are available upon request to the email of the corresponding author william.villegas@udla.edu.ec.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lyu, S.; Liu, J. Convolutional Recurrent Neural Networks for Text Classification. *J. Database Manag.* **2021**, *32*, 65–82. [\[CrossRef\]](#)
2. Balasubramaniam, N.; Kauppinen, M.; Hiekkänen, K.; Kujala, S. Transparency and Explainability of AI Systems: Ethical Guidelines in Practice. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Birmingham, UK, 21 March 2022; Volume 13216. [\[CrossRef\]](#)
3. Lee, C.H.; Cha, K.J. FAT-CAT—Explainability and Augmentation for an AI System: A Case Study on AI Recruitment-System Adoption. *Int. J. Hum. Comput. Stud.* **2023**, *171*, 102976. [\[CrossRef\]](#)
4. Bopaiah, K.; Samuel, S. Strategy for Optimizing an F1 Car's Performance Based on FIA Regulations. *SAE Int. J. Adv. Curr. Prac. Mobil.* **2020**, *2*, 2516–2530. [\[CrossRef\]](#)
5. Lv, L.; Li, H.; Wu, Z.; Zeng, W.; Hua, P.; Yang, S. An Artificial Intelligence-Based Platform for Automatically Estimating Time-Averaged Wall Shear Stress in the Ascending Aorta. *Eur. Heart J.-Digit. Health* **2022**, *3*, 525–534. [\[CrossRef\]](#)
6. Markus, A.F.; Kors, J.A.; Rijnbeek, P.R. The Role of Explainability in Creating Trustworthy Artificial Intelligence for Health Care: A Comprehensive Survey of the Terminology, Design Choices, and Evaluation Strategies. *J. Biomed. Inform.* **2021**, *113*, 103655. [\[CrossRef\]](#)
7. Hamon, R.; Junklewitz, H.; Sanchez, I.; Malgieri, G.; De Hert, P. Bridging the Gap between AI and Explainability in the GDPR: Towards Trustworthiness-by-Design in Automated Decision-Making. *IEEE Comput. Intell. Mag.* **2022**, *17*, 72–85. [\[CrossRef\]](#)
8. Amann, J.; Blasimme, A.; Vayena, E.; Frey, D.; Madai, V.I. Explainability for Artificial Intelligence in Healthcare: A Multidisciplinary Perspective. *BMC Med. Inf. Decis. Mak.* **2020**, *20*, 310. [\[CrossRef\]](#)
9. Kuiper, O.; van den Berg, M.; van der Burgt, J.; Leijnen, S. Exploring Explainable AI in the Financial Sector: Perspectives of Banks and Supervisory Authorities. In *Artificial Intelligence and Machine Learning. BNAIC/Benelearn 2021. Communications in Computer and Information Science*; Springer: Cham, Switzerland, 2022; Volume 1530.
10. Alsaigh, R.; Mehmood, R.; Katib, I. AI Explainability and Governance in Smart Energy Systems: A Review. *Front. Energy Res.* **2023**, *11*, 1071291. [\[CrossRef\]](#)
11. Leventi-Peetz, A.M.; Östreich, T.; Lennartz, W.; Weber, K. Scope and Sense of Explainability for AI-Systems. In *Intelligent Systems and Applications. IntelliSys 2021. Lecture Notes in Networks and Systems*; Springer: Cham, Switzerland, 2022; Volume 294.
12. Ren, B.; Zhang, Z.; Zhang, C.; Chen, S. Motion Trajectories Prediction of Lower Limb Exoskeleton Based on Long Short-Term Memory (LSTM) Networks. *Actuators* **2022**, *11*, 73. [\[CrossRef\]](#)

13. Haimed, A.M.A.; Saba, T.; Albasha, A.; Rehman, A.; Kolivand, M. Viral Reverse Engineering Using Artificial Intelligence and Big Data COVID-19 Infection with Long Short-Term Memory (LSTM). *Env. Technol. Innov.* **2021**, *22*, 101531. [\[CrossRef\]](#)
14. Zhou, Y.; Zheng, H.; Huang, X.; Hao, S.; Li, D.; Zhao, J. Graph Neural Networks: Taxonomy, Advances, and Trends. *ACM Trans. Intell. Syst. Technol.* **2022**, *13*, 1–54.
