
Predicting the future: Game momentum prediction model based on attention LSTM neural network

Summary

In the 2023 Wimbledon men's singles final, 20-year-old Spanish rising star Carlos Alcaraz used extraordinary momentum to defeat 36-year-old Novak Djokovic. It was Djokovic's first loss at Wimbledon since 2013 and ended a remarkable run by one of the greatest players in Grand Slam history. Athletes, coaches, spectators, and the media are accustomed to using the concept of "momentum" to commentate on sporting events, describe and explain actions and situations, and make game predictions based on past and/or ongoing performance of individuals or teams. However, there is little consensus among sports viewers about what momentum actually is and how it is measured. Researchers and coaches pointed out that it is necessary to study existing game data and prediction models to determine how to identify relevant factors to adjust strategies and game techniques for the next game.

To address the above challenges, we propose a game momentum prediction model based on attention LSTM neural network to describe and predict the game state and momentum of tennis matches. We used a method of evaluating a tennis player's momentum using a weighted multiplication of points and match factors. In designing an LSTM neural network with tennis match characteristics, we divide the neural network into LSTM layer, Attention layer, connection layer, dropout layer and MSE_loss layer. And select the appropriate activation function and optimization algorithm, carefully adjust the hyperparameters, especially the learning rate, and gradually train the model by adjusting the parameters. By visualizing the model results, a line chart showing the change of game momentum over time is obtained. In this line chart, we can clearly identify which player had better momentum at a certain point in the game and how big their performance was.

Through ablative analysis, the extent to which each factor contributes to the game situation is determined. By observing changes in model performance, we can infer the impact of this factor on the outcome, that is, the game situation. By comparing the average differences of different features, we can understand the importance of each feature to the model and the degree of contribution to the predicted results. We also use this model to conduct personalized analysis on a certain athlete, infer his competition habits by deriving the most affected characteristics of the athlete, and give recommendations.

The robustness of the model was tested by analyzing momentum from other matches. Finally, we came to the conclusion that teamwork and tactical arrangements in basketball games are similar to doubles cooperation and strategic arrangements in tennis games to a certain extent. This similarity may allow the model to more easily transfer and apply knowledge and skills learned in tennis when adapted to the game of basketball.

Keywords: Momentum, tennis, LSTM neural network, attention, ablative analysis, prediction, personalization

Contents

1 Introduction	4
2 Assumptions and Justifications	5
3 Notations	6
4 Data analysis	6
4.1 Data pre-processing	6
4.2 Data visualization	7
5 Match momentum prediction model based on attentional lstm neural network	9
5.1 Quantitative forms of assessing momentum	9
5.2 Establish an attention LSTM model with the goal of predicting changes in the game situation	10
5.2.2 Establish an LSTM neural network with tennis match characteristics	10
5.2.3 Introduction of attention mechanisms	11
5.2.4 MSE loss function	11
5.2.5 Add dropout layer to prevent overfitting	12
5.2.6 Full connectivity using two sense layers	12
5.3 Momentum model training results for far mobilization simulations using the model ..	12
5.4 Analyzing the Randomness of Achieving a Winning Streak in Relation to Momentum	13
5.5 Analysis of Key Factors in Predicting Turns in a Match Based on Ablativity Analysis	13
5.6 Research and recommendations on the playing habits of specific players - the example of the player Alejandro Davidovich Fokina	14
5.6.1 Training Personalized Neural Network Models with Ablative Analysis	15
5.6.2 Match tips for player Alejandro Davidovich Fokina and his opponents	16
6 Real-world testing and generalizability analysis - the case of boxing and basketball games	17
6.1 Data selection and analysis	17
6.2 Evaluation using models	18
7 Model Evaluation	20
7.1 Strengths	20
7.2 Weaknesses	21
7.3 Further Discussion	21
8 Conclusion	22

9 Memo to the Tennis Coaching Team	23
References	24
Appendices	24

1 Introduction

We used a method of evaluating a tennis player's momentum using a weighted multiplication of points and match factors. This is in tennis, momentum should not just be about the score of the game (for example, it may happen that although the player scores, his opponent gradually grasps the player's movement habits and tries to break from his weak points, which does not represent the player's momentum is better), but is fact-based that requires comprehensive consideration from multiple aspects.

After scientifically quantifying momentum, we built an attention-based LSTM neural network to predict game momentum. By visualizing the model results, a line chart showing the change of game momentum over time is obtained. In this line chart, we can clearly identify which player had better momentum at a certain point in the game and how big their performance was.

In the process of designing the attention-based LSTM neural network, we rationally designed the architecture of the neural network and divided it into LSTM layer, Attention layer, connection layer, dropout layer and MSE_loss layer. And the appropriate activation function and optimization algorithm are selected, which lays a solid foundation for the stable learning of the model. Second, we carefully adjusted the hyperparameters, especially the learning rate, to ensure that the model would neither oscillate due to too high a learning rate nor converge slowly due to too low a learning rate during training. With carefully chosen learning rates, we achieve smooth improvements in model performance. In addition, in order for the model to have good generalization ability, we gradually train the model by adjusting parameters. As the number of iterations continues to increase, our model not only gradually adapts to the distribution of training data, but can also effectively handle the validation set or test set. Unseen data, maintaining consistent performance improvements. Overfitting is one of the common problems in neural network training, which causes the model to perform well on the training data but degrade on the validation set or test set. In order to prevent our model from over-fitting problems, we add a dropout layer to randomly "turn off" or "discard" some neurons in the network and introduce noise to reduce the complex co-adaptation relationship between neurons to prevent over-fitting. fitting.

The remainder of this article is organized as follows:

In Chapter 5.6, the key factors affecting the game situation and the issue of giving personalized advice to different players are addressed. We adopted an ablative analysis approach. Predictions are made by replacing data points one by one and comparing the differences between predicted and actual values to determine the extent to which each factor contributes to the match situation. In order to express the model operation results more intuitively, we used cluster analysis to visualize the eight most influential factors and all factors separately. When it comes to giving personalized advice to different players, we take the player Alejandro Davidovich Fokina as an example, filter out his relevant data, create a data set, extract the unique name and its corresponding row number, and use a personalized neural network to provide each player with personalized advice. Individuals are trained, and each individual's data set is analyzed and visualized using ablation analysis methods. Based

on the influence of each factor on the player's game state, suggestions are given to the player Alejandro Davidovich Fokina and his opponents respectively.

In Chapter 6, we verified the actual combat testing and universality of this model. We selected data from the UFC Ultimate Fighting Championship and the 2020 NBA from 1993 to 2021 to make the above predictions. And analyze the prediction results to eliminate possible causes and improvement measures.

In Section 7, we evaluate the model, analyze its strengths and weaknesses, and propose future improvements. In the final section, we wrote a letter to the tennis coaching team based on the conclusions drawn from the article.

