Problem Chosen	2024 MCM/ICM Summary Sheet	Team Control Number
C		2419815

Momuntum in Tennis

Summary

Momuntim is defined in dictionaries as "strength or force gained by motion or by a series of events," but this force is challenging to measure in practical phenomena. Tennis, as a typical one-on-one individual sport, is not influenced by other team members during the competitive process. Instead, opponents directly impact each other, providing an excellent environment for modeling momentum.

In Task 1, starting from the intuitive idea that a player's previous state influences subsequent states in a match, our team first established a machine learning model based on LSTM for time-series performance prediction. The training data are derived from the processed match samples through feature engineering. Considering the significant impact of serving on the win rate, we used the statistically calculated serving win rate as a penalty term to adjust the labels, achieving a point prediction model for match data. This model predicts the player's performance score at a specific moment, and the predicted performance score difference is used as the output. The model successfully captures the flow of play as points occur, demonstrating high predictive confidence with low RMSE (0.0638) and R^2 (0.1159) values. Furthermore, we perform real-time performance score EWMA processing to obtain a quantified trend of momentum. Swings of momentum are visualized for presentation. Additionally, we visualize some statistical indicators of players (e.g., serving error rate) based on real data to achieve match visualization.

In Task 2, we first decoupled the task requirements, primarily judging the role of momentum through two indicators: swings in play and runs of success. Swings in play can be quantified and defined based on the output data of the LSTM point prediction model. After binarization, the randomness is assessed through a run test. Finally, we calculate the model's Z-score (-6.9791) and P-value (2.9714e-12), significantly rejecting the null hypothesis, thus proving that these two indicators are not random and indicating the role of momentum.

In Task 3, our objective is to predict fluctuations in the match to assess changes in momentum. We analyze the features most correlated with these changes and provide effective recommendations for players based on the analysis. Initially, we categorize momentum changes using thresholding and tri-classification. We define samples where momentum category transitions occur as target samples, turning the task into a classification problem. Given the data imbalance, an improved random forest model is employed for fitting. The top few features in terms of importance are Error (19.37%), Good_shot (18.39%), Distance_run (11.44%), Game_sit (9.46%), and others. Finally, we provide recommendations to players based on the prediction results and feature importance judgments.

Our established model has practical implications for both players and coaches in tennis matches. By predicting the real-time performance of players, we can identify the weaknesses of players during the match. Through the quantification of momentum, predictions of future momentum trends at specific moments in the game can be made. Coaches can use these predictions to provide targeted guidance to players. By predicting turning points, players can be alerted to negative changes in momentum in advance and make preparations accordingly.

Keywords: Symmetric feature representation; LSTM ; EWMA; Random forest

Contents

1 Introduction

1.1 Background

In last year's Wimbledon Gentlemen's singles final, 20-year-old Spanish prodigy Carlos Alcaraz defeated 36-year-old Novak Djokovic, breaking Djokovic's extraordinary winning streak in Grand Slam events since 2013. The process of this match was full of twists and turns, heart-stopping moments, as players frequently shifted control trends during the intense contest. The remarkable swings in scores and even the course of the match can largely **be attributed to momentum.**

Unlike the precisely defined momentum in physics, the momentum referred to here is defined as **"strength or force gained by motion or by a series of events."** Although this phenomenon is intuitively understandable, such as the commonly mentioned "hot hand effect" in basketball, it is challenging to practically measure and quantify. Furthermore, the impact and role of various events occurring during a match on momentum remain unknown.

In comparison to team sports,**tennis, as a one-on-one competitive sport, provides an environment where the influence of momentum is more easily observable.** In tennis, opponents directly impact each other, and there are no intermediaries (such as other team members) to bias the results. This one-on-one situation is significant in sports momentum research, both for model simplification and interpretability.

1.2 Restatement of the Problem

***Problem* 1:** Develop and apply a model that **capable of capturing the flow of play as points occur**. The model should be able to identify, at specific moments in a match, **which player is performing better and to what extent they have an advantage**. Additionally, provide a **visualization based on the model to illustrate the flow of the match**. It is essential to consider the higher probability of winning points or games for the **serving player** in the model.

***Problem* 2:** Use the model to evaluate a coach's viewpoint. He questions **the role of "momentum"** in the game and believes that **swings in play and runs of success by one player are random.**

***Problem* 3:** Develop a model based on match data to **predict turning points in a game**. Identify which events lead to a shift in control trends from one player to another, **analyzing the factors with the highest correlation**. Provide constructive advice for players entering new matches based on past fluctuations in match momentum.

***Problem* 4:** Assess and test the predictive capabilities of the model using other match data. Analyze instances where the model may perform poorly and identify factors for future improvement. **Evaluate the generalizability of the model** to other matches, tournaments, different court surfaces, and sports such as table tennis.

***Problem* 5:** **Compose a memorandum summarizing the conclusions of the model.** Offer constructive advice to the coach regarding the role of match momentum and provide recommendations for player training to respond effectively to events influencing the flow of a tennis match.

2 Preparation for Modeling

2.1 Assumptions and Explanations

Assumption 1: **Momentum can be quantified**

Explanation : Believing that momentum can be quantified through multiple indicators

Assumption 2: **Participants all try their best to complete the competition**

Explanation : It is believed that players all want to win the match, and there is no intentional situation of allowing opponents to win.

Assumption 3: **Data values that cannot be determined are all valid**
Explanation : For data that cannot be accurately determined, we all consider it to be valid and correct

2.2 Notations and Glossary

Several important mathematical notations used in this paper are listed in Table 1.

Table 1: Notations used in this paper

Notation	Description	Unit
M_t	The momentum value of the t-th sample in the time series	-
τ	The threshold for using momentum ternary	-
α	the smoothing parameter	-
β	Quantitative expression of serving advantage	-

*There are some variables that are not listed here and will be discussed in detail in each section.

3 Feature Engineering

The feature engineering of the given tennis match data is a crucial step in the modeling process, because it forms the basis for both the quantitative analysis of momentum and the utilization of models such as evaluation and regression prediction, all reliant on the modeling of relevant features. In the domain of feature engineering, we initiated the process with data cleansing, addressing the detection and handling of missing values and outliers. Subsequently, the method of utilizing feature differences was employed to maintain the symmetry of the model. In the aspect of feature scaling, we applied a normality test and normalization operation. For the preprocessed data features, feature selection is conducted during the modeling process based on the specific problem at hand, aiming to maximize the alignment between the model and features.

3.1 Data Clean

Cleaning the data is not only the first but also a crucial step in ensuring the reliability of the model. For the given detailed data of a tennis match, we determine which data points are missing or anomalous.

3.1.1 Missing Value Handling

We found missing data in speed_mph (10.3%), serve_width (0.7%), serve_depth (0.7%), and return_depth (17.9%). Typically, missing data is handled through methods like deletion or interpolation. However, considering that the latter three are not quantitative data and, in order to preserve the temporal features of the sequences, handling the missing parts is challenging. Therefore, we have decided not to use the four features in subsequent model training and will not perform any processing on them.

