Problem Chosen	2024 MCM/ICM Summary Sheet	Team Control Number
C		2403774

Uncovering the Hidden Momentum: A Data-Science Exploration of Tennis Match Dynamics

Tennis, as a comprehensive sport, has always been a subject of great interest. Besides the physical demands on athletes, exceptional skills and outstanding tactical strategies are key to victory. This study delves into the concept of "**athletic potential**" in competitive sports, an inherent vitality and advantage that becomes apparent during matches, utilizing data science and machine learning techniques. Ultimately, the data-driven insights derived from our models will provide valuable guidance and reference for coaches

This article primarily establishes two models: the **XGBoost-based Momentum Quantification Model** and the **Tennis Match Point Time-Convolution Model**.

Before building the models, we initially conducted data preprocessing, including the identification of outliers and handling missing values using strategies such as the **Interquartile Range (IQR)**. We also selectively removed or imputed data based on practical significance. Subsequently, we designed variables based on different timeframes (**Current-Past**) and various temporal scales (**Short-Medium-Long**). Finally, we used **Spearman's correlation coefficient** to eliminate variables with high correlations to the dependent variable and those exhibiting autocorrelation.

Addressing the first problem, this study introduces improvements to the **XGBoost-based momentum estimation** of players. We establish a relationship between momentum and short-term match outcomes, achieving a precision **MSE (Mean Square Error) indicator of 0.02** through a 5-fold cross-validation approach.

Regarding the second problem, we randomly generated match scenarios based on game rules and original data. We applied the model developed for problem one to estimate the momentum for both real and random matches. Ultimately, by comparing various metrics that measure the correlation between momentum and match outcomes, we provide **strong evidence** supporting the impact of momentum on match scores.

For the third problem, we utilized **Bayesian online change-point detection methods** to identify crucial turning points in matches, and we visualized the results using two sample matches. To accommodate the different temporal characteristics of match data, such as capturing immediate situations in short timeframes and understanding overall trends in long timeframes, we designed the **Tennis Match Point Temporal Convolutional Network**. This model achieved a **Top-1 accuracy of 75.70%** in predicting turning points.

Concerning the fourth problem, we conducted domain transfer testing using tennis match data from the **2023 Wimbledon Women's Singles**. The momentum prediction achieved a precise **MSE indicator of 0.025**, demonstrating the high generalization performance of our model.

Additionally, we performed sensitivity analysis to explore the impact of input variations on the model. Finally, the **strengths** and **weaknesses** of our model are summarized. In conclusion, we include a letter to tennis coaches at the end of the article, introducing the overall concept and results of our research.

Keywords: Tennis Athletic Potential, XGBoost, Momentum Quantification, Temporal Convolutional Network

Contents

1 Introduction

1.1 Background

"To become the first man in a decade to defeat Novak Djokovic on Centre Court at Wimbledon is truly a remarkable feat I'll always cherish," Alcaraz reflected on his historic win.

In the gripping Wimbledon men's singles final of 2023, the prodigious 20-year-old Spaniard, Alcaraz, outplayed the venerable 36-year-old Serbian champion, Djokovic, in a five-set thriller to claim the prestigious title. Despite a rocky start, where Alcaraz dropped the first set 1-6, he tenaciously fought back to secure the match, leaving the audience in awe of the stunning upset.

Competitive momentum, within the realm of sports, represents the intrinsic vigor and edge that becomes apparent during a match. It denotes the collective strength and prowess of either a team or an individual athlete, which is instrumental in asserting dominance within the game.

To harness this competitive momentum, a diverse set of skills is crucial, encompassing technical prowess, tactical acumen, psychological resilience, and rigorous physical conditioning. Possession of such attributes enables athletes to elevate their performance, dominate the competition, and achieve superior results.

Despite its frequent mention in sports narratives, competitive momentum remains an elusive, subjective concept, one that defies direct measurement through conventional statistical methods. Therefore, sports analysts and coaches are often inclined to leverage specific in-game data, such as points scored, service speed, error rates, and more, to construct models that endeavor to quantify the effects of momentum and forecast game trends and outcomes.

1.2 Restatment of the Problem

Through the analysis and investigation of the background and negotiators' guidance of the problem, the restate of the problem can be expressed as follows:

- **Build a mathematic model for Match Flow Analysis:**
  - ✓ Create a model that captures the progression of a tennis match based on the scorning events.The model should be capable of identifying who is well-performed at any given moment, and quantify their level of superiority.
  - ✓ Include a visualization that depicts the flow of the match, as informed by the model.
  - ✓ Incorporate the advantage that players have while serving into the model.
- **Use the model to evaluate the claim by a tennis coach who doesn't believe the role of "momentum".**
- **Predict Shifts in "Momentum":**
  - ✓ Construct a model to forsee the pivotal moments that shifts the "momentum" of the game,and identify the key indicators that drive these shifts.
  - ✓ Offer advise to a player entering a new match against a different opponent, based on the dynamics of "momentum" swings.
- **Model Testing:**
  - ✓ Test the model against additional matches to evaluate its effectiveness in forecasting momentum swings and the outcomes of games.
  - ✓ Should the model display occasional inaccuracies, pinpoint elements that could be incorporated in future versions to enhance its predictive accuracy.
  - ✓ Consider how the model might be adapted to other contexts, such as women's matches, different court surfaces, or other racket sports like table tennis.
- **Compose a one-to-two-page memo to be included in the report, offering advice to tennis coaches according to the study.**

1.3 Our Work

In line with the background and research questions, our work encompasses the following aspects:

(1) Data preprocessing to obtain the required independent variables for subsequent modeling. Establishing an XGBoost-based momentum quantification model to quantify and assess a player's performance at specific times, and visualizing the momentum in relation to match outcomes.

(2) Simulating a random match to compare the momentum of a random match with that of a real match. Using Pearson correlation coefficients, we confirmed that a player's fluctuations and success during a match are not random but influenced by momentum.

(3) Utilizing Bayesian online changepoint detection methods to identify turning points in each match and obtaining the importance ranking of independent variables through Spearman rank correlation. We then developed the Tennis Match Point Time-Convolution Model, which can predict when turning points are likely to occur.

(4) Validating the model using data from other matches and conducting an evaluation of the model's performance.

2 Assumptions And Notations

2.1 Assuptions

- **The sum of momentum of two players is constant.** Momentum reflects a player's ability to score against an opponent. Given the momentum of player 1, the momentum of player 2 can be derived symmetrically, which is conducive to describing the player's state.
- **The effect of momentum is primarily short-term.** A player's momentum at a given moment may remain stable in a local time window and change with subsequent wins and losses.
- **There is a certain degree of momentum inheritance between each game, set and match.** At the beginning of each game, the player will be somewhat affected by the situation of the previous game or the overall set and court, thus obtaining different initial momentum.
- **The impact of data outside the table on the match is ignored.** The physical quality of athletes and other factors can be reflected in the information of the match given in the table, and other factors have a small impact and can be ignored.
- **The number of promotions a player makes may be unrelated to momentum.** Both sides in each match have the same number of advances and similar physical exertion, so the effect of advance records on momentum is not taken into account.
- **The total distance run to some extent represents the value of physical exertion.** It is common sense that the more an athlete runs, the greater the physical exertion.

2.2 Notations

Tabel 1:Symbols and Descriotions

Symbol	Description
b	Current iteration step
M	Ensemble of models
η	Learning rate
i	Number of samples in the dataset
m	Total number of samples in the dataset
γ,λ	Use to adjust the complexity of the tree.
y_i	Momentum based on match outcomes
$Y_i^{(t)}$	Model's prediction of momentum at iteration t
T	Current number of leaf nodes in the regression tree
Y_i	Momentum of player i at time t

M_{m0}	Maximum momentum
M_f	Serving momentum factor Mf=0.1

3 Data Preprocessing

3.1 Data Cleaning

Due to the randomness of data and missing records from devices, it is necessary to clean the data. This involves correcting anomalies and filling in missing values to ensure the accuracy of subsequent analyses and the reliability of modeling. Taking the current serve speed column(speed_mph) and Player 1's running distance column (p1_distance_run) as examples, the methods and process of data cleaning are detailed as follows.

3.1.1 Correction of Anomalous Values

Due to the randomness in data collection, there may be outliers in the samples that significantly differ from the majority of the values, thus it requirs repair or removal. Use the Interquartile Range (IQR) method to identify the lower quartile Q1 at the 25% position and the upper quartile Q3 at the 75% position in the data, then calculate the interquartile range (IQR), with the calculation formula as follows:

$$Q1 = \frac{n+1}{4} \tag{1}$$

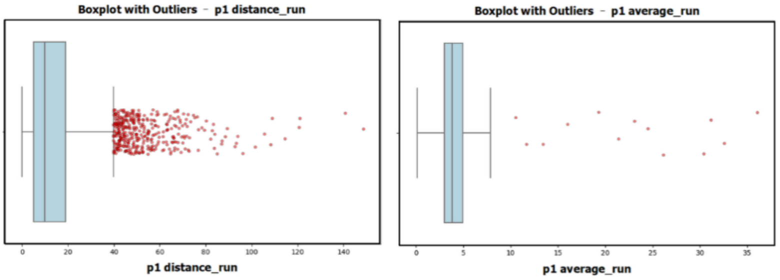
$$Q3 = \frac{3(n+1)}{4} \tag{2}$$

$$IQR = Q3 - \mathrm{Q}1 \tag{3}$$

In the formula, n is the size of the dataset. If n is not an integer, linear interpolation can be performed between the two data points closest to n to estimate Q1 and Q3. The interquartile range (IQR) represents the distance between Q3 and Q1.

In this paper, data outside the range [Q1 - 1.5 IQR, Q3 + 1.5 IQR] is defined as an outlier and is subjected to correction. Taking speed_mph as an example, the box plot of its outliers is shown in Figure 1.