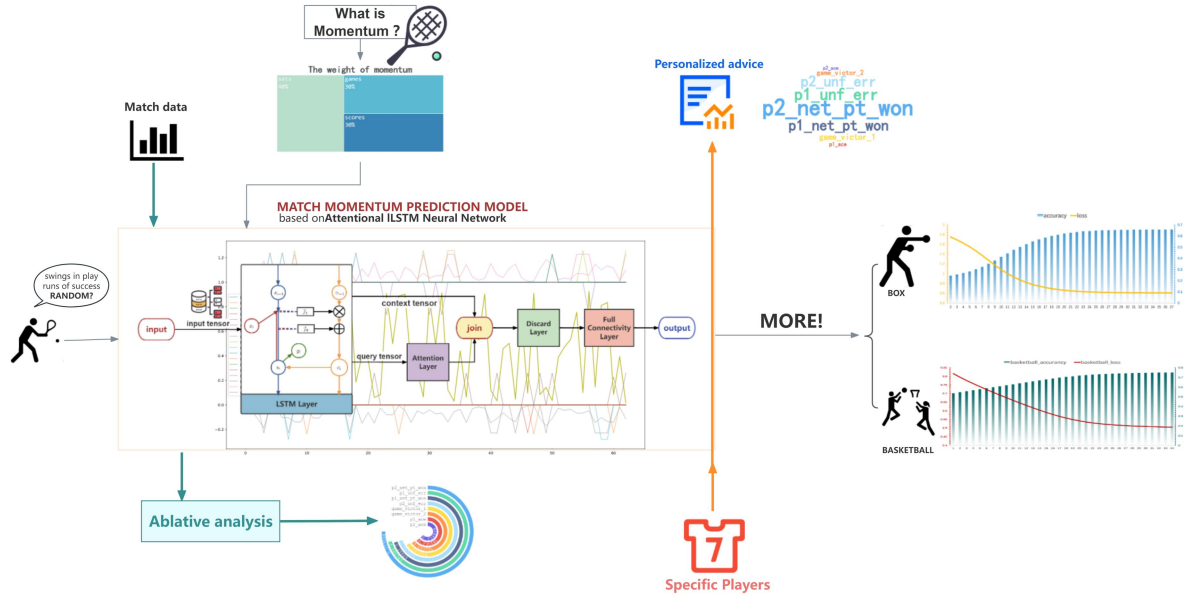


Figure 1: Our work

2 Assumptions and Justifications

- **Assumption 1:** We assume that complete and accurate tennis match data is available, including every score Detailed information (e.g. server, receiver, score type, etc.).

Justifications: Despite the incompleteness of the data and some tolerant error in statistics, we make this assumption to guarantee one valid solution.

- **Assumption 2:** :Within a period of time (such as one game or several games), the player's performance is relatively stable. Historical data can be used to predict future performance.

Justifications: This assumption allows us to use historical data to build predictive models. While player performance may fluctuate, it is assumed that such fluctuations are predictable in the short term.

- **Assumption 3:** During the game, external conditions (such as weather, court conditions) remain relatively stable, and It will have a significant impact on the outcome of the game.

Justifications: This assumption is to ensure that we can exclude external factors when analyzing match data. interference, focusing on the analysis of player performance and scoring process.

- **Assumption 4:** We assume that the prediction results of the model will not be affected by external factors (such as referee decisions, audience etc.) significant interference.

Justifications: In tennis, external factors may have a certain impact on the flow of the game and the performance of the players. Influence. However, to keep the model simple and predictable, we assume that these factors have. The impact on the results is negligible. This does not mean that these factors are not important, but that at this point in model construction. At this stage we chose to exclude it.

3 Notations

In this work, we use the nomenclature in Table 1 in the model construction. Other nonefrequent-used symbols will be introduced once they are used.

Table 2: Notations used in this paper

Symbol	Description	Unit
M_1	Momentum of Player1	String
M_2	Momentum of Player2	String
S_1	The number sets of Player1 had won	Number
S_2	The number sets of Player2 had won	Number
G_1	The number games of Player1 had won	Number
G_2	The number games of Player2 had won	Number
C_1	The number scores of Player1 had won	Number
C_2	The number scores of Player2 had won	Number

4 Data analysis

In this section, we describe the process of data preprocessing and data visualization in detail. Corresponding preprocessing measures were taken to address issues such as missing values, redundant values, unquantified natural language, and data leakage in the original data. Specifically, it includes cleaning up missing data, deleting redundant information, encoding and converting natural language, and normalizing according to data characteristics.

In addition, we visualized the preprocessed data to verify the preprocessing effect and visually display the data distribution and clustering patterns, providing strong support for subsequent model establishment.

Through these processes, the training efficiency and overall performance of the neural network model can be improved, thereby providing a better insight into the fluctuation trends and potential patterns of game data.

4.1 Data pre-processing

From studying the given data, it can be observed that the original data has missing

values, redundant values, natural language has not yet been quantified, and some data have data leakage and data scale inconsistencies, so the data needs to be preprocessed.

Regarding the problem of missing values, in order to maintain data alignment as much as possible and eliminate the impact of the difference between the processed data and the original data on subsequent analysis, we cleaned up the data of several games with a large amount of missing data.

Regarding the problem of redundant values, it is observed that each match corresponds to different participating players from both sides, so we group according to match_id, convert match_id into serial numbers, and delete redundant information player1 and player2;

For natural language, by observing the variable characteristics of game_victor, winner_shot_type, serve_width, serve_depth, return_depth and considering the data types required by the subsequent balabal model, one-hot encoding is used to convert the categorical variables into a format that the machine learning algorithm can understand and use. . In particular, we perform differential processing on elapsed_time to more intuitively display how much time each game (note the translation) consumes. According to the interpretation of natural language in the data_dictionary.csv file, it was found that p1_points_won, p2_points_won, server had data leakage problems, so the above data was deleted in the neural network input.

For the problem of inconsistent data scales, linear mapping and nonlinear mapping are used to normalize the data according to different data characteristics. For example: rally_count and p1_distance and p2_distance use non-linear mapping.

4.2 Data visualization

Due to the need to build a model, it is necessary to first conduct intuitive data insights. We set the ordinate to the value of the normalized variable, that is, from 0 to 1, to make the data more uniform, which is conducive to more uniform input to the neural network, which can improve training efficiency, avoid numerical problems during the training process, and improve the model overall performance. The meaning of the abscissa is that one row of the table is used as the smallest unit as the change sequence, which represents the fluctuation trend of each variable as the game progresses.

Data visualization has the following advantages:

- Verification preprocessing: You can help detect outliers through data visualization (for example, you can clearly observe in the figure that the 2023-wimbledon-1301 race is missing rally_count and speed_mph data) and further more intuitively verify whether the data preprocessing is effective. Achieve a more uniform effect for neural network understanding. For example, you can observe the nonlinear mapping of p2_distance_run after preprocessing, which makes the distribution more uniform according to the density of the data. Running distance rally_court, on 0~1)
- Data distribution visualization: Visually see the data distribution to help determine whether the data is concentrated in a certain area, whether there are outliers, etc. For example, a scatter plot can be used to show the distribution of the scores of both sides and the changes as the game progresses.

- Data cluster analysis: Scatter plots help identify the clustering patterns of data, and infer the characteristics and potential meanings of these clusters by observing the similarity of multiple clustered data clusters, thereby helping to



infer the associated attribute characteristics and potential meanings between variables. For example, the relationship between a player's running distance and the number of points can be inferred from a scatter plot.

Figure 2: speed_mph before data preprocessing

From Figure 2, you can see that the data distribution of speed_mph is very uneven. Most of the data is concentrated in the range of 80~140, while it is relatively sparse in other ranges. In addition, there is a gap in Figure 2 due to missing data or limitations of data collection, and these missing data greatly reduce the accuracy of the neural network.

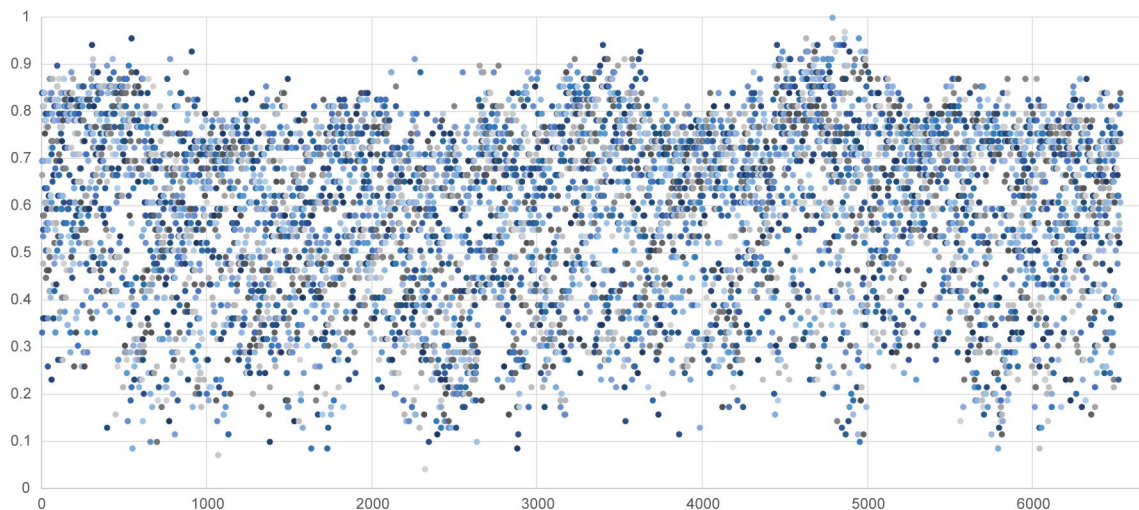


Figure 3: speed_mph after data preprocessing

After data preprocessing for speed_mph, the data points on the scatter plot are now evenly distributed in the range of 0-1, which not only eliminates the sparseness and aggregation of the data in the early stage, but also makes the overall distribution of the data smoother and more continuous. Preprocessing specifically includes: First, we thoroughly cleared the missing values in the original numbers to ensure the integrity and consistency of

the data set. The originally existing data blank areas also completely disappeared, which further proved that the data The effectiveness and necessity of pretreatment. Secondly, in order to eliminate the nonlinear characteristics and scale differences of the data, we used a nonlinear mapping method to effectively compress and map the value range of speed_mph to the interval of 0-1.