3.1.2 Outlier Handling

In the detection and handling of outliers, the features p1_score and p2_score represent the scores of different players within the current game. According to the described rules of the match, these two features should consist of data categorized as 0, 15, 30, 40, and AD. However, at various moments in different matches, scores such as 1, 2, 3, 4, etc., were observed. Based on our observation and analysis, it is inferred that these anomalous data points represent the number of games won by the player in that particular game, rather than the conventional scoring system (0, 15, 30, 40, AD).

According to information from Baidu Baike, the reason tennis scoring is not an arithmetic progression is solely based on pronunciation issues. Therefore, mapping the score to the number of winning balls is reasonable. Thus, we map p1_score and p2_score as shown in Table 2.

Table 2: Mapping Scores to Winning Points

Score	**Points Won in a Game**
15	1
30	2
40	3
AD	4

This mapping seems to pose an issue when measuring the winning ball count for individual players. However, as seen in the subsequent data processing, when we use the difference of score to represent the p1_score and p2_score columns, such mapping does not pose any problem.

3.2 Difference Representation: Symmetric

To ensure the effectiveness of the model, it is essential to consider the feature information of both players involved in the match. Therefore, the features of interest often have two values, corresponding to each player. We refer to such features as "Symmetric Features." For example, for the within-game scoring feature (px_score), we can create a new feature by taking the difference between the features, a method discussed in Michal Sipko's work[1]. Although intuitively modeling tennis matches by simultaneously using features of both players can retain more information and make the model more accurate, in practice, the difference of player features usually provides sufficient information while reducing the number of features.

Another significant advantage of using the difference in feature variables as features is **the guarantee it provides for the symmetry of the model.** We define a symmetric model here as a type of model that produces identical match outcome predictions even when player labels are swapped (i.e., Player 1 and Player 2 are exchanged). Due to noise in the data, an asymmetric model might assign more importance to a feature for Player 1 than for Player 2, resulting in different predictions based on player labeling. For example, a simple logistic regression model might assign a higher weight to "p1_score" relative to "p2_score". Using a single "score_difference" feature to represent the difference helps avoid bias.

3.3 Feature Extraction

For feature extraction, we discard irrelevant features (e.g., match_id) and strive to retain features closely related to the match trends. This approach allows the machine learning model to autonomously learn the importance and correlations of features comprehensively.

Simultaneously, we include the ATP rankings for each player as of July 3, 2023.[1]

To accommodate subsequent model training, in conjunction with the Symmetric Feature processing method mentioned in Section 4.2, we perform the following processing and extraction on the given dataset "Wimbledon_featured_matches":

Quantitative Features:

1. Symmetric Feature: The difference is computed according to the equation ??.

2. Not a Symmetric Feature: No processing.

Categorical Features:

1. Symmetric and Binary Feature: Construct quantitative data through the new feature (new_feature = p1_feature - p2_feature) , where the values are transformed into ordinal data levels.

2. Not a Symmetric Feature: Utilize one-hot encoding.

The summary of the extracted data features is presented in Table 3:

[1]Ranking data is sourced from ATP rankings, available at `https://www.atptour.com/en/rankings/singles?DateWeek=2023-07-03&Region=all&RankRange=0-100&SortField=FullName&SortAscending=True`.

Table 3: Feature Summary

Feature	**Explanation**
Rank	Differences in ATP ranking on July 3rd
Sets	Differences in winning sets
Games	Differences in winning games
Points	Differences in score in this game
Error	Differences in double_fault and unforced error of this point
Distance_run	Differences in distance_run
Rally_count	Number of shots during the point
Game_sit	The situation of this game (one-hot)
Set_sit	The situation of this set (one-hot)
Good_serve	Differences in untouchable winning serve of this point
Good_shot	Differences in untouchable winning shot of this point
Net_pt	Differences in net_pt
Net_pt_won	Differences in net_pt_won
Break_pt	Differences in Break Point
Break_pt_won	Differences in successful Breakpoint
Break_pt_missed	Differences in Missed Breakpoints

3.4 Feature Scaling

We initially conducted a **normality test on the distribution of features using the Jarque-Bera test, as outlined in Equation 1.** This test relies on sample skewness and kurtosis, and calculates the JB statistic based on their values. During hypothesis testing, we set a significance level of 0.1 and observed that the majority of JB statistics significantly exceeded the critical values determined by the significance level and degrees of freedom. Consequently, we find that **the characteristics of the data do not adhere to a normal distribution for the most part.**

Therefore, we opt for **normalization rather than standardization when processing the features of the data.**

$$JB = \frac{n}{6}\left(\frac{(\text{Skewness})^2}{2} + \frac{(\text{Kurtosis} - 3)^2}{4}\right) \tag{1}$$

Where the sample skewness and kurtosis are calculated as:

$$\text{Skewness} = \frac{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2\right)^{3/2}}$$

$$\text{Kurtosis} = \frac{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2\right)^{2}}$$

Where X_i represents the observed values in the sample, and $\bar{X}$ is the sample mean.

4 LSTM Performance Prediction and Momentum Quantification

4.1 Intuition: How to Quantify Momentum

Through the analysis and decoupling of Problem 1, we can summarize two explicit requirements.

(1) Construct a model capable of estimating and quantifying the current performance and momentum of players.

(2) Visualize the process of the match.

We attempt to further decouple the model in requirement (1) by characterizing the player's performance at a specific moment through a point prediction model based on the player's winning probability at that moment. However, understanding only the player's performance and winning probability at a specific moment is insufficient. To conduct a practical quantification analysis of momentum, we perform statistical analysis and state transition processing on the real-time winning probability prediction model. By fitting the estimated global winning probability of a player at a specific moment, we quantify their momentum at that moment.

4.2 Model Workflow

Our model established for Task 1 consists mainly of three components: **LSTM Performance Prediction, Momentum Quantification, and Match flow Visualization.**

The overall workflow of the model is presented in Figure 1

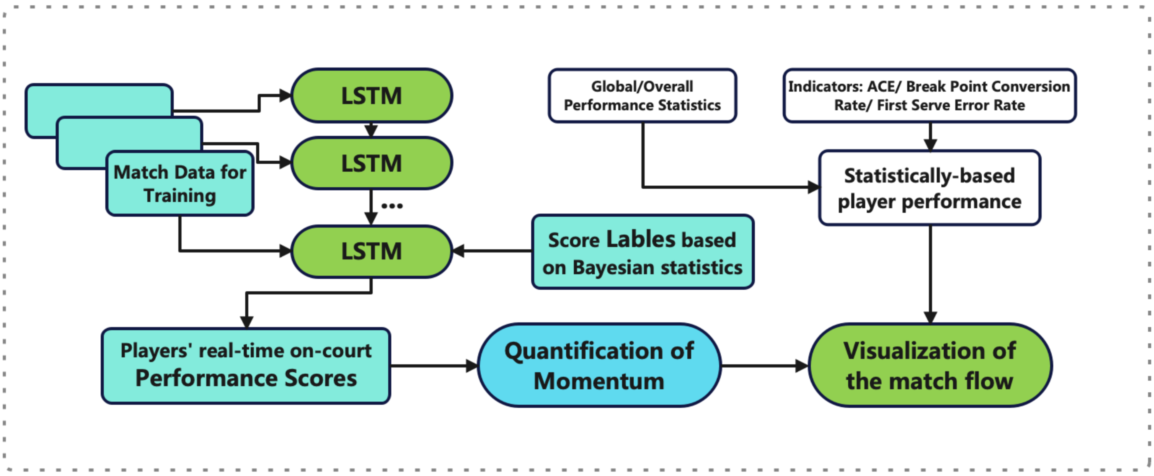

Figure 1: The model for Task 1

Model	Feature Interpretability	Temporal	Model Complexity
Logistic Regression	Strong	No	Low
Non-linear Regression	Moderate	No	Moderate
LSTM (RNN Variant)	Moderate	Yes	Moderate to High

