Problem Chosen	2021	Team Control Number
C	MCM Summary Sheet	2428897

Champion's Code: Unraveling the Impact of Momentum in Tennis Matches

Summary

In the 2023 Wimbledon Gentlemen's final, newcomer Carlos Alcaraz defeated Grand Slam veteran Novak Djokovic, marking the end of Djokovic's decade-long reign. This event has rekindled interest in the significance of "Momentum" in tennis competitions, suggesting that in-depth research into this phenomenon could be both intriguing and significant.

Before delving into the research questions, it was observed that the data, while relatively accurate, required **data preprocessing** due to variations in recording rules. To obtain high-quality data, the **PCA algorithm** was used for dimensionality reduction, from which data with main features was obtained.

For the first question, the research employed a **Markov chain model** to simulate and predict flow of play at the point-scoring moments, calculating the probabilities of turning the table as the match progresses. Flow of play is visualized in **Fig4**. Furthermore, a **performance quantification system**, which is based on previous studies, was established to evaluate player performance. Multivariate logistic regression confirmed the reliability of this system.

To address the second question, a momentum quantification system was established based on previous studies. However, it requires de-quantification of the weights of features. To do that, data must be processed by the **GWO-BP** neural network model. This system's **reliability was accredited** by multiple logistic regression. In addition to **comparison with other models, Random Forest model, optimized via grid search and cross-validation,** yielded satisfactory results with an **F1 score of 0.894,** which indicates momentum's significant impact in a contest.

For problem three, Random Forest prediction model is chosen**. Gain Importance methods** were used to identify the most significant features, such as **total running distance, scoring of the player and scores ahead of others. Spearman's** correlation model **re-confirmed** the finding of **K-S test** that data did not follow a normal distribution. When it comes to changes in "momentum", the **ChangeFinder algorithm** is employed to find out points of sudden changes and recommendations is given to players in combination with the overall model outcomes.

Concerning the fourth question, data from various sports, including women's tennis and table tennis, were analyzed. It is shown that the model is accurate in prediction with R2 values being 0.923, higher than between 0.75 and 0.85. Therefore, this model passed robustness tests for generalization.

At last, the model passed sensitivity analysis. A report on the role of momentum is issued for coaches to consult.

Keywords: Markov Chain, GWO-BP, Random Forest, K-S Test, ChangeFinder

Contents

1 Introduction

1.1 Problem Background

In the 2023 Wimbledon Gentlemen's final, Spanish rising star Carlos Alcaraz defeated the Grand Slam favorite, Novak Djokovic, ending Djokovic's decade-long dominance in Grand Slam tournaments. This result was in stark contrast to the initial expectation of an easy victory for Djokovic. The favored player at the start ultimately lost the match, suggesting that the concept of "Momentum" played a crucial role. Alcaraz may have fortuitously gained more momentum, highlighting the importance of studying momentum factors on the court. Understanding how to define momentum, investigating the factors influencing its shifts, and analyzing its impact among athletes are of significant importance to players, coaches, and sports departments worldwide.

1.2 Restatement of the Problem

Considering the context and constraints presented in the problem description, the following problems will be solved:

- ***Performance Model*:** Develop a model that captures the progression of tennis match considering server's advantage, enabling quantification of performance fluctuations on a point-by-point basis. This model discerns and quantifies different phases of player performance throughout the match.
- ***Momentum Identification*：** Formulate a model to assess the influence of momentum on a player's performance during a match, to determine whether it leads to patterned fluctuations in their results.
- ***Momentum Impact:*** Identify specific momentum indicators to ascertain the transfer of momentum between athletes during a competition. Investigate the strength of these influences and utilize findings to provide recommendations for subsequent matches.
- ***Model Evaluate:*** Evaluate the effectiveness of the established model and assess its performance in diverse data application scenarios to verify its generalizability.
- ***Memo:*** Compose a memo to the coach summarizing the model research and, based on findings regarding momentum, provide tactical recommendations for players' in-game strategies.

1.3 Literature Review

Just as in 2023 Carlos Alcaraz overcame an initial poor performance to ultimately defeat Novak Djokovic, victory in tennis matches is closely related to psychological pressure on the players[1], match duration, and scoring opportunities. This phenomenon closely interacts with specific events (such as aces and double faults) in prior points[2], significantly affecting larger issue analyses (Richardson P. A., et al., 1988; Moss B., 2015). Consequently, exploring these intricate relationships is not only traceable but also of significant importance.