For the outliers on the left side of the figure, they are replaced by performing data fitting interpolation within the group of matches they belong to. It is worth noting that for the p1_distance_run data column, since p1_distance_run is related to rally_count, the more rallies played, the greater the running distance of the player. Therefore, it is necessary to first remove the impact of the number of rallies on running distance. First,divide p1_distance_run by rally_count,and then proceed to correct the outliers. The box plots of outliers before and after preprocessing are shown in Figure 2. The processed outliers are then subjected to data fitting interpolation, and multiplied by rally_count to serve as the new corrected values for p1_distance_run.



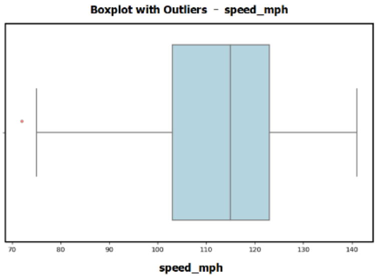


Figure 1:Outlier boxplot of speed mph

Figure 2: Box plots of outliers for p1_distance_run before and after preprocessing

3.1.2 Filling Missing Data

Firstly, it is necessary to identify missing values in the dataset. Taking speed_mph as an example, a heatmap of missing values can be generated using the Heatmap Function from the Seaborn Library in Python. This heatmap provides a visual representation of missing data within the dataset. The result of generating a heatmap using the heatmap function is shown in Figure 3. In the figure, there are large yellow bands with gaps, indicating continuous missing data in this segment. Upon comparison, it is observed that this missing data all originates from the match_id = 2023-wimbledon-1310, which is attributed to system errors such as missing device records and the importance of the speed_mph parameter. Therefore, the data from this match is not considered, and the remaining 30 sets of data will be used for subsequent analysis and modeling.

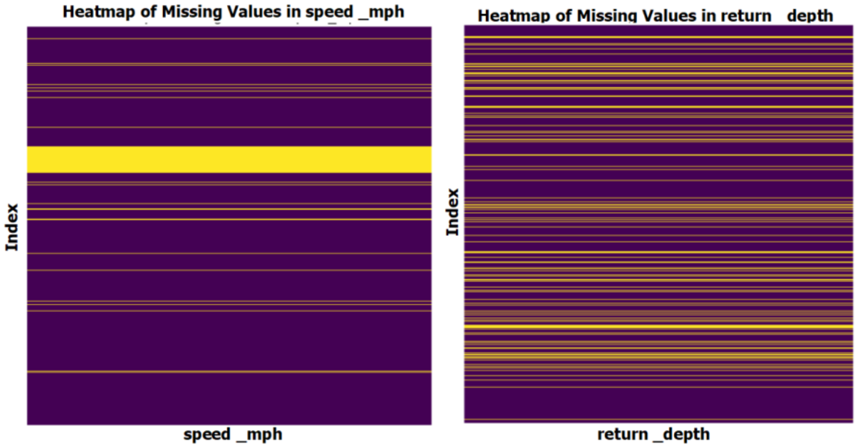


Figure 3: The heatmap of missing values for "speed_mph."

Additionally, considering the real circumstances of the matches, when there is a double fault situation in serving, meaning p1_double_fault + p2_double_fault = 1, both speed_mph and rally_count should be equal to 0. Taking the match_id=1301 as an example, all the data with speed_mph=NA is selected, as shown in Table 2.

Tabel 2: Partial data when match_id=1301 race speed_mph=NA

elapsed_time	p1_double _fault	p2_double _fault	p1_double_fault+p2_double _fault	rally_count	speed_mph
0:32:57	0	1	1	0	NA
2:06:20	1	0	1	0	NA
2:14:53	0	1	1	0	NA
2:28:21	1	0	1	0	NA
2:53:10	1	0	1	0	NA
2:55:10	1	0	1	0	NA
3:05:04	0	0	0	3	NA
3:13:12	1	0	1	0	NA
3:22:58	1	0	1	0	NA
3:28:08	0	1	1	0	NA
3:33:45	1	0	1	0	NA

From the table, it can be observed that when elapsed_time is 3:05:04, neither side committed a double fault, and thus, the recorded speed_mph as NA is considered a genuine missing value. The rally_count of 3 also supports this conclusion. In contrast, for other instances where speed_mph is NA, double faults occurred. Although speed_mph is recorded as missing, it should be imputed.

The nearest-neighbor interpolation method is employed for filling in missing values. This method is suitable for imputing missing values in time series data and typically does not introduce new data structures or models. It effectively preserves the original data distribution or characteristics. Similarly, the same method was used for data cleaning in other variables within the dataset.

3.2 Data Transformation

In this study, some variables are unordered multicategorical variables represented as text in the dataset. Taking "winner_shot_type" as an example, it is represented as "F" for "Forehand Winner," "B" for "Backhand Winner," and "0" for "Other Winner." Analysis reveals that it belongs to an unordered multicategorical variable. From a numerical perspective, assigning values 1, 2, and 3 would imply an ordinal relationship, which is not accurate. In reality, the methods of winning are independent and do not have a hierarchical order. Assigning values of 1, 2, and 3 and incorporating them into the model would be inappropriate, so they need to be transformed into dummy variables.

For example, with regard to the "winner_shot_type" variable, there are three possible values: "F" is transformed into {0, 1}, "B" into {0, 1}, and "0" into {0, 0}. Similarly, other variables are transformed using the same method.

3.3 Feature Recombination

As shown in Table 3, based on the dataset and in accordance with the rules of tennis matches, the following 36 sets of variables are selected as potential independent variables that may affect momentum. These independent variables are categorized into three types based on the time scale: current time, previous shot time, and long-term time. The independent variables included in the long-term time category are calculated based on existing data and encompass various information related to the same game, set, and match.

Tabel 3: Variable - Symbol comparison table

Instantaneous Calculated Variables	**Symbols**
Instantaneous Local Variable	
Current server	server
Current serve count	serve_no
Current serve speed	speed_mph
Current serve width	serve_width
Current serve depth	serve_depth
Rounds taken for current game score	rally_count
Current return depth	return_depth
Current set count	set_no
Current game count	game_no
Historical Instant Variable	
Previous point's score	Point_Score
Previous point's score without touch during serve, option 1	Serve_No_Touch_1
Previous point's score without touch during serve, option 2	Serve_No_Touch_2
Previous point's score without touch during rally, option 1	Rally_No_Touch_1
Previous point's score without touch during rally, option 2	Rally_No_Touch_2
Previous point's no-touch win type	Win_Type
Previous point's running distance, option 1	Serve_No_Touch_1
Previous point's running distance, option 2	Serve_No_Touch_2

Time Interval Calculated Variables	**Symbols**
Set Level Interval Variable	
Current set's athlete's consecutive points won, option 1	Consecutive_1
Current set's athlete's consecutive points won, option 2	Consecutive_2
Number of games won in the current set, option 1	Games_1

Number of games won in the current set, option 2	Games_2
Number of double faults in this set, option 1	Double_Faults_1
Number of double faults in this set, option 2	Double_Faults_2
Number of unforced errors in this set, option 1	Unforced_Errors_1
Number of unforced errors in this set, option 2	Unforced_Errors_2
Number of net touches in this set, option 1	Net_Touches_1
Number of net touches in this set, option 2	Net_Touches_2
Current match's points lead progress	Lead_Progress
Game Level Interval Variable	
Total distance covered in this match, option 1	Total_Distance_1
Total distance covered in this match, option 2	Total_Distance_2
Number of break points won by the athlete in this match, option 1	Break_Points_Won_1
Number of break points won by the athlete in this match, option 2	Break_Points_Won_2
Number of break points lost by the athlete in this match, option 1	Break_Points_Lost_1
Number of break points lost by the athlete in this match, option 2	Break_Points_Lost_2
Point Level Interval Variable	
Total distance covered in this match, option 1	Total_Distance_Match_1
Total distance covered in this match, option 2	Total_Distance_Match_2

3.4 Variable Selection

In the initial selection of variables, comprehensive coverage of the model was considered, but there may be some correlations among certain variables. To further improve the operational efficiency and simplicity of the model, a correlation analysis[1] was conducted on the variables intended for inclusion in the model. As shown in Figure 4, the confusion matrix[11] of variables clearly reveals the presence of high correlations among variables. Therefore, variable selection[2] based on correlation is necessary.

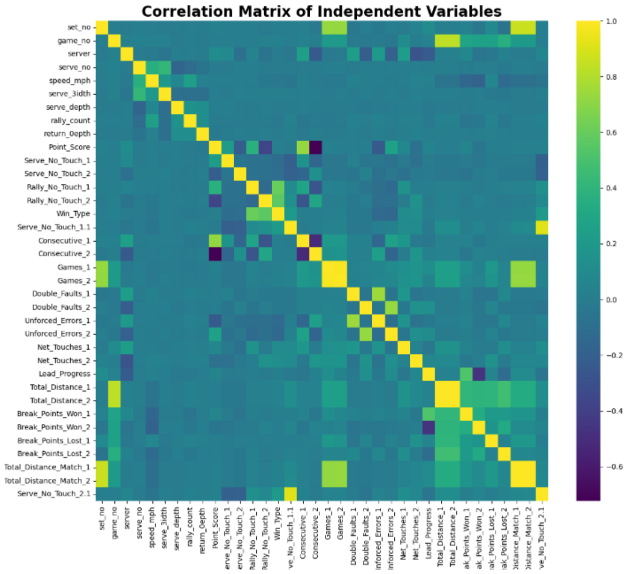


Figure 4: Confusion matrix plot of variables

3.4.1 Correlation Analysis

The Spearman rank correlation coefficient[10] is a non-parametric statistical measure suitable for testing the correlation of abnormal distribution or ordered categorical data variables. The autocorrelations among the independent variables are calculated as shown in Figure 5. Based on this, some variables[3,4] with high autocorrelations in the figure were removed. For example, "set_no" showed strong correlations with "Games_1," "Games_2," "Total_Distance_Match_1," and "Total_Distance_Match_2." We chose "Total_Distance_Match_1" as the final independent variable, reflecting the player 1's energy consumption, and removed the other variables.