Table 4: Comparison of Different Models

4.3 Real-time Performance Point Prediction Model Based on LSTM

4.3.1 Why We Use LSTM

Although the logistic regression model has strong feature interpretability, where learned weights represent the impact of each feature, when trained on given tennis match data, it doesn't consider the temporal information of the matches. Instead, the logistic regression model independently trains samples, which is inconsistent with the impact of momentum stated in the problem (i.e., the influence of the previous moment's state on the subsequent moment's state). Non-linear regression models, despite having higher model complexity, still overlook the temporal information in the match data and are therefore not considered.

Based on this, we attempt to use recurrent neural networks(RNN) to simultaneously capture both the feature information and temporal information of the match data, which is theoretically feasible.

Considering that the current state during a match is significantly influenced by both nearby moments and moments that occurred further in the past, it's essential to address the common issue of long-term dependency in the context of predicting tennis match outcomes using Recurrent Neural Networks (RNNs). The long-term dependency problem is illustrated in Figure 2, where the predicted value h_t for the current state is minimally affected by the initial match result X_0. This challenge arises because RNNs struggle to effectively capture information far from the current time step. In each time step, the model can only retain and utilize limited prior information. Consequently, in tennis match prediction, this limitation results in the loss of information about states that occurred further in the past compared to the current moment.

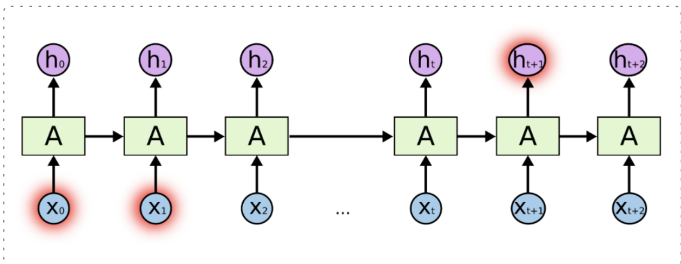

Figure 2: The Long-term Dependency Problem

Therefore, we opted for a variant of RNN known as Long Short-Term Memory (LSTM[5]) as the foundational model for real-time winning rate prediction. The basic structure of LSTM is illustrated in Figure 3. LSTM incorporates gate mechanisms including forget gates, input gates, and output gates, allowing the network to decide when to remember, update, and output information. This enables the model to learn and maintain dependencies on match states, addressing the challenge of long-term dependency.

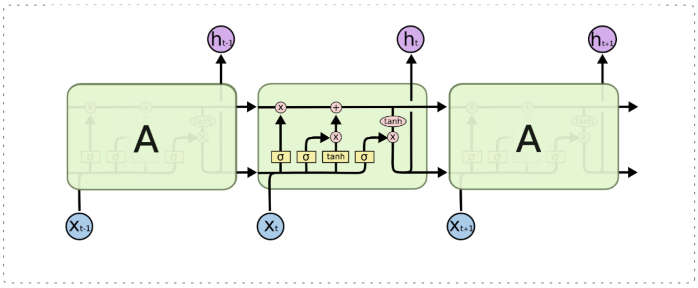

Figure 3: Basic Structure of LSTM

Specifically, for the task of predicting tennis match outcomes over time, the gating mechanism in LSTM operates as follows:

1. Input Gate i_t:

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \tag{2}$$

2. Forget Gate f_t:

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \tag{3}$$

3. Output Gate o_t:

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \tag{4}$$

4.3.2 Labeling: Considering the Serving Advantage

Choosing Labels for the LSTM Regression Model: To fit the current winning probability ($p1$ winning probability at a specific moment), using the outcome of the next point as the true label for LSTM regression is feasible. Thus, for the i-th sample, we can define:

$$y_i = \begin{cases} 1, & W_{1i} = 1 \\ 0, & W_{1i} = 0 \end{cases}$$

where W_{1i} represents the winning outcome of $p1$ in the i-th sample, $W_{1i} = 1$ indicates $p1$ wins, otherwise loses.

Considering the Advantage of Serving: Taking into account the assumption of the prompt that **"in tennis, the player serving has a much higher probability of winning the point/game,"** implying a positive impact of serving on the winning probability. If 'serving' is considered as a feature input to the LSTM model, it cannot guarantee this assumption. Therefore, we modify the label values to reflect the positive impact of serving on the winning probability:

In the case of $p1$ serving, the probability of winning the next point is higher than when $p1$ does not serve. In the case of $p1$ serving, the probability of losing the next point is lower than when $p1$ does not serve.

This can be achieved by modifying the values of y as follows:

$$y_i' = \begin{cases} 1, & W_{1i} = 1, S_{1i} = 1 \\ \beta, & W_{1i} = 0, S_{1i} = 1 \\ 1 - \beta, & W_{1i} = 1, S_{1i} = 0 \\ 0, & W_{1i} = 0, S_{1i} = 0 \end{cases}$$

where $\beta > 0$ represents the positive impact of serving on the winning probability; W_{1i} indicates whether $p1$ wins in the i-th sample, $W_{1i} = 1$ indicates $p1$ wins, otherwise loses; S_{1i} represents whether $p1$ serves in the i-th sample, $S_{1i} = 1$ indicates $p1$ serves, otherwise does not serve.

A more general expression is:

$$y_i' = y + (S_{1i} - W_{1i}) \cdot \beta$$

Determination of β: Rather than manually determining β, statistical results would be more convincing.

We calculate the winning frequency of $p1$ when serving for all samples, approximating this frequency as the probability:

$$P = \frac{f(W_{1i} = 1, S_{1i} = 1)}{f(S_{1i} = 1)}$$

where $f(W_{1i} = 1, S_{1i} = 1)$ represents the frequency of samples where $p1$ serves and wins, and $f(S_{1i} = 1)$ represents the frequency of samples where $p1$ serves.

Then, we define β to describe the serving advantage as follows:

$$\beta = P - 0.5$$

According to the statistics, $P = 0.6731$ and $\beta = 0.1731$.

Final Expression for Labels:

$$y_i' = y + (S_{1i} - W_{1i}) \cdot 0.1731$$

4.3.3 Fitting and Prediction

Regarding data features, we have conducted feature extraction operations in Section 3 of feature engineering, and the selected features are presented in Table 3.

For the LSTM data samples, we initially randomly divided all 31 matches into training and testing sets. Specifically, 5 matches (with match-ids 2023-wimbledon-1305/1309/1313/1402/1504) were designated as test set samples and were not involved in model training. For the samples used in training, we employed a sliding window approach. All windows from the same match in the training set were extracted and shuffled to form batches, which were used as inputs for the model. The labels were extracted considering the serving advantage, as discussed in Section 4.3.2.