In studying basketball, researchers and scholars have employed various models to predict and assess the performance and scoring potential of basketball players, achieving noteworthy results. Among these, the homogeneous Markov model has demonstrated strong performance[3] (Štrumbelj E., P., et al., 2012), showcasing its capability to explore the transitions and states of scoring throughout a game. Similarly, the National Football League (NFL) employs random forest methods combined with pre-match influencing factors to calculate potential victory probabilities[4], accurately predicting match outcomes (Lock D., et al., 2014). The random forest, a robust ensemble model, excels in multi-class prediction tasks. Optimizing model hyperparameters with genetic algorithms enhances the predictive or classification performance of BP neural networks[5] [6], which are also frequently used in this area of research (Zhang J., et al.,2020; Eberhart R. C., et al.,1998). Building on the efforts of our predecessors, this study integrates the advantages of various methods in modeling and solving problems encountered in the competition paper.

1.4 Our Work

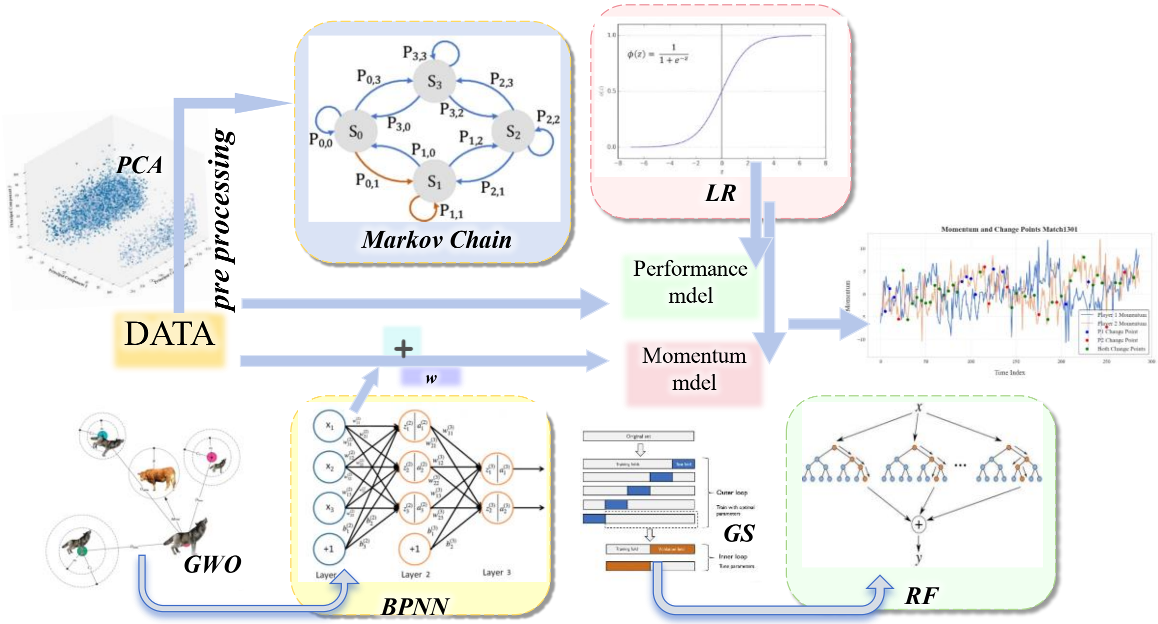


Fig 1 Our work

2 Assumptions and Justifications

To simplify the model and facilitate the proposal and implementation of a mathematical model, we have made the following assumptions with corresponding justifications:

- **Assumption 1:** It is assumed that if data classification or regression tasks can be accurately performed through scientific mathematical methods, then the momentum of athletes and fluctuations in performance, as well as success achievement, are measurable and not random.

Justification: If scores can be scientifically classified or predicted by mathematical models, they must follow certain scientific principles and regularities, making them interpretable and hence, not indeterminable.

- **Assumption 2:** It is assumed that only the conditions provided by the problem statement are considered, neglecting other factors that may influence athletes' performance and momentum during the competition.

  Justification: There are numerous factors influencing the outcome of a match. Considering all these factors over a short period and collecting data beyond what is already provided are challenging tasks. To facilitate model establishment, this assumption is made.

- **Assumption 3:** It is assumed that the data provided by the problem statement, along with data collected independently, are accurate and reliable after undergoing data preprocessing and feature engineering.