This uniform and continuous data distribution provides a solid foundation for our subsequent data analysis and model building. It can not only improve the accuracy and reliability of analysis results, but also help reveal potential correlations and patterns between data. Therefore, as a key step in the data analysis process, the importance of data preprocessing cannot be ignored.

5 Match momentum prediction model based on attentional lstm neural network

5.1 Quantitative forms of assessing momentum

In order to combine the assessment of game situation changes with the influence of other factors, we need to quantify the concept of game situation changes. The outstanding performance of a tennis player in each round is ultimately measured by the game score. The game score reflects the player's technical level, physical condition, psychological quality and other comprehensive competitive abilities, so we use whether the player scores in the next unit game. Determine how good or bad the player's momentum is during this period.

Taking into account the efficiency and stability of the neural network, we weighted the score and introduced time series into the model to show the impact of the number of winning streaks in the game on the player's momentum. The change of this value shows the player's momentum. In each game, we give different weights to the matches.

Table 3: Weight table

Belong to	Name	Proportion
The weight of sets	S_1	40%
	S_2	
The weight of games	G_1	30%
	G_2	
The weight of scores	C_1	30%
	C_2	

Based on the calculation for each player 1, we also defined a concept called the bias indicator, which is used to express the direction of the game situation. This indicator is related to the player's momentum.

$$Skew_{index} = (S_2 \times 0.4 + G_2 \times 30\% + C_2 \times 30\%) - (S_1 \times 40\% + G_1 \times 30\% + C_1 \times 30\%) \quad (1)$$

5.2 Establish an attention LSTM model with the goal of predicting changes in the game situation

In this part, we build an attention LSTM model with the goal of predicting the game situation. The goal is to map the input sequence (including factors that affect the game) to the output sequence (winner of this point) and use it when processing the sequence. An attention mechanism is used to extract the dynamic weight of each factor on the result change, and based on this, the ablation analysis of these factors and the value of point_vector is performed.

This model consists of a five-layer architecture, including the input layer, LSTM layer, attention layer, connection layer, dropout layer and fully connected layer.

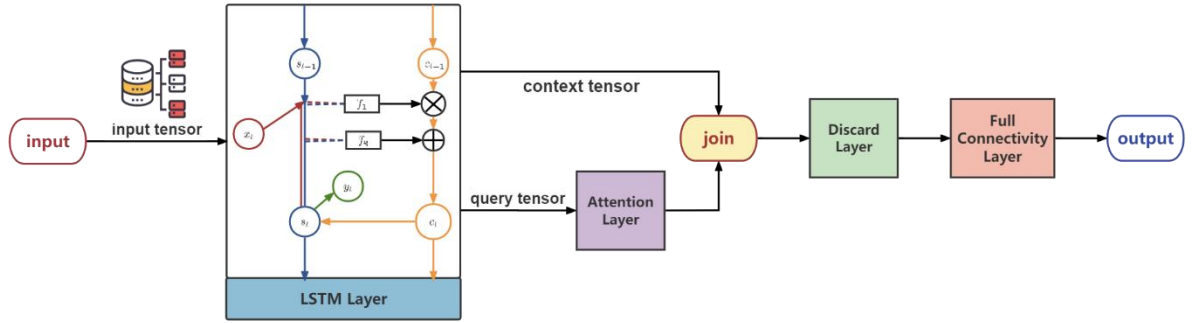


Figure 4: The architecture of LSTM neural network

In addition to the conventional loss and training functions, we introduce an attention mechanism to measure the correlation of each element in the input sequence with the query vector.

5.2.2 Establish an LSTM neural network with tennis match characteristics

Considering the hugeness of the given data and the long time span of the data, we use a special RNN neural network, the LSTM neural network, to solve the gradient disappearance and gradient explosion problems during long sequence training. Considering the actual situation of the tennis match, we take the player's score (point_vector) as the input value and other parameters besides the score as the input value. Treat each match as a unit block, use three match blocks as a test set, and the rest as a training set.

First, use LeCun Normal to initialize the weights of the LSTM layer:

$$W \sim N\left(0, \frac{1}{fan_{in}}\right) \quad (1)$$

The internal structure of LSTM neural network is as follows:

- **Oblivion Gate:** The role of this part is to filter important features and ignore irrelevant information. After the information has been input and trained, it needs to be decided that this existing information is retained for selection. We use the sigmoid function for quantitative representation, outputting a real number between 0 and 1 to the next cell state $C_{(t-1)}$, with 0 indicating complete discard and 1 indicating complete retention. where σ represents the sigmoid activation function, w_1 represents the weight matrix, and b_1 is the bias vector.

$$f_1 = \sigma \left(w_1 \left[\frac{s_{t-1}}{x_t} \right] + b_1 \right) \quad (2)$$

- **Input Gate:** The role of this part is to select the new information to be added to the mainline. There are two steps included here, firstly deciding which information needs to be updated by sigmoid function, and in generating the alternative to be used for updating by tanh function, the mainline is updated by joining these two parts and doing a multiplication operation. Where we use (w_2) and (b_2) to denote the updated content, i.e., the new input data

$$f_2 = \sigma \left(w_2 \left[\frac{s_{t-1}}{x_t} \right] + b_2 \right) * \tanh \left(\widehat{w_2} \left[\frac{s_{t-1}}{x_t} \right] + \widehat{b_2} \right) \quad (3)$$

- **OUTPUT GATE:** Based on the analysis above, we will use the scorer of the match, i.e. `point_victor`, as the output value. This output will be based on our mainline state, but is also a filtered version. First, we run the sigmoid function to determine which part of the cell state will be output. Next, we run the mainline state through the tanh function (to get a value between -1 and 1) and multiply it by the output of the sig-moid gate, and we end up just outputting the portion we determined to be output.

$$c_t = f_1 * c_{t-1} + f_2 \quad (4)$$

5.2.3 Introduction of attention mechanisms

In order to better deal with the relevance and importance of the data, and to analyze the most relevant factors to the current game situation, we introduce an attention mechanism, which allows the model to focus its attention on different input locations in order to better capture the correlation between the inputs. Here, different input locations are set as different variables, such as the depth of a player's serve. The attention mechanism allows the sequence model to adaptively adjust the weights based on the contextual information of the inputs. Attention is paid to different parts of the input sequence by calculating the attentional weights. It takes the output of the LSTM layer as a query vector and the input sequence as a key-value pair, and then computes the attention vector for weighted aggregation of information from the input sequence.

We get the output `listm_out` at the LSTM layer and select the hidden state of the last time step as the query vector `query` (which captures the aggregated information of the whole sequence and thus draws attention to different time steps in different sequences in the attention mechanism). Then, using the query vector `query` and `listm_out` as inputs, the attention mechanism module `self.attention` is called to compute the context vector `context_vector` and return the attention weights `attention_weights`.

At the join layer, we join the query vector with the attention vector in order to fuse the attention weights with the query vector.