15. Le, X.H.; Ho, H.V.; Lee, G.; Jung, S. Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. *Water* **2019**, *11*, 1387. [\[CrossRef\]](#)
16. Grover, R. Analysing the importance of qualifying in formula 1 using the fastf1 library in python. *Int. J. Adv. Res.* **2022**, *10*, 1138–1150. [\[CrossRef\]](#) [\[PubMed\]](#)
17. García, A.; Martínez, B.; Ramírez, C. Machine Learning and Artificial Intelligence for Predictive Maintenance in Industrial Applications. *Sensors* **2022**, *22*, 677. [\[CrossRef\]](#)
18. Satpathi, A.; Setiya, P.; Das, B.; Nain, A.S.; Jha, P.K.; Singh, S.; Singh, S. Comparative Analysis of Statistical and Machine Learning Techniques for Rice Yield Forecasting for Chhattisgarh, India. *Sustainability* **2023**, *15*, 2786. [\[CrossRef\]](#)
19. Aversa, P.; Cabantous, L.; Haeffliger, S. When Decision Support Systems Fail: Insights for Strategic Information Systems from Formula 1. *J. Strateg. Inf. Syst.* **2018**, *27*, 221–236. [\[CrossRef\]](#)
20. Patil, A.; Jain, N.; Agrahari, R.; Hossari, M.; Orlandi, F.; Dev, S. Data-Driven Analysis of Formula 1 Car Races Outcome. In *Artificial Intelligence and Cognitive Science. AICS 2022. Communications in Computer and Information Science*; Springer: Cham, Switzerland, 2023; Volume 1662.
21. Petróczy, D.G.; Csátó, L. Revenue Allocation in Formula One: A Pairwise Comparison Approach. *Int. J. Gen. Syst.* **2021**, *50*, 243–261. [\[CrossRef\]](#)
22. Weiss, T.; Chrosniak, J.; Behl, M. Towards Multi-Agent Autonomous Racing with the DeepRacing Framework. In Proceedings of the 2021 International Conference on Robotics and Automation (ICRA 2021)—Workshop Opportunities and Challenges with Autonomous Racing, online, 31 May 2021. Available online: <https://linklab-uva.github.io/icra-autonomous-racing/> (accessed on 5 April 2023).
23. Lapré, M.A.; Cravey, C. When Success Is Rare and Competitive: Learning from Others' Success and My Failure at the Speed of Formula One. *Manag. Sci.* **2022**, *68*, 8741–8756. [\[CrossRef\]](#)
24. Abdellah, A.R.; Mahmood, O.A.; Kirichek, R.; Paramonov, A.; Koucheryavy, A. Machine Learning Algorithm for Delay Prediction in IoT and Tactile Internet. *Future Internet* **2021**, *13*, 304. [\[CrossRef\]](#)
25. Adamu, M.; Haruna, S.I.; Malami, S.I.; Ibrahim, M.N.; Abba, S.I.; Ibrahim, Y.E. Prediction of Compressive Strength of Concrete Incorporated with Jujube Seed as Partial Replacement of Coarse Aggregate: A Feasibility of Hammerstein–Wiener Model versus Support Vector Machine. *Model. Earth Syst. Environ.* **2022**, *8*, 3435–3445. [\[CrossRef\]](#)
26. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional Neural Networks: An Overview and Application in Radiology. *Insights Imaging* **2018**, *9*, 611–629. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Yamazaki, K.; Vo-Ho, V.K.; Bulsara, D.; Le, N. Spiking Neural Networks and Their Applications: A Review. *Brain Sci.* **2022**, *12*, 863. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent Advances in Convolutional Neural Networks. *Pattern Recognit.* **2018**, *77*, 354–377. [\[CrossRef\]](#)
29. Cuomo, S.; Di Cola, V.S.; Giampaolo, F.; Rozza, G.; Raissi, M.; Piccialli, F. Scientific Machine Learning Through Physics-Informed Neural Networks: Where We Are and What's Next. *J. Sci. Comput.* **2022**, *92*, 88. [\[CrossRef\]](#)
30. Abedin, B. Managing the Tension between Opposing Effects of Explainability of Artificial Intelligence: A Contingency Theory Perspective. *Internet Res.* **2022**, *32*, 425–453. [\[CrossRef\]](#)
31. Sand, M.; Durán, J.M.; Jongsma, K.R. Responsibility beyond Design: Physicians' Requirements for Ethical Medical AI. *Bioethics* **2022**, *36*, 162–169. [\[CrossRef\]](#)
32. Yang, G.; Ye, Q.; Xia, J. Unbox the Black-Box for the Medical Explainable AI via Multi-Modal and Multi-Centre Data Fusion: A Mini-Review, Two Showcases and Beyond. *Inf. Fusion* **2022**, *77*, 29–52. [\[CrossRef\]](#)
33. Villegas-Ch, W.; Palacios-Pacheco, X.; Luján-Mora, S. Artificial Intelligence as a Support Technique for University Learning. In Proceedings of the IEEE World Conference on Engineering Education (EDUNINE), Lima, Peru, 17–20 March 2019; pp. 1–6.