  Justification: The data provided are thoroughly audited and verified. Given that the competition process is observed by many, recording errors are generally unlikely, though recording rules may influence. However, after data preprocessing, such data can be considered accurate and reliable. The probability of computational errors is minimal, and for the feasibility of the problem, the impact of computational errors is not considered.

3 Notations

The key mathematical notations used in this paper are listed in Table 1.

Table 1: Notations used in this paper

Symbol	Description
$\Omega(f)$	Regular term
g_i	The first derivative of a function
h_i	The second derivative of a function
$\mathrm{cov}(X, Y)$	Covariance
σ	Standard deviation
γ	Number of leaf nodes
λ	Leaf node ratio
T	Number of leaf nodes

4 Data preprocessing and Feature engineering

Upon reviewing the entire dataset and gaining a comprehensive understanding of its structure, we undertake preprocessing steps and data feature engineering to facilitate the problem-solving process and ensure accurate solutions.

4.1 Data Preprocessing

Initially, upon examination of the match score data, it is noted that scores are not consistently recorded using traditional tennis scoring terms (15, 30, 40, AD). Instead, several instances utilize numerical values from 1 to 9. Drawing on knowledge of tennis scoring rules, these numerically represented points are converted into the conventional scoring system (i.e., 1-9 to 15, 30, 40, AD). Furthermore, due to the computational incompatibility of text-based representations like 'AD', this term is replaced by an equivalent numeric value, specifically 45, for seamless integration in mathematical calculations. Additional transformations and treatments are applied to other relevant data as required for the specific problem-solving context. Such as experiments commonly used to standardize the treatment.

Standardize the data to have zero mean and unit variance. This is done for each feature independently

$$Z = \frac{X-\mu}{\alpha} \tag{4-1}$$

where Z is the standardized data, X is the original data, μ is the mean, and σ is the standard deviation.

4.2 PCA data dimensionality reduction

As shown in Figure 1 below, the following figure shows the flowchart of our PCA algorithm, which first performs empirical judgment to determine those characteristic factor indicators are reliable, and then performs PCA data dimensionality reduction processing.

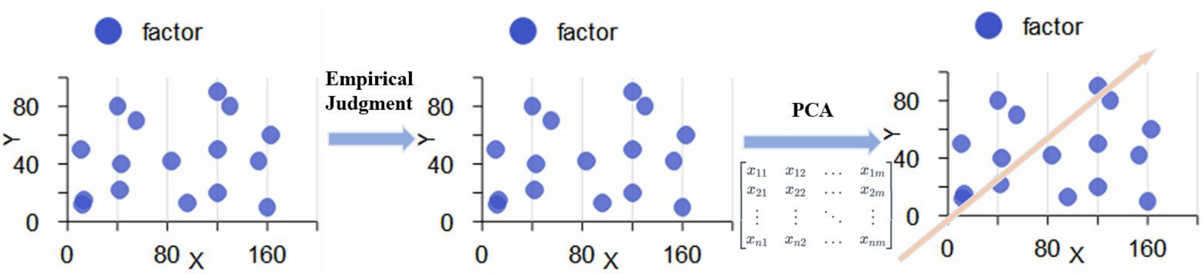


Fig 2 PCA data dimensionality reduction

Let X be the data matrix with dimensions $m \times n$, where m is the number of samples (matches) and n is the number of features (set count, game victories, running distance, scores). And standardize the data to have zero mean and unit variance as (4-1). This is done for each feature independently.

Step1: Eigenvalue Decomposition: Calculate the covariance matrix C of the standardized data. Perform eigenvalue decomposition on the covariance matrix C

$$C = \frac{1}{m} Z^T Z = V \Lambda V^T \tag{4-2}$$

where Z is the standardized data, X is the original data, μ is the mean, and σ is the standard deviation. V is the matrix of eigenvectors, and Λ is the diagonal matrix of eigenvalues.

Step2: Cumulative Explained Variance: The explained variance for each principal component is given by the eigenvalues. Calculate the cumulative explained variance to decide on the number of principal components to retain

$$CEV = \sum_{i=1}^{k} \frac{\lambda_i}{\sum_{i=1}^{n} \lambda_i} \tag{4-3}$$

where CEV Cumulative Explained Variance

Step3: Choose the number of principal components (k) that captures a sufficiently high percentage of the total variance, e.g., 95%.