5.2.4 MSE loss function

MSE squares the difference between the predicted and true values and averages them, so it is able to accurately measure the magnitude of the error in continuous numerical predictions, which is compatible with the fact that the role of this model is a regression task.

$$MSE = \frac{\sum_{i=1}^n (f(x) - y)^2}{n} \quad (1)$$

5.2.5 Add dropout layer to prevent overfitting

Next, the query vector query and the context vector context_vector are spliced to obtain concat_out. the spliced vectors are then regularized through the dropout layer and deactivated on-the-fly by the formula shown below, which randomly discards a portion of the neuron's outputs in order to achieve the effect of reducing overfitting.

$$\begin{cases} r_j^{(l)} \sim \text{Bernoulli}(p), \\ \tilde{\mathbf{y}}^{(l)} = \mathbf{r}^{(l)} * \mathbf{y}^{(l)}, \\ z_i^{(l+1)} = \mathbf{w}_i^{(l+1)} \tilde{\mathbf{y}}^l + b_i^{(l+1)}, \\ y_i^{(l+1)} = f(z_i^{(l+1)}). \end{cases} \quad (1)$$

5.2.6 Full connectivity using two sense layers

Finally, the final output output is obtained by linear transformation and activation function processing of concat_out by two Dense layers self.sense1 and self.sense2.

It is used to multiply the input tensor with the weight matrix and add the bias vector, and then nonlinearly transformed by the activation function.

5.3 Momentum model training results for far mobilization simulations using the model

We obtained data on the momentum of the athletes after training on momentum using the model, and below are the training results from a set of matches that we extracted, where we calculated the momentum as well as the bias metrics.

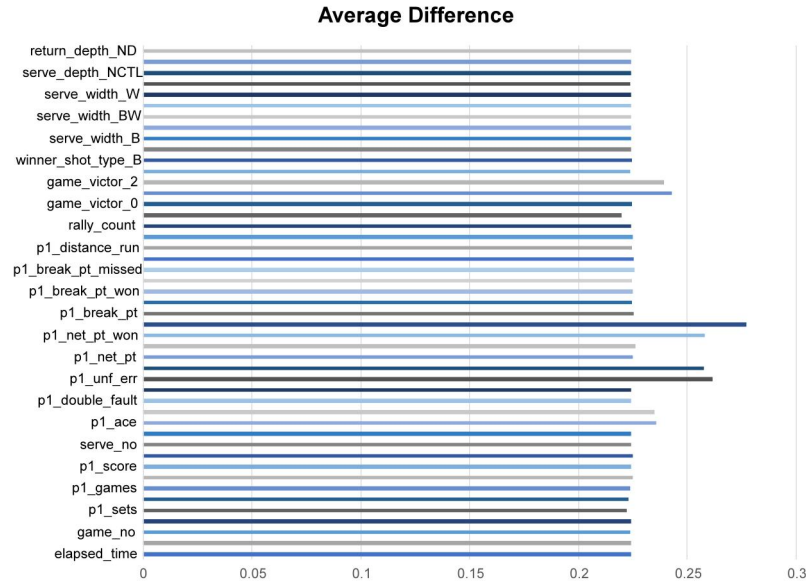
Table 4: Selected Training Results

p1_ace	p1_unf_err	p1_distance_run	y_pred
1	0.736227324	1.036662551	0.9147499
1	1	1.032098063	0.5767203
1	0.736227324	1.084323866	0.8972379
1	1	1.171509611	0.11855964
1.238394872	1	1.003965666	0.10896986
1	1	1.032330259	0.40069884
1.238394872	1	1.041654768	0.17204781
1	1	1.10695077	0.8434998
1	1	1.15119978	0.13188735

After obtaining the training results, we visualized the training results and obtained the following line graph, which contains the trained bias indicators used to visualize the situation of the game, and also includes the data of the athletes during the game on the field. The black line in the graph is the bias indicator, when the black line is close to 1, it means that player 2 has a better momentum and the game progresses in the direction of player 2's victory, when the black line is close to 0, it means that player 1 has a better momentum and the game

For each feature/factor, replace its value with the mean value to reach the point where the ablated dataset will contain only some of the features/factors. Using the ablated dataset, model predictions are made for each sample. By comparing the model predictions before and after the ablation, the degree of ablation effect or influence of each feature/factor can be calculated. Finally, we measure the difference between the calculated predictions by the

squared difference. The larger the mean difference, the greater the effect of the feature's ablation on the predicted results; the smaller the mean difference, the less the effect of the feature's ablation on the predicted results. By comparing the mean differences of different



features, we can understand how important each feature is to the model and how much it contributes to the prediction results.

Figure 6: Quantitative results of the impact of factors on the game situation

In order to more intuitively show the size of the impact of each factor on the turn of the situation, we visualize the final bar chart form, and through the cluster analysis method to find out the top 4 most influential factors.

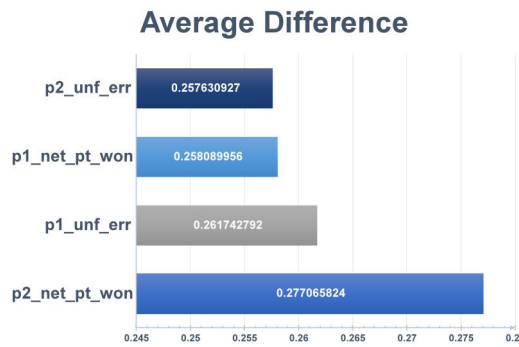


Figure 7: The top eight most influential factors in the game situation

5.6 Research and recommendations on the playing habits of specific players - the example of the player Alejandro Davidovich Fokina

In this part, we filter the data of a particular player in the given data and create a dataset to extract the unique name and its corresponding row number. The dataset of each individual is read from the saved dataset in the order of the player's name and a personalized neural network is trained for each individual and the dataset of each individual is analyzed using ablative analysis. Based on the derived impact of each factor on that player's game form and

use this to give recommendations. Here we take the example of player Alejandro Davidovich Fokina, and the same analysis can be applied to other players.

5.6.1 Training Personalized Neural Network Models with Ablative Analysis

After extracting all the match data of a particular player, we use the `solution_1` function to train a personalized neural network model.

Parameter name	expressed content
suffix	The name of the individual, which will be used as a suffix to the model name
model_name	The name of the model where the name of the individual is added as a suffix to the schema
first_epoch	Initial number of training rounds for the model

Table 5: Some important parameters in the `solution_1` function

After personalizing the neural network by training positively on a particular player, we use `solution3_1` to implement an ablative analysis for evaluating the impact of each factor athlete's playing status. Here we use `player1` to represent the player Alejandro Davidovich Fokina and `player2` to represent the opponent of the player Alejandro Davidovich Fokina. The effect of a factor on Alejandro Davidovich Fokina was assessed using an ablative analysis to derive the magnitude of the absolute value of the difference between the model's prediction and the true value of a factor after ablating it. The difference is quantified by the mean difference, where the greater the value of the mean difference, the greater the impact of the factor on the outcome, i.e. the greater the impact on the momentum of the player Alejandro Davidovich Fokina's game will be.

We visualized the results of the resulting data and represented them in the form of radar charts in order to observe more clearly and intuitively the magnitude of the impact of each factor on that player.

