

# Swinging Success: Applying Stock Momentum Models to Capture Tennis Momentum

## Summary

In the world of sports, we often hear the term *momentum* being used to describe the state of success for a player over an opponent, as a result of psychological and emotional shifts. The member who possesses this momentum at any given time through a contest is considered to be the likely victor given the current state, or at a turning point of favor. This is prevalent in the world of Tennis, where the back and forth game style, and constant exchange between two opponents give rise to changes in momentum between opponents.

However, due to the anecdotal nature of momentum and its lack of quantifying metrics, many who are involved directly with the sport doubt its impact on the outcome of any contest. Thus to analyze this claim, we conducted a data analysis and mathematical model to determine the effects of momentum on the outcome of a given match.

Firstly, we developed a momentum expression by tracking the point difference between players in a tennis match. Due to the similarities in point difference and a financial market, we utilized MACD (moving average convergence divergence) a common stock indicator for calculating the momentum. The momentum was calculated by taking the difference between two EMA's (exponential moving averages) of different window sizes. This captured which player was performing better and to what extent. Then, we were able to run correlation tests to determine if certain psychological events would have an effect on the momentum. The **statistically significant** correlations to the momentum was scoring **break points, double faults, and committing unforced errors**. We then plot these events on the momentum in order to visualize the expected increases and decreases of the momentum.

Secondly, we have developed a **Long Short Term Memory (LSTM) Prediction Model** to predict the point difference throughout the remainder of a match. The LSTM algorithm is optimized using Grid-Search and an Adaptive Moment Estimator using the **the best combination of hyper parameters**. We used all of the match metrics as the input into our model, training it to recognize patterns in the input features and predict point difference, and thereafter compute momentum flow and swing throughout the final set in the match. The models training results show a **MSE of 2.325 and MAE of 1.399, indicating great predictive performance** 5 The predicted results for the 2023 Wimbledon's Gentlemen's Final are **Carlos Alcaraz over Novak Djokovic** in an upset victory, we were also able to make predictions on multiple matches. We conducted a **sensitivity analysis** on our model by means of Permutation Importance. From the results of our Permutation Importance we derived a feature impact on momentum for each match, **determining which factors contribute largely to momentum**.

Thirdly, upon analysis of our results we found that **momentum flow plays a strong factor in determining the outcome of a given contest** in some cases more so than point difference. We explored the events in a match for a given player that affect their momentum, and the overall momentum of the match to advise them in future matches.<sup>1</sup>

<sup>1</sup>**Keywords:** LSTM, Permutation Importance, Adam, EMA, MACD

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Problem Background . . . . .	2
1.2	Restatement of the problem . . . . .	2
1.3	Data . . . . .	2
1.4	Our Work . . . . .	4
<b>2</b>	<b>Momentum in Tennis</b>	<b>4</b>
2.1	Momentum Equations . . . . .	4
2.2	Momentum Analysis . . . . .	5
2.2.1	Correlations . . . . .	6
2.2.2	Player's Advice . . . . .	7
<b>3</b>	<b>LSTM Deep Learning Prediction Model</b>	<b>9</b>
3.1	Description of LSTM Algorithm . . . . .	9
3.1.1	Parameter Tuning and Optimization . . . . .	10
3.2	Predictions on Momentum Swing . . . . .	10
3.2.1	Model Expansion Generalized Predictions . . . . .	13
3.3	Prediction Evaluation Analysis . . . . .	13
<b>4</b>	<b>Model Sensitivity Permutation Importance</b>	<b>14</b>
<b>5</b>	<b>Model Evaluation and Further Discussion</b>	<b>15</b>
5.1	Strengths . . . . .	16
5.2	Weaknesses . . . . .	16
5.2.1	Further Discussion . . . . .	16
<b>6</b>	<b>Letter</b>	<b>17</b>
<b>A</b>	<b>Appendix</b>	<b>18</b>
A.1	Player Performance Analytics . . . . .	18
A.1.1	Server's Probability of Winning . . . . .	18
<b>B</b>	<b>AI Reporting</b>	<b>19</b>

# 1 Introduction

## 1.1 Problem Background

In the world of sports the concept of momentum may be considered a quantifiable psychological factor, which has potential implications on the outcome of any contest. In the physical world momentum is the strength gained by a series of events, in sports, a player may feel they have the momentum during a contest, however it is difficult to measure because it is not apparent how various events during the match act to create or change momentum if it exists.

Tennis is a popular sport which is showcased globally in tournaments, where players from around the world compete, engaging in fast paced rallies to determine each point. Momentum throughout any tennis match may vary greatly between any consecutive set of points, games, sets, or matches.

Moreover, in the 2023 Wimbledon Gentleman's final, a world renown tennis tournament, an unlikely battle between young player Carlos Alcaraz and seasoned veteran, long time Wimbledon champion, Novak Djokovic unfolded, to the surprise of many. The match was a great battle, where Djokovic initially possessed momentum, earning a convincing 6 – 1 victory in set 1. However, thereafter rising star Alcaraz prevailed for a gritty win in a 7 – 6 tie breaker set. Entering the third set, Alcaraz, who now possessed momentum achieved a great victory 6 – 1 over Djokovic. Despite all momentum gained by Alcaraz, Djokovic managed to swing momentum in his favor, earning a 6 – 3 win. Finally, initially Djokovic managed to carry his momentum into the final set, however the young star rallied to conquer the great Djokovic in a triumphant 6 – 4 victory. By tracking the back and forth nature of tennis, and analyzing the changes in momentum throughout the matches and sets, we may be able to gain a better understanding of the patterns of momentum, to predict when it may change, and analyze its effect on the outcome of any contest.

## 1.2 Restatement of the problem

We need to analyze the data provided from Wimbledon 2023 men's matches after the first 2 rounds and answer the following questions:

1. Develop a model that captures the flow of play as points occur and apply it to one or more of the matches more of the matches.
2. Assess the claim of momentum having a role in the outcome of a contest
3. Predict swings in momentum throughout a contest, analyze contributing factors and advise players based on previous swings of momentum against new opponents.
4. Test and assess model on accuracy of predictions in momentum.

## 1.3 Data

**Data errors** There were several discrepancies that rose when converting the match's elapsed time to seconds. This conversion was necessary in order to calculate the time between each

point. The most common discrepancies appeared during a game when the time between points being over five minutes. Although rest breaks occur in tennis, the only breaks allowed are: 90 second break after the 3<sup>rd</sup> game, at least 2 minutes of rest between each set, medical timeouts and one bathroom breaks. Using the tennis's rules for break, we went through each instance where the time between points was greater than three minutes and replaced any values that were not consistent with the expected flow of the match. An example of an invalid time difference is in Table 1.

<i>match_id</i>	<i>elapsed_time</i>	<i>set_no</i>	<i>game_no</i>	<i>point_no</i>
2023-wimbledon-1303	00:54:22	2	7	83
2023-wimbledon-1303	24:56:34	2	8	84

Table 1: Example of invalid elapsed time between points in a match. The *set\_no* and *game\_no* were utilized to determine the accuracy of a time difference.

