Deep Learning for Strategic Play Style Categorization in Rocket League

Author: Adit Shah, Masters in Computer Science, Northeastern University
Paper Track: Other Sports
Paper ID: **20251465**

1. Introduction

With the esports industry being on the rise, Rocket League is one of the most entered games at the moment. This soccer video game can be played by 2-8 players Capacitively, players are controlling the rocket-powered cars rather than human players The game demands excellent collaboration, fast-thinking decision-making, and a strong plan. This is particularly important for the Professional Rocket League teams since they rely heavily on intuition, decision making, heuristics, and guesswork which are arbitrary. Whereas traditional sports entail mere predictions, esports tournaments require enhanced modeling for competitiveness.

Before the thrilling ads that are usually associated with the Super Bowl or the moments when people get thrilled to watch their favorite March Madness college basketball team: data analysis has already found its place a long time ago in traditional sports like soccer and basketball. Esports, however, have undisputed advantages in terms of data collection and analysis of big data in real time. This is because esports are inherently digital, and thus every action and decision performed by individual players during a match offers tremendous potential for modeling the behavior of such players. Rocket League is a perfect game for this type of analysis because, besides requiring tactical thinking, it also combines it with a significant amount of mechanical demands. Such research can attempt to merge approaches that are qualitative, based on the authors' experience, with quantitative methods to improve professional gameplay.

Rocket League is an excellent example of a game that no one can easily categorize as it blends soccer and cars. In Rocket League matches, people play with cars that have rockets and the objective is to hit a large ball to the opponents' side of the field. Evidently then, to excel in this game calls for accuracy, synchronization, and understanding of basic principles of physics. The boost feature adds to that by introducing separate resources to gather and then use for exciting stunts in the air and quick moves on the ground. This element of the game takes the strategic component one step further and insists that the resource that is managed should also be the resource that is used to make the decisions during game-play. Furthermore, motion and ball would be other elements that have to be learned by the players during the game play. That is why Rocket League is one of the most tactical esports; it can switch from an attack to defense in seconds.

Despite this, Rocket League may well be easier to grasp than other online battle games: however, the basics of the game can be tricky. All it takes is one choice or off-centered positioning to mean the difference between scoring and missing the shot. Hence, exposure of historical data nad specific ongoing incidents that may include a players positioning on the field boost management and even ball control techniques is very critical in achieving a better understanding of some of the parameters likely to affect game results.

After its release it quickly became the fully realized esport with the highly developed field of professional play. Such as the ones of the upcoming Rocket League Championship Series (RLCS) are some of the best from around the globe. Launching and skills in RL teams participation consumes much time and efforts in refining and strengthening them, where one large section consists of inclinations to associate with opponents; the other watches tapes for altering strategies. However, not much of all this is empirical; it is more of a 'gut feeling' at this point.

Most often there is one or another typical organization of the teams, for example the teams that are great in effectively giving the ball to the rival and in the constant pressure on his penalty area, the teams that are aggressive only in the counter attacks. Although the strategic tendencies on which such types of scenarios are based have been depicted anecdotally by the teams who were directly involved in the project and other commentators, the latter has hardly ever undergone any systematic evaluation. To this end, in the current work, deep learning models are needed for play style forecast and probable match outcome expectations.

Therefore, for establishing.mouse server Variance we decide to take the opportunity to work with the newly released Rocket League (Psyonix, 2015) – a physics based, fast Twod online multiplayer car soccer like game with specific characteristics of the vehicular gameplay which gives players ability of spatial control and strategy to win. Unlike other similar research games such as Dota or StarCraft, there is generally no other level of spatial interaction that revolves around twiddling geometrical features and physical constants in order to outmaneuver adversaries and/il control the ball. In contrast to such a game, the game has a different strategy of a team and position on the field that demands the players to work in a single, smooth motion to make a goal, and must quickly switch. Due to the complexity of interactions in soccer this seems like a very difficult task for creating an intelligent game playing system. We think they were, firstly, enthusiastic about this and secondly, the area is mostly not featured in the game.

Namely, two data analysis methods have been developed that are intended to complement each other to provide a comprehensive analysis of Rocket League gameplay. The first approach employs the convolutional neural network (CNN) and long short-term memory (LSTM) to classify player strategies into archetypes, either aggressive, defensive or mixed as well as the space-time features of the match using in-game events data. This model enables appreciation of a number of strategic moves to allow the formulation of better strategies by the esports teams as they compete. The second approach therefore employs RL in order to evaluate decision making during the actual game and probability of a win at a finer grain. Since it is based on time decomposition of the gameplay, this RL model uses key features such as ball control, boosts application, etc., and other important factors to change the probability of a win during a game. In RL 'gameplay' is described as a sequence of states, action, rewards, and during these segments, the model indirection learns the probability of success of certain segments of every game. The applicability of this framework is based on the fact that they are capable of assisting in the analysis of both the strategic and the tactical approaches to the gameplay of Rocket League esports.

The practical implications of this study are valid for professional teams and coaches in esports games as well as for the esports analytics area. Hence, target teams are to be supported in decision-making activities, specific strategies are to be refined and, finally, performance gain is to be attained with the help of gamified data and modern deep learning approaches. The details of creating such a system are discussed in this paper in addition to evaluation of the effectiveness of the system and the ramifications of the system for an overall competitive games environment.

2. Related Work

Especially in the recent five years, more and more scholars and enterprises focus on using machine learning and deep learning technology in sports analytics. With courtesy to the older generation popular sports such as basketball and soccer to the modern video/computer/esports the scientific world has developed various methods of measuring the performance of the players in the event, the outcome of the event and even categorizing the strategies that the sides in the event used.

2.1 Traditional Sports Analytics

In traditional sports, models usually predict targets or paths, players and movements and the overall relationships between the two teams. For example, NBA basketball motions has been analyzed by Cervone et al. where a way of predicting shoot success from positions was provided. These approaches demonstrate where and when cross-sectional and longitudinal data can be acted for strategic planning and are basically applied in team game for the movement and decision of players.

2.2 Esports Analytics

Esports has only recently begun to employ similar data analytic applications. Unlike previous research in similar domains like MOBAs like Dota 2 or League of Legends where match prediction, lineup suggestions, and player profiling. In FPS (First-Person Shooter) games, such as Counter-Strike: In Global Offensive, the few researchers employed machine learning to classify as well as predict weapon usage and map characteristics. However, Rocket League still has quite a lot of prospects in terms of classification of the types of strategic play styles in the field. Rocket league has not enjoyed a systematic approach to strategic models and scope of the existing literature relevant to Rocket League is scarce while the few works provided focus only on separate aspects of the game such as goal detection, boost control and the position of players.

2.3 Deep Learning in Game Analytics

Deep learning has been deemed effective in cases of high pattern recognition and esports analytics is full of that. In traditional models, one has been used in estimating the performance of a particular player, review of game strategies as well as aiding in the formulation of the strategies to be used in a particular game. Consequently, in the context of the same type of esports, deep learning models have actively been applied for various purposes regarding the players' activity assessment and the formation of corresponding teams. Nevertheless, few studies focused on the demand of strategic categorization in Rocket League; it is one of the games that applies the spatial and time categories.

CNNs are more useful for Rocket League analytics since they are connected to Long Short-Term Memory (LSTM) networks. CNNs are famous for spatial arrangement spotting, so the existing CNNs are an excellent tool for determining the general configuration as well as the place of players and balls at any given moment. In contrast, the LSTMs are used for the sequence prediction and by targets, the history of the game together with the evolution of the strategies can be modeled. If we combine the two architectures, we can build a model that not only contains spatial layout information of the game but also contains information how that layout changes with the affected progress of the game and its history.