Step4: Project the original data onto the selected principal components.

$$Y = ZV_k \tag{4-4}$$

where Y is the reduced-dimensional data, Z is the standardized data, and V_k contains the first k eigenvectors.

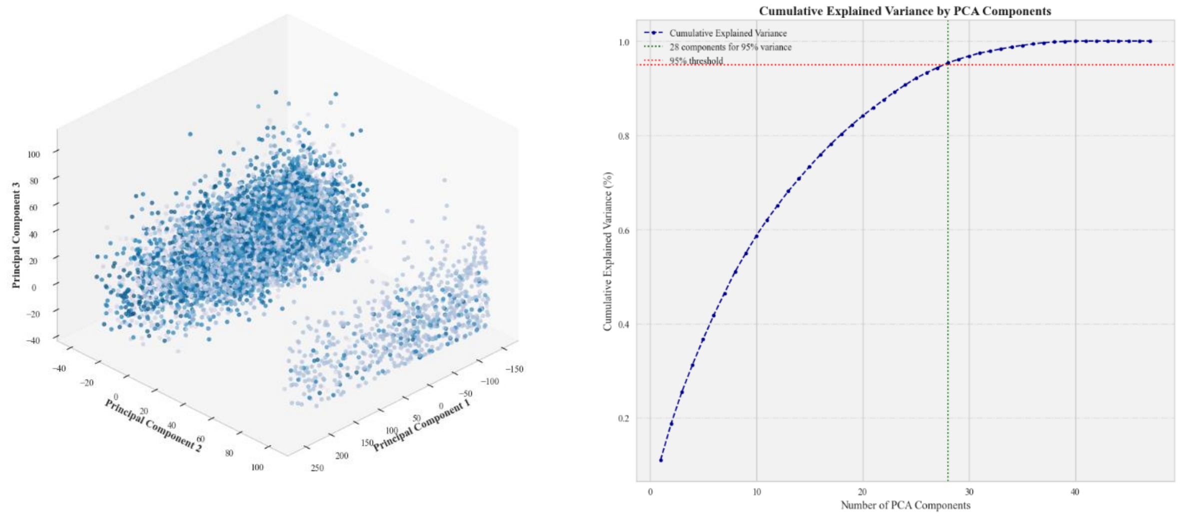


Fig 3 PCA Data Dimensionality Reduction Visualization

Through dimensionality reduction, we obtain dominant dimensions that significantly influence performance metrics such as athletes' scores, opponents' scoring patterns, number of matches won in the current tournament, and running distances, among others

5 Data preprocessing and Feature engineering

To investigate and visualize the progression of match scores, we employ a Markov chain model to compute transition probabilities based on in-game scenarios. Developing a Performance Model that quantifies actual performance, elucidating how athletes improve over time.

5.1 Markov Chain Modeling and Solution

5.1.1 Establishment of the Markov Chain Model

In tennis matches, a Markov chain model is employed to describe the process of match set score transitions. The calculation involves determining the probability of transitioning from one set-score state to another.

Step1: Define the State Space S, Each state in S is represented as $(S1, S2)$, where $(S1, S2)$ denote the number of sets won by Player 1 and Player 2, respectively. The possible states are given by $S = \{(0,0), (0,1), (1,0), (1,1), \dots\}$.

Step2: Definition of State Vector Each element in a probability vector

$$X^{(n)} = \left(x_1^{(n)} x_2^{(n)} \cdots x_k^{(n)}\right) \tag{5-1}$$

Step3: Definition of State Vector Each element in a probability vector represents a probability, with the sum of all elements equaling unity. There are k possible states in the scoring system. Each element in the vector indicates the probability of being in the i-th state at the n-th observation.

Step4: We use the following transition probability matrix X to represent the probabilities of moving from one match score state to another: Transition Probability Matrix $P_{ij}(i, j = 1,2, \dots, k)$ signifies the probability that the match score was in state i before this observation and is now observed to be in state j.

$$X = \begin{pmatrix} p_{11} & \cdots & p_{1k} \\ \vdots & \ddots & \vdots \\ p_{k1} & \cdots & p_{kk} \end{pmatrix} \tag{5-2}$$

Step5: According to the Markov property of no aftereffect and iteration::

$$X^{n+1} = X^0 P^n \tag{5-3}$$

Iteration Property The transition of the match score status at any point solely relies on the previous score status.