With these training samples and appropriate label data, we can proceed to train the point prediction model defining real-time player performance. The set hyperparameters are detailed in Table 5.

The learning curve of the LSTM is illustrated in Figure 4. It can be observed that the training error consistently decreases with increasing epochs, indicating that the model

Table 5: Hyperparameters of Model

Parameter	Value
Input Size	20
Hidden Size	180
Number of Layers	2
Output Size	1
Number of Epochs	80

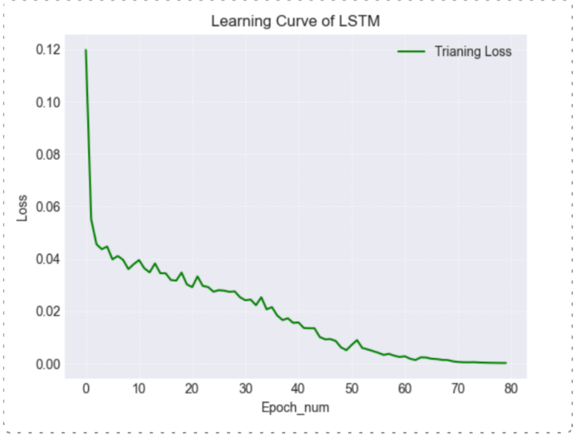

Figure 4: Learning Curve of LSTM

is learning the distribution of the training data. After the model converges, we halted the training to prevent overfitting to the training data.

Utilizing the real-time performance point prediction model we developed, we made predictions on the test set, and the visualization of the results is presented in Figure 5. The solid green line represents the actual performance of players in matches, while the dashed red line represents the model's predicted scores for player performance. It is visually evident that the two curves largely follow each other, indicating a favorable predictive performance of the model.

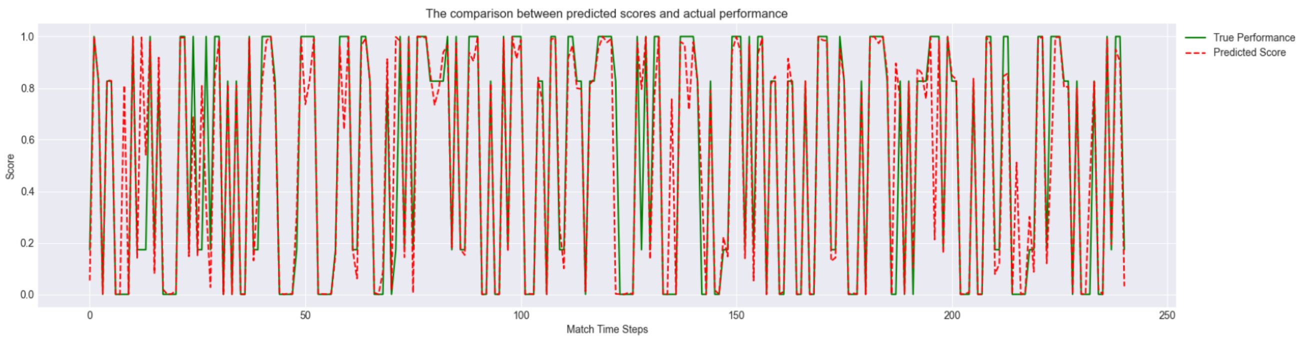

Figure 5: Model Predictions on the Test Dataset

4.3.4 Model Error Analysis

There are various metrics for error analysis of the model's predictive capability, and we have chosen three commonly used ones for our analysis. We trained the model multiple times, and the final analysis results are computed as the average values of the selected metrics.

MAE: The weighted average of the absolute differences between the predicted values and the true values.

Mean Absolute Error does not consider the direction of errors but focuses solely on the average magnitude of prediction errors. Therefore, this metric avoids the problem of errors canceling each other out, providing a more accurate reflection of the actual prediction error magnitude. The formula is as follows:

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}|Y_t - \hat{Y}_t| \tag{5}$$

RMSE: The square root of the average of the squared differences between the true values and the predicted values.

It measures the deviation between predicted and true values and provides a more intuitive sense of the magnitude. The formula is as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=t}^{n}(Y_t - \hat{Y}_t)^2} \tag{6}$$

R^2: The Coefficient of Determination

It is used to measure the goodness of fit of the model predictions to the actual values. The formula is as follows:

$$\text{R}^2 = 1 - \frac{\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2}{\sum_{t=1}^{n}(Y_t - \bar{Y})^2} \tag{7}$$

The variable t in the above three metrics represents the time steps of a tennis match. The total length n of each match prediction is the match duration minus the window length. The final error analysis results are presented in Table 6.

Table 6: Mean Values of Model Evaluation Metrics

Metric	Mean*
MAE	0.0638
RMSE	0.6829
R^2	0.1159

* We conducted multiple model training sessions and took the average of all test results.

4.4 Quantifying Momentum Through EWMA of Performance

Our model is based on the assumption that at a specific moment in a tennis match, the predicted global winning rate, obtained by statistically analyzing the local point winning rates leading up to that moment, represents the player's momentum at that moment. This allows for the quantification of real-time momentum, with the change in momentum trends reflecting the direction of the match.

The state transition equation for momentum is as follows:

$$M_t = f(M_{t-1}, EP_t) \tag{8}$$

Here, M_t represents the predicted momentum quantification value at time t, M_{t-1} represents the momentum quantification value at the previous moment, EP_t represents the estimated real-time winning rate at time, and f is the function representing the impact of real-time performance prediction on the momentum state transition.

Taking into account the inconsistency in the impact of states at different time intervals on the current state, we utilize the concept of Exponential Weighted Moving Average (EWMA) to quantify momentum. The state transition is as follows:

$$M_t = (1 - \alpha) \cdot M_{t-1} + \alpha \cdot EP_t \tag{9}$$

Here, the parameter α serves as the smoothing parameter, representing the degree to which the previous moment's performance influences the current momentum.

As momentum modeling inherently lacks any dimensional units, and our analysis focuses on the trend of momentum changes rather than its absolute values, we have normalized it.

We arbitrarily choose the parameter α as 0.05 for the quantification calculation and visual presentation of momentum.

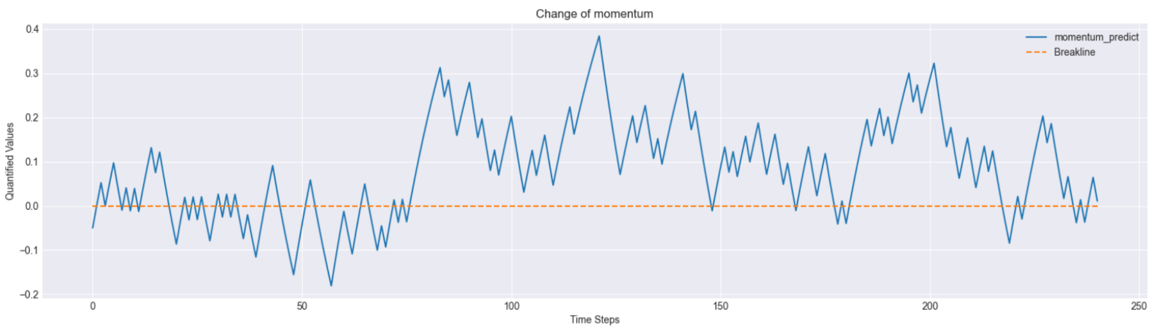

Figure 6: Quantification of Momentum Trends

4.5 Statistically-based Performance Vistualization

5 The Role of Momentum

5.1 Decoupling of Indicators

Momentum, as a non-explicit characteristic, is indeed challenging to quantify, but its impact on a player during a match is intuitive. We aim to assess whether swings in play and runs of success by one player are random using the performance prediction and momentum quantification models defined in Task 1.