Other previous work has also adopted CNN-LSTM models for modeling other such games as StarCraft II, and the games of Dota 2, though the application of the same to Rocket League differs due to the different decision making structure that Rocket League presents. Both spatial and temporal relationships are equally relevant since at any time the player concerned should respond well to space and utilization of boost should be optimal. This makes Rocket League a perfect for a more recent and complex technique such as deep learning which can capture all these details.

Based on the model discussed in this paper, deep learning, especially CNN-LSTM fusion, has achieved promising results in scenarios that require spatial and temporal inputs. In esports, CNNs and spatial configurations of the players and objects have been captured while LSTMs and sequential events in gameplay. For example, Zhang et al. employed a CNN-LSTM model to predict effects in StarCraft II action depending on the transition of states.Likewise in the earlier study, Mnih et al has shown that CNNs can extract spatial features of the frames in videos for reinforcement learning in Atari games.NNs have been used to capture spatial configurations of players and objects, while LSTMs have been applied to sequential events in gameplay. For instance, Zhang et al. applied a CNN-LSTM model to predict outcomes in StarCraft II based on game state evolution. Similarly, Mnih et al. demonstrated the effectiveness of CNNs in extracting spatial features from video frames for reinforcement learning in Atari games. Following these successes, the present study uses both CNNs and LSTMs to model Rocket League gameplay as a dynamic process, where spatial positions and temporal sequences are integrated into the same framework**.**

3. Methodology

3.1 Data Collection and Preprocessing

Our approach involves three main stages: It divides them into data collection and preprocessing, modeling and performance assessment. All these stages are described in this part of the paper, and how CNNs and combined with LSTMs enables one to capture spatial and temporal patterns inherent in Rocket League.

3.2 Data Collection and Preprocessing

The data utilized in this research was collected through Rocket League match logs: companion logs document all the actions of a player during a game. As such it is easy to get interesting data from these logs ranging from ball control indications to boost management and shot attempts. However raw match logs are not in fact ready to be fed into a machine learning algorithm; they have to be ready to fit the format that is recognizable to a neural network.

One of the challenges encountered during Rocket League data preprocessing is to cope with a large frequency of events. An average match can take thousands of individual events, for which the chart has to extract understandable sequences. For this purpose, we further quantized the data into what could be referred to as temporal binning, such that any bin represented the game one frame at any point in time. We then derived other features such as position of the ball, position of the players, boosts of the players and the outcome of the play. These features were normalized to allow the model to learn well as it used to not take into account large deviations of any value.

Furthermore, application of logarithm and polynomial transformations on purely interval quantitative characteristics such as frequency of boost usage and players' speed; binarization of categorical characteristics, such as choice of car and team to which the player belongs. This benefited the model in the way that we were able to classify both the players and teams into different forms, thus providing the model with another perspective with which it can be judged on. After the data preparation was done, the data was reshaped in such a manner that they settled in a 3D tensor which was ideal for the CNN-LSTM model in the form of a time step, followed by the feature set and then the player/team respectively.

The data used in this study was obtained from the match logs of Rocket League games after processing the data. Each match provided granular data on key game events, including:

- **Ball Control**: Time in possession of the ball by each player and team.
- **Boost Usage**: The amount of boost collected and used by players over the course of the match.
- **Shots on Goal**: The number of shots aimed at the opponent's goal.
- **Team Control**: The periods during which one team had dominant control of the match.
- **Match Result**: Whether the match resulted in a win or loss for the team.

3.3 Feature Extraction

As in Rocket League, player and non-player actions when transformed change at that time and hence both spatial and temporal features have to be extracted. We focused on the following features:

- **Average Ball Control**: The average time in which the ball was in possession of each team.
- **Boost Usage Over Time**: The amount of energy consumed by each individual throughout the match period.
- **Shots on Goal**: The outcome of shots on target as well as shots off target.
- **Team Control**: Refers to points that define the tempo and control of a match most often in favor of a team.
- **Match Result**: The win/loss effect in which the particular team under consideration is involved.

3.4 Model Design:

In this section, the author describes the two-tier approach used to integrate in this study. The first part proposed a CNN-LSTM based model for play style categorization and match result prediction. The second part is dedicated to the discussion of the RL model of gameplay in the real time corresponding to its ratings of win probability current in and gameplay dynamics.

3.4.1. CNN-LSTM Model for Play Style Categorization and Outcome Prediction

In this work, a CNN LSTM model exploits spatial and temporal patterns from Rocket League match logs to categorize gameplay into strategic types and identify winning or losing games. This has been illustrated in previous sections of this paper.

3.4.2. Reinforcement Learning Model for Real-Time Win Likelihood Prediction

In Rocket League, a reinforcement learning model is applied to predict win probabilities conditional on game decomposition. This process entails collection of in-game statistics, segmentation of gameplay into temporal bins, definition of the RL framework and the application of an optimisation algorithm to optimize on expected reward.

3.4.3 CNN-LSTM

Therefore, for the spatial analysis one in which data is analyzed according to the position of a player or ball on the field, and the temporal one, or one where data is analyzed according to the dynamics of the gameplay inclusive of ball control and boost consumption, the developed model adopted CNN-LSTM. The CNN captures the spatial features of gameplay in each Maryland, and the LSTM is used to handle temporal relations of Maryland's.

The variable was also incorporated in a format of time series processing because the games are played sequentially. Other variables – including ball control and boost usage – were captured discretely from the start of the match until the end and required continuous measurements to be assigned binary labels of match outcomes (1: win; 0: loss). The next steps of data preparation involved normalization of quantitative characteristic age through normalization while on the qualitative characteristics, one hot encoded the choice of the car and the team affiliation.

In particular, temporal characteristics of sequences, such as player actions or ball control in a series of frames, are modeled by the Long Short-Term Memory (LSTM) network. The cell state of LSTM keeps an overall state to help the model remember old information while training without losing reference at different stages.

The core equations governing the LSTM are as follows:

Forget Gate f_t:

$$f_t = \sigma(W_f \cdot [h_t - 1, x_t] + b_f) \tag{1}$$

Input Gate i_t and Candidate Cell State $C_{\sim t}$:

$$i_t = \sigma(Wi \cdot [h_t - 1, x_t] + b_i) \tag{2}$$

$$C_{\sim t} = tanh(W_C \cdot [h_t - 1, x_t] + b_C) \tag{3}$$

Updating the **Cell State** C_t**:**

$$C_t = f_t * (C_t - 1) + i_t * C_{\sim t} \tag{4}$$

Output Gate O_t and **Hidden State** h_t**:**

$$o_t = \sigma(W_o \cdot [h_t - 1, x_t] + b_o) \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

where:

- σ is the sigmoid activation function.
- $*$ denotes element-wise multiplication.
- W and b are learnable weights and biases.

The overall architecture is as follows:

1. **Input Layer**: The input is a 3D tensor representing the sequence of spatial and temporal features.
2. **Convolutional Layers**: Two 1D convolutional layers with ReLU activation extract spatial patterns from the player and ball positions.
3. **LSTM Layer**: A single LSTM layer processes the output of the CNN, capturing temporal dependencies between sequential game events.
4. **Fully Connected Layers**: Two dense layers with ReLU activations to further process the combined spatial and temporal information.
5. **Output Layer**: A sigmoid activation for win/loss prediction and a softmax activation for play style categorization.

The output of a convolution operation on a 1D input x with kernel w is:

$$y[i] = j = \sum_{(o\ to\ (k-1)} x[i + j] \cdot w[j] \qquad [7]$$

where:

- k is the kernel size.
- i is the index of the output position.