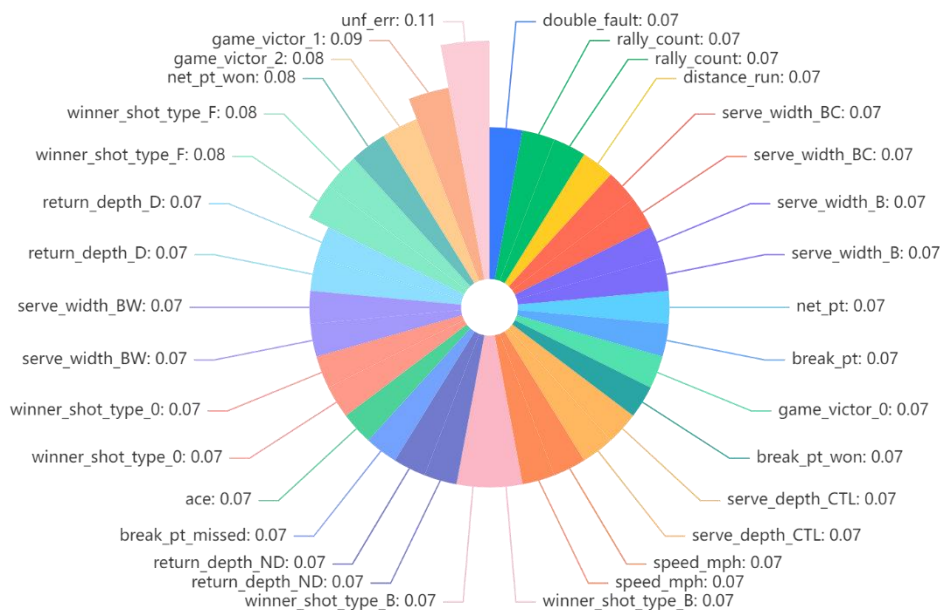


Figure 8: Effect size of each factor on the player Alejandro Davidovich Fokina

In Figure8 it can be clearly observed that there are some particularly salient points that have a particularly significant impact on this player, and we extract the top ten factors to show this.

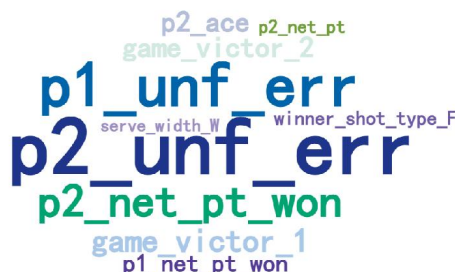


Figure 9: 10 Factors Most Influential to Player Alejandro Davidovich Fokina

Table 6: Notes on the meaning of the first 10 factors and specific data

Feature	Average Difference
player 2 made an unforced error	0.123026949
player 1 made an unforced error	0.109757273
player 2 won the point while at the net	0.096090198
play1 won a game this point	0.086699134
play2 won a game this point	0.08323251
player 2 hit an untouchable winning serve	0.082968642
player 1 won the point while at the net	0.080047497
untouchable shot is forehand	0.076456964
player 2 made it to the net	0.075394971
direction of serve is wide	0.075113215

5.6.2 Match tips for player Alejandro Davidovich Fokina and his opponents

Based on the data obtained, we can give objective and effective advice to the player Alejandro Davidovich Fokina and his opponents.

For Alejandro Davidovich Fokina:

1. Reduce unforced errors: According to the data, Alejandro Davidovich Fokina has a high level of unforced errors. He can reduce the number of unforced errors by playing more focused and consistent.
2. Using the net to win: According to the data, Alejandro Davidovich Fokina has a higher probability of winning when he goes to the net. He can try to go to the net more often and use this advantage to win more points.
3. Increase variety in serve direction: According to the data, Alejandro Davidovich Fokina's serve direction is more concentrated in the "wide" position. He could try to add more variety to his serves, including inward or center serves, to make it more difficult for his opponents.

For Alejandro Davidovich Fokina's opposing players:

1. capitalize on Alejandro Davidovich Fokina's unforced errors: based on the data,

opposing players can notice that Alejandro Davidovich Fokina has higher unforced errors. They can force him to make more unforced errors by putting more pressure on the opponent.

2. Avoid letting Alejandro Davidovich Fokina go to the net: According to the data, Alejandro Davidovich Fokina has a higher probability of winning when he goes to the net. Opposing players can try to avoid giving him easy access to the net and take steps to keep him near the baseline.

3. Countering Alejandro Davidovich Fokina's serve: According to the data, Alejandro Davidovich Fokina's serve direction is mostly focused on the "wide" position. Opponents can prepare for this position when preparing for the serve and move to the right position in advance to increase the success rate of the response.

6 Real-world testing and generalizability analysis - the case of boxing and basketball games

In this section, in order to verify the generalizability of the model, we choose two very different sporting events: boxing and basketball games. Boxing is known for its intense physical confrontation and rapid movement changes, while basketball is known for its teamwork and fast-paced offensive and defensive transitions. Despite the fact that these two competitions differ significantly from tennis in terms of rules, number of participants, and sports styles, testing the model's performance under both boxing and basketball matches reveals that the model shows an impressive ability to adapt to new data and extract meaningful information.

6.1 Data selection and analysis

For the selection of the dataset we selected the historical data of UFC Ultimate Fighting Championship from 1993-2021 and pre-processed the data according to the methods mentioned in Section IV. Compared with tennis matches, boxing matches are characterized by more direct physical confrontation, higher technical and strategic requirements, stronger physical fitness requirements, and faster match tempo and intensity. Therefore the collection and processing of raw data should be adapted to the characteristics of boxing.

In the selection of data, we chose the data of UFC Ultimate Fighting Championship 1993-2021 and NBA 2020 as the original data of the model.

Based on the fact that boxing has characteristics such as more intense physical confrontation, we added variables to the model to identify boxing-specific actions and tactics such as striking, defense, and pace movement. The variables with typical boxing characteristics and their explanations collected in are partially shown here. The "****" can be different types of attacks, such as: SUB_ATT: Submission attempts, REV: Counterattacks, SIG_STR: Effective strikes, TOTAL_STR: Total strikes, TD: Submissions, HEAD: Head attacks, BODY: Body attacks, DISTANCE. HEAD: Head Attack, BODY: Body Attack, DISTANCE: Long Range Attack, GROUND: Ground Attack.

Table 7: Variables that typically characterize boxing matches and their interpretation

variable name	interpretations
Height_cms	Height of players (centimeters)
Reach_cms	Player's arm span (centimeters)
Weight_lbs	Weight of players (lbs)
avg_*_att**	Average number of attacks of a certain type launched (e.g. SIG_STR_att indicates the number of valid strike attempts)
avg_opp_CTRL_time(seconds)	Average time opponents are controlled (in seconds)
win_by_*	Different types of victories (e.g., majority decision, unanimous decision, KO/TKO, etc.)

Similarly, in the application of basketball game, we added variables to the model that represent the fast pace and teamwork of a basketball game, such as the data of players' movement trajectories, shooting percentages, and passing networks. The variables and their characteristics that characterize the basketball game are partially shown here for cell phones.

Table 8: Typical Variables in Basketball and Their Interpretation

variable name	interpretations
Team	The player's team
3p_shot_rate	Three-point field goal percentage
Serve_rate	percentage of hits on serve
F_throws	Free throws
Defen	defense

6.2 Evaluation using models

After we applied the model to boxing matches and basketball matches, respectively, we collected a series of data about the performance of the model. The values of LOSS and ACCURACY are chosen here to provide direct evidence about how well the model performs in the new domain of boxing matches.