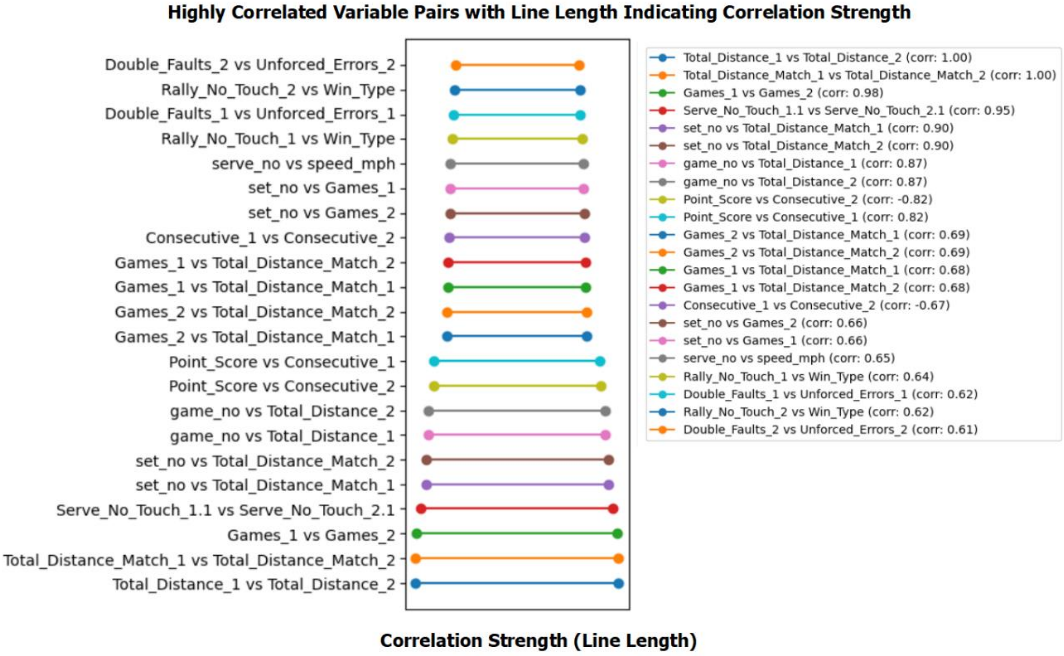
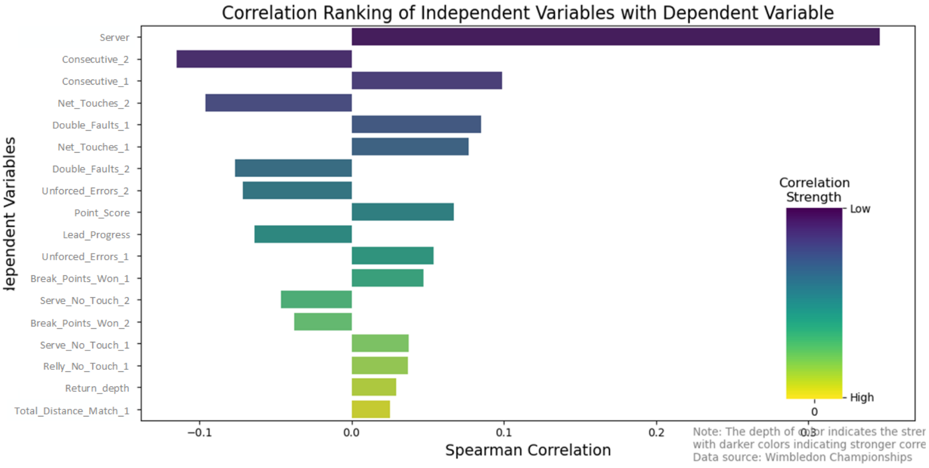


Figure 5: The degree of self-correlation between independent variables

3.4.2 Variable Importance Ranking

After removing linearly correlated variables, there were still many variables, and not all of them had a significant impact on the model's results. To improve computational efficiency and further rank the importance of independent variables, the variables were sorted based on their impact on the dependent variable, which is whether the match is scored. Variables were sequentially removed from the smallest impact factor to the largest until a significant difference in model accuracy occurred.

When the variable "set_no" was removed, the classification accuracy decreased significantly from 95.5% to 66.1%. Therefore, the removal of independent variables was terminated at this point. The final selection of independent variables includes the following: server, Consecutive_2, Consecutive_1, Net_Touches_2, Double_Faults_1, Net_Touches_1, Double_Faults_2, Unforced_Errors_2, Point_Score, Lead_Progress, Unforced_Errors_1, Break_Points_Won_1, Serve_No_Touch_2, Break_Points_Won_2, Serve_No_Touch_1, Rally_No_Touch_1, return_depth, and Total_Distance_Match_1.



Note: The depth of color indicates the strer with darker colors indicating stronger corre Data source: Wimbledon Championships

Figure 6: The correlation ranking of different independent variables with the dependent variable

4 Task 1: Momentum quantization and prediction model

In this section, we first define and quantify momentum. Then, we identify which player performs better and to what extent at specific moments in the game by building an XGBoost momentum prediction model.

4.1 Definition and Quantification of Momentum

We define momentum as follows: Momentum is implicitly embedded in the game state, explicitly reflecting the continuous scoring and losing in the game results. We make the following assumption: if the game state is related to scoring and losing, it can be reflected through momentum, completing the mapping from the distribution of game states to the distribution of scoring situations, establishing a relationship between the two distributions; conversely, if there is no relationship between them, such a mapping would be impossible to achieve.

Let the current game result sequence be $P = \{P_0, P_1, \dots, P_N\}$, and the current moment be *i*, with N samples in total. We aim to estimate a player's momentum at moment i by considering the game results in the vicinity of moment *i*. First, we need to create a window around moment *i* to capture the game's state within a short time frame. Next, we assign weights to these game results within the window, with results closer to the moment i contributing more significantly to momentum estimation, while data from more distant matches have a smaller contribution.

Considering that in tennis matches, the serving player has a higher probability of winning points, we introduce a serving momentum factor, denoted as M_f, to adjust the momentum labels. When Player 1 serves, they gain an initial energy boost, and vice versa, they lose some energy when receiving. Both the gained and lost momentum are controlled by the momentum factor. Therefore, the momentum label formula can be expressed as follows:

$$Y_i = \left(\frac{m}{M_{m0}}\right)(1 - M_t) + M_t, M_t = 0.1 \tag{4}$$

4.2 XGBoost-Based Momentum Quatification Model

Due to the non-linearity between independent variables and the final outcome of the game, we chose to use machine learning methods to discover the underlying relationship between independent and dependent variables. After considering various machine learning models, we opted for XGBoost for momentum prediction. XGBoost[8,9] utilizes an optimized distributed gradient boosting algorithm that allows rapid training of large-scale datasets while maintaining efficient resource utilization.

The construction and solution steps of the XGBoost momentum prediction model are as follows:

(1) Data Preprocessing

Before constructing the model, while maintaining the consistency of data distribution, the original dataset is initially divided into a training set S (80%) and a test set T (20%). The model is trained on S and the model's generalization ability is tested on T. For the training set S, further p-fold k-fold cross-validation is applied, where S is randomly divided into k subsets, and in each iteration, k-1 subsets are used for training, while the remaining one is used for validation. To reduce errors due to different sample divisions, this process is repeated p times, resulting in the average of p+k validation results as the model evaluation result. In this study, p=5 and k=5, meaning 5 times 5-fold cross-validation was utilized.

(2) Basis Function Construction

XGBoost gradually adds new models in each iteration to expand its understanding of new data. Its formula is as follows:

$$M_b = M_{b-1} + \eta \cdot T_b \tag{5}$$

(3) Definition of Loss Function

The goal of XGBoost training is to minimize the loss function, which includes both traditional loss functions and regularization terms that describe model complexity.

$$\min Obj = \sum_{i=1}^{m} l\left(y_i, \hat{y}_i^{(t-1)} + f_i(x_i)\right) + \Omega(f_k) \tag{6}$$

In the formula, the first term represents the difference between the momentum predicted by the model and the momentum calculated based on game results, while the second term represents the model's complexity. In the above formula, {i} represents the number of samples in the dataset, and {m} represents the total number of samples in the dataset. {γ} and {λ} are used to adjust the complexity of the tree. The regularization term can smooth the final learning weight and prevent overfitting, improving the generalization ability of momentum prediction across matches and players.

(4) Momentum Error Prediction

Predicting the momentum can be considered a regression task. In each optimization step, the model minimizes the mean squared error (MSE) between the predicted momentum and the labeled momentum, defined as follows:

$$l(y_i, \hat{Y}_i^{(t)}) = \frac{1}{2}\left(y_i - \hat{Y}_i^{(t)}\right)^2 \tag{7}$$

In the formula, $\{y_i\}$ represents the label of momentum, $\{\hat{Y}_i^{(t)}\}$ represents the momentum prediction of the model in t iterations

(5) Structural Regularization Term Calculation

The formula for calculating the regularization term is as follows:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{8}$$

In the formula, the hyperparameters {γ} and {λ} control the strength of the penalty, {T} represents the current number of leaf nodes in the regression tree, and the last term is the L2 norm of the leaf node values.