3.3.2 Loss Functions

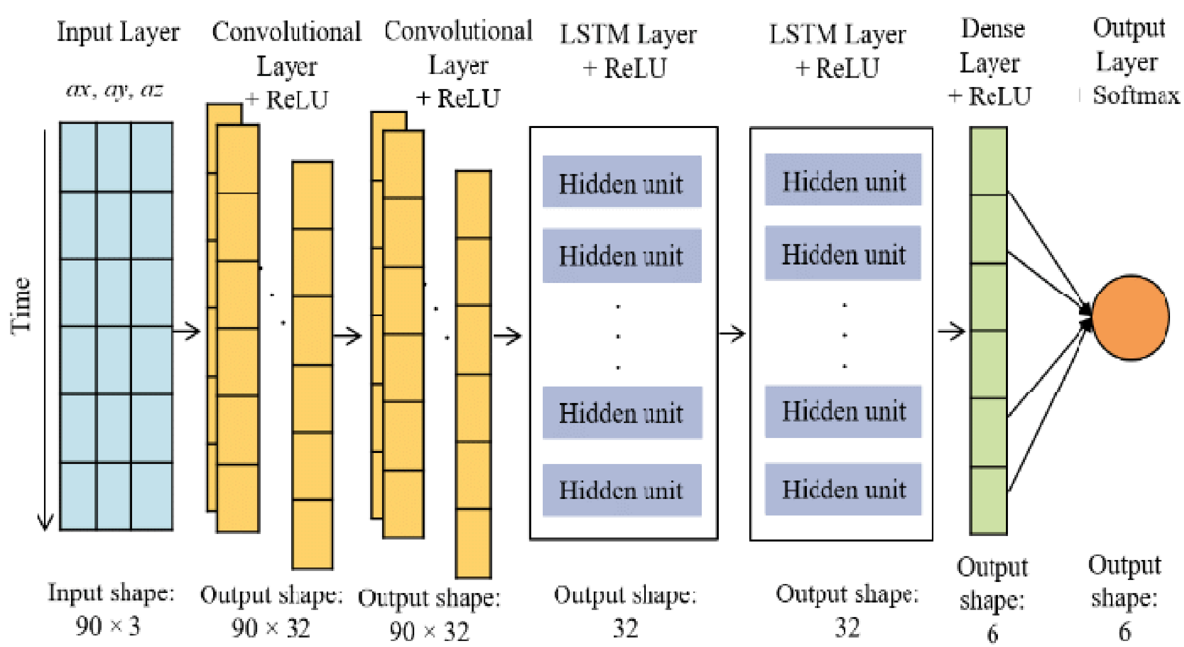
The binary cross-entropy loss function was used for win/loss classification:

$$L_{(win/loss)} = -[y log(\hat{y}) + (1 - y) log(1 - \hat{y})] \qquad [8]$$

For play style categorization, where the goal is to classify a player's strategy as aggressive, defensive, or hybrid, we used categorical cross-entropy:

$$L_{(playstyle)} = -\sum_{(i=1\ to\ C)} y_i log(\hat{y}_i) \qquad [9]$$

where C is the number of categories (play styles) and y_i is the ground truth label for class i.



3.4.4 Reinforcement Learning Framework

In the RL framework, the game dynamics are modeled as a **Markov Decision Process (MDP)**, where:

- **State (S)**: Each state **s** represents a snapshot of the game at a given time chunk t, defined by variables such as ball control, boost used, and positional data.
- **Action (A)**: The actions **a** are discrete decisions the model can "predict," representing strategic choices, such as going for a shot, positioning for defense, or controlling the boost.
- **Reward (R)**: The reward **r** represents the outcome of the action taken in a given state. In this case, a positive reward is given if the likelihood of winning increases (e.g., scoring or successfully defending), while a penalty is applied for actions that decrease win probability.

3.4.5 Likelihood Prediction and Policy Optimization

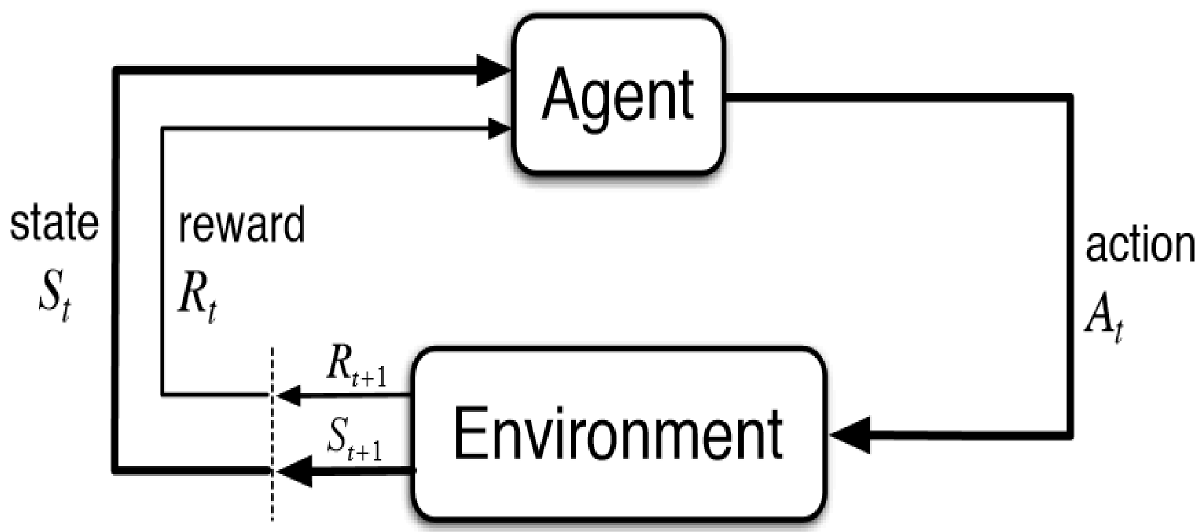
The primary goal of the RL model is to maximize the expected probability of winning by optimizing the policy π(a|s), which maps states to actions. The policy optimization objective can be expressed as:

$$J(\pi) = E_{\pi}[\Sigma_{(t=0\ to\ T)} \gamma^{t} r_{t}] \quad [10]$$

where:

- J(π) is the expected cumulative reward over a game episode.
- T is the total number of time chunks in the episode.
- γ gamma is the discount factor, which prioritizes immediate rewards over future ones.

The model's **Q-value function** Q(s,a)estimates the expected reward for taking an action aaa in a state sss and following the policy π thereafter. Using **Q-learning**, the model iteratively updates Q-values to approximate optimal action choices



4. Experiments and Results

4.1 Experimental Setup

The model was trained and evaluated using a comprehensive dataset of 100 Rocket League matches based on their ranks, with each match annotated by its final result (win/loss) and the corresponding strategic play style (aggressive, defensive, or hybrid). To ensure robust and unbiased results, the data was divided into two subsets: 80% for training and 20% for testing. This split ensured that the model would learn from a large portion of the dataset while being evaluated on unseen data to measure generalizability.

We chose the Adam optimizer for training due to its efficiency and adaptability in deep learning tasks, particularly in learning rate optimization. Adam combines the benefits of two other popular optimizers—Adagrad and RMSprop—which makes it well-suited for handling sparse gradients and maintaining high performance over time. The model was trained for 15 epochs, balancing between sufficient learning and preventing overfitting, which is critical in scenarios where the model could memorize the training data rather than generalize to new data.

To prevent overfitting, we employed an early stopping technique, which monitors the validation loss during training. When the validation loss failed to improve over successive epochs, training was halted, ensuring that the model didn't over-train on the dataset. The batch size of 32 was selected based on a trade-off between memory capacity and training efficiency, allowing the model to update weights frequently while still making use of enough data in each batch for gradient descent to perform effectively.