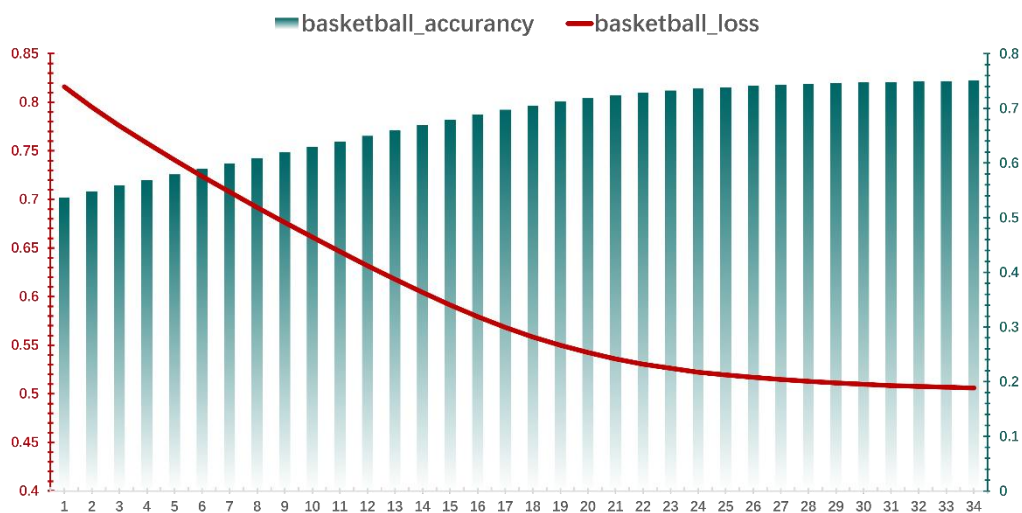


Figure 10: Accuracy of Momentum in Predicting Basketball Games

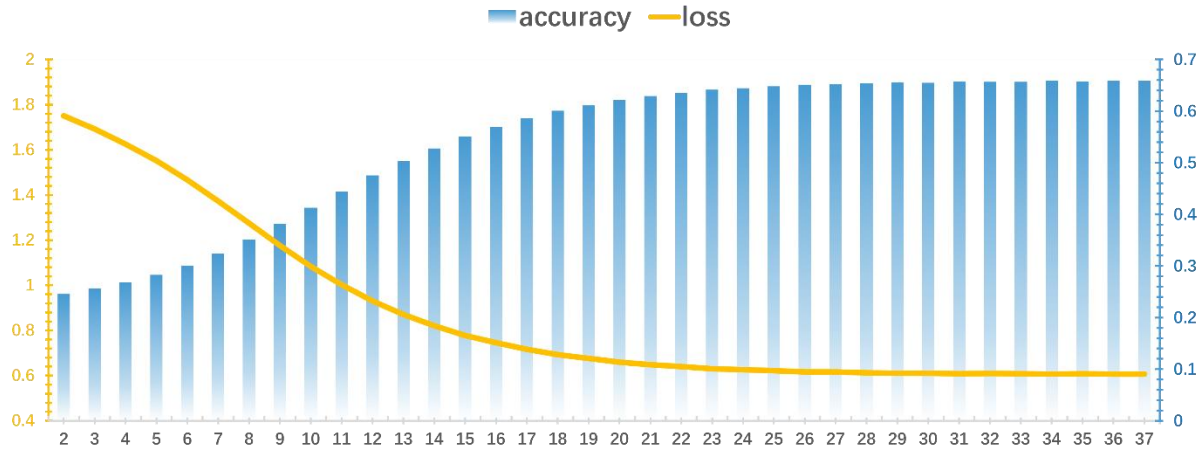


Figure 11: Accuracy of Momentum in Predicting Boxing

From the illustration, we can see that the learning curve of ACCURACY is quite smooth, i.e., the performance of the model is steadily improving during the training process without any major fluctuations or abrupt changes, which indicates that the training is stable and is not seriously affected by noisy data, outliers, or instability of the model architecture.

Interestingly, although the model showed some adaptation in both of these very different sporting events, higher accuracy was obtained in basketball. Upon analysis, we suggest the following possible reasons.

Starting from the facts, there are significant differences between tennis, boxing and basketball in terms of game flow and athlete performance. Tennis focuses on individual skills and reaction speed, boxing emphasizes physical confrontation and tactical strategies, while basketball combines individual skills, teamwork, and tactical placement. Therefore, the main challenge faced by models when applied across domains is how to capture and adapt to the key features and dynamics in these different tournaments.

When applying the model to boxing matches, we observed that the model had some difficulty in identifying strikes, defenses, and pace movements.

This may be due to the fact that the currently available data variables are not sufficiently detailed to describe the rapid movement changes and tactical diversity of athletes in boxing, making it difficult for the model to adequately learn and extract effective features. In addition, compared to the other two ball games, studies have shown that boxing is more susceptible to the influence of the environment and the athletes' own psychological state [6]. However, these factors are not considered in our proposed model, which may also contribute to the lower accuracy of boxing compared to tennis and basketball.

Secondly, teamwork and tactical arrangements in basketball are similar to some extent to doubles coordination and strategic arrangements in tennis. This similarity may allow the model to more easily transfer and apply the knowledge and skills learned in tennis when adapting to basketball.

These findings not only provide useful insights for our subsequent model improvement and application, but also reveal possible common laws and patterns among different sports events.

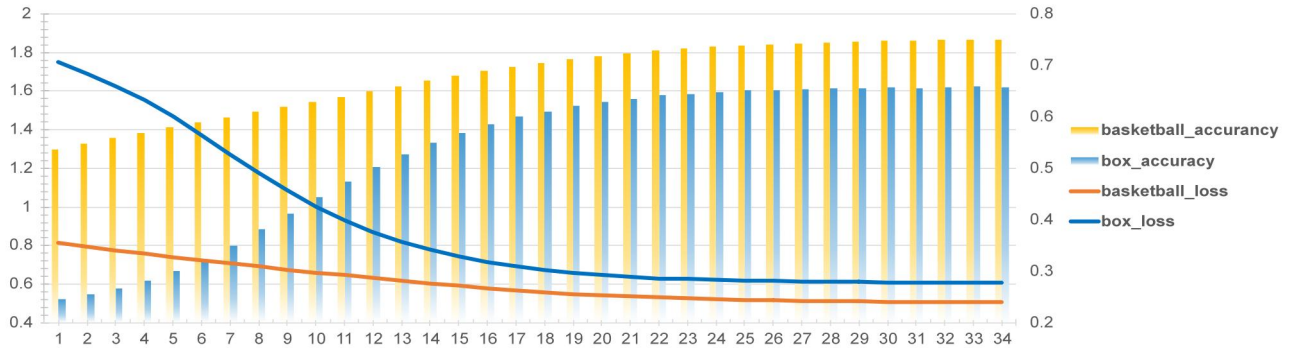


Figure 12: Accuracy of Momentum in Predicting Boxing and Basketball

7 Model Evaluation

In this section we analyze the strengths and weaknesses of the model and propose future improvements.

7.1 Strengths

In the study of race momentum prediction model based on attentional LSTM neural network, we have used various advanced techniques to enhance the performance and generalization ability of the model. The following are the advantages of our model design and implementation process:

- **Enhancement of generalizability:** our model demonstrates strong generalization ability. As long as there is an intrinsic correlation between the data, regardless of the source, the model is able to make effective predictions after appropriate data cleaning and preprocessing. This generalizability is not only reflected in the adaptability to different tennis players, but also in the flexibility to respond to different match scenarios and conditions.
- **Application of ablative analysis:** In order to improve the interpretability of the neural network, we employ the Ablation Analysis technique. By systematically removing certain components or features of the model and observing their impact on the performance, we are able to more accurately understand the contribution and importance of each factor in the model
- **Integration of Temporal Neural Networks:** Tennis is a highly dynamic and temporal activity. In order to better capture this temporal information, we integrated a temporal neural network. This network structure can effectively process time series data and extract long-term dependencies and periodic patterns in it, thus improving the accuracy of the model in predicting the course of the match.
- **The introduction of the attention mechanism:** In the neural network design, we have introduced an attention mechanism. This mechanism enables the model to automatically focus more attention on high-impact and information-rich features. By dynamically

adjusting the weight allocation of features, the attention mechanism improves the model's sensitivity to key information and processing efficiency.

- Training of player-specific models: Considering the significant differences in the styles, techniques and tactics of different tennis players, we train proprietary prediction models for each player. These models are able to analyze the nuances between players in greater depth and provide personalized prediction results and tactical recommendations. In this way, we not only improve the overall performance of the models, but also provide coaches and players with more targeted training and match strategy guidance.

7.2 Weaknesses

Although the race momentum prediction model based on attentional LSTM neural network shows significant advantages in terms of prediction accuracy and capturing complex temporal relationships, it still has some limitations that cannot be ignored.

First of all, this model requires a relatively large amount of data. In addition, the model needs to incorporate multiple features in order to extract enough information to support accurate prediction. This requirement may pose a challenge in certain scenarios where data is scarce or feature access is limited.

Second, the present model relies on human data preprocessing steps. Raw data that has not been processed or cleaned is often difficult for the model to converge, which may be due to issues such as outliers, missing data, or inconsistent data formats. Therefore, it is important to perform an exhaustive data preprocessing exercise to ensure the quality and consistency of the input data before applying the present model.