Initially, we attempt to decouple the problem by evaluating whether the following two indicators are random:

(1) Swings in play, representing the fluctuation in the match.

(2) Runs of success, indicating a player's consecutive scoring situations.

Although these two indicators seem closely related to the match's trends, our analysis reveals that they can be assessed from different dimensions or even different methods.

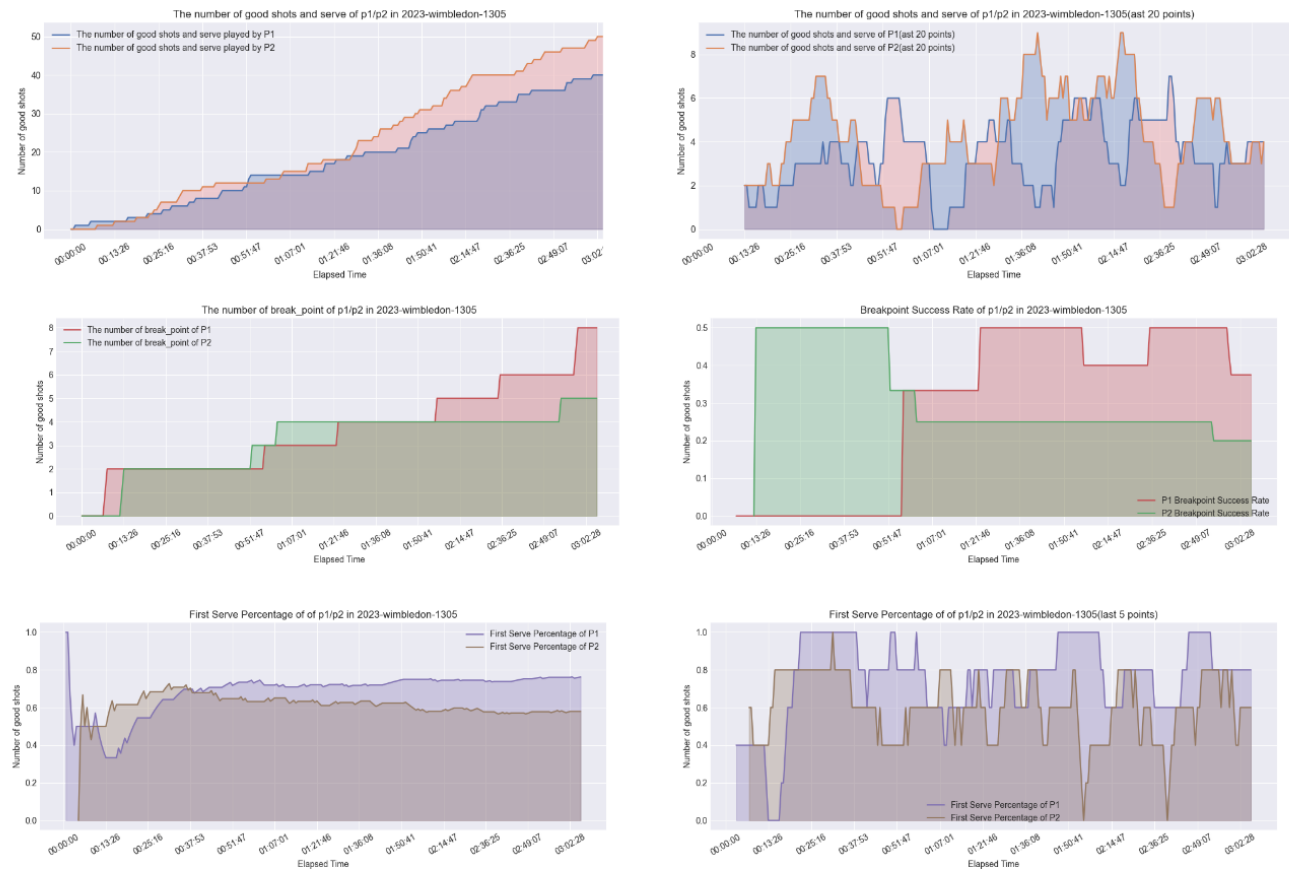

Figure 7: Visual Summary of Player Performance

First is the indicator (1) swings in play. We believe or define swings in a match as the temporal variation sequence of momentum. In fact, in our Task 1 model, we have conducted a detailed quantification analysis of momentum (including training a real-time prediction model and using EWMA for state transitions and quantification), and its definition is given in Equation 9.

The derivation for swings is as follows:

$$Swings = \Delta M_t = M_t - M_{t-1} \tag{10}$$

Substituting Equation 9 into Equation 10, we obtain:

$$\Delta M_t = \alpha \cdot (EP_t - M_{t-1}) \tag{11}$$

Clearly, swings (ΔM_t) is related to EP_t and M_{t-1}, meaning that in this model, the real-time prediction results for player performance and winning probability at time t by the LSTM-based predictive model determine the state transition of swings. In other words, we can assess the randomness of swings' distribution based on the stochasticity of the EP_t sequence. If EP_t is not random, then according to our derivation, swings are certainly not random. The complete randomness testing process for swings in play is presented in the next section.

Next is indicator (2) runs of success. Intuitively, runs of success in tennis matches refer to a player's consecutive scoring situations. To simplify the model, we define a winning streak as two consecutive scores. Thus, if a player scores at both time t and time $t-1$, we determine that they are "on runs of success." Therefore, we can detect and record these winning streak data based on real data. To identify whether these winning streak data are random, further randomness testing can be conducted, and this part of the content will also be presented in the next section.

5.2 Model Workflow

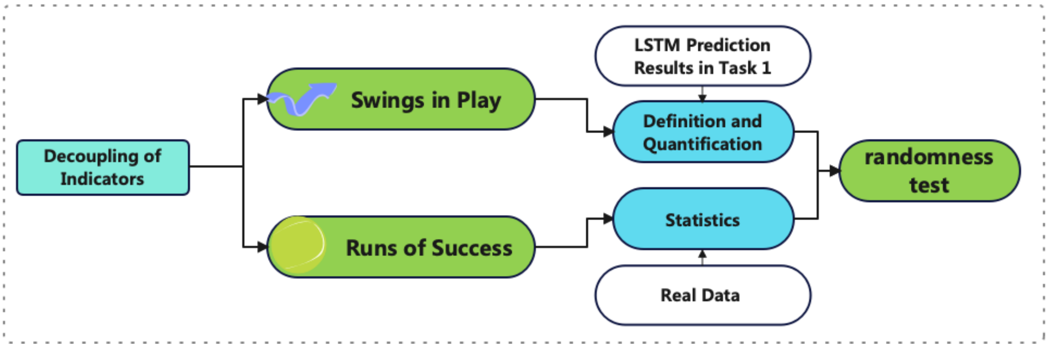

Figure 8: The model for Task 2

5.3 Randomness Testing

5.3.1 Swings of Play: Binarization and Runs Test

In Section 5.1, we have discussed the rationale behind the following conclusion:

The real-time predictive results (EP_t) of player performance at time t, based on the LSTM prediction model, determine the state transition of swings of play. We can assess the randomness of the distribution of swings by examining the randomness of the distribution of the EP_t sequence. If EP_t does not exhibit a random distribution, according to our deduction, the swings are undoubtedly non-random.