In this example, we see that for this match, the difference between the points is about a full day. We used the lengths of match recorded on Wimbledon website [4] to check the accuracy of the time difference. There were twelve other occasions where the elapsed time between a point was removed and replaced by the average for that match.

**Point Difference** An important value that was used for calculating and predicting momentum was the point difference, *point\_diff*, between the players. The point difference was calculated by taking the difference between each player's total points sequentially along the match. Positive values represent *player1* and negative values represent *player2*.

**Psychological Factors** These are values that we categorized as psychological factors as they are expected to have an impact on a players confidence when they occur.

<i>variable</i>	<i>definition</i>
<i>ace</i>	untouchable winning server
<i>untouch_shot</i>	untouchable winning shot
<i>fault</i>	missed both serves
<i>unforced_error</i>	unforced error
<i>point_net</i>	winning a point at the net
<i>point_break</i>	winning a point while receiving

Table 2: Variable names and definitions for the expected psychological factors on momentum.

## 1.4 Our Work

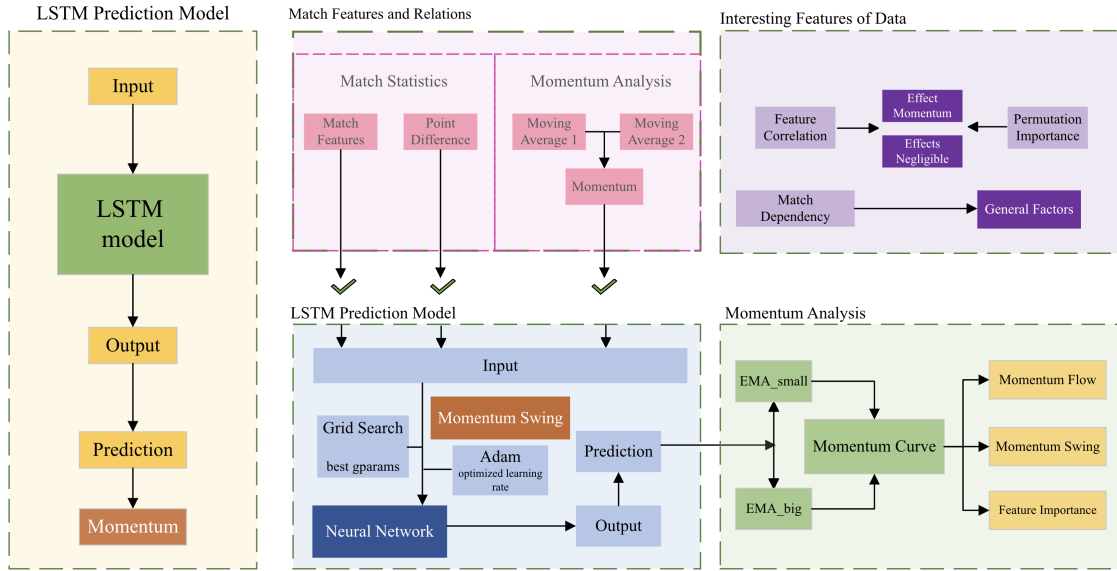


Figure 1: General Work Approach

## 2 Momentum in Tennis

Although tennis is a physical demanding sport, the mental side of the game is also a big factor. The best tennis players are able to dominate on the court while controlling their mind. Its important for the player to constantly stay in their own rhythm and understand what they are physically feeling [7]. In tennis, events will happen that will either boost or diminish player's confidence. Overtime these events help develop a psychological factor known as momentum, as players brains will commonly categories certain events as positive and negative. The obvious momentum indicator is when a player is on a streak of scoring consecutive points in a row. Additionally, there are more events that are expected to have an effect on momentum: Table 2. While formulating a plan on how to model momentum, we plotted the point difference,  $point_{diff}$  throughout a game in hopes to capture the game flow, Figure 2. We instantly thought of treating the point difference like a stock price as they oscillate in linearly. For stocks, a Moving Average Convergence/Divergence (MACD) [2] is commonly used to calculate a momentum oscillator for the price. We implemented a similar model of the MACD to our point difference to calculate the momentum throughout a match.

### 2.1 Momentum Equations

The momentum is calculated is by taking the difference between two exponential moving averages with different window sizes. These exponential moving averages were calculated using the

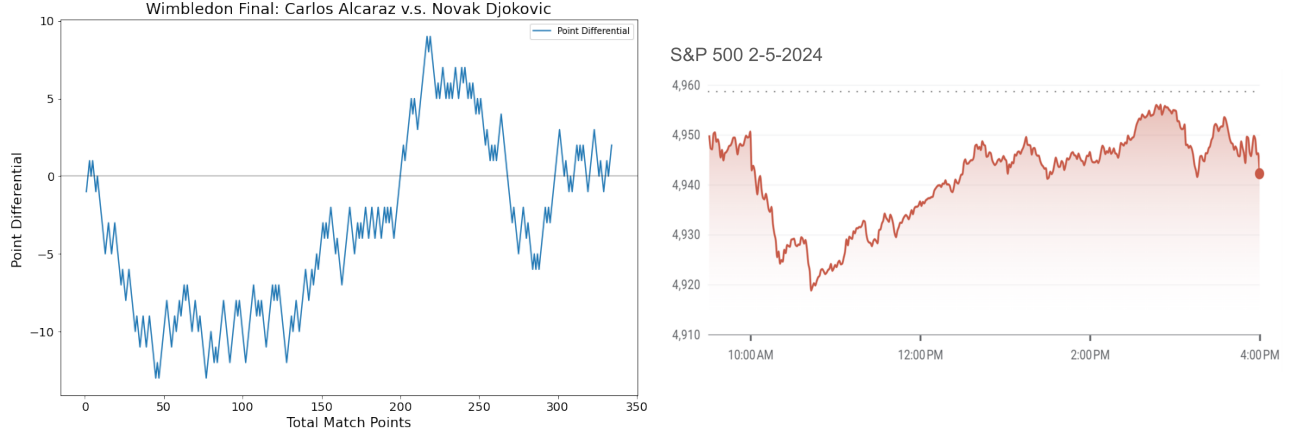


Figure 2: Comparison between a tennis game point differential and a stock market price. They both have the same shape and only make linear movements for every new data entry. The point differential is from the final match at Wimbledon between Carlos Alcaraz and Novak Djokovic. The photo of the stock prices is from the S&P 500 on 2-5-2024 [3].

following formula,

$$EMA = \frac{2(point_{diff} - EMA_{prev})}{window + 1} + EMA_{prev} \quad (1)$$

where,  $EMA$  represents the exponential moving average,  $EMA_{prev}$  is the previous average and  $window$  is the window size. The window sizes that we used were 12 for  $EMA_{small}$  and 26 for  $EMA_{big}$  [5]. The momentum was then calculated by taking the difference between the exponential moving average of the smaller window and the larger window.

$$momentum = EMA_{small} - EMA_{big} \quad (2)$$

When the point difference,  $point_{diff}$ , is increasing or decrease rapidly, this indicates that a player is building a streak of point, which can indicate a increase or decrease in momentum. Utilizing this moving average, we are able to now capture which player has the momentum and visualize it.