Based on the MSE loss function, pseudo-residuals can be derived and expressed as:

$$r_{ib} = -\left[\frac{\partial l\left(y_i, \hat{y}_i^{(b-1)}\right)}{\partial \hat{y}_i^{(b-1)}}\right] = y_i - \hat{y}_i^{(b-1)} \tag{9}$$

In each iteration of the model, these pseudo-residuals are used as gradients of the loss function with respect to the model predictions to train new decision trees. This way, each iteration of the model attempts to correct the prediction errors of the previous iteration.

(6) Momentum Prediction Model Construction

In summary, our model can be summarized as follows:

$$\begin{cases} min\ Obj = \sum_{i=1}^{m} l(y_i, \hat{y}_i^{(t-1)} + f_i(x_i)) + \Omega(f_k) \\ \hat{y}_i = \sum_{k=1}^{K} f_k(x_i)\, , f_k \in \varphi \\ \Omega(f_k) = \gamma T + \frac{1}{2}\lambda w^2 \\ l(y_i, \hat{Y}_i^{(t)}) = \frac{1}{2}(y_i - \hat{Y}_i^{(t)})^2 \\ Y_i = \left(\frac{m}{M_{m0}}\right)(1 - M_t) + M_t, M_t = 0.1 \end{cases} \tag{10}$$

Based on the XGBoost model described above, we model the game data, where the input is the known information at the current time point. The optimization of the objective function aims to minimize the difference between the predicted probability of winning the next time step and the actual outcome.

Algorithm: Momentum Quantization

```
Input: $P_{[i]}, s_{[i]}, N, W, X_{[i]}$
Output: $q_{[i]}, G_{[i]}$
$M_{mo} = \sum_{w=-W}^{W} e^{w}$
Initialize the model Mo with a constant value
e.g. the mean of the labels
for $i \leftarrow 0$ to $N$ do
    $m \leftarrow 0$;
    for $j \leftarrow -W$ to $W$ do
        if $0 \leqslant i+j < N$ and $|s[i+j]-s[i]| \leqslant 1$ then
            $m \leftarrow m + e^{|i+j|} \cdot P[i+j]$;
    end
    $Y_i \leftarrow \frac{m}{M_{m0}} . (1-M_f) + M_f$;
    Get the predicting $\hat{Y_i} \leftarrow \sum_{k=1}^{t-1} f_k(x_i)$
    Get the pseudo-residual $r_i \leftarrow Y_i - \hat{Y_i}$
    Fit a regression tree $T_i$ to the pseudo-residuals $r_j$.
    Update the model $M_i \leftarrow M_{i-1} + \eta \cdot T_i$.
end
```

In the above algorithm, we use the match data S for each time step and differentiate whether it is the same match based on the provided number of match sessions. In the algorithm, W represents the window size, and we use exponential weighting to weigh the match data within the window and accumulate contributions to the momentum estimation in m. To normalize the estimate of momentum, we calculate the normalization factor M_{mo} as the maximum momentum and map m to the range of 0-1. According to the above algorithm, we can obtain momentum labels for each time step based on match results, which are used to fit an XGBoost model to map from match data to estimated player potentials. The fitting of the XGBoost model first obtains predictions of the previous model on the data and then calculates pseudo-residuals based on the output and labels. Afterward, a new tree is obtained based on the pseudo-residual and the specified learning rate and added to the tree ensemble.

(7) Model Evaluation

To assess the predictive accuracy of the model for momentum, we selected the Mean Square Error (MSE) as a performance metric. The MSE measures the difference between the model's predicted momentum and the momentum generated based on the matches. We performed 5-fold cross-validation, and the average MSE for the model was found to be 0.02.

4.3 Results Visualization and Analysis

As shown in Figure 7, we present the estimation of Player1's momentum and their score situation, which can be symmetrically deduced for Player2. Using the XGBoost-based momentum prediction model, we calculate the changes in their momentum. Due to significant fluctuations in the results and considering that momentum quantification results can be affected by external disturbances, the use of a Kalman filter effectively eliminates noise and jitter. Compared to the original momentum results, the results after Kalman filtering are smoother. The orange line in the graph represents records from matches starting from a score of 0 and scoring 100 times, where 1 indicates a score and 0 indicates no score. According to our definition, when the momentum remains consistently above 0.5, Player1 has a greater advantage; conversely, when it falls below 0.5, Player2 has a greater advantage. At the beginning, as shown in Box 1, Player1's estimated momentum increases, indicating continuous scoring in the game with fewer losses. Subsequently, there is a turning point in the estimated momentum, following multiple losses. Then, the momentum returns to a higher level, accompanied by multiple scores. In the later stages of the match, as shown in Box 2, the estimated momentum experiences a significant decline, followed by continuous losses for Player1. Even though there is 1 score, the estimated momentum continues to decline. This phenomenon demonstrates that our designed model can describe the potential scoring situation of players near a certain moment, which is the concept of momentum.

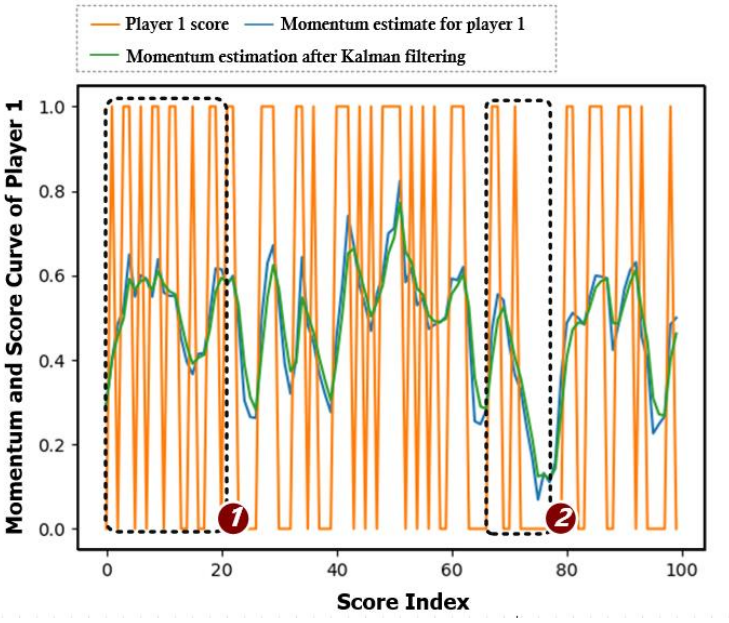


Figure 7: Visualization of score and momentum quantification

5 Task 2: The Momentum-Outcome Connection

In this section, we first generate random match scenarios, including match feature states and match scores. Then we apply the XGBoost momentum prediction model established in question 1 to calculate potential energies for both real matches and random matches. Finally, we compare and evaluate the correlation between match states (match conditions, serve speed, etc.) and momentum by selecting multiple indicators.The results can demonstrate how momentum affects match scores.

5.1 Random Match Scenario Generation

The coach believes that match scores are independent of match states, such as player serves, which affect momentum. Based on this, we set out to generate random match states and random scores to simulate random match scenarios. As shown in Figure 8, to make random match scenarios as close as possible to real match rules, we randomly select values and options for features from a uniform distribution to create new random match states. Regarding match outcomes, we set the probability of random scores to be 50%, and each setting is independently and identically distributed. The generation of random matches is incremental, and match results, determining player victories and match completion, are calculated based on the number of sets, games, and points in a Wimbledon match.

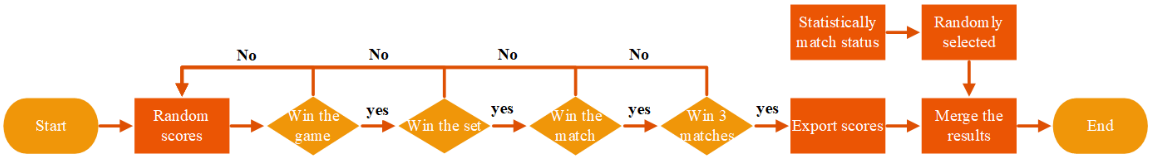


Figure 8: The process of generating random match scenarios

5.2 Analysis of the Impact of Momentum on Scores

We utilize multiple evaluation metrics to comprehensively assess and compare the predictive performance of the model. These metrics include Mean Square Error (MSE), Root Mean Square Error

(RMSE), Mean Absolute Error (MAE), R-squared (R^2), Adjusted R-squared (Adjusted R^2), and Pearson Correlation Coefficient (Pearsonr). Their definitions and meanings can be found in Table 4.

Tabel 4: Explanation of Evaluation Metrics and Numerical Significance

Indicators	Describes
Mean Square Error (MSE)	A smaller MSE indicates that the model's predictions are closer to the actual values
Root Mean Square Error (RMSE)	A smaller RMSE indicates higher accuracy in the model's predictions
Mean Absolute Error (MAE)	A smaller MAE implies higher precision in the model's predictions
Mean Absolute Error (R^2)	An R^2 value closer to 1 indicates a better fit of the model.
Adjusted R-squared (Adjusted R^2)	A higher Adjusted R^2 value suggests stronger explanatory power of the model.
Pearson Correlation Coefficient (Pearsonr)	The Pearsonr value ranges from -1 to 1, where 1 represents a perfect positive correlation, -1 represents a perfect negative correlation, and 0 represents no linear correlation.

As shown in Table 5, the numbers from 1301 to 1701 represent the real scenarios used, with scenario 10 excluded from the analysis due to missing data (details can be found in Section 3.1.2). The label "random" represents simulated random scenarios. Our model demonstrates significantly lower levels of estimation error, with MSE, RMSE, and MAE reflecting errors within the ranges of 3.554e-3 to 8.621e-3, 5.961e-2 to 9.285e-2, and 4.625e-2 to 7.478e-2, respectively. These values are much lower than the corresponding values under random scenarios, which are 0.125576, 0.354367, and 0.297229. This indicates that our model significantly outperforms traditional methods in terms of prediction accuracy and can provide more accurate predictions of actual values.