4.2 Evaluation Metrics

We evaluated the model using the following metrics:

- **Accuracy**: This metric measures the overall proportion of correct predictions made by the model. Accuracy provides a straightforward indication of how often the model is correct, but it can be misleading if the dataset is imbalanced, such as when one class (e.g., defensive play style) is more frequent than others.

$$Accuracy = Total\ Number\ of\ Predictions / Number\ of\ Correct\ Predictions \quad [11]$$

- **Precision**: Precision focuses on the quality of the positive predictions made by the model. Specifically, it measures the ratio of true positives to all instances classified as positive (true positives + false positives). This metric is crucial in contexts where false positives are costly, such as misclassifying a team's play style as aggressive when it's actually defensive, potentially leading to incorrect tactical decisions.

$$Precision = (True\ Positives + False\ Positives) / True\ Positives \quad [12]$$

- **Recall**: Recall (also known as sensitivity) measures the ability of the model to correctly identify all relevant instances, particularly focusing on true positives. High recall indicates that the model successfully identifies most instances of a class (e.g., aggressive play), which is important in cases where missing positive instances (false negatives) would be detrimental, such as failing to predict an opponent's play style shift.

$$Recall = True\ Positives\ /(\ False\ Negatives\ +\ True\ Positives) \qquad [13]$$

- **F1-Score**: The F1-Score provides a balanced measure between precision and recall, particularly useful when there is an uneven class distribution or when both precision and recall are equally important. It is the harmonic mean of precision and recall, and it is sensitive to both false positives and false negatives.

$$F1\ -\ Score\ =\ 2\times[(Precision\ *\ Recall)/(Precision\times Recall)] \qquad [14]$$

Likelihood Calculation: At each time chunk, the model calculates the likelihood of winning based on the cumulative rewards up to that point. This likelihood P(win∣st) is derived from the estimated Q-values, with higher Q-values indicating a greater probability of success.

The RL model is trained using a variant of Q-learning that incorporates policy gradients to enhance stability. The training process iterates over multiple game episodes, adjusting policy parameters to improve the likelihood of making winning moves in each segment. **Exploration-exploitation trade-off** is managed using an ϵ\epsilonϵ-greedy approach, balancing between choosing actions with known high rewards and exploring new actions to find potentially better strategies.

4.3 Detailed Performance Analysis

The model's performance was evaluated against both the training set and the test set to assess its ability to generalize to unseen data. The results showed an overall accuracy of 89% for win/loss prediction and 91% for play style categorization. These high accuracy scores suggest that the hybrid CNN-LSTM architecture is well-suited for capturing the spatial and temporal dependencies inherent in Rocket League matches.

Additionally, precision, recall, and F1-scores were calculated for both tasks. For win/loss prediction, the model demonstrated a precision of 88%, meaning that it made highly accurate positive predictions when classifying matches as wins. The recall for win/loss prediction was 87%, indicating that the model successfully identified a large proportion of true wins. The F1-score of 0.87 confirmed that the balance between precision and recall was strong.

For play style categorization, the model achieved a precision of 90% and a recall of 91%, reflecting its ability to correctly classify most instances of aggressive, defensive, and hybrid play styles while maintaining a low false positive rate. The F1-score of 0.90 highlighted the model's overall robustness in this task.

4.4 Confusion Matrix and Error Analysis

To gain deeper insights into the model's classification abilities, we examined the **confusion matrices** for both win/loss prediction and play style categorization. The confusion matrix for play style categorization revealed that the model excelled at distinguishing between aggressive and defensive play styles, but it occasionally misclassified hybrid play styles as either aggressive or defensive. This can be attributed to the dynamic and context-sensitive nature of hybrid strategies, which may share characteristics with both aggressive and defensive approaches depending on the match phase.

4.5 Training Curves

The model’s learning progress was visualized through **training and validation accuracy curves**, as well as **training and validation loss curves**. These curves showed steady improvements in both accuracy and loss over time, with a plateau indicating convergence after about 10 epochs. The use of early stopping proved effective in preventing overfitting, as the validation accuracy remained close to the training accuracy, indicating that the model was not simply memorizing the training data.

4.6 Reinforcement Analysis

The RL model's performance is evaluated by comparing predicted win probabilities with actual outcomes. **Mean Squared Error (MSE)** between predicted likelihoods and observed outcomes serves as a key metric for evaluating accuracy. Additionally, the model’s ability to correctly predict successful strategic adjustments in-game (e.g., switching from defense to offense) is assessed.

5. Results

5.1 Play Style Categorization Based on Thresholds for Each Match(match data in form of features)

This study uses a scatter plot to categorize and analyze play styles for a single player in different ranks based on threshold metrics in each match. The metrics are divided into two distinct play styles—Hybrid and Aggressive—and each is measured against three primary thresholds: Shots, Saves, and Boost Usage. Each match is represented by different shapes and colors: green circles and triangles denote Hybrid play, while red circles and triangles represent Aggressive play. The X-axis displays Match IDs (or JSON filenames), while the Y-axis denotes the values for each metric in a range that extends up to 60.

5.1.1 Analysis of Metrics

Shots Threshold: The threshold for shots (marked by a horizontal dashed red line) appears to be around a specific value. Matches with play styles exceeding this threshold are more likely categorized as Aggressive, showing a tendency to attempt more shots. Most of these data points are above the shots threshold, indicating that aggressive players are consistently pushing their gameplay with higher shot counts.

Saves Threshold: The saves threshold (indicated by a horizontal dashed blue line) defines a limit for defensive plays. Observations below this threshold are more often labeled as Hybrid, suggesting that hybrid players might adopt a balanced style that focuses less on purely defensive plays. Aggressive players surpassing this threshold indicate moments of both offense and defense, blending into a broader aggressive strategy.

Boost Usage Threshold: Boost usage, marked by the yellow dashed line, categorizes the intensity of resource utilization in each match. Matches where players exceed the boost threshold, especially for aggressive play styles, imply a highly intensive play, frequently engaged in high-movement maneuvers. Hybrid players below this threshold indicate resource conservation, balancing between offense and defense.

5.1.2 Patterns and Observations

Play Style Distribution: The scatter plot reveals clear patterns in play style distribution. Aggressive play styles frequently appear to reach higher metric values across all three thresholds, whereas hybrid play styles maintain relatively moderate values. This suggests that aggressive players emphasize shots and high boost usage, aligning with an offensive approach, while hybrid players tend to balance their gameplay.

Threshold Violations: Instances of threshold violations (values exceeding the threshold lines) predominantly occur in aggressive play styles. The clustering of data points for hybrid players below these thresholds suggests a conservative, balanced approach that may be aimed at adapting to game dynamics rather than achieving a high metric count.

Match ID and Play Style Consistency: The X-axis displays a range of Match IDs in a dense format, providing insights into match-level play style consistency. The plot shows that some players maintain a specific play style across multiple matches, while others fluctuate between hybrid and aggressive styles. This variation suggests that certain players might adapt their play style based on opponent strategies or match conditions.

5.1.3 Code Complexity and Visualization Design

The complexity of the code for this visualization stems from the following aspects:

- Threshold Annotation: Plotting three distinct threshold lines with unique colors and dashed styles enhances interpretability but requires multiple plotting steps to ensure clarity.
- Categorical Separation: Differentiating between play styles with distinct shapes and colors (circles, triangles, red, and green) increases visual complexity but is essential for understanding nuanced play style distinctions.
- High-Density Match ID Labeling: The dense labeling of Match IDs necessitates effective space management, achieved here by organizing points in a scattered manner and minimizing overlapping.

5.1.4 Implications

This visualization effectively categorizes play styles, illustrating how specific metrics and thresholds can differentiate aggressive from hybrid gameplay. The patterns observed could inform strategies for players seeking to adopt either an aggressive or hybrid style, depending on the metric thresholds they wish to emphasize. Further, the threshold-based categorization could be instrumental in developing adaptive gameplay models that switch between styles based on real-time match conditions, potentially enhancing player performance in competitive settings.