## 2.2 Momentum Analysis

The momentum curve will visualize what player has the momentum, the flow of the game and lastly, how much a player is performing better by the magnitude of the momentum. We ran the model on every game provided in the data set and Figure 3a presents the final match of Wimbledon between Carlos Alcaraz and Novak Djokovic and a fourth round match between Holger Rune and Alejandro Fokina, Figure 3b. This fourth round match was included due to the events that took place. Alejandro Fokina was able to establish a substantial lead in the point difference, that observing the point difference plot one would conclude that Alejandro Fokina won the match. However, Holger Rune won the match and the momentum plot is able to capture the flow of the game. From these momentum curves, it is apparent how momentum plays a role in winning games, as when a player has the momentum they are winning more games, which can be seen from the win pillars.



(a) Final: Carlos Alcaraz versus Novak Djokovic

(b) Fourth Round: H. Rune versus A. Fokina

Figure 3: The top plot is the point difference ( $point_{diff}$ ) throughout the match, where there is constant changes. The small pillars represents when a game is won by a player and the larger pillars is when a set is won by a player. The bottom plot is the momentum curve and at each spot the  $EMA_{small}$  and  $EMA_{big}$  intersect the momentum shifts. The star by the players name indicates the winner of the match.

### 2.2.1 Correlations

We then looked into the psychological factors that is expected to affect a players momentum (Table 2). We ran a Pearson correlation test on each variable against the momentum for all the data, Figure 4. The winning untouchable shot, break points, unforced errors and double faults all had p-values below 0.05, which infers that their correlations are statistically significant. However, the correlation values for the winning untouchable shots and double faults are so small that they may not be a reliable predictor for momentum swings. The effect of when a player wins a break point and unforced errors is visualized in Figure 5a and exemplifies when a player scores the break point the momentum increases towards their way. The break point has the largest correlation among all values, which is expected as the player at that time had the lower probability of winning [Appendix A.1]. When a player commits an unforced error, Figure 5b, displays how a player's opponent's momentum will increase. This is indicated by the momentum going in the opposite direction of that player (positive for  $player_1$  and neg for  $player_2$ ). It is expected that when double faults occur, the momentum will shift towards the other player,

Figure 5c captures this, but it is only clear for *player*<sub>2</sub>. This infers that, based on our data, players in *player*<sub>1</sub> column were less affected by double faults compared to those in *player*<sub>2</sub> column.

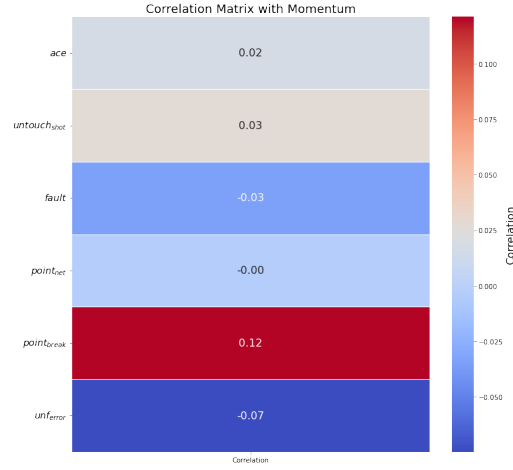
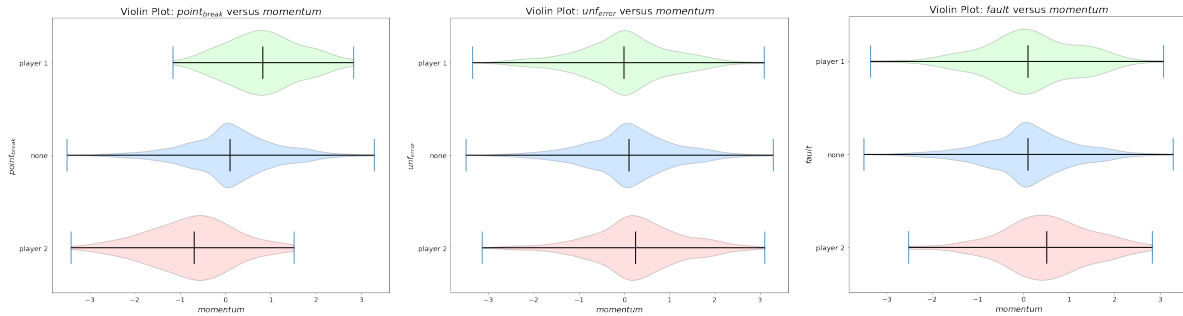


Figure 4: Heat map of the Pearson correlations between the psychological variables (Table 2) and *momentum*



(a) Violin plot of the *point<sub>break</sub>* versus *momentum*. (b) Violin plot of the *unf<sub>error</sub>* versus *momentum*. (c) Violin plot of the *fault* versus *momentum*.

Figure 5: Violin plots to visualize the correlation between the psychological variables and momentum. A positive momentum is for *player*<sub>1</sub> and a negative momentum is for *player*<sub>2</sub>. **Figure 5a** displays the relationship between them, which indicates that when the a player wins a break point their momentum increases as well. **Figure 5b** displays the relationship between them, which indicates that when a player commits an unforced error their momentum decreases. **Figure 5c** displays how *player*<sub>2</sub> is the affected more by committed double faults compared to *player*<sub>1</sub>.

### 2.2.2 Player's Advice

From the previous section, it is clear that the best advice to give for building momentum is to capitalize on scoring points when you are receiving the serve (*point<sub>break</sub>*). This is the general



advice that can be given to all players, however, if we were to give advice to a specific player it would be specialized to that player. For specific player advice, we observed Carlos Alcaraz's final match in the Wimbledon tournament. The Pearson correlation between the variables in Table 2 and the momentum, resulted in a statistically significant positive weak correlation for the break points and a statistically significant negative weak correlation for the unforced error (Table 3). Using Figure 6, we were able to visualize the change in momentum for when these events occurred. From these plots, the general trend is when a "positive" event occurred, ace and break point, the momentum of that player increases, where as, for a "negative" event, double fault and unforced error, the momentum of that player decreases. When a negative event takes place close to each other, it is possible that the previous event had an impact on the mentality of the player, which results in another negative event. The advice for Carlos Alcaraz would be the obvious, to minimize the double faults and unforced errors, but also, a short term memory because if you focus on the negative event, your rhythm will be lost and so will your momentum. Attempt to stay neutral throughout the match and do not get over excited about positive events as it could diminish your momentum later. In sports, when a player becomes overly confident they are more prone to start to think too much when they are faced with adversity [7].

variable	Pearson	p-val
$point_{break}$	0.1266	0.021
$unf_{err}$	-0.1181	0.031

Table 3: Final Match's correlation between psychological variables and momentum.