The R^2 and Adjusted R^2 values of our model fall within the ranges of 0.8602 to 0.9073 and 0.8594 to 0.9071, respectively, which are far higher than the values under random scenarios, which are -2.2744 and -2.28424. This result not only demonstrates the model's strong explanatory power for data variability but also shows its stable performance even when considering model complexity.

The Pearson correlation coefficient further confirms the strong positive linear relationship between our model and actual values, with values ranging from 0.9408 to 0.9591, deviating by only 0.002911. This indicates a very high linear consistency between our model's predictions and actual label values. In other words, as actual label values increase, the model's predictions also increase, and vice versa. This high level of linear relationship suggests that the model not only accurately predicts the magnitude of actual values but also maintains a consistent trend in predictions across different ranges of label values.

In contrast, the Pearson correlation coefficient for random scenarios is close to zero, indicating almost no linear relationship between its predictions and actual label values, failing to reflect the changing trends in actual label values.

In summary, a player's score in the match is not random, and momentum plays a significant role in the game.

Tabel 5: Evaluation of Consistency between Predicted Potential Energy and Match Results

match	mse	rmse	mae	r_squared	adjusted_r_squared	ks_real
1301	0.005301	0.072808	0.057363	0.839388	0.838847	0.052323
1302	0.004725	0.06874	0.055849	0.866766	0.866097	0.331034
1303	0.005613	0.074917	0.059082	0.868931	0.867938	0.002346
1304	0.005553	0.074516	0.058671	0.880302	0.879945	0.052706

1305	0.005473	0.073977	0.059918	0.872491	0.871969	0.065444
1306	0.005082	0.071287	0.057995	0.890007	0.889674	0.0915
1307	0.006592	0.081189	0.066523	0.851584	0.850939	0.018196
1308	0.007664	0.087545	0.069851	0.835557	0.834683	0.156663
1309	0.004817	0.069407	0.055093	0.871828	0.87122	0.365668
1311	0.006264	0.079147	0.062545	0.86502	0.864593	0.11801
1312	0.005775	0.075997	0.061332	0.844124	0.843553	0.206051
…	…	…	…	…	…	…
1701	0.003554	0.059614	0.046257	0.907333	0.907054	0.226841
random	0.136022	0.368812	0.301388	-2.54679	-2.55744	9.25E-06

Using the XGBoost momentum prediction model built in question one, we calculate potential energies by substituting random match data into the model. If the predicted momentum from random data matches the variation in match scores, it proves that the score situation is random. Conversely, if there is no match between the predicted momentum and the score situation, it demonstrates that momentum can reflect a player's scoring situation over a certain period of time. As shown in Figure 9, the left-side data points represent predicted momentum, the right-side represents label momentum, which is the momentum reflected by match results, and the lines in the middle indicate that the two numbers belong to the same moment. By comparing, it is observed that in real match situations, the lines in the graph are mostly parallel to the x-axis, indicating that predicted momentum can be well-mapped to match situations. This suggests that momentum to some extent reflects match outcomes. However, in the coach's perceived random match scenarios, the lines do not exhibit a clear correspondence, indicating that the model cannot map from match states to match scores. Consequently, it implies that momentum can be used to reflect a player's scoring changes over a certain period of time.

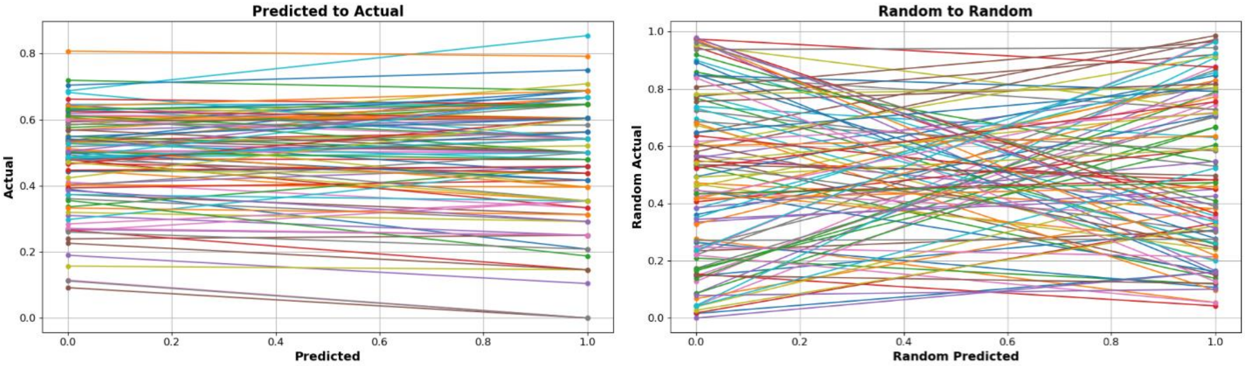


Figure 9: The mapping between the predicted mometum and the momentum defined in the contest

6 Bayesian Inflection point Detection and Prediction

6.1 Bayesian Online Inflection point Detection Method for Identifying Inflection points

6.1.1 Principle

The core principle of the Bayesian online inflection point detection method[5,6,7] is to use Bayesian statistics to identify structural changes, or inflection points, in time series data. This method calculates the conditional probability of observed time series values at each time point to assess the likelihood of a inflection point occurring. Specifically, it assumes that the time series can be divided into several segments, with data within each segment following a certain statistical distribution, and differences in distribution parameters between different segments. By applying Bayesian rules to all observed data before each time point, the posterior probability of that point being a inflection point can be estimated. Subsequently, these posterior probabilities are used to infer the locations of inflection points: points where the probability significantly increases are considered as inflection points. This process is performed online, meaning that data can be processed one by one without waiting for the collection of the entire data sequence, enabling real-time inflection point detection.

1.Initialize

$$P(r_0) = \tilde{S}(r_0) \text{ or } P(r_0 = 0) = 1$$

$$v_1^{(0)} = v_{prior}$$

$$x_1^{(0)} = x_{prior}$$

2.Observe New Datum x_t

3.Evaluate Predictive Probability

$$\pi_t^{(r)} = P(x_t | v_t^{(r)}, x_t^{(r)})$$

4.Calculate Growth Probabilities

$$P(r_t = r_{t-1} + 1, x_{1:t}) = P(r_{t-1}, x_{1:t-1}) \pi_t^{(r)} (1 - H(r_{t-1}))$$

5.Calculate Changepoint Probabilities

$$P(r_t = 0 + 1, x_{1:t}) = \sum_{r_{t-1}} P(r_{t-1}, x_{1:t-1}) \, \pi_t^{(r)} H(r_{t-1})$$

6.Calculate Evidence

$$P(x_{1:t}) = \sum_{r_t} P(r_t, x_{1:t})$$

7.Determine Run Length Distribution

$$P(r_t | x_{1:t}) = P(r_t, x_{1:t}) / P(x_{1:t})$$

8.Update Sufficient Statistics

$$v_{t+1}^{(0)} = v_{prior}$$

$$x_{t+1}^{(0)} = x_{prior}$$

$$v_{t+1}^{(r+1)} = v_t^{(r)} + 1$$

$$x_{t+1}^{(r+1)} = x_t^{(r)} + u(x_t)$$

9.Perform Prediction

$$P(x_{t+1} | x_{1:t}) = \sum_{r_t} P(x_{t+1} | x_t^{(r)}) P(r_t | x_{1:t})$$

10.Return to Step 2

Here are the meanings of the variables involved:

- $P(r_0)$: The probability of initializing a inflection point (inflection point). δ(r) is the Dirac delta function, which equals *1* at *r=0*, and *0* otherwise. Alternatively, $P(r_0 = 0)$ is set to 1, indicating that there are no inflection points initially.
- $v^{(0)}$and$x^{(0)}$: Initialization parameters for sufficient statistics, corresponding to prior knowledge v_{prior} and x_{prior}.
- $\pi_t^{(r)}$: The predicted probability calculated after observing new data x_t. It represents the probability of data x_t occurring given the current sufficient statistics and inflection point location.
- $P(r_t = r + 1 | x_{1:t})$: Growth probability at time *t* for a inflection point.
- $P(r_t = 0 | x_{1:t})$: Inflection point probability at time *t* for resetting to *0*.
- $P(x_{1:t})$: Evidence probability, representing the probability of observing the data sequence $x_{1:t}$ at time *t*.
- $P(r_t | x_{1:t})$: Run-length distribution, indicating the probability of the current run length being $r_t$given the data sequence $x_{1:t}$.
- $v^{(r+1)}$ and $x^{(r+1)}$: Updated parameters for sufficient statistics.

- $P(x_{t+1}|x_{1:t})$: The probability of predicting the next data point x_{t+1} at time $t+1$.

6.1.2 Implementation Process

From the perspective of Player 1, we assigned one point for each victory and deducted one point for each loss. We conducted inflection point detection on Player 1's scoring trend using Bayesian online inflection point detection. We plotted the scoring trends and inflection point graphs for two matches with match IDs 1407 and 1503, as shown in Figure 10 and Figure 11.