Finally, it is worth noting that the training time of the present model is significantly longer compared to traditional statistical methods. This is mainly due to the complexity of the LSTM network structure and the computational demands of the attention mechanism. Although the longer training time may limit the model's ability to be applied immediately, we believe that this time gap is expected to be reduced in the future through means such as optimized algorithms and hardware acceleration.

In summary, despite the significant advantages of the attention LSTM-based match score prediction model in terms of prediction performance, there are still some challenges in terms of data volume requirements, data preprocessing, and training time. Future research will aim to address these issues to further improve the model's utility and generalization ability.

7.3 Further Discussion

In this paper, we constructed an attentional LSTM-based game score prediction model, which effectively captures the key information during the game by introducing an attentional mechanism and realizes a more accurate score prediction. However, we also realize that the current model has some limitations in terms of data requirements and data processing, so in our future work, we will focus on the following two aspects for improvement.

First, to address the problem of the model's dependence on large data volumes, we plan to introduce a small sample learning approach. Small sample learning aims to achieve effective model training by utilizing limited sample data, which is important for scenarios

where data acquisition is difficult or data labeling is costly in practical applications. By combining small-sample learning techniques, we expect to be able to train score prediction models with stable performance on smaller datasets, thus broadening the scope of application and lowering the threshold of its practical use.

Second, in order to solve the problem that current models require human data preprocessing, a model that can automatically perform data preprocessing can be developed in the future. Data preprocessing is an indispensable part of the machine learning process, but manual data preprocessing is often time-consuming, labor-intensive and error-prone. Therefore, building a data preprocessing model that can automatically handle common problems such as missing values, outliers, data format conversion, etc. will greatly improve the efficiency and ease of model application. By automating the data preprocessing, we expect to further simplify the process of using the model, so that non-specialists can also easily apply the model for game score prediction.

In summary, future work will be centered around the introduction of small-sample learning methods and the development of automated data preprocessing models. We believe that these improvements will further enhance the practicality and ease of use of the game score prediction model based on attentional LSTM, and provide more possibilities for research and application in related fields.

8 Conclusion

Athletes, coaches, spectators and media are used to employing the notion of “momentum” to comment sport events, describe and explain behaviors and situations, and make game predictions on the basis of the past and/or ongoing performance of an individual or team. However, there is low agreement amongst sports spectators as to what momentum actually is and how it can be measured.

In order to be able to better simulate the momentum of the players for better performance on the court, we have done the following.

1. We build an attentional LSTM model with the goal of predicting the situation of the match, the goal is to map the input sequence (including the factors affecting the match) to the output sequence (the winner of the point of the game), and use an attentional mechanism in processing the sequence to extract the dynamic weights of the factors on the change of the result, based on which we do the ablativity analysis of these factors with the value of `point_vector`. This model consists of a five-layer architecture, including the structure of Input layer, LSTM layer, Attention layer, Connection layer, Dropout layer and Fully connected layer. Finally we use this model to implement the simulation of momentum and can visualize the course of the game.

2. In order to explore the impact of a particular factor, such as the position of a player's serve, on the match situation, we use ablativity analysis to observe the changes in the model performance by eliminating or modifying the features/factors one by one, so as to speculate on the magnitude of the impact of this factor on the outcome, i.e., the match situation, and by comparing the average difference of the different features we can understand the importance of each feature for the model and how much they contribute to the prediction of the the

degree of contribution to the result.

3. Practical test and generalizability analysis, in this part, we test the robustness of the model by analyzing the momentum of other games. Finally, we get the following conclusion that teamwork and tactical arrangements in basketball are similar to some extent to doubles coordination and strategic arrangements in tennis. This similarity may make it easier for the model to transfer and apply the knowledge and skills learned in tennis when adapting to basketball matches.

9 Memo to the Tennis Coaching Team

February 2, 2024

To: The Tennis Coaching Team

From: MCM Team #2405192

Subject: Insights on the Role of “Momentum” and strategies for Player Preparation in Resource to Events Impacting Match Flow

Young Carlos Alcaraz showed his strong willpower to defeat the experienced Novak Djokovic in the final of Wimbledon Open 2023. Our team analyzed the given data and conceived a match momentum prediction model based on attentional LSTM neural networks that accurately describes and predicts changes in momentum during tennis matches. By testing boxing and basketball matches, we verified that the model is generalizable.

Possible understanding of the role of “Momentum”

We constructed an attentional LSTM neural network-based game momentum prediction model for capturing the flow of the game at the time of scoring and applied it to real games. Momentum can reflect a player's performance on the court and can be used to predict the outcome of a game, helping players to improve their performance on the court and helping them to score better.

Proposal to players

To improve tennis players' comprehension and practical application of momentum, we recommend implementing the following coping strategies, which will assist players in adjusting their strategies and reacting to the flow of the game.

- Winning by utilizing the net stroke: according to the data, the probability of winning is higher when going to the net. He can try to go to the net more often and use this advantage to win more points.
- Increase variety in the direction of the serve: for a player who serves in a single direction he can try to add more variety to his serve, including inward or center serves, in order to make it more difficult for his opponent.
- Capitalizing on the opponent's unforced errors: according to the data, it is possible to force the opponent to make more unforced errors by putting more pressure on him.

References

- [1] Walid Briki, Rethinking the relationship between momentum and sport performance: Toward an integrative perspective, *Psychology of Sport and Exercise*, Volume 30, 2017.
- [2] Peter O'Donoghue & Emily Brown (2009) Sequences of service points and the misperception of momentum in elite tennis, *International Journal of Performance Analysis in Sport*, 9:1, 113-127, DOI: 10.1080/24748668.2009.11868468.
- [3] Dietl, Helmut; Nessler, Cornel (2017). Momentum in tennis: controlling the match. UZH Business Working Paper Series 365, University of Zurich.
- [4] S.E. Iso-Ahola, C.O. Dotson Psychological Momentum-A key to continued success: *Frontiers in Psychology*, 7 (2016), p. 1328.
- [5] Braidwood, J. (2023), Novak Djokovic has created a unique rival – is Wimbledon defeat the beginning of the end, *The Independent*,
- [6] <https://www.independent.co.uk/sport/tennis/novak-djokovic-wimbledon-final-carlos-alcarazb2376600.html>.
- [7] Slimani, Maamer, et al. "Effect of acute boxing training session on cognitive function and psychological states." *Medicina Dello Sport* 75.3 (2022): 404-412.

Appendices

Appendix 1
Introduce: Code implementation of an attentional LSTM neural network
<pre>def custom_accuracy(y_true, y_pred): # Calculate the absolute difference between the predicted value and the true value diff = K.abs(y_true - y_pred) # Calculate accuracy: 1 minus the average of the difference accuracy = 1 - K.mean(diff, axis=-1) return accuracy def train_step(model, X_batch, y_batch, loss_function, optimizer): with tf.GradientTape() as tape: X_batch = tf.convert_to_tensor(X_batch, tf.float32) predictions = model(X_batch, training=True) # Reshape predictions to match the shape of y_batch predictions = tf.squeeze(predictions, axis=[1, 2]) loss = loss_function(y_batch, predictions) gradients = tape.gradient(loss, model.trainable_variables) optimizer.apply_gradients(zip(gradients, model.trainable_variables)) # Calculate accuracy using custom_accuracy function accuracy = custom_accuracy(tf.cast(y_batch, tf.float32), predictions) return loss, tf.reduce_mean(accuracy) def train_model(model, X_train, y_train, X_test, y_test, epochs=10, batch_size=32, history=None, lea</pre>