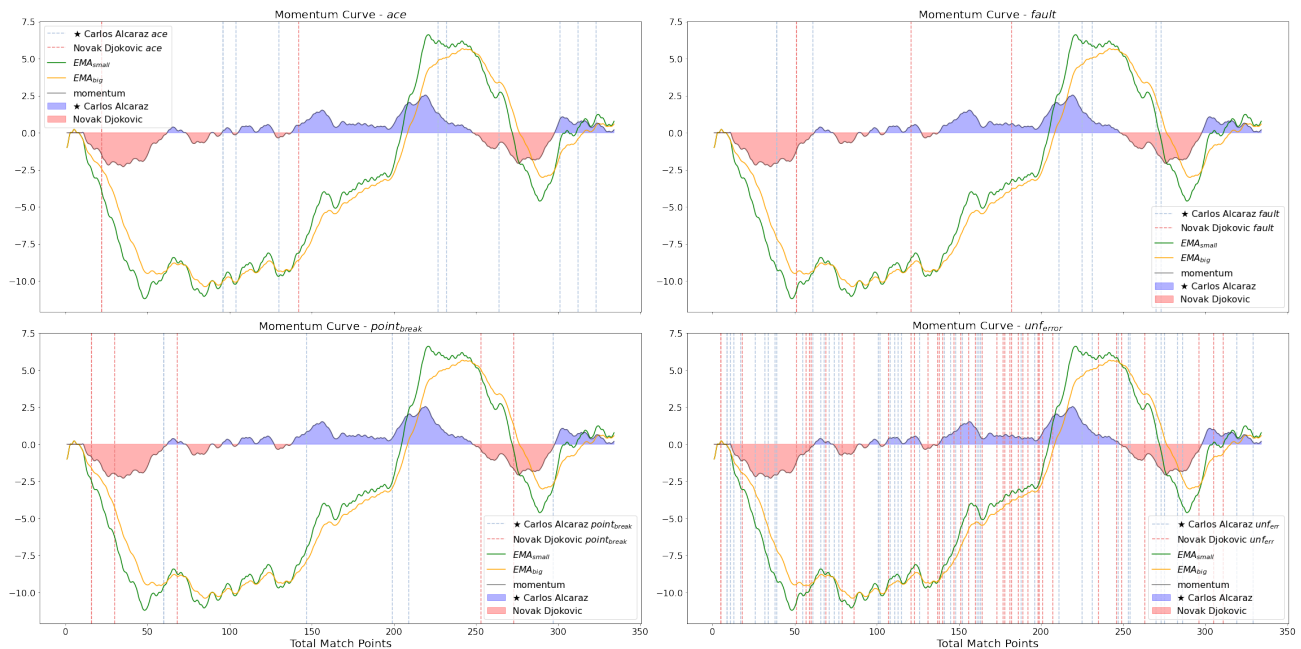


Figure 6: Momentum Curve for Final match between Carlos Alcaraz and Novak Djokovic including markers where an event occurs that could impact the momentum. There is a plot for the following psychological variables:  $ace$ ,  $fault$ ,  $point_{break}$ ,  $unferr$

### 3 LSTM Deep Learning Prediction Model

In section 1.3 we calculated  $point_{diff}$ , as expected  $point_{diff}$  varies throughout any given contest. By analyzing the changes in this data throughout the entirety of a match and studying trends in our momentum calculations we can make informed predictions about future swings in momentum throughout a given match. In this section we applied the Long Short Term Memory (LSTM) deep reinforcement learning algorithm to perform machine learning on the provided point difference up until a specific point into the match, and ultimately make a prediction for the changes in momentum in the remainder of the match.

#### 3.1 Description of LSTM Algorithm

LSTM (Long Short Term Memory) is a type of recurrent neural network (RNN) that is commonly used for time series analysis, and excels in capturing long term dependencies in sequential data which is ideal for capturing momentum trends throughout a contest.

The main idea behind the LSTM architecture is the cell state, where information flow is regulated by three sigmoid gates. An output between zero and one from the sigmoid gate indicates how much of the information to let through, 0 being none and 1 being all, they protect and control the cell state. The first step, filter out unwanted information via the first sigmoid gate called the *forget gate layer*[6]. Secondly we decide what new information will be kept in the cell state by, the *input gate layer*, thereafter a *tanh* layer creates a vector of potential new information, then combining the two the LSTM creates an update to the cell state.

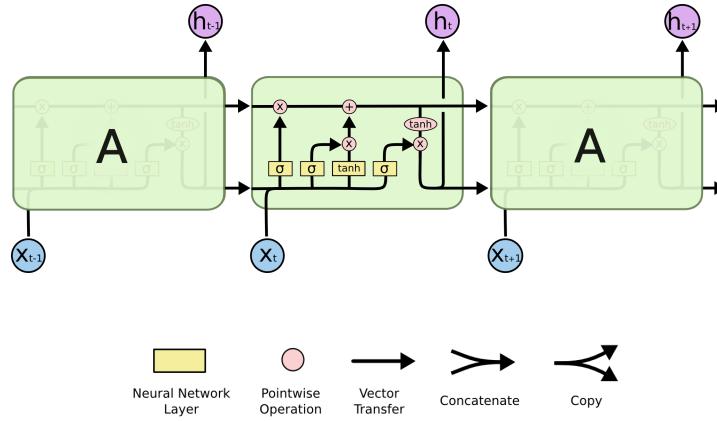


Figure 7: An Overview of the Algorithm Flow of LSTM [6]

The update equations of the LSTM are as follows:

$$\begin{aligned}
 f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
 i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
 \tilde{C}_t &= \tanh(W_C \cdot [h_t, x_t] + b_C) \\
 C_t &= f_t \cdot C_t + i_t \cdot \tilde{C}_t
 \end{aligned} \tag{3}$$

Where:

Symbol	Description
$f_t$	the forget gate
$i_t$	the input gate
$\tilde{C}_t$	potential information vector
$C_t$	new cell state
$\sigma$	the sigmoid function
$*$	the Hadamard product
$h_t$	the hidden state at time t
$h_{t-1}$	the hidden state at time t-1 or at time 0
$x_t$	the input
$b_f, b_i, b_C$	the parameters that need to be trained
$W_f, W_i, W_C$	the parameters that need to be trained

Table 4: Notations used in Equation (2)

In this approach, the sequential time series data of a match is fed into a LSTM model to learn the patterns in the sequence, which can then be used to make predictions on future data points.

### 3.1.1 Parameter Tuning and Optimization

The LSTM algorithm consists of GridSearch and Adaptive Moment Estimation (Adam) for tuning and optimization.