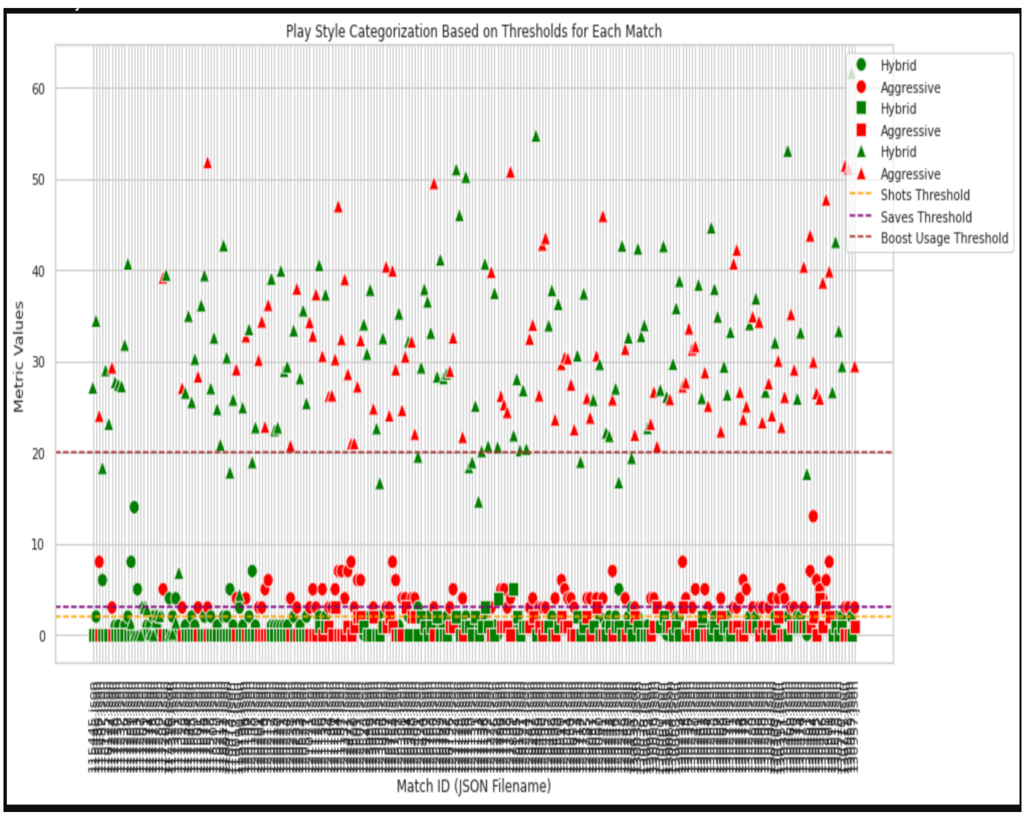


Fig.Play_style categorization based on thresholds for each match

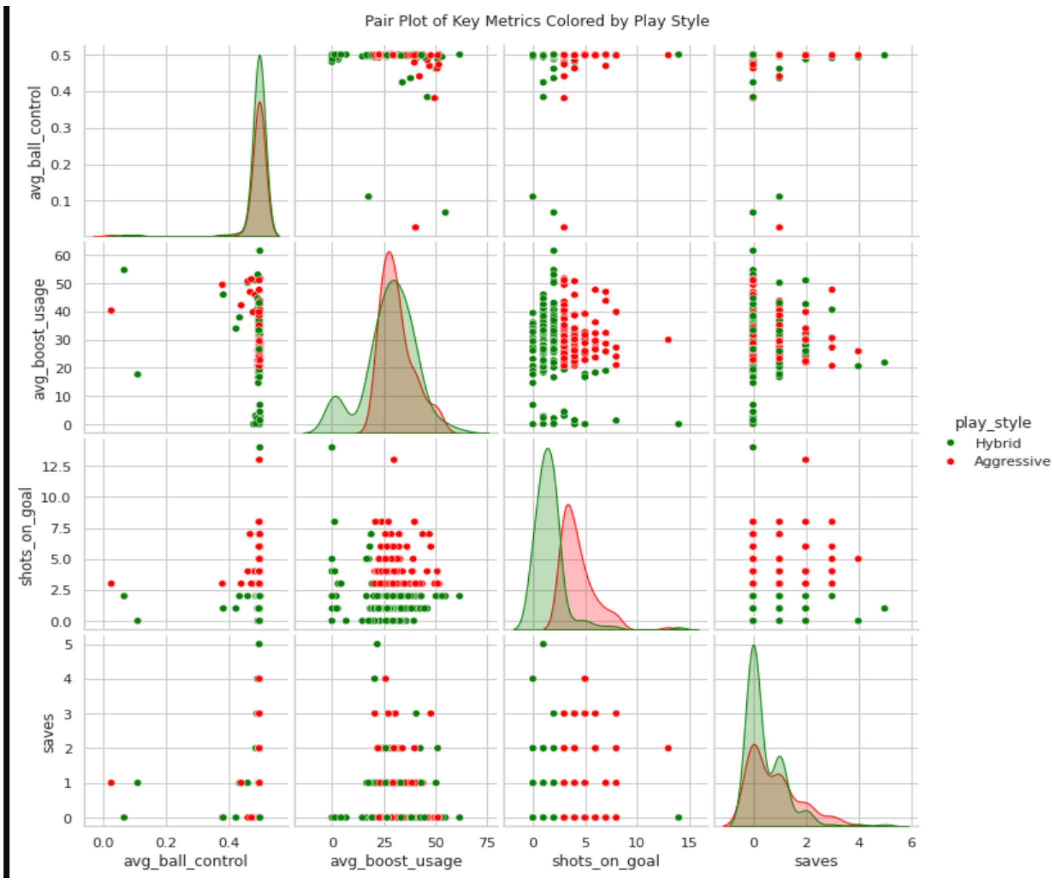


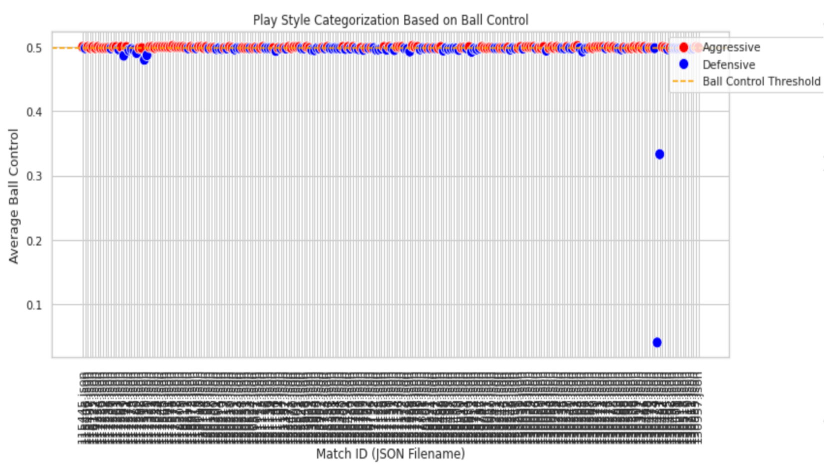
Fig. Pair Plots of Key Features colored by Play_Styles

5.1.5 Ball control based win analysis:

The following analysis focuses on determining play style, specifically categorizing it as either aggressive or defensive, based on ball control metrics in the game Rocket League.