34. Budiharto, W. Data Science Approach to Stock Prices Forecasting in Indonesia during Covid-19 Using Long Short-Term Memory (LSTM). *J. Big Data* **2021**, *8*, 47. [\[CrossRef\]](#)
35. Ma, M.; Liu, C.; Wei, R.; Liang, B.; Dai, J. Predicting Machine's Performance Record Using the Stacked Long Short-Term Memory (LSTM) Neural Networks. *J. Appl. Clin. Med. Phys.* **2022**, *23*, e13558. [\[CrossRef\]](#)
36. ArunKumar, K.E.; Kalaga, D.V.; Mohan Sai Kumar, C.; Kawaji, M.; Brenza, T.M. Comparative Analysis of Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM) Cells, Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving Average (SARIMA) for Forecasting COVID-19 Trends. *Alex. Eng. J.* **2022**, *61*, 7585–7603. [\[CrossRef\]](#)
37. Sen, S.; Sugianto, D.; Rochman, A. Komparasi Metode Multilayer Perceptron (MLP) Dan Long Short Term Memory (LSTM) Dalam Peramalan Harga Beras. *Ultimatics* **2020**, *12*, 35–41. [\[CrossRef\]](#)
38. Moch Farryz Rizkilloh; Sri Widiyanesti Prediksi Harga Cryptocurrency Menggunakan Algoritma Long Short Term Memory (LSTM). *J. RESTI (Rekayasa Sist. Dan Teknol. Inf.)* **2022**, *6*, 25–31. [\[CrossRef\]](#)
39. Kratzert, F.; Klotz, D.; Brenner, C.; Schulz, K.; Herrnegger, M. Rainfall-Runoff Modelling Using Long Short-Term Memory (LSTM) Networks. *Hydrol. Earth Syst. Sci.* **2018**, *22*, 6005–6022. [\[CrossRef\]](#)

40. Zuo, H.M.; Qiu, J.; Jia, Y.H.; Wang, Q.; Li, F.F. Ten-Minute Prediction of Solar Irradiance Based on Cloud Detection and a Long Short-Term Memory (LSTM) Model. *Energy Rep.* **2022**, *8*, 5146–5157. [[CrossRef](#)]
41. Ho, C.H.; Park, I.; Kim, J.; Lee, J.B. PM2.5 Forecast in Korea Using the Long Short-Term Memory (LSTM) Model. *Asia Pac. J. Atmos. Sci.* **2022**, *1*, 1–14. [[CrossRef](#)]
42. Fang, L.; Shao, D. Application of Long Short-Term Memory (LSTM) on the Prediction of Rainfall-Runoff in Karst Area. *Front. Phys.* **2022**, *9*, 790687. [[CrossRef](#)]
43. Matam, B.R.; Duncan, H. Technical Challenges Related to Implementation of a Formula One Real Time Data Acquisition and Analysis System in a Paediatric Intensive Care Unit. *J. Clin. Monit. Comput.* **2018**, *32*, 559–569. [[CrossRef](#)]
44. Laghrissi, F.E.; Douzi, S.; Douzi, K.; Hssina, B. Intrusion Detection Systems Using Long Short-Term Memory (LSTM). *J. Big Data* **2021**, *8*, 65. [[CrossRef](#)]
45. Angelov, P.P.; Soares, E.A.; Jiang, R.; Arnold, N.I.; Atkinson, P.M. Explainable Artificial Intelligence: An Analytical Review. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2021**, *11*, e1424. [[CrossRef](#)]
46. Molnar, C. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*; Packt Publishing: Birmingham, UK, 2020.
47. Tjoa, E.; Guan, C. A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4793–4813. [[CrossRef](#)]
48. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.I. From Local Explanations to Global Understanding with Explainable AI for Trees. *Nat. Mach. Intell.* **2020**, *2*, 56–67. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.