GridSearch is a parameter optimization algorithm used for fine-tuning hyper-parameters in machine learning models to optimize their performance. It iterates through a specified parameter grid and trains and evaluates the model for each possible parameter combination. Outputting the best parameters and corresponding model performing metrics (Equation MSE MAE).

The Adam optimizer is commonly used in training neural network models like our LSTM model. It combines Momentum optimization, to help accelerate optimization process, and Root Mean Square Propagation, which scales the learning rates for each parameter as well as maintain two moving averages for each parameter. In an LSTM involved in handling dependencies in sequential data, Adam can help in navigating through the gradients more effectively.

Using Gridsearch and Adam in the LSTM model algorithm allow for systematically exploring different combinations of hyper-parameters, and find the optimal configuration that results in best performance for our LSTM model.

## 3.2 Predictions on Momentum Swing

Leveraging the comprehensive and powerful machine learning library in Python, we opted to use the LSTM model provided by TensorFlow. TensorFlow is a machine learning library based

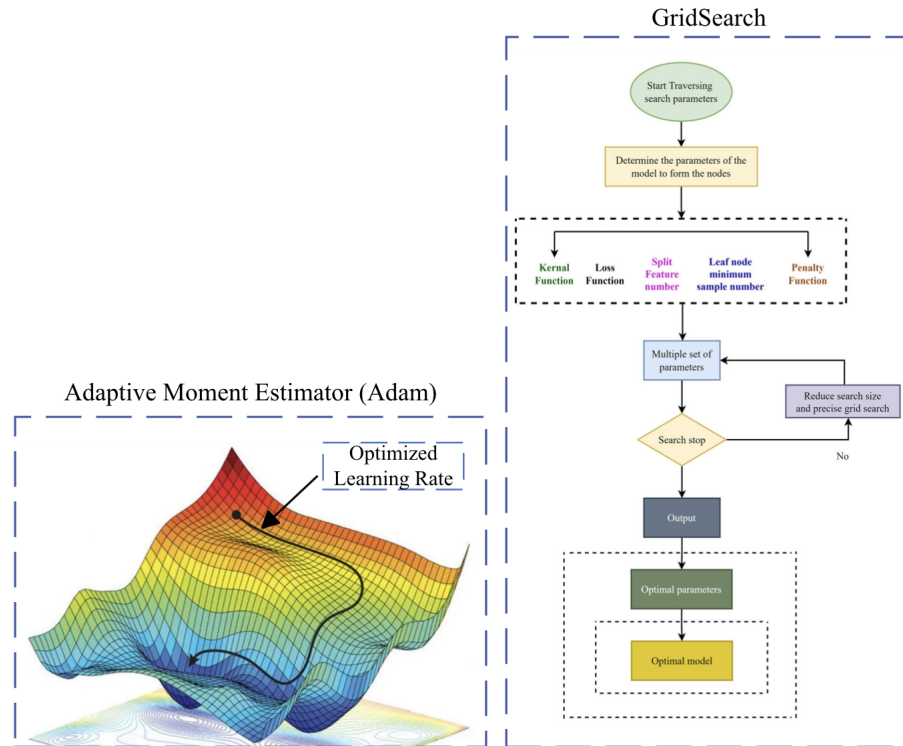
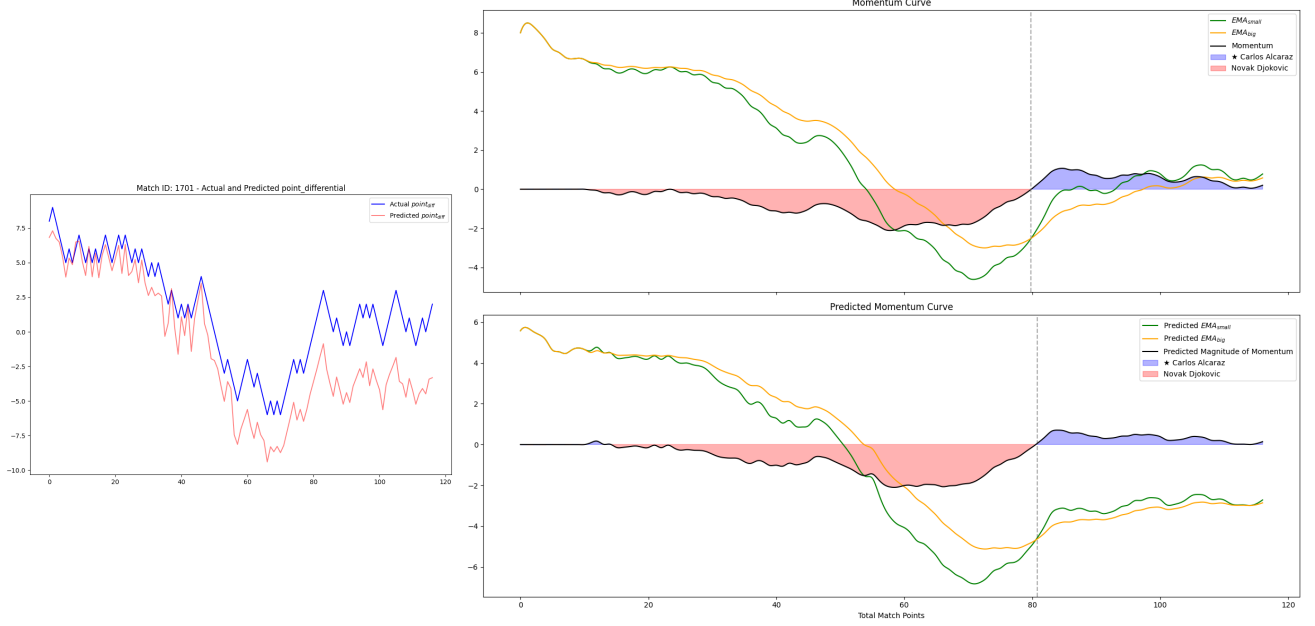


Figure 8: Visual of GridSearch algorithm flow and, visual representation [8] of how the learning rate changes over time during the training process. Learning rate decreases gradually to allow the model to converge to a minimum without oscillations or overshooting.

in Python, it offers seamless Keras integration, which is a high-level neural networks API. This provides a powerful simple method of building comprehensive LSTM neural networks. A distinguishing feature of TensorFlow is eager execution enabled by default. Eager execution allows for immediate evaluation of operations, making it easier to debug and iterate over model development. It's particularly beneficial when working with dynamic architectures like LSTM's, and is well suited for making predictions. In TensorFlow we can utilize the `tf.keras.models[1]` module to easily construct and train LSTM models as well as use the model for making predictions. This method enables an elegant simple solution to making a powerful comprehensive model, capable of accurate predictions.

We specifically modeled a sequence-to-vector (one) architecture. This LSTM structure is effective for tasks where the goal is to predict a single dimension vector of values ( $point_{diff}$ ) based on a sequence of input data. In sequence-to-one architecture, the LSTM layers act as hierarchical feature extractors. Each LSTM layer processes the input sequence, extracting relevant features and patterns. The subsequent layers build upon the hierarchical representation, learning more abstract and complex features. LSTM's excel in capturing sequential dependencies, which can then be used to predict future data points.