Appendix 2
Introduce: Code implementation of ablative analysis
<pre># Create a copy to avoid modifying the original DataFrame modified_df = df.copy() # Replace the current ablation column's values with the mean of that column, simulating ablation</pre>


```
modified_df[col] = df[col].mean()

# Prepare the data
X, y = [], modified_df['point_victr'].values
for i in range(len(modified_df)):
    # Build an 8-row data window consisting of the current row and the previous 7 rows for the current
    sample
    if i < 7:
        # Use mean filling when there are less than 8 rows
        padding_rows = 7 - i
        padded_array = np.array([list(column_means.values()) for _ in range(padding_rows)])
        window_data = np.vstack((padded_array, modified_df.iloc[:i + 1][columns].values))
    else:
        window_data = modified_df.iloc[i - 7:i + 1][columns].values
    X.append(window_data)

X = np.array(X).astype(np.float32) # Convert to a NumPy array and set the type

# Make predictions using the model
predictions = model.predict(X, verbose=0) # Use the predict method instead of calling the model directly

# Calculate the difference between the predicted value and the true value, and record it
difference = np.abs(predictions.flatten() - y) # Ensure consistent dimensions
differences_no_nan = difference[~np.isnan(difference)]
ablation_effects[col] = differences_no_nan.mean()
```

OpenAI ChatGPT (Dec 2023 version, ChatGPT-4)

Query1: Data is provided for every point from all Wimbledon 2023 men's matches after the first 2

rounds. You may choose to include additional player information or other data at your discretion,

but you must completely document the sources. Use the data to:

- Develop a model that captures the flow of play as points occur and apply it to one or more of the matches. Your model should identify which player is performing better at a given time in the match, as well as how much better they are performing. Provide a visualization based on your model to depict the match flow. Note: in tennis, the player serving has a much higher probability of winning the point/game. You may wish to factor this into your model in some way.
- A tennis coach is skeptical that "momentum" plays any role in the match. Instead, he postulates that swings in play and runs of success by one player are random. Use your model/metric to assess this claim.
- Coaches would love to know if there are indicators that can help determine when the flow of play is about to change from favoring one player to the other.
 - o Using the data provided for at least one match, develop a model that predicts these swings in the match. What factors seem most related (if any)?
 - o Given the differential in past match "momentum" swings how do you advise a player going into a new match against a different player?
- Test the model you developed on one or more of the other matches. How well do you predict the swings in the match? If the model performs poorly at times, can you identify any factors that might need to be included in future models? How generalizable is your model to other matches (such as Women's matches), tournaments, court surfaces, and other sports such as table tennis.

Query2: variables,explanation,example

match_id,match identification,"2023-wimbledon-1701 ("7" is the round, and "01" the match number in that round)"

player1,first and last name of the first player,Carlos Alcaraz

player2,first and last name of the second player,Novak Djokovic

elapsed_time,time elapsed since start of first point to start of current point (H:MM:SS),0:10:27

set_no,set number in match,"1, 2, 3, 4, or 5"

game_no,game number in set,"1, 2, ...,7"

point_no,point number in game,"1, 2, 3... etc."

p1_sets,sets won by player 1,"0, 1, or 2"

p2_sets,sets won by player 2,"0, 1, or 2"

p1_games,games won by player 1 in current set,"0, 1,...,6"

p2_games,games won by player 2 in current set,"0, 1,...,6"

p1_score,player 1's score within current game,"0 (love), 15, 30, 40, AD (advantage)"

p2_score,player 2's score within current game,"0 (love), 15, 30, 40, AD (advantage)"

server,server of the point,"1: player 1, 2: player 2"

serve_no,first or second serve,"1: first serve, 2: second serve"

point_victor,winner of the point,"1 if player 1 wins, 2 if player 2 wins"

p1_points_won,number of points won by player 1 in match,"0, 1, 2... etc."

p2_points_won,number of points won by player 2 in match,"0, 1, 2... etc."

game_victor,a player won a game this point,"0: no one, 1: player 1, 2: player 2"

set_victor,a player won a set this point,"0: no one, 1: player 1, 2: player 2"

p1_ace,player 1 hit an untouchable winning serve,0 or 1

p2_ace,player 2 hit an untouchable winning serve,0 or 1

p1_winner,player 1 hit an untouchable winning shot,0 or 1

p2_winner,player 2 hit an untouchable winning shot,0 or 1

winner_shot_type,category of untouchable shot,"F: Forehand, B: Backhand"

p1_double_fault,player 1 missed both serves and lost the point,0 or 1

p2_double_fault,player 2 missed both serves and lost the point,0 or 1

p1_unf_err,player 1 made an unforced error,0 or 1

p2_unf_err,player 2 made an unforced error,0 or 1

p1_net_pt,player 1 made it to the net,0 or 1

p2_net_pt,player 2 made it to the net,0 or 1

p1_net_pt_won,player 1 won the point while at the net,0 or 1

p2_net_pt_won,player 2 won the point while at the net,0 or 1

p1_break_pt,player 1 has an opportunity to win a game player 2 is serving,0 or 1

p2_break_pt,player 2 has an opportunity to win a game player 1 is serving,0 or 1

p1_break_pt_won,player 1 won the game player 2 is serving,0 or 1

p2_break_pt_won,player 2 won the game player 1 is serving,0 or 1

p1_break_pt_missed,player 1 missed an opportunity to win a game player 2 is serving,0 or 1

p2_break_pt_missed,player 2 missed an opportunity to win a game player 1 is serving,0 or 1

p1_distance_run,player 1's distance ran during point (meters),"5.376, 21.384, etc."

p2_distance_run,player 2's distance ran during point (meters),"6.485, 12.473, etc."

rally_count,number of shots during the point,"1, 2, 4, etc. (includes serve)"

speed_mph,speed of serve (miles per hour; mph),"81, 124, etc."

serve_width,direction of serve,"B: Body, BC: Body/Center, BW: Body/Wide, C: Center, W: Wide"

serve_depth,depth of serve,"CTL: Close To Line, NCTL: Not Close To Line"

return_depth,depth of return,"D: Deep, ND: Not Deep"

Query3:

match_id,elapsed_time,set_no,game_no,point_no,p1_sets,p2_sets,p1_games,p2_games,p1_score,p2_score,server,serve_no,point_victor,p1_points_won,p2_points_won,set_victor,p1_ace,p2_ace,p1_winner,p2_winner,p1_double_fault,p2_double_fault,p1_unf_err,p2_unf_err

err,p1_net_pt,p2_net_pt,p1_net_pt_won,p2_net_pt_won,p1_break_pt,p2_break_pt,p1_break_pt_won,p2_break_pt_won,p1_break_pt_missed,p2_break_pt_missed,p1_distance_run,p2_distance_run,rally_count,speed_mph,game_victor_0,game_victor_1,game_victor_2,winner_shot_type_0,game_victor_0,game_victor_1,game_victor_2,winner_shot_type_0,winner_shot_type_B,winner_shot_type_F,serve_width_B,serve_width_BC,serve_width_BW,serve_width_C,serve_width_W,serve_depth_CTL,serve_depth_NCTL,return_depth_D,return_depth_ND

Which listings are reflected here? Players. Information about the match. For example, the distance traveled, the speed of serve, the number of errors, and so on. Note that this is not directly reflected. Current point winners. I need, I need you to list those. The ones that do not reflect the current point winner. Player match information. V

Output.

Columns of data that do not directly reflect the winner of the current point but do reflect player race information include:

1. `elapsed_time` - the elapsed time from the first point to the current point, reflecting the duration of the match and possible physical exertion.
2. `set_no` - the current set number, showing at what stage the match has progressed.
3. `game_no` - the number of games in the current set, providing information on the progress of the match.
4. `point_no` - the score number of the current set, showing how far the set has progressed.
5. `p1_sets` and `p2_sets` - the number of sets won by both players, showing the general trend of the match.
6. `p1_games` and `p2_games` - the number of games won by both players in the current set, reflecting how the match is going in the current set.
7. `p1_score` and `p2_score` - the scores of both players in the current set, providing an indication of the tension in the set.
8. `server` - the player currently serving, which can reflect the advantage of the serving side.
9. `serve_no` - Number of serves (first or second), affects the risk and strategy of the serve.
10. `p1_ace` and `p2_ace` - Ace serves (direct scoring serves) by both players, showing the efficiency of the serve.
11. `p1_winner` and `p2_winner` - the number of winners for both players, showing offensive ability.
12. `p1_double_fault` and `p2_double_fault` - The number of double serve errors for both players, showing consistency of serve.
13. `p1_unf_err` and `p2_unf_err` - The number of unforced errors by both teams, showing the stability of the serve.