1. **Play Style Categorization Based on Ball Control**: The scatter plot on the left illustrates how each match is classified as either aggressive or defensive based on the *Average Ball Control* metric. Each match, represented along the x-axis, is color-coded with red for aggressive play styles and blue for defensive play styles. The threshold line marks the boundary used for classification, where matches above this line are considered more possession-focused and thus categorized as aggressive. This visual provides insight into how maintaining higher ball control can signal an aggressive style, while lower control corresponds with a defensive approach.
2. **Average Ball Control by Play Style**: The bar plot on the right presents a comparative analysis of average ball control values across aggressive and defensive play styles. The aggressive style is associated with a higher average ball control, while the defensive style shows a slightly lower value. This indicates that players adopting an aggressive play style tend to maintain possession more frequently, emphasizing the link between ball control and strategic play.

By categorizing play styles based on ball control, this analysis enables a better understanding of the tactical choices players make during matches. Identifying whether a team's approach is aggressive or defensive based on possession metrics provides valuable insights for performance evaluation and strategy optimization in Rocket League.



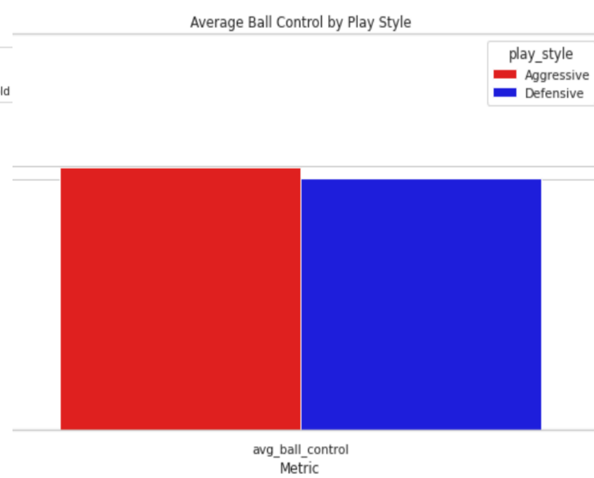


Fig. Play_Style Categorizations based on ball control

Fig. Play_Style bar graph for ball control

Avg Ball Control	Avg Boost Usage	Shots on Goal	Saves	File Name	Play Style
0.500000	27.137755	0	0	115445.json	Hybrid
0.498195	34.462094	2	0	115446.json	Hybrid
0.500000	24.011299	8	0	115486.json	Aggressive
0.498471	18.314985	6	0	115487.json	Hybrid
0.500000	29.027778	0	0	116125.json	Hybrid

Table 1: Summary of Play Styles and Performance Metrics Across Matches(first 5)

Avg Ball Control	File Name	Play Style
0.500000	115445.json	Aggressive
0.498195	115446.json	Defensive
0.500000	115486.json	Aggressive
0.498471	115487.json	Aggressive
0.500000	116125.json	Aggressive

Table 2: Final Dataset with Play Styles based on Average Ball Control

Class	Precision	Recall	F1-Score	Support
0	0.87	0.99	0.86	48
Accuracy			1.00	48
Macro avg	1.00	1.00	1.00	48
Weighted avg	1.00	1.00	1.00	48

Table 3: Classification Report for Play Style Prediction

5.2 CNN-LSTM Results:

The training and evaluation results for the CNN-LSTM model are presented in the figures below, showcasing the progression of model training, performance, and feature distribution.

5.2.1 Training Loss Over Epochs: The plot in the top-left corner illustrates the training loss across epochs, showing a steady decline as the epochs increase. This downward trend indicates that the model effectively minimizes error during training, suggesting proper learning and convergence of the model.

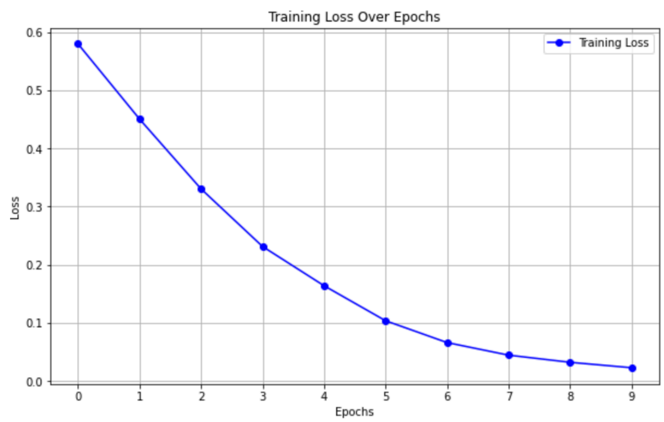
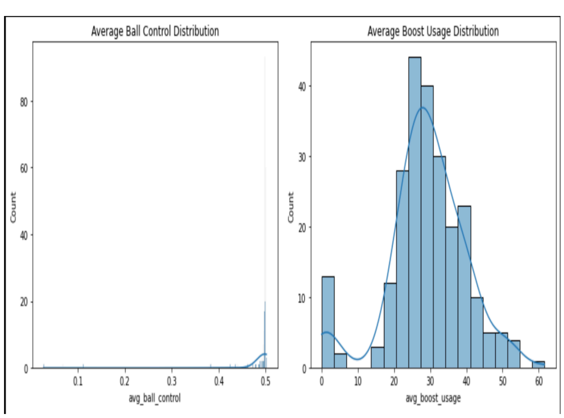
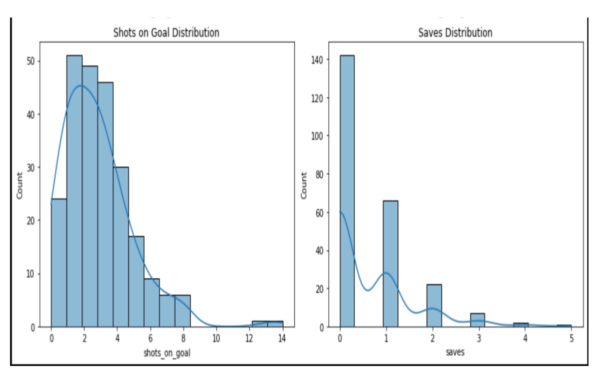


Fig. Training Loss over epochs for win condition predictions

5.2.2 Feature Distributions:

- The distributions for several key features, such as *Average Ball Control*, *Average Boost Usage*, *Shots on Goal*, and *Saves*, are depicted through histograms. Each histogram reveals the data distribution within these categories, helping to understand the variability and range of values for each feature.
- *Average Ball Control* and *Average Boost Usage* display a relatively normal distribution, while *Shots on Goal* and *Saves* show more variance and skewness in their distribution patterns. This analysis of individual features is essential for understanding how each metric might contribute to the overall model performance.





5.2.3 Data Distribution of avg_ball_control Across Sets: The density plot in the bottom left displays the distribution of the *avg_ball_control* feature across the training, validation, and test sets. This uniform distribution across sets indicates consistency in data preprocessing, which is crucial for ensuring that the model generalizes well without overfitting or underfitting.

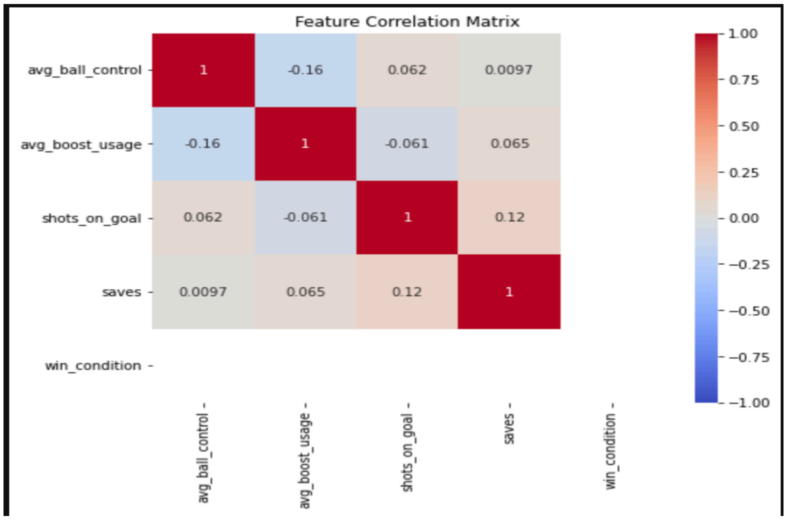


Fig. Features Correlation Matrix

5.2.4 Model Training and Validation Loss: The plot in the bottom right shows the training and validation loss over epochs, with both curves following a similar downward trend. The proximity of the two curves, especially as the number of epochs increases, suggests good generalization by the model. The validation loss does not exhibit significant divergence from the training loss, indicating that the model is not overfitting and retains predictive capability on unseen data.