From Figure9b, it can be seen that our model quite accurately predicts the events of the 2023 Wimbledon Gentleman's final, where Carlos Alcaraz in blue, successfully gained momentum over Novak Djokovic in the final games of the last set. Our momentum swing indicator,

(a) Predicted and Actual  $point_{diff}$ 

(b) Predicted and Actual Momentum Curve

Figure 9: Momentum on last set in match, from prediction on  $point_{diff}$ , of the last set. Momentum curve represents momentum of last 120 points in the match between Novak Djokovic and Carlos Alcaraz, dashed indicator represents moving average intersection, predicting swing point of momentum.

the intersection of  $EMA_{small}$  and  $EMA_{big}$ , as illustrated by the dashed line, accurately predicts the moment where momentum would swing thereafter. An analysis on **Pearson correlation reveals a linear agreement of 90.096%** between our prediction and actual. Furthermore the correlation coefficient of 0.90, is supported by a **p-value significantly less than 0.05**, therefore this correlation is statistically significant.

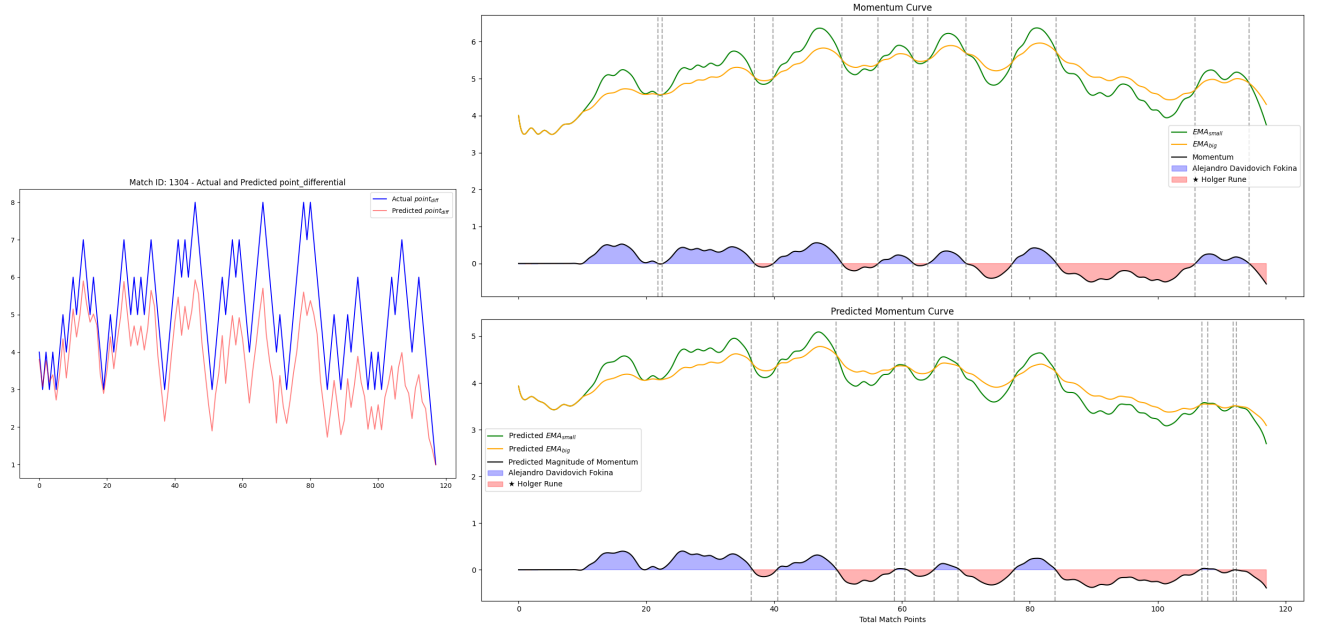
To further assess the performance of our LSTM model we can calculate Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2}{N}} \quad (4)$$

Where:  $y_i$  is our actual value and  $\hat{y}_i$  is our predicted value. We find that **RMSE= 1.525% with accepted values below 10%, indicating a small error**. This supports acceptable accuracy from our LSTM model

### 3.2.1 Model Expansion Generalized Predictions

Using the former LSTM model described we can generalize the model to predict the momentum curve of and outcomes of any tennis match. An interesting match to analyze; Alejandro Davidovich Fokina v Holger Rune in the 2020 Wimbledon Gentleman's final, where the point difference favored Alejandro for almost the entirety of the match, but the victor was Holger (Figure 3b). We can evaluate how our model preforms in the unlikely scenario by trying to predict its outcome by predicting match flow.



(a) Predicted and Actual  $point_{diff}$

(b) Predicted and Actual MACD

Figure 10: Momentum on last set in match, from prediction on  $point_{diff}$ , of the last set. Momentum curve represents momentum of last 120 points in the match between Alejandro Davidovich Fokina and Holger Rune, dashed indicator represents moving average intersection, predicting swing point of momentum.

The LSTM model performs well on this anomaly of a match, we can see this match has many swings in momentum, and in the last games of the set it favored Holger. This prediction supports the claim that momentum has an impact on deciding the outcome of a match, furthermore this prediction supports ability of our model to be generalized to other matches, certainly Women's tennis and potentially other sports.

### 3.3 Prediction Evaluation Analysis

To evaluate our models performance, we used the common method of Mean squared Error (MSE) and Mean Absolute Error (MAE). MSE represents the average of the squared differences between prediction and actual, while MAE is the average of the absolute differences between the prediction and actual data. They were calculated as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (5)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (6)$$

Where:  $n$  is the number of samples,  $y_i$  denotes the origin data, and  $\hat{y}_i$  is the prediction data.

According to the results of our prediction, **the LSTM Prediction Model had a very good prediction performance for the swing in momentum throughout the last set of the match in the 2023 Wimbledon Gentleman's Final, with a MSE of 2.325 and MAE of 1.399.** This indicates that the model was able to accurately predict the momentum swing and outcomes of the set, demonstrating the effectiveness of the LSTM algorithm for this task.

Based on the above analysis, we consider our **prediction model** to be **comprehensive and accurate.**

Considering that a set and match outcome have a clear correlation with momentum,<sup>2</sup> we should begin to consider events and features that have strong impact on the momentum swing throughout a match. We can evaluate the impact certain features have on the magnitude and changes of momentum by looking at feature importance.

## 4 Model Sensitivity Permutation Importance

We conducted a sensitivity analysis on the input parameters of the LSTM prediction model to test its sensitivity to changes in the input parameters in predicting *point<sub>diff</sub>* from which the momentum flow was directly calculated. The specific method we used was Permutation Importance to assess feature importance of our model.

Feature importance is a major part of model evaluation. Without determining what model inputs are the most important, any model is a “black-box” model, and given the magnitude of our input parameters feature importance is essential. It gives us the ability to learn what inputs affect model predictions and those that do not.