For the EP_t sequence, we employ a binarization method to generate a binary sequence (01 sequence). Specifically, moments where a player's performance score exceeds 0.5 are considered as 1, and otherwise, they are considered as 0. We then utilize the runs test method to assess the randomness of this binary sequence.

The runs test is a method primarily used for assessing the randomness of binary sequences. A run refers to a segment in the sequence where the same sample consecutively appears. The main approach of this test is to identify the runs and their respective quantities formed by the arrangement of sample indicators in the binary sequence. Random sequences tend to exhibit runs of varying lengths, while non-random sequences may have a more uniform distribution, enabling the test to evaluate randomness.

Ultimately, we obtained the results as shown in Figure 7. The Z-score is negative, indicating that the observed number of runs is less than the expected number, suggesting that the data tends to cluster together to form continuous trends. The p-value approaching zero indicates that the run patterns in the data are non-random and statistically significant. This suggests that we can reject the null hypothesis, asserting that the data does not follow a random process.

Algorithm 1: Run Test for Binary Sequence

```
Data: Binary sequence binarySequence
Result: Acceptance or rejection of the null hypothesis with P-value
Initialize variables;
n1 ← count of 0s in binarySequence;
n2 ← count of 1s in binarySequence;
ERL ← (2·n1·n2)/(n1+n2) + 1;
SD ← sqrt( (2·n1·n2·(2·n1·n2−n1−n2)) / ((n1+n2)^2·(n1+n2−1)) );
Z ← calculateZStatistic(binarySequence, ERL, SD);
P ← calculatePValue(Z);
while condition do
    Perform operations in the loop body;
    if condition then
        Perform operations when the condition is true;
    else
        Perform operations when the condition is false;
if P ≤ α then
    return Reject the null hypothesis (Sequence is not random) with P-value P;
else
    return Accept the null hypothesis (Sequence is random) with P-value P;
```

Conclusion: The run patterns of the binary sequence generated by binarizing the EP_t sequence differ from a random process, indicating the presence of some trend or pattern. Therefore, the indicator of Swings of play, determined by the EP_t sequence, is undoubtedly non-random. This underscores the role of momentum in match flow.

Table 7: Run Test Results

	Z Score	P Value
Test Result	-6.9791	2.9714×10^{-12}

6 Swings Prediction and Indicator Analysis

In the third task, we aimed to predict the fluctuations in the match, analyze factors strongly correlated with match swings, and provide potential positive guidance for coaching and on-court performance for players.

To provide better recommendations for the observation indicators of the flow of play change time, we believe that fluctuations significant enough to cause a change in momentum are more analytically meaningful. Combining the quantification model for momentum proposed in Section 5.4, we can analyze and predict the turning points of momentum.

6.1 Labeling: Threshold Segmentation and Ternarization Processing

The prediction of turning points can be considered as a classification task. Due to the fact that classification tasks typically involve supervised machine learning models, it is necessary to assign labels to all data points that could potentially be turning points.

Definition of Momentum Turning Points:

According to the quantification model for momentum proposed in Section 5.4, when momentum equals 0, it signifies a balanced situation. The turning points of momentum are defined as instances where momentum changes from negative to positive or from positive to negative, which is intuitive and reasonable.

However, a significant issue arises when the match momentum fluctuates around zero during intense competition between two players. The previously described method may result in numerous turning points during these intense moments, even though there might not be a clear shift in the match momentum at those instances.

Through research, we found inspiration from the threshold processing work in digital image processing[4].

Trinarization of Momentum and Redefinition of Momentum Turning Points:

The data representing the temporal fluctuations of momentum is subjected to threshold segmentation. Specifically, values with absolute magnitudes less than the threshold τ are set to 0, positive values exceeding the threshold τ are set to 1, and the rest are set to -1. In other words, we discretize and categorize the temporal fluctuations of momentum using threshold segmentation and trinarization, as shown in Equation 12.

$$M_t' = \begin{cases} 0, & \text{if } |M_t| < \tau \\ 1, & \text{if } M_t \geq \tau \\ -1, & \text{if } M_t < -\tau \end{cases} \tag{12}$$

Our approach is interpretable, defining moments when the momentum value exceeds the threshold as strong momentum moments, moments with absolute momentum values less than the threshold as balanced moments, and otherwise as weak momentum moments. The redefined turning points of momentum are the time points where transitions occur among these three discrete states.

The threshold is chosen to be 0.09, i.e., τ = 0.09. This choice is interpretable, as it aligns with the model proposed in Section 5.4, where momentum increases after winning two consecutive points. The visualization of threshold segmentation and trinarization of momentum is as Figure 9 and 10

Classification Task and Label Formulation:

The prediction of momentum turning points boils down to a binary classification task, determining whether a given moment is a momentum turning point. The label formulation for this classification task is as follows:

$$Y_t = \begin{cases} 1 & \text{if } X_t \text{ is a positive turning point} \\ -1 & \text{if } X_t \text{ is a negative turning point} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