5.1.2 Solution of the Markov Chain Model

We construct a transition probability matrix X based on multiple match records for both athletes. Using a Markov chain method, we analyze the hidden patterns in the changes between states. A visual representation of these transitions—a precedence diagram—is provided, which shows the probabilities of different score outcomes and how the match might unfold when Players 1 and 2 begin with a score of (0,0) in their next contest.[i]

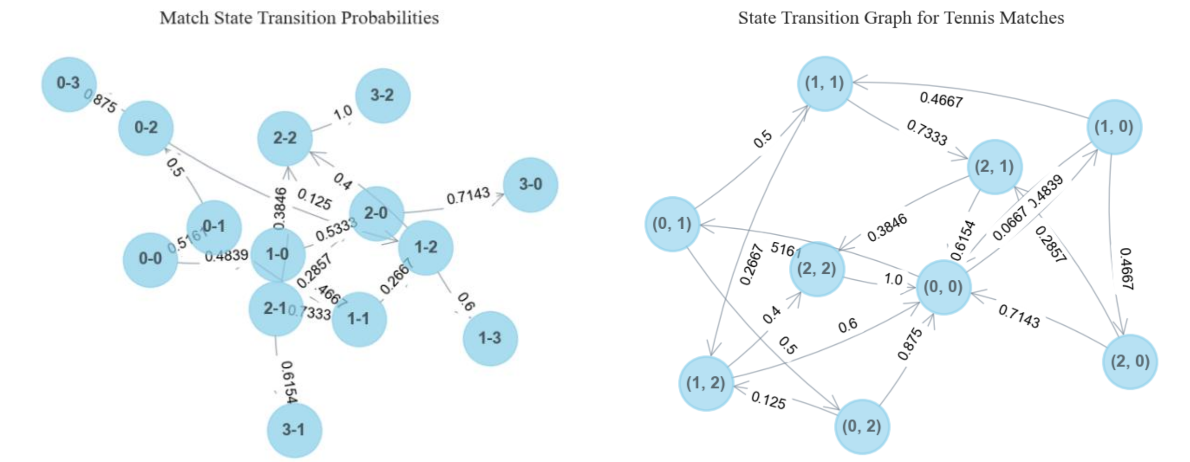


Fig. 4 Game flow charts and associated transition probabilities of distinct conditions

Fig. 3 illustrates the game flow charts and associated transition probabilities under distinct conditions, revealing a marked change in score probabilities when the impact of serving is taken into account versus neglected. This finding underscores the relevance of this factor in the resolution of the addressed issues, hence its inclusion in our approach to ensuing questions. Our experimental investigation leverages data from match 2023-wimbledon-1301. And the chart above is of the left is a chart of the results of a first mover advantage, the right has no first mover seeking advantage, a first mover advantage does affect the game and makes it easier for a player to succeed.

5.2 Performance Modeling and Solution

Step1: To better observe and evaluate athletes' performance in this match, drawing upon literature [1][2][5], we aim to establish a quantitative standard for processing the athletes' game situations. We define an athlete's performance score as being related to factors such as the number of games won, points scored, errors committed, serve quality, and running distance covered. The calculation formula for the performance score is as follows:

$$Performance = PG + PS + PP + PA + PF + PE - PD \qquad (5\text{-}4)$$

where, PG denotes the number of games won by the athlete, PS represents the total points scored, PA signifies whether the athlete serves first, PP indicates points scored when serving first, PE denotes double faults, and PD stands for the athlete's total running distance.

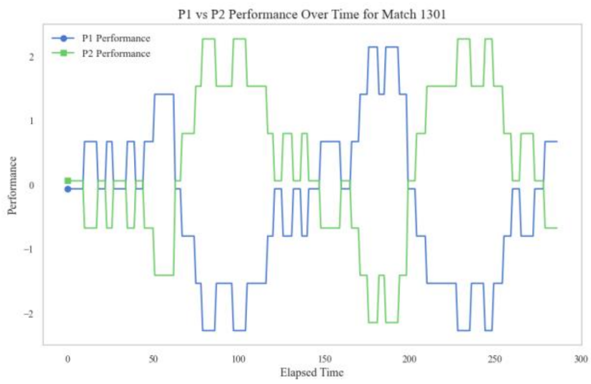
Step2: To investigate the relationship between an athlete's performance and their current scoring situation, we introduce the following formula

$$p_{i-j}\_games = p_i\_games - p_j\_games \qquad (5\text{-}5)$$

where, within this context, $(i, j = 1,2)$, refer to Player 1 and Player 2 respectively, and P_games signifies the number of games won by player$i$ up to the current point in the match.