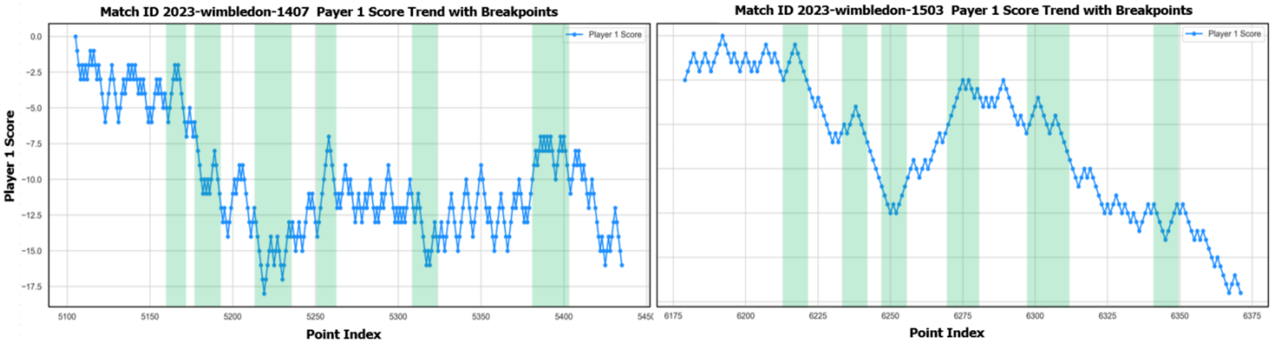


Figure 10: The scoring trend and inflection point graph for Player 1 at matchid=1407&1503

The inflection points in the graph are presented as regions (through) rather than individual points because inflection point detection typically provides a possible interval where structural changes might occur over a period of time. This represents a transitional phase rather than a discrete moment because real data may contain noise, and actual changes may occur gradually.

Taking Figure 10 as an example, based on the data in the table, when the first inflection point occurs, Player 2 made a double fault; when the second inflection point occurs, Player 1 scored a break point; when the third inflection point occurs, Player 2 scored a break point; when the fourth inflection point occurs, Player 1 scored a point without any touch. Subsequent inflection points also correspond to specific events, aligning with the facts. Therefore, we consider the approach of using Bayesian online inflection point detection to find inflection points as feasible.

6.2 Correlation Analysis of Inflection Points

Following the method outlined in Section 6.1, we detected the inflection point regions in the matches. We replaced the dependent variable with a binary variable, where 1 represents being in the inflection point region and 0 represents not being in the inflection point region. We then used the Spearman rank correlation coefficient method as described in Section 3.4.1 to calculate the correlation between independent variables and the dependent variable. Finally, we obtained the importance ranking of the 18 independent variables as shown in Figure 12.

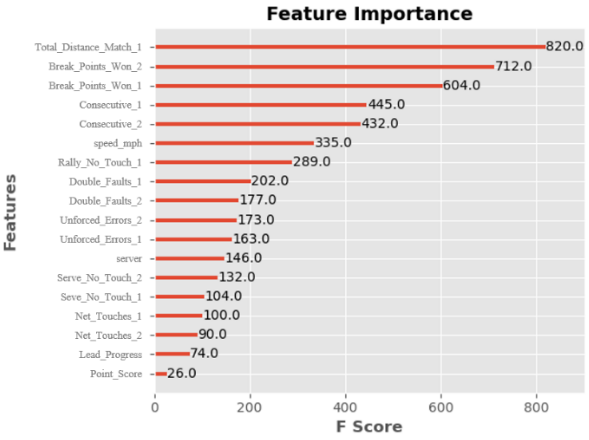


Figure 11: Independent Variable Importance Ranking

From the graph, it can be observed that the most important variable is the distance covered by the athlete, followed by whether both sides won a break point, and subsequently variables like the consecutive scoring streak. This result aligns with our expectations. Firstly, the distance covered by the athlete reflects the degree of physical exertion, with increasing distance indicating growing fatigue, which can lead to shifts in the game's momentum. Secondly, whether both sides won a break point is also a crucial factor affecting the game's trend. Winning a break point provides a boost to the player's morale and confidence while delivering a blow to the opponent's confidence. Therefore, it is considered a significant factor. Based on this, we believe that calculating the importance ranking of independent variables using this method is feasible.

6.3 Tennis Match Point Temporal Convolutional Network

According to the Bayesian online inflection point detection model, we have obtained inflection point regions that are close to the Ground Truth in advance, as shown in Figure 13. We represent the inflection points using a binary model, where points within the inflection point regions are marked as 1, and the rest are marked as 0. Considering the time series nature of the data, we use a Temporal Convolutional Network (TCN) for inflection point prediction, treating it as a binary classification problem. TCN is a type of 1D convolutional model that incorporates the concept of convolution from CNN into sequential networks similar to RNN.

In the TCN network, dilated causal convolutions are primarily used. This convolution is characterized by the inclusion of dilation in the regular convolution, aiming to increase the receptive field, as shown in Figure 13:

Causal Convolution refers to the type of convolution where the output at time t depends only on the information at and before time t and does not depend on information after time t. The diagram illustrates a typical causal convolution schematic.

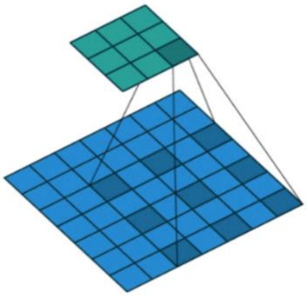


Figure 12: Dilated Conv

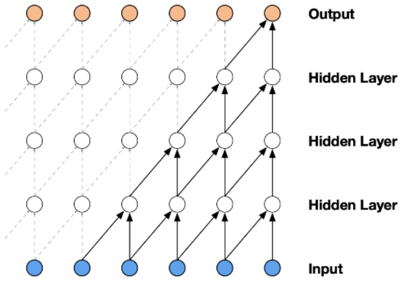


Figure 13: Causal Conv

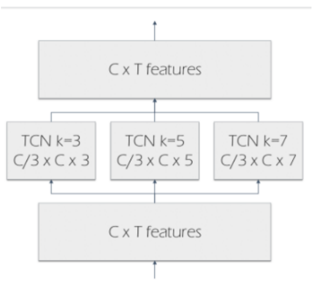


Figure 14: MSTCN

Due to variations in the number of rounds in different matches, it is influenced by factors such as the skill gap between the competing sides and the game's status, both subjectively and objectively pleasing. Furthermore, the presence of winning streaks due to being in a hot state or losing streaks due to being in a low state is critical information for match outcomes. Therefore, in this chapter, based on TCN and considering the characteristics of tennis matches, we have designed a Multi-Stage Temporal Convolutional Network (MSTCN), as shown in Figure 15 below. The introduction of MSTCN aims to incorporate match information across multiple time scales simultaneously for model prediction.

Multi-time scale temporal convolution, when used in tennis match prediction, offers several advantages by integrating information from different time scales. These advantages can improve prediction accuracy, enhance model generalization capabilities, and better handle complex time series data. The summarized advantages are as follows:

6.3.1 Design Details

In this study, we employed a two-layer multi-scale temporal model as described above. We have input time series data X, where X[t] represents the observation at time step t. The model extracts features from the time series through convolution operations.

Multi-Scale Temporal Convolution Layer:

The size of the first convolution kernel is k_1=3, designed to capture features at shorter time scales;The size of the second convolution kernel is k_2=5, intended for capturing features at intermediate time scales;The size of the third convolution kernel is k_3=7, aimed at capturing features at longer time scales.

The formula for performing the convolution operation is as follows:

$$Z_i[t] = f(w_i * X[t:t+k_i] + b_i) \text{ for } i = 1,2,3 \tag{11}$$

$$P_i[t] = g\big(max(f(w_i * X[t:t+k_i] + b_i)[t:t+s])\big) \text{ for } i = 1,2,3 \tag{12}$$

The equations represent a two-scale convolutional neural network layer followed by a max pooling operation. The convolutional layer computes features Z_1, Z_2,and Z_3 by applying kernel weights W_1,W_2,and $W_3$and adding biases b_1, b_2,and b_3,respectively, with a Re LU activation function f. The subsequent pooling layer reduces dimensionality with a max pooling function g over a window sizes, resulting in pooled features P_1, P_2,and P_3.

After the pooling layer, a fully connected layer is added to learn high-level representations of the time series.

$$Y[t] = h(W_f c * P_1[t] + b_f c) \tag{13}$$

In this context:

$Y[t]$ is the model's output.

h is the activation function of the fully connected layer.

$W_f c$ and $b_f c$ are the weights and bias terms of the fully connected layer.

The two-layer multi-scale temporal convolutional network follows the same design approach.

The model's hyperparameters are defined as follows:

Tabel 6:experimental parameter setting details

Name	Illustration	Value
Epoch	Total epoches in training	10
Batch_Size	The number of input video in a batch	6
Leaning-Rate	The learning rate for optimizer	le-4
Optimizer	The optimizer used in our experiment	Adam

6.3.3 Results

In this section, we conducted experiments to classify whether tennis matches belong to the transition point region. Model evaluation was performed using five-fold cross-validation, and the predictive accuracies for fivea runs were 75.49%, 74.94%, 75.70%, 73.98%, and 73.83%, with an average accuracy of 74.79%.

6.4 Suggestions

Based on the performance evaluation of the multi-scale temporal convolutional network model mentioned above, we can observe that the model achieves an average accuracy of approximately 74.79% in predicting tennis match turning points. This indicates that the model has a certain degree of reliability and can predict key turning points in matches relatively accurately. Based on this, I suggest that tennis coaches can use this model to assist in training and match strategy development:

- Match Preparation: Coaches can use the model's predictions to prepare strategies for potential turning points. Understanding possible turning points in a match can help players better prepare mentally and technically to face these critical moments. For example, if a break point has a significant impact on turning points, coaches can advise players to stay focused when facing break points from their opponents, delaying the arrival of a turning point.
- Targeted Training: Using the features identified by the model as turning points, coaches can design specific training plans to enhance a player's performance during these critical moments. For instance, by simulating training scenarios related to turning points, coaches can improve a

player's mental resilience and adaptability.

- Strategy Adjustments: During the course of a match, coaches can also make timely adjustments to the game plan based on the model's predictions. For example, when approaching a predicted turning point, coaches can guide players to adopt a more conservative or aggressive playing style to respond to potential changes in the match dynamics.