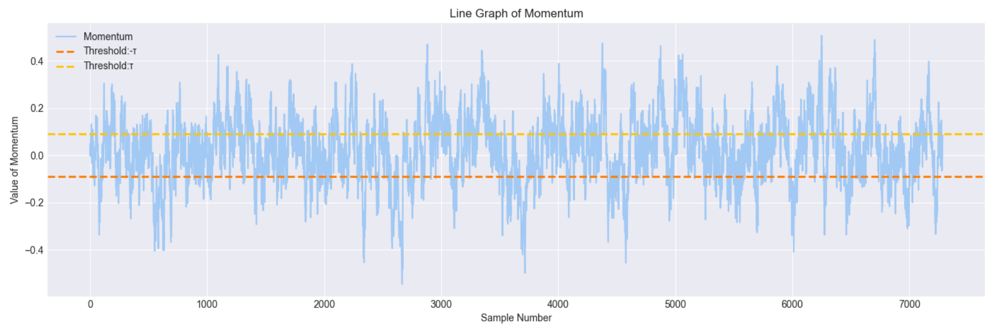

Figure 9: Visualization of Threshold Segmentation

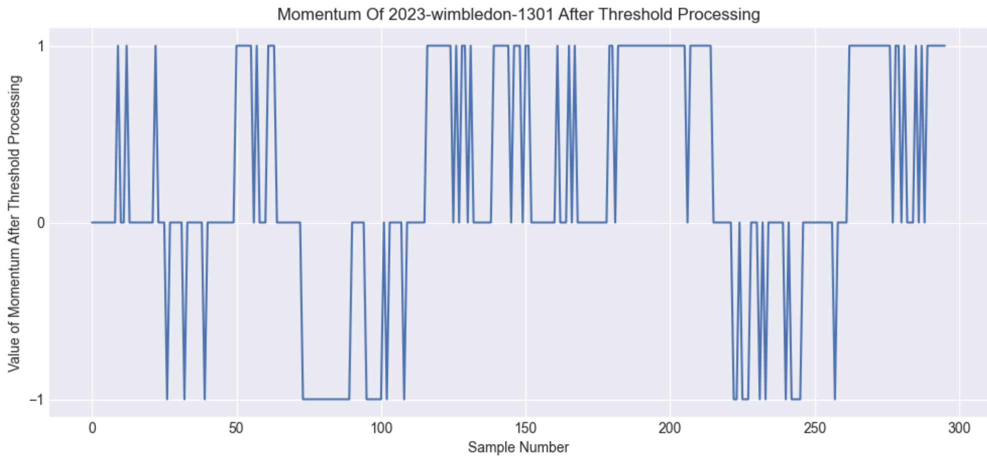

Figure 10: Trinarization of Momentum

6.2 The Turning Point Classification Model Based on Random Forest

6.2.1 Why We Use RF

Based on the classification task of momentum turning points and the determination of relevant features:

(1) Effectively predicts turning points.

(2) Indicates the importance of features in the training data.

With these explicit requirements, we opt for a supervised machine learning model for the classification task of turning points, aiming for its ability to generalize the learned predictive capability from the training set to unknown datasets. Considering the need for feature importance selection, we intuitively prioritize models like decision trees and ultimately choose the Random Forest model based on its performance.

6.2.2 Fitting and Prediction

Sample Structure: Features extracted in Section 4.3 are used for each sample, corresponding to a binary label (0 or 1).

Training and Testing Set Split: The dataset is divided into training and testing sets with an 80:20 ratio. To ensure visualization on a per-match basis, random samples from 5 matches are selected as the testing set, while the rest are used for training.

Handling Imbalanced Data: After analyzing the dataset of 7284 samples, it is found that the ratio of label 0 is as high as 77.2%, while labels 1 and -1 account for only 11.2% and 11.4%, respectively. This indicates an imbalanced data distribution in the classification problem. Random Forest (RF) inherently has advantages in handling imbalanced data. To enhance RF's capability in dealing with imbalanced data, the following optimizations are implemented:

Different weights are assigned to samples of different classes during node splitting: fewer samples receive higher weights. The weight W_i for class i is calculated using the formula:

$$W_i = \frac{S}{S_i}$$

where S is the total number of samples, and S_i is the total number of samples in class i.

Entropy is chosen as the evaluation criterion for node splitting. Compared to Gini impurity, entropy is more sensitive to impurity and tends to create more balanced trees, demonstrating relatively strong adaptability to imbalanced data.

The prediction results for a randomly selected match from the training set are illustrated in Figure 11.

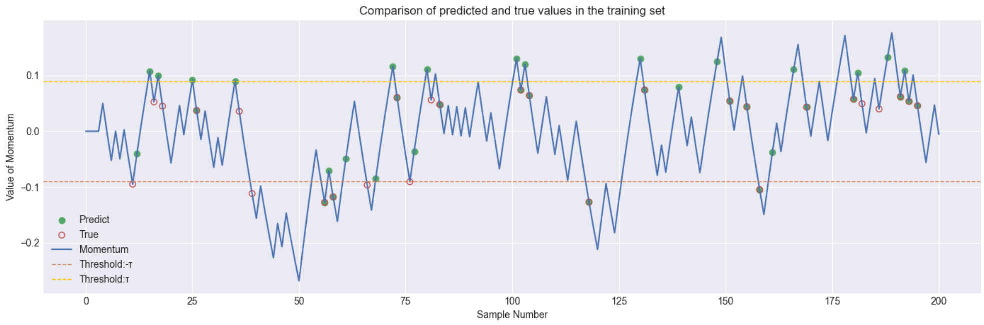

Figure 11: Turning Points Prediction on Training-set

The predicted turning points overlap with the actual turning points, and the non-overlapping predicted points are also close to the actual points.

6.2.3 Model Evaluation

Table 8: Performance Metrics for the RF Classification Model on Training Data

Evaluation Metric	Metric Explanation	Calculated Value
Accuracy	Proportion of correctly classified instances	0.926
Precision	Accuracy of positive class predictions	0.950
Recall	Proportion of true positive instances	0.726
F1 Score	Harmonic mean of Precision and Recall	0.805

From the two major categories of metrics above, it can be seen that the model has achieved a good fit on the training set. There are instances where samples labeled as "good momentum" are classified as "balanced momentum," or samples labeled as "bad momentum" are classified as "balanced momentum," while very few or almost no samples labeled as "good momentum" are classified as "bad momentum," or vice versa. This phenomenon intuitively aligns with reasonability and interpretability.

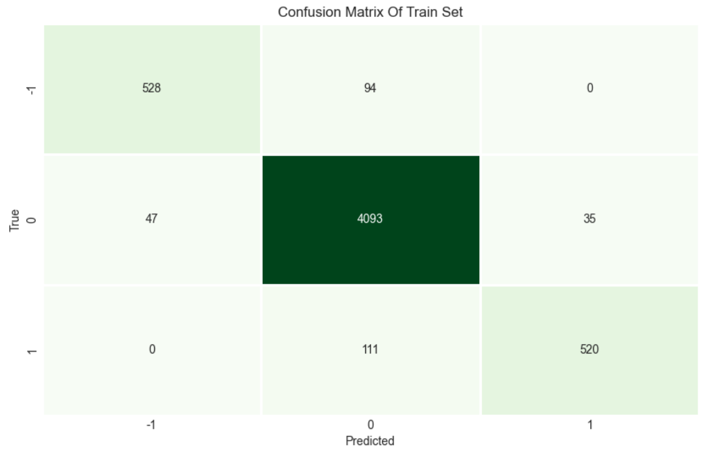

Figure 12: Heatmap of the Confusion Matrix for the Training Sett

6.3 Indicator Correlation Analysis

Based on the training set, Random Forest (RF) assesses the importance of the features extracted in Section 4.3 as follows:

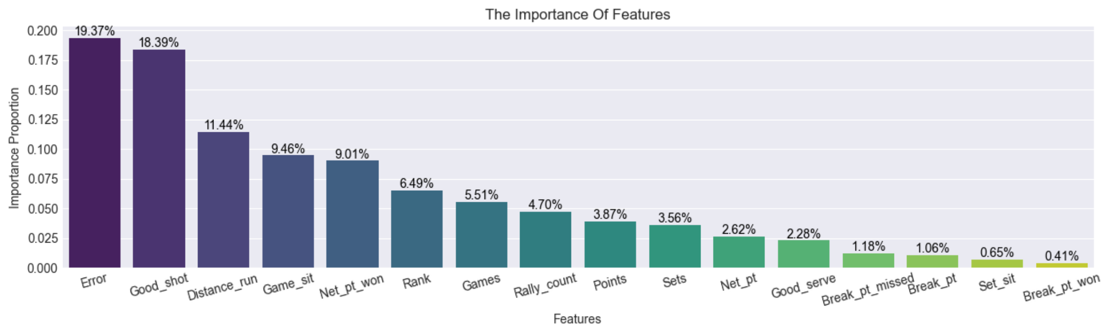

Figure 13: Importance of the Features Based on RF

From the graph, it can be observed that factors such as Error, Good_shot, Distance_run, and Game_sit have relatively high importance, indicating that these factors have a significant impact on momentum transitions.