Permutation importance measures the change in model error after a single model's features values have been shuffled. Our iteration of permutation importance measures changes in error by calculating differences in MSE, Equation 6.

---

<sup>2</sup>This claim is made on the two matches analyzed, and our models performance. More matches can be predicted with our model to strengthen this claim.

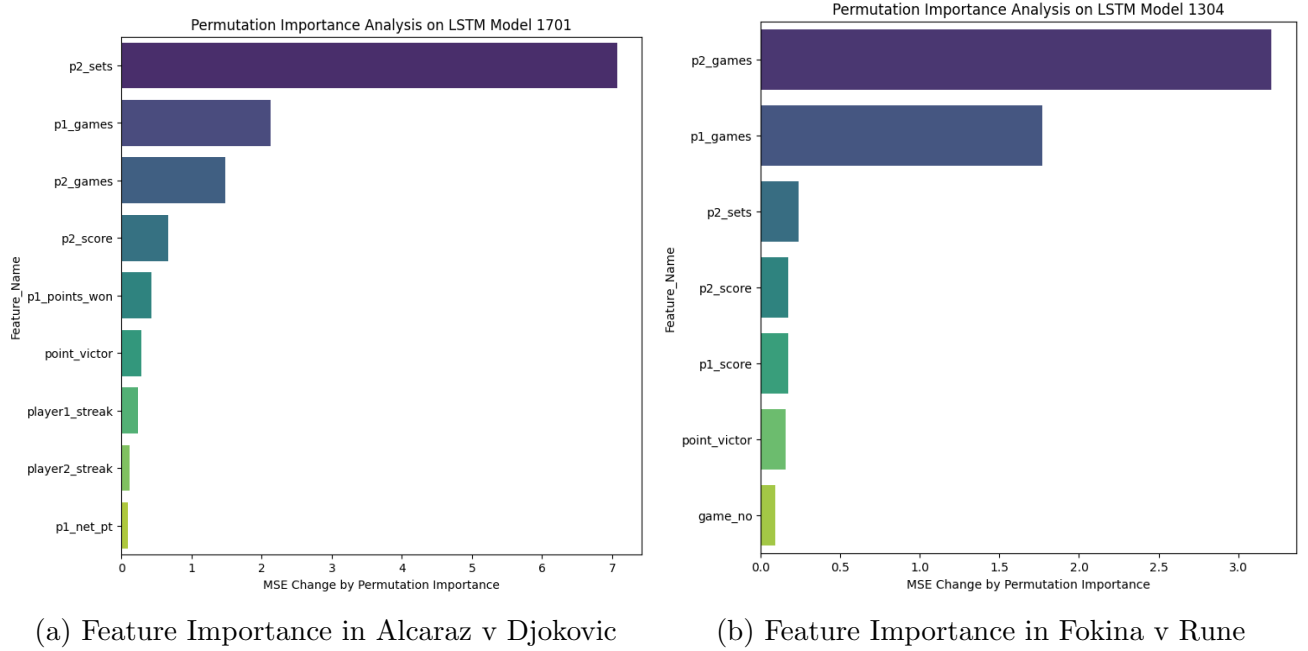


Figure 11: Permutation Importance for the last set of each match, where the magnitude of importance is calculated by the difference in MSE after shuffling. Only Features with an MSE of  $\geq 0.008$  are displayed.

This method is useful given our LSTM model<sup>3</sup> and problem since our momentum curve is calculated from *point\_diff* predictions. It allows for effective and computationally efficient sensitivity analysis. Since the model does not need to be rerun in order to compute permutation importance, rather it is derived directly from the output of our model.

Analyzing Figure 11 we see that different features have varying importance in determining the momentum throughout a match, and is match dependent. These results support the findings in Figure 6. Not only does permutation importance provide a useful sensitivity analysis of our LSTM, it provides insight into the behaviors of momentum and how it manifests throughout a match.<sup>4</sup>

Conducting a permutation importance evaluation on our model, trained for each match we would be able to average the feature importance's and see if there are any dominating features which impact momentum more often.

## 5 Model Evaluation and Further Discussion

Based on the performance of our model and evaluating the necessary metrics, we can assess its strengths, weaknesses and expand on its implementation.

<sup>3</sup>This is not a general statement. Caution should be used when using permutation importance on LSTM's, given the objective of our model and its architecture, permutation importance is comprehensive.

<sup>4</sup>Note that MACD, EMA, and *point\_diff* were dropped when computing importance. These features would dominate MSE. We want only direct match feature correlation.



## 5.1 Strengths

1. **Momentum Evaluation:** Momentum expression has the abilities to calculate the momentum through the match utilizing the point difference and exponential moving averages. The moving averages are able to detect when players start outperforming their opponent. This provides the base calculation for the momentum and the prediction model is able to find the variables that influence this momentum.
2. **LSTM:** The LSTM algorithm in Section 3 has several advantages when used for prediction tasks, especially those given a high magnitude of features. LSTM excels in handling sequential data and identifying complicated trends. This algorithm provides a simple approach and solution to complex problems, with easily scale-able model complexity to capture difficult predictions accurately.
3. **GridSearch and Adam:** Adopting a GridSearch and Adam optimizer into our model greatly improves its performance and combats any weaknesses associated with LSTM parameter complexity. After running our GridSearch and optimizer our algorithm is hyper-tuned to our problem and avoids over and under fitting.
4. **Momentum Curves and Feature Importance:** From our predictions on  $point_{diff}$ , a simple solution to predicting momentum is a direct result. Our model is able to derive a considerable amount of information given its simplicity. We are also able to efficiently conduct further analysis with permutation importance. Our model directly builds off of itself, **running one model, one time we can accurately predict the momentum flow and swing, outcome, and variables that effect momentum the most.**

## 5.2 Weaknesses

1. For real time analytics and predictions it's computational efficiency should be further optimized for run time.
2. Widow size in  $EMA_{big}$  and  $EMA_{small}$  moving averages naturally cause a point *delay* before predicting starts in our model.
3. Training time and Data Requirements, because we are limited in the amount of data per match, and LSTM's prefer large datasets, our model must be trained longer to achieve our results.

### 5.2.1 Further Discussion

The LSTM model[3] can accurately predict more than one match, and based on our evaluation we are confident it can predict any tennis match, men or women's. However we are not certain in the generalization of our model and approach to all sports. Contests with similar play style and scoring to tennis such as, but not limited to, Squash, Racquet Ball, and Table Tennis, should perform well in our model.

Further research and analysis will need to be conducted to strengthen this claim. And assess its capabilities predicting sports of different nature, such as Football, Baseball, or Soccer. Where point differences are less consistent.

## 6 Letter

Dear Tennis Coaches,

I am writing to you to inform you about recent results regarding a study on momentum in tennis.