7 Task 4: Model Validation and Evaluation

7.1 Model Validation with Other Matches and Accuracy

We obtained tennis match data for the 2023 Wimbledon Women's Singles[12,13] from the internet. We initially performed autocorrelation analysis on the independent variables in this dataset using the Spearman correlation coefficient matrix method. The results of this analysis are shown in Figure X:

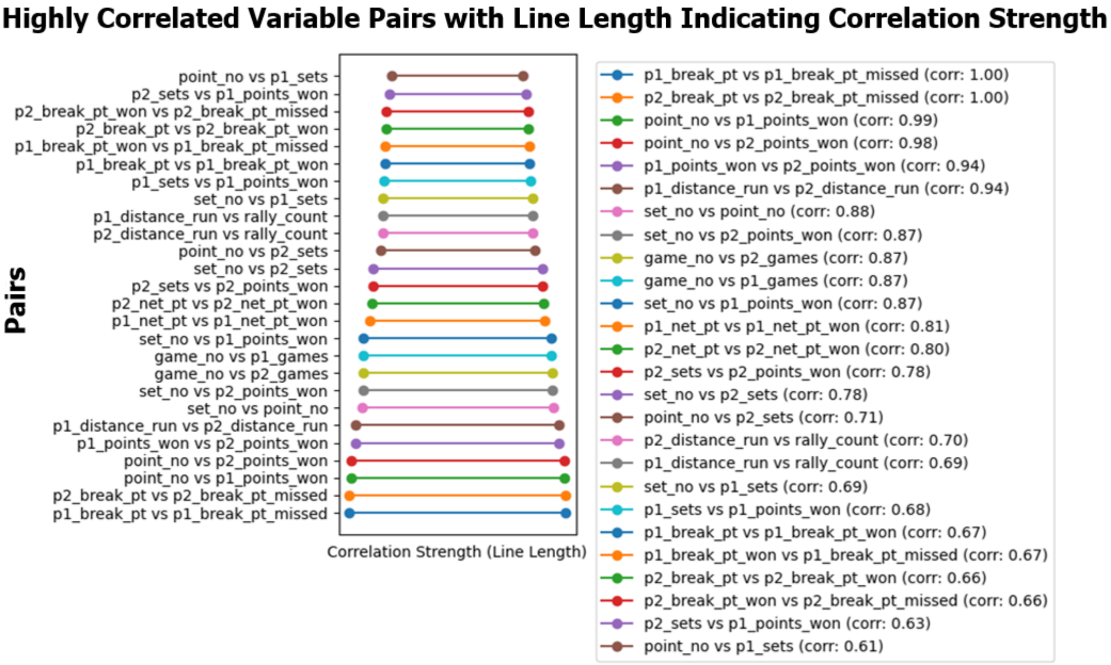
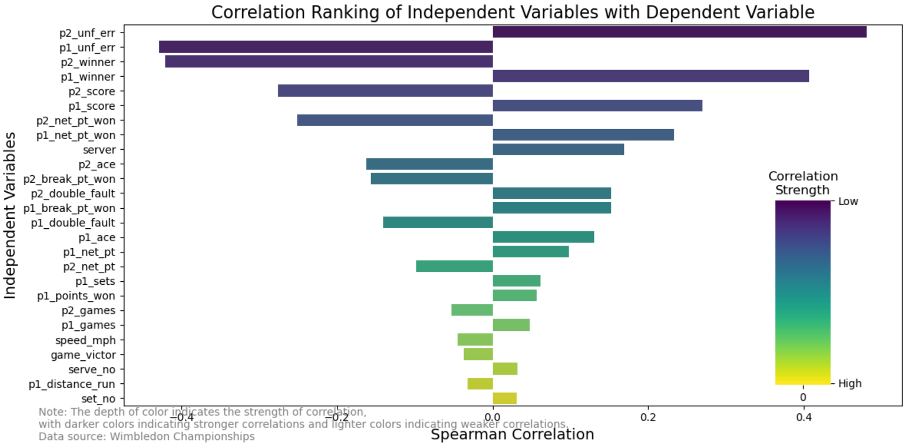


Figure 15: Autocorrelation among independent variables in women's singles

After removing strongly autocorrelated independent variables, we further filtered the independent variables based on their correlation with the dependent variable. The final ranking of the correlation between the selected independent variables and the dependent variable is shown in Figure X:



Note: The depth of color indicates the strength of correlation, with darker colors indicating stronger correlations and lighter colors indicating weaker correlations.
Data source: Wimbledon Championships

Figure 16: Correlation ranking of different independent variables with the dependent variable

Using the selected independent variables, we reran the XGBoost-based potential quantification model and obtained momentum estimates and their corresponding scores for the 2023 Wimbledon women's singles matches, as shown in Figure 17:

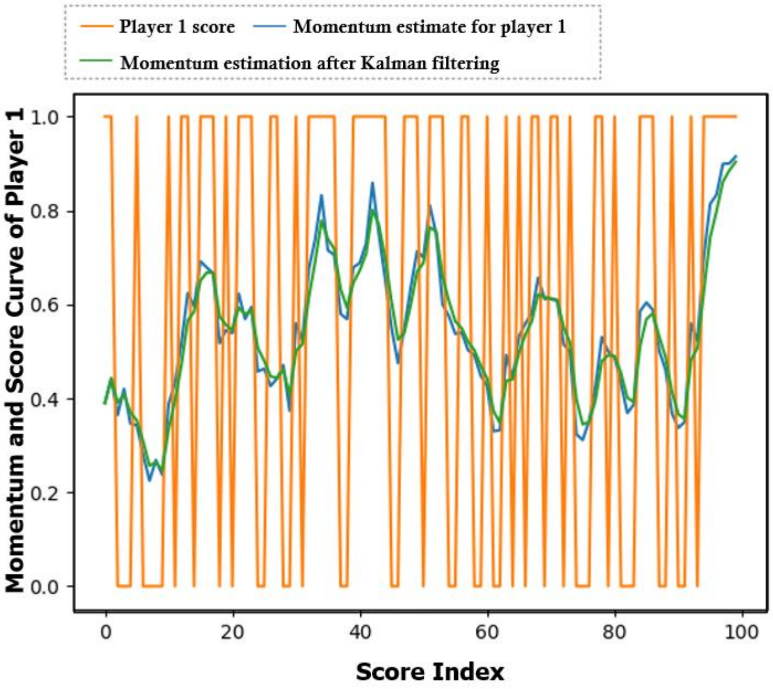


Figure 17: Visualization of women's singles scores and potential quantification

In this simulation, the XGBoost potential energy prediction model achieved an MSE (Mean Squared Error) metric of 0.025, indicating a high level of accuracy.

7.2 Model Evaluation

From the results in section 7.1, it is evident that the model also demonstrates high accuracy in simulating other matches. If the model's accuracy were to decrease, we would consider the following scenarios: the partial elimination of influential independent variables during variable selection, the neglect of other highly influential independent variables in other matches, and more. We would address these issues through the following approaches:

1. Conduct residual analysis on the model to identify systematic biases or uncaptured nonlinear relationships within the model.

2. Introduce external data to retrain the model and enhance its predictive capabilities.

3. Perform error analysis on the model's performance to determine if issues arise from data quality problems such as missing values, outliers, or incorrect data labels.

4. Collaborate with domain experts to explore whether there are essential, yet overlooked factors or relationships.

Our model exhibits strong generality for other matches, as demonstrated in section 7.1, where women's singles data was used to verify its applicability. In various matches, as long as we possess sufficient independent variable data, we can retrain a suitable model with high precision.

8 Sensitivity Analysis

To validate the sensitivity of our model to variable changes, i.e., the impact of input variations on the output momentum estimates, we randomly selected some data points from the entire dataset and modified the values of certain variables, including serve speed and running distance. We compared the changes in predicted momentum and the probability of a turning point occurrence to investigate the model's sensitivity to input changes.

We randomly selected 10 data points from all the data, used the XGBoost momentum prediction model to estimate momentum, and altered the "speed_mph" variable by ±5%. The experimental results are shown in the left figure, indicating that increasing serve speed can enhance the kinetic energy of players by 2.9%, while decreasing it can reduce the kinetic energy of players by 2.1%.

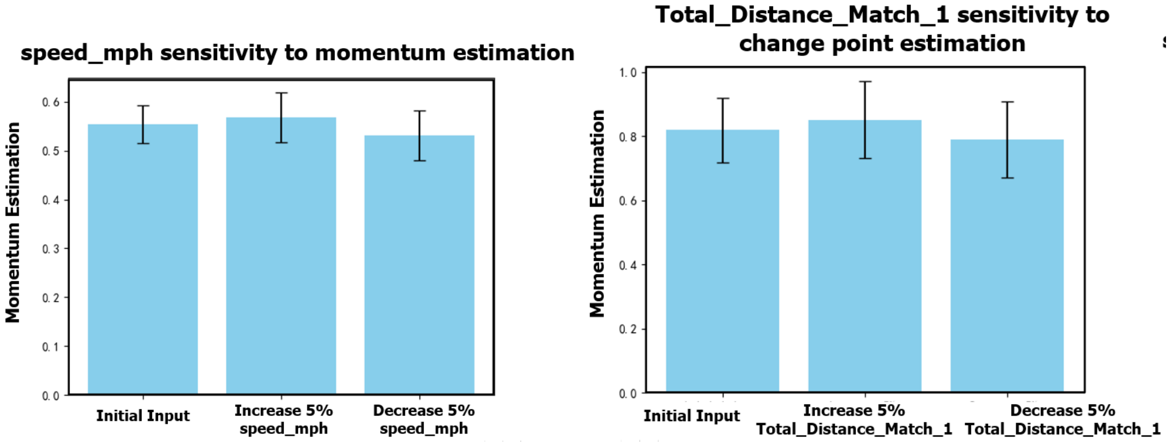


Figure 18: Sensitivity Analysis

We also selected 10 data points where a turning point occurred and altered "Total_Distance_Match_1" for Player 1 by ±10%. We then used the model from Task 3 to compare the changes in the probability of a turning point occurrence. The experimental results indicate that when athletes increase their energy expenditure, it may increase the probability of a turning point occurring by 3.4%. Conversely, decreasing energy expenditure may reduce the probability of a turning point occurring by 4.1%.