5.2.1 Solution of the Performance Model

We have depicted the respective changes in scores and the progression of game numbers in the following illustration, data from match 2023-wimbledon-1301:



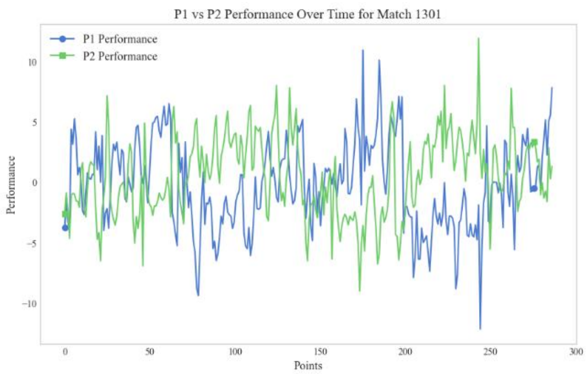


Fig.4 Difference in Player's Game Counts **Fig. 5 Changes in Performance Metrics**

Figure 5 shows the variation in the number of games played by a player, while Figure 6 presents fluctuations in performance scores. In this analysis based on data from the 2023-wimbledon-1301 event, we find a strong association between changes in performance metrics and the player's standing in terms of game victories. Specifically, an increase in performance scores is typically accompanied by an increase in the number of games won.

5.3 Multivariate Regression Model Validation

Given that the observed graphical correlation in performance may not guarantee theoretical reliability, we establish a multiple logistic regression model to examine if there exists a significant link the performance scores of players and their current point outcomes.

5.3.1 Establishment of the Multivariate Regression Model

The logistic regression equation for the probability of the outcome (scoring situation)， and the probability of each scoring category is calculated using the softmax function

$$P(Y_i = k) = \frac{exp(\beta_{0k} + \beta_1 X_{1i} + \cdots + \beta_p X_{pi})}{\sum_{i=1}^{k} exp(\beta_{0k} + \beta_1 X_{1i} + \cdots + \beta_p X_{pi})} \tag{5-6}$$

where, K is the total number of scoring categories. Y_i : Scoring situation for the i-th point. k : Specific scoring category. β_{0k} : Intercept for category k. $\beta_1, \beta_2, \dots, \beta_p$: Coefficients associated with predictor variables. $X_{1i}, X_{2i}, \dots, X_{pi}$: Predictor variables for the i-th point. $P(Y_i = k)$: Probability of scoring category k for the i-th point. $P(Y_i = K)$: Total probability across all scoring categories for the i-th point.

5.3.2 Validation of the Multivariate Regression Model

Multivariate regression model results show that p1_performance has a statistically significant positive relationship **(coefficient: 0.2062, p-value ≈ 0)** with point_victor, while p2_performance exhibits a significant negative correlation **(coefficient: -0.1915, p-value ≈ 0)**. Therefore, both variables have a substantial impact on predicting point_victor.Additionally, the classification evaluation metrics are computed as shown in Table 2 below.

Table2 Multiple regression modeling results

	Accuracy	Recall	F1 Score	Precision

Value	0.678	0.695	0.688	0.681

The ROC curve in the figure exhibits an AUC of 0.75, suggesting substantial correlation. This strong relationship is substantiated by examining and validating it against relevant assessment criteria and statistical significance.