Momentum is known as a important psychological factor that plays a role in all sports. It is hard to properly quantify the momentum within a sport especially a game of tennis, that requires a athletic physic and a strong mind. Utilizing the data from the 2023 Wimbledon tournament, we set out to model the momentum, which captures the flow of the game, and what are the important factors that play a role. We were able to achieve this by observing the point difference between two players over the course of a match. The point difference is the difference between the total points each player has won along the match.

Firstly, we established a Momentum Model that calculated two exponential moving averages on the point difference throughout the match. Then the difference between those exponential moving averages represented the momentum of the game. Since averages are moving in small windows, the model is capable of capturing when players are starting to perform better than their opponent. This is used to then quantify the momentum and the magnitude of this difference is how much the momentum a player has at that point. We found that the statistically significant correlations between psychological factors and momentum was scoring break points, double faults, and committing unforced errors.

Secondly, we developed a LSTM Prediction Model to predict the point difference throughout the remainder of a match. We used all of the match metrics as the input into our model, training it to recognize patterns in match statistics and predict point differences thereafter. Capturing momentum flow and swing throughout the final set in the match. The models training results indicate great predictive performance and support the game that momentum plays a role in determining match outcome. The predicted results for the 2023 Wimbledon's Gentlemen's Final are Carlos Alcaraz over Novak Djokovic in an upset victory, we were also able to make predictions on multiple matches. Thirdly, we conducted a sensitivity analysis on our model. From the results we derived a feature impact on momentum for each match, determining which factors contribute largely to momentum.

Upon analysis of our results we found that momentum flow plays a strong factor in determining the outcome of a given contest in some cases more so than point difference.

We would advise to consistently inform the athletes to stay neutral throughout the match. What we mean is to just observe the event from a non bias point of view as this will allow the player to not categorize an event as negative nor positive. Of course, enjoy the positive moments, but use them as more motivation and to not get complacent. As we were able to see in multiple matches when a player had a large lead and got complacent, the opposing player was able to gain momentum, which was present on the momentum curve. If a negative event happens, focus on the current point, ignore the past and the future. It was apparent in multiple

matches, that when multiple negative events happens close to each other, the inner mental game of tennis was the reason for the player losing momentum.

## A Appendix

### A.1 Player Performance Analytics

#### A.1.1 Server's Probability of Winning

In tennis, it is expected that the server has an advantage over the receiver, to prove this statement we the calculated probability for a player to win the point while serving. This was achieved by summing the total points won by a server over the total points for the whole dataset provide on Wimbledon.

$$p_{win} = \frac{\sum serverWinPoints}{\sum points} \quad (7)$$

Using equation 7, we calculated the probability of a server winning the current point [5].

<i>serverWinPoint</i>	<i>points</i>	<i>p<sub>win</sub></i>
4903	7284	0.673

Table 5: Values and Results for Calculation of Probability a Server Wins the Point.

A hypothesis test was ran with these values and a *p-val* of 1 was calculated, therefore, we fail to reject the null hypothesis and we can use this value to represent the player's probability of winning the point when serving. We are also assuming that each serve is independent of each other, although, this is not the case as it will be proven wrong when we model the momentum.

## References

- [1] Keras: The high-level api for tensorflow. <https://www.tensorflow.org/guide/keras>.
- [2] MACD - Wikipedia. <https://en.wikipedia.org/wiki/MACD>.
- [3] S&p 500 google finance. <https://www.google.com/finance/quote/.INX:INDEXSP?sa=X&ved=2ahUKEwi-lMOEoZWEAxXnk4kEHXI1DxgQ3ecFegQIKxAX>.
- [4] Wimbledon 2023 results. [https://www.wimbledon.com/en\\_GB/scores/results/day14.html?event=MS](https://www.wimbledon.com/en_GB/scores/results/day14.html?event=MS).
- [5] Gerald Appel. *Technical analysis: power tools for active investors*. FT Press, first edition, 2005.
- [6] Colah. Understanding lstm networks. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>, 2015. Published on August 27, 2015.
- [7] Timothy Gallwey. *The Inner Game of Tennis*. New York, NY, 1974.

- [8] Tech-AI-Math. Optimization techniques in machine learning. <https://ai.plainenglish.io/optimization-techniques-in-machine-learning-8b4f7325295>, September 2023.

## B AI Reporting

**User:** Explain to me how the MACD work for a stock price like I am a 5 year old.

**ChatGPT:** Alright, imagine you have a magical pair of glasses that can show you if a stock is having a good or a not-so-good time. Now, these glasses have two special lenses, let's call them Fast Lens and Slow Lens.

Fast Lens: This lens looks at how fast the stock is moving recently. If it sees the stock is running really fast, it says, "Hey, the stock might be doing well!" If it's not moving much, it says, "Hmm, the stock might not be so happy right now."

Slow Lens: This lens takes its time to look at the overall movement of the stock. It's like the wise old owl of the glasses. If it sees the stock has been consistently going up, it says, "Ah, things seem good for the stock." If it's going down a lot, it says, "Hmm, the stock might not be having a great time lately."

Now, the special thing is, when you put these two lenses together, they create a magic number that helps you understand if the stock is generally happy or not. If the Fast Lens is running ahead of the Slow Lens, it might mean the stock is feeling extra positive. But if they are not really agreeing, the stock might be a bit uncertain.

That magic number is called the MACD, and it's like your glasses telling you a story about how the stock is feeling right now. So, when you hear people talk about MACD in the stock market, just remember it's like magical glasses helping them see if the stock is having a good or not-so-good time.

**User:** Explain the proper implementation of LSTM and how it works

**ChatGPT:** Alright, let's break down LSTM (Long Short-Term Memory) in a simpler way:

Imagine you have a special robot assistant (LSTM) that helps you remember important things over a long period. Here's how it works:

Memory Cell:

The robot has a memory cell, like a tiny notebook, to jot down information. It can read from and write to this notebook, helping it remember things for a long time. Three Gates:

The robot has three gates: an input gate, a forget gate, and an output gate. Think of these gates like filters that control the flow of information. Input Gate:

When new information comes in, the input gate decides what to write into the notebook. It carefully selects which parts are important and need remembering. Forget Gate:

Sometimes, it's essential to forget old stuff to make room for new things. The forget gate decides what to erase from the notebook, making space for fresh information. Output Gate:

When you need to recall something, the output gate decides what information to share. It carefully selects what to tell you based on what's in the notebook. Now, let's put it all together:

Reading Information:

When the robot encounters new data, the input gate decides what to remember and adds it to the notebook. Forgetting Old Stuff:

The forget gate checks if any old information is no longer useful and erases it from the

notebook. Remembering for Later:

The memory cell holds onto the important information for a long time. Recalling Information:

When you need the robot to tell you something, the output gate selects the relevant information from the notebook. This way, the LSTM robot is excellent at managing information for a long time, deciding what to remember, what to forget, and what to share when you need it. It's like a smart memory assistant! In the world of computers and data, LSTMs are fantastic for tasks like predicting future values, understanding sequences, and making sense of time-dependent patterns.