9 Advantages and Disadvantages Analysis

9.1 Advantages

(1) The model employed in this paper combines machine learning and deep learning technologies, demonstrating remarkable adaptability and versatility. These models can handle inconsistent game data parameter settings and optimize model parameters to match the characteristics of specific game data through an automated training process.

(2) This paper uses an XGBoost-based quantitative prediction model, which boasts fast training speed and efficient execution performance, optimizing computational speed through parallel processing. Additionally, XGBoost offers high flexibility, allowing for customized optimization goals and evaluation criteria, reducing overfitting risk through regularization, and enhancing model generalization.

(3) The experimental design of this paper utilizes a five-fold cross-validation strategy, which offers three main advantages: reducing overfitting risk, improving model generalization, and providing reliable performance estimates.

(4) The advantages of the multi-timescale temporal convolutional model in tennis turning point prediction are summarized as follows: it can capture features at different time scales, enhance model generalization, improve the capture of time dependencies, and support irregular data sampling, ultimately improving prediction accuracy and model adaptability.

9.2 Disadvantages

(1) The model may not consider all relevant independent variables comprehensively, potentially leading to information omissions that could affect model accuracy.

(2) The multi-timescale temporal convolutional network may increase computational complexity. Since this model needs to simultaneously process information at multiple time scales, it may result in higher computational costs and resource requirements.

References

[1] Shrestha N. Detecting multicollinearity in regression analysis[J]. *American Journal of Applied Mathematics and Statistics*, 2020, 8(2): 39-42.

[2] Daoud J I. Multicollinearity and regression analysis[C]//*Journal of Physics: Conference Series. IOP Publishing,* 2017, 949(1): 012009.

[3] Yu H, Cui P, He Y, et al. Stable learning via sparse variable independence[C]//*Proceedings of the AAAI Conference on Artificial Intelligence.* 2023, 37(9): 10998-11006.

[4] Lee K Y, Li B, Zhao H. Variable selection via additive conditional independence[J]. J*ournal of the Royal Statistical Society Series B: Statistical Methodology,* 2016, 78(5): 1037-1055.

[5] Wang F, Li W, Padilla O H M, et al. Multilayer random dot product graphs: Estimation and online inflection point detection[J]. arXiv preprint arXiv:2306.15286, 2023.

[6] Cao Z, Seeuws N, De Vos M, et al. Inflection point Detection in Multi-Channel Time Series Via a Time-Invariant Representation[J]. I*EEE Transactions on Knowledge and Data Engineering*, 2023.

[7] Sellier J, Dellaportas P. Bayesian online inflection point detection with Hilbert space approximate Student-t process[C]//*International Conference on Machine Learning. PMLR,* 2023: 30553-30569.

[8] Kadra A, Lindauer M, Hutter F, et al. Well-tuned simple nets excel on tabular datasets[J]. *Advances in neural information processing systems, 2021*, 34: 23928-23941.

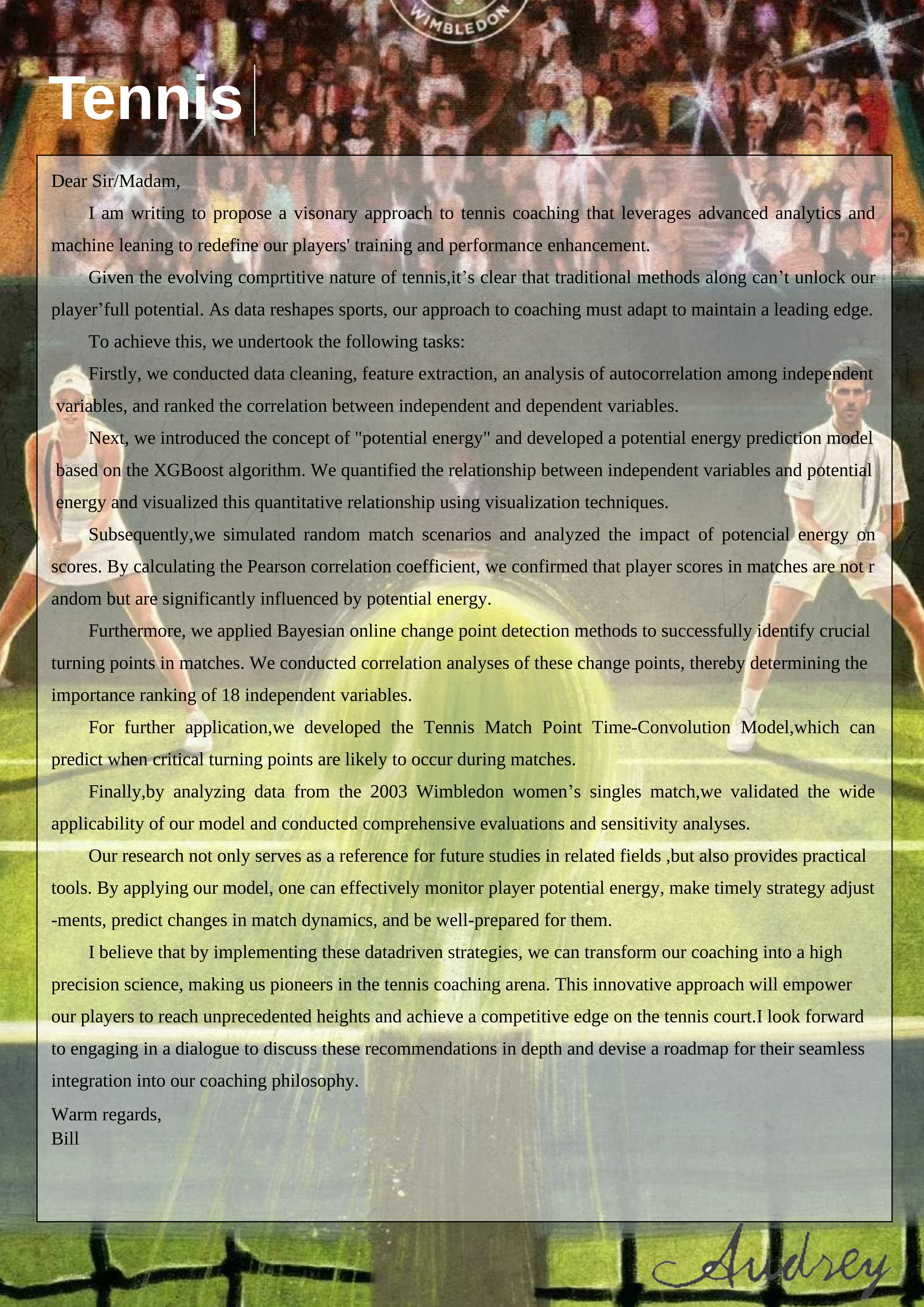
[9] Kadra A, Lindauer M, Hutter F, et al. Well-tuned simple nets excel on tabular datasets[J]. *Advances in neural information processing systems, 2021*, 34: 23928-23941.

[10] Sedgwick P. Spearman's rank correlation coefficient[J]. B, 2014, 349.

[11] Heydarian M, Doyle T E, Samavi R. MLCM: Multi-label confusionmatrix[J]. IEEE Access, 2022, 10: 19083-19095.

[12] Takahashi H, Okamura S, Murakami S. Performance analysis in tennis since 2000: A systematic review focused on the methods of data collection[J]. 2022.

[13] Data-driven analysis of point-by-point performance for male tennis player in Grand Slams https://revistas.rcaap.pt/motricidade/article/view/16370.



Tennis

Dear Sir/Madam,

I am writing to propose a visonary approach to tennis coaching that leverages advanced analytics and machine leaning to redefine our players' training and performance enhancement.

Given the evolving comprtitive nature of tennis,it’s clear that traditional methods along can’t unlock our player’full potential. As data reshapes sports, our approach to coaching must adapt to maintain a leading edge.

To achieve this, we undertook the following tasks:

Firstly, we conducted data cleaning, feature extraction, an analysis of autocorrelation among independent variables, and ranked the correlation between independent and dependent variables.

Next, we introduced the concept of "potential energy" and developed a potential energy prediction model based on the XGBoost algorithm. We quantified the relationship between independent variables and potential energy and visualized this quantitative relationship using visualization techniques.

Subsequently,we simulated random match scenarios and analyzed the impact of potencial energy on scores. By calculating the Pearson correlation coefficient, we confirmed that player scores in matches are not random but are significantly influenced by potential energy.

Furthermore, we applied Bayesian online change point detection methods to successfully identify crucial turning points in matches. We conducted correlation analyses of these change points, thereby determining the importance ranking of 18 independent variables.

For further application,we developed the Tennis Match Point Time-Convolution Model,which can predict when critical turning points are likely to occur during matches.

Finally,by analyzing data from the 2003 Wimbledon women’s singles match,we validated the wide applicability of our model and conducted comprehensive evaluations and sensitivity analyses.

Our research not only serves as a reference for future studies in related fields ,but also provides practical tools. By applying our model, one can effectively monitor player potential energy, make timely strategy adjustments, predict changes in match dynamics, and be well-prepared for them.

I believe that by implementing these datadriven strategies, we can transform our coaching into a high precision science, making us pioneers in the tennis coaching arena. This innovative approach will empower our players to reach unprecedented heights and achieve a competitive edge on the tennis court.I look forward to engaging in a dialogue to discuss these recommendations in depth and devise a roadmap for their seamless integration into our coaching philosophy.

Warm regards,
Bill