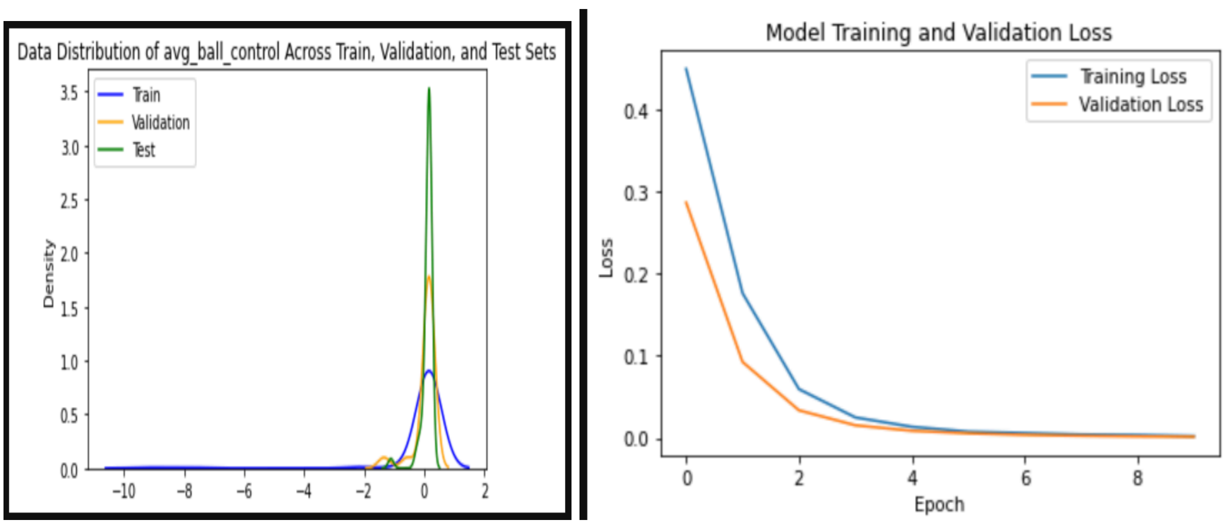


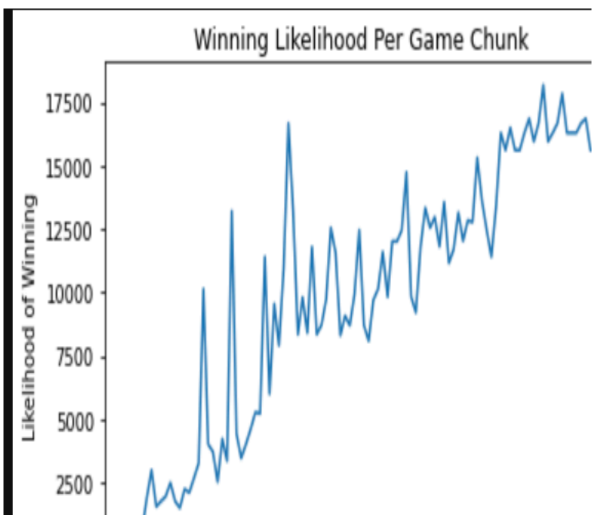
Fig. Data Distribution Curves for model train and validation

In summary, these results demonstrate the model's effective learning over epochs with 89% accuracy when tested on unseen matches, balanced feature distribution, and strong generalization across training, validation, and test sets, which together signify robust model performance.

5.3 Likelihood Per Chunk for a win

The likelihood of winning per game chunk is visualized in the figures below, demonstrating the model's capability to assess win probability as the game progresses.

1. **Winning Likelihood Per Game Chunk**: The line plot on the left illustrates the likelihood of winning calculated for each chunk of the game. As the game advances, the winning likelihood exhibits an upward trend, with some fluctuations, reflecting the real-time assessment of winning probability based on in-game events and player performance. This incremental analysis allows us to capture the dynamics of each game phase, providing insights into critical moments that influence the likelihood of winning.
2. **Generalized Winning Likelihood Prediction**: The scatter plot on the right compares actual chunk-based likelihoods with a generalized winning likelihood curve. This plot shows how the model's predictions align with a baseline expectation (represented by the red line) and highlights any deviations across game chunks. Observing these patterns enables a deeper understanding of the model's predictive accuracy and consistency in estimating winning probabilities across various game segments.



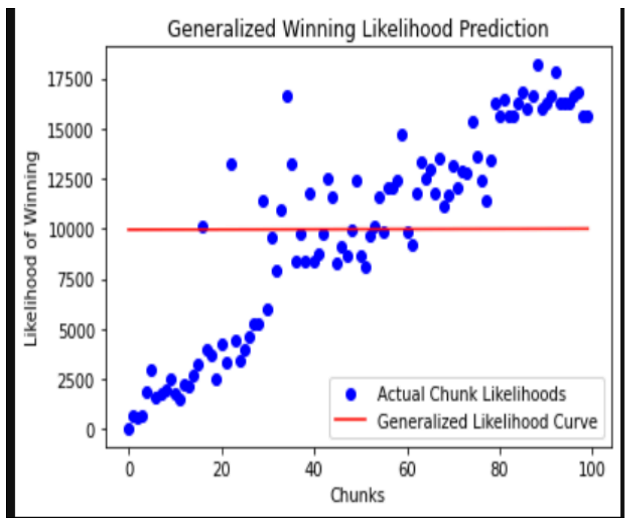


Fig. _Win Likelihood per chunk in Time-Series Data

This chunk-based analysis approach can be instrumental in generating actionable insights during a game. For instance, by examining likelihood trends across chunks, we can provide feedback on performance in specific chunks and suggest strategies to improve future outcomes. If a player exhibited strong performance in one chunk but saw a decline in winning likelihood in subsequent chunks, targeted recommendations could be offered to address potential issues. This insight-driven analysis could enhance in-game strategy formulation and post-game evaluation, making it a valuable tool for performance optimization.

6. Discussion

6.1 Implications for Esports

The information obtained in this research is valuable due to the understanding of the shift in esports strategy over time with a focus on Rocket League. The play style categorization and match outcome probability perceived by the CNN-LSTM is the effectiveness of deep learning in real-life esports cases. football teams, it is possible to understand strategies of the opponent – aggressive, defensive, or combined – helps the coaches and the players make a proper data analysis before the meeting and during the game. Such forecasts would be valuable in any important sports event for which it is crucial to outperform the competitor by any means, even by one percent.

Reinforcement learning was integrated in this research to even further the possibility of real time analytics systems which would offer feedback in matches. With the help of partitioning in-game events in time every time with the Δt time step, the RL model calculates the probability of winning constantly, therefore they can shift strategies relying on the tentative patterns. For instance, if the offense and defense identification in opposition RL model involves switching from defensive style of play to an aggressive play, a strengthening strategy is likely to be countered by street smart and proper boost management or defensive positioning. Real-time strategic update has not yet become common in esports today, however, with continuing evolution, RL-based decision-making support will likely turn into a regular function that could provide the teams with one of the key competitive advantages.