This result is reasonable and interpretable. For example, Error and Good_shot directly reflect the player's technical proficiency and performance state, while Distance_run may reflect the player's endurance level and offensive-defensive capabilities. Game_sit can represent various factors such as court conditions and psychological aspects resulting from the end of a game. All these aspects are closely related to changes in momentum. The relatively lower importance of factors like Rank suggests that there is no complete disparity in skill on the field, and performance is influenced by various comprehensive factors.

7 Model Evaluation

7.1 Predictive Capability and Enhancements

Regarding the competition trend reversal classifier trained in Task 3 based on a random forest, we attempted its actual classification on the test set to assess the model's predictive capability. The evaluation employed the same metrics as those used for assessing the classification model on the training set, and the results are presented in Table 9.

Table 9: Performance Metrics for the RF Classification Model on Testing Data

Evaluation Metric	Metric Explanation	Calculated Value
Accuracy	Proportion of correctly classified instances	0.746
Precision	Accuracy of positive class predictions	0.484
Recall	Proportion of true positive instances	0.385
F1 Score	Harmonic mean of Precision and Recall	0.393

By comparing Figure 8 and Table 9, it can be observed that the model exhibits poor generalization on the test set. Particularly, the precision and recall are low, indicating a small True Positive (TP) value calculated by the classifier. This implies that the model's predictive performance on positive instances, i.e., predicting trend reversal points, is subpar.

Improvements to the Model:

- Based on Model Performance: Explore new factors to be included in future models based on performance evaluation results. For instance, if the model performs significantly worse on women's match data compared to men's matches, considering the influence of gender-specific factors may be necessary. If the accuracy on grass court matches is lower than on hard court matches, investigating the impact of playing surface type on match momentum may be essential.
- More Complex Features: Consider incorporating more complex features such as player fatigue, psychological pressure during critical moments in a match, and court factors (e.g., grass, hardcourt, clay).
- Model Adjustments: Fine-tune model parameters based on different types of datasets or experiment with different machine learning algorithms.
- Cross-Sport Generalization: Analyze the applicability of the model across various sports and explore whether universal patterns of momentum transitions can be captured across different sports.

7.2 Generalization Analysis

The generalization of the model depends on the design and training process. In our modeling of momentum in tennis matches, we conducted a relatively comprehensive analysis of the features of the match data to ensure the complexity and inclusiveness of the model. We retained most of the factors that we deemed could influence momentum in matches. However, the cost of doing so is the potential loss of some generalization ability of the model.

For other similar types of matches, such as women's tennis matches or championships, our model exhibits relatively strong generalization. This is because these matches share similar features, and with sufficient training, the model may possess decent generalization capabilities. However, due to potential differences in rules and formats, targeted adjustments and optimizations of the model are necessary.

The generalization ability of our model may be limited for matches on different court surfaces or in other sports. Different types of matches involve distinct game focuses and tactics. For example, basketball games may involve rapid transitions between offense and defense, while table tennis games may prioritize controlling the spin and speed of the ball—features not as prominent in tennis matches. Additionally, our consideration of unique features in tennis match data (such as break points) makes it challenging to generalize to other types of matches.

To enhance the model's generalization to other sports, further feature engineering is required. Specific data features for each type of sport should be collected and analyzed. For instance, basketball may require considerations of metrics like rebounds and assists, while table tennis may need to factor in metrics related to successful receives, forehand and backhand strokes[8], etc.

MEMORANDUM

To: Tennis Coaches and Players

From: A MCM Team

Subject: Sincere Recommendations from Momentum Modeling

Date: Tuesday, February 6th, 2024

In tennis, the momentum effect refers to a phenomenon where a player who wins a point, hits a crucial shot, or delivers another outstanding performance gains a psychological advantage over the opponent, thus boosting confidence and performance. It can also be understood as ”success breeds success.” The dictionary defines momentum as ”strength or force gained by motion or by a series of events.” Research indicates that momentum significantly influences match outcomes, and players often leverage momentum to achieve a series of successful matches.

In the modeling process, we explored many conclusions about momentum, hoping to inspire and assist in your training.

Coach’s Recommendations:

(1) Swings of Match flow and runs of success are not random, so the influence of momentum should not be ignored during matches. Strengthening attention to momentum is necessary.

(2) At a specific moment in the match, our time-series model can capture and output real-time final winning probability information of the player based on previous match data, and this prediction result has high confidence. Therefore, when the real-time predicted winning probability is too low, it is advisable to make appropriate choices in the match, ensuring the conservation of energy.

(3) At a specific moment in the match, it is possible to make real-time predictions of momentum fluctuations and turning points based on previous match data. This is advantageous for timely prediction and prevention of potential adverse effects (such as a continuous decline in momentum).

(4) Considering the impact of momentum, psychological resilience training for players can be incorporated into regular training sessions.

Player’s Recommendations:

(1) Train players to maintain focus and composure, regardless of individual scores or match results. This involves some techniques such as concentration, visualization, and positive self-talk to maintain mental strength and focus during matches.

(2) Ensure physical readiness to maintain high intensity and focus during matches. Proper adjustments, such as appropriate halftime breaks and adjustments, can help maintain performance levels even in the face of momentum fluctuations or fatigue.

(3) Momentum Management: Encourage players and coaches to actively monitor specific indicators reflecting match momentum, such as score differentials and effective use of serve rights.

References

[1] Sipko, M., & Knottenbelt, W. (2015). Machine learning for the prediction of professional tennis matches. MEng computing-final year project, Imperial College London, 2.

[2] Clarke, S. R., & Dyte, D. (2000). Using official ratings to simulate major tennis tournaments. International transactions in operational research, 7(6), 585-594.

[3] Klaassen, F. J., & Magnus, J. R. (2001). Are points in tennis independent and identically distributed? Evidence from a dynamic binary panel data model. Journal of the American Statistical Association, 96(454), 500-509.

[4] Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, 9(1), 62-66.

[5] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[6] Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

[7] Silva, J. M., Hardy, C. J., & Crace, R. K. (1988). Analysis of psychological momentum in intercollegiate tennis. Journal of sport and exercise psychology, 10(3), 346-354.

[8] Hegazy, H., Abdelsalam, M., Hussien, M., Elmosalamy, S., Hassan, Y. M., Nabil, A. M., & Atia, A. (2020). Ipingpong: a real-time performance analyzer system for table tennis stroke's movements. Procedia Computer Science, 175, 80-87.

[9] Thibodeaux, J., & Winsler, A. (2020). Careful what you say to yourself: Exploring self-talk and youth tennis performance via hierarchical linear modeling. Psychology of Sport and Exercise, 47, 101646.