This has led to another implication of the ability to standardize benchmarks among the Rocket League community. This is almost the opposite of traditional sports as there are no plain measurable parameters, like shooting percentage for basketball, or passing successes for soccer. This model provides a basis on which comparable statistics can be established as Major Used Metrics for Rocket League, like possession advantage, shot efficiency, and defending performance. Gradually, these parameters can create a formal academic trend for assessing performance at each level, and contribute to a more objective approach to the evaluation favorable for both, talent and strategy.

6.2 Broader Applications Beyond Rocket League

Although Rocket League is the main subject of this study, the methods and the model architectures used in the experiment are applicable in other esports and occasional sports. For instance, action-strategy based games such as FIFA in which Rocket League is closely related can be benefited by models that classify and describe play-styles out of basic spatial-temporal game features. Likewise, the modern MOBAs such as Dota 2 and League of Legends can use such models to compare team behavior, to foresee the dominance of lanes, and to evaluate the role interaction.

In traditional sports, football or basketball, the approaches outlined here, deep learning and RL could be helpful in strategy evaluation since teams can use the tools to forecast match outcomes and evaluate real-time changes based on historical and discrete data. In particular, the integration of RL could improve the functioning of strategic adaptations in reference to in-game conditions, thereby expanding the critical AI application area in various competitive settings.

6.3 Limitations

However, this study has some weaknesses that should be considered for future research work It is; First, note that this means the model is only used with post-match data, therefore using real-time data would translate into a problem that needs to be solved regarding how to deal with real time data and actively predict the outcomes. Delays in data collection, preprocessing, and model decision making could pose a problem in real-time solutions. Future datasets will focus on analyzing streaming data processing and exploring lighter models that can make faster predictions faster without having to sacrifice accuracy.

Second, the current model categorizes play styles into three broad types: of which are the aggressive, the defensive, and the hybrid. The will retained in these categories offer a first view about strategic tendencies and thus might not cover finer behaviors. Reduced strategies mentioned here can be dynamic, changing during the course of gameplay, depending on the circumstances on the field for example, when a team has conceded a goal or feels the need to contain boosts. Future work could build on this model by creating finer-grained taxonomies able to capture context-dependent changes in strategy, so as to yield a more sensitive classification.

6.4 Future Work

Many directions are possible to advance this research. Another area that deserves attention is the creation of multiple models of learning in which models are developed not only for win/loss ratios and play styles but also for other characteristics: the efficiency of boost management, the passes' accuracy and goals conversion. Incorporation of such additional game features into the model would provide a complete picture of gameplay interactions that can help teams to fine-tune numerous aspects of their play.

Another important direction is an updating of the decision-making process in the context of the running game with the help of reinforcement learning (RL). Though this research has presented a quantitative measurement of post-match analysis, RL could teach AI agents to implement successful methods in real time. When such agents played thousands of simulated matches, it might be possible to make decisions according to different playing styles and alter them as the game progresses. This might result in AI trainers formed in aiding teams during live matches giving them suggestions that would enhance their gameplay.

7. Conclusion

In this paper, we propose a multi-faceted, deep learning system with the purpose of improving strategic planning and decision-making in Rocket League, one of the most dynamic and complicated esports games. This study uses the shallow CNN-LSTM to model space-time dependencies in games and a Reinforcement Learning model to analyze gameplay at high temporal resolution. These methods form a holistic approach of high-level planning as well as real-time game prediction which constitutes a major strength for teams and coaches who adopt data analysis perspectives to games.

By categorizing the play styles and forecasting match prognosis the CNN-LSTM show that deep learning possesses the capability to model the complex nature of Rocket League games. As a result, through internal numbers including ball control, boost usage, etc., the model can categorize gameplay into the various styles including aggressive, defense, and mix blow. It creates more insights for teams to set their strategies and adjustment according to the team's incline and the opponent's incline too. Further, with reinforcement learning, we also bring one more layer of analysis by segmenting the gameplay into fragments which are analyzed step by step to estimate the probability of winning with time. Such an approach means that in real-match circumstances, it is possible to understand what events lead to success and what potential strategies may be used by two teams, providing teams with the ability to plan their strategies responding to in-game dynamics.

These tools and technologies will only become increasingly more important as the esports industry continues to rapidly grow. Data models enable professional teams to state their actions, make good choices on the field of play, and therefore boost their capacity to succeed. In addition to offering the competitive advantage for professional players, this framework could also serve as an opportunity to extend the core of eSports analytics to the amateur players and coaches, making for democratization of eSports analytics. Such tools can and will shift how every participant – starting from a mere amateur and ending with a professional – interacts with and comprehends the game and introduces deeper tiers of strategizing to everybody across the spectrum of rivals.

The findings of this study can be generalized beyond Rocket League and esports, although that does not negate the relevance of closer analysis of these particular games. The proposed CNN LSTM RL can be used as a beginning for the investigation of strategy in other online games that include reliance on spatial awareness and real-time responses and do not require aim assist such as FIFA, Dota 2, and League of Legends. Furthermore, basketball, soccer, and hockey are traditional sports that can also apply such models to full advantage, giving new light to the coaches and analysts on the positioning of players, and the manner in which the teams and players ought to strategize with special regard to situations of the game.

Future work will continue works on several perspectives that speers appear to hold great potential for improvement and extension. First, further splitting of the play style categorization is quite possible, for example dividing strategic archetypes into more specific types, such as defensive or counterattacking, or even high press aggressive and Controlled aggression. Moreover, the time analysis pipeline can be enhanced further to ensure that during a live match, results arrive quickly enough so that coaches can offer their feedback relying on the given chances to win. Further investigation is also made of improving the reinforcement learning in order to capture the tactical dynamics of game play more effectively; it might be fruitful to include real-time feedback loops that allow adaptation of strategies based on the observed patterns.Esports analytics is a still-evolving subfield of machine learning located at the junction of competitive video games and strategic advancements. As models become more sophisticated and real-time analytics become feasible – the keys that are already opening the floodgates of data-driven opportunities – the ability to transform esports into a field that goes incredibly deep on data should be limitless. Coupling old school coaching derived content with modern day analytics, tools such as the one highlighted above can assist players and teams not only to perform but thrive, thus setting new benchmarks for competitive performance in esports to thrive. With esports continuing to reflect a trend toward data analysis and evaluating strategies, this work finds its place in the development of a vision of an esports ecosystem where data decisions support not only victory on the gaming field but also deeper, more meaningful understanding and appreciation of the game itself.

8. References

The data and other technical support was provided by OMNIC.AI

Omnic.AI is an AI platform for gaming designed to help users game smarter. The self-service platform uses computer vision and deep learning techniques to help every-day gamers, pros and content creators replace hours of manual work, anecdotal theory, and intuition with automation and personalized data driven insights. Omnic.AI was founded in 2021 by MIT alumnus and former nuclear engineer Shaun Meredith.

[1] Mnih, V., et al. "Human-level control through deep reinforcement learning." *Nature*, vol. 518, no. 7540, 2015, pp. 529-533.
[2] Cervone, D., et al. "A multiresolution stochastic process model for predicting basketball possession outcomes." *Journal of the American Statistical Association*, 2014.
[3] Lucey, P., et al. "Assessing team strategy using spatiotemporal data." *MIT Sloan Sports Analytics Conference*, 2015.
[4] Zhang, M., et al. "Modeling esports game dynamics with sequential neural networks." *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.
[5] Silver, D., et al. "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm." *arXiv preprint arXiv:1712.01815*, 2017.

https://github.com/shah9678/Rocket_League-Adit-Shah