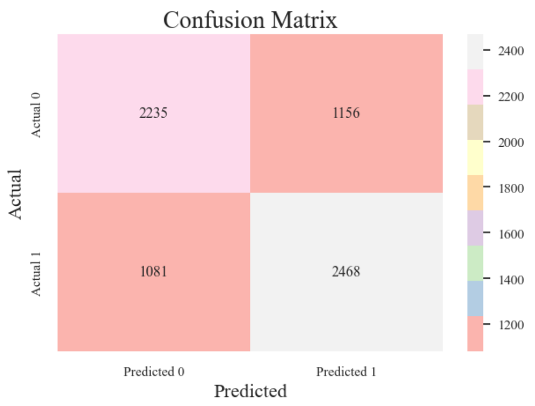


Fig. 6 Confusion Matrix

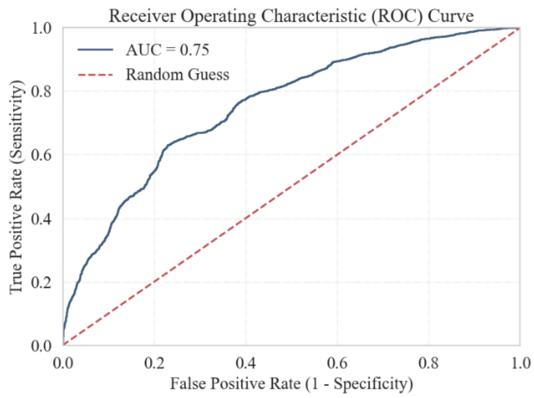


Fig. 7 ROC

6 Validation and Analysis of Momentum's Reliability

To investigate the efficacy of Momentum, we initially conducted extensive literature review and found that experts have previously defined it in this context. We thus referred to their methods to quantitatively define Momentum. However, determining the weights of its influencing factors was identified as a necessity. To address this issue, we constructed a Grey Wolf Optimized Backpropagation Neural Network model. The model is defined based on the output weights from the first layer of the neural network. Subsequently, through comparative analysis of numerous models, we sought out and analyzed the optimal model for analyzing Momentum.

6.1 GWO-BP Modeling and Solution

6.1.1 Establishment of the GWO-BP Model

The following figure illustrates our establishment of a BPNN model for determining the weights affecting Momentum, where GWO is used to tune the BPNN's hyperparameters.

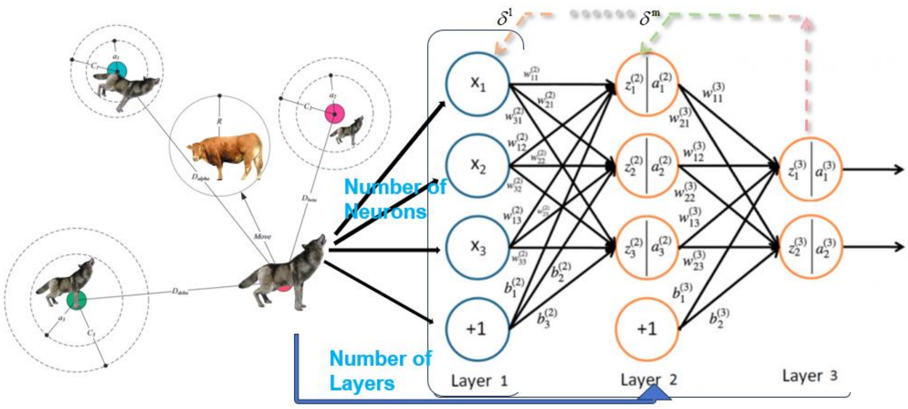


Fig. 8 Structure of the GWO-BP model

We aim to study the weight values of various features in tennis matches, such as serving first, the previous game's scoring situation, and the player's physical condition. To achieve this, we propose optimizing a BP (Backpropagation) neural network model using the Grey Wolf Optimization (GWO) algorithm.

PART1： Target Output and Mean Squared Error

Step1: Target Output: The target output for the i-th training sample is denoted as d_i.

Step2: Mean Squared Error (MSE): The objective function is defined as the MSE between the neural network output and the target output:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \tag{6-1}$$

Where: N is the total number of training samples

PART2： BP Neural Network Model:

Step1: Input Layer: The input layer of the BP neural network consists of features related to tennis match characteristics. Let X_i represent the feature vector for the i-th training sample.

Step2: Hidden Layers: The neural network may have one or more hidden layers, each with a set of weights and biases denoted as $W^{(l)}$ and $b^{(l)}$ for the l-th layer.

Step3: Activation Function: The activation function for the l-th layer is $\sigma^{(l)}(\cdot)$.

Step4: Output Layer: The output layer produces the network's final output y_i

PART3：Grey Wolf Optimization (GWO):

Step1: Search Agents: Let A represent the grey wolf population, where each grey wolf is a potential solution in the weight space.

Step2: Objective Function (Fitness):The fitness function is the negative of the MSE, as GWO seeks to maximize the fitness.

Step3: Grey Wolf Position Update: The position of each grey wolf is updated iteratively using the following formula:

$$X(t+1) = X(t) - A \bullet Fitness \bullet Rand(\ ) \tag{6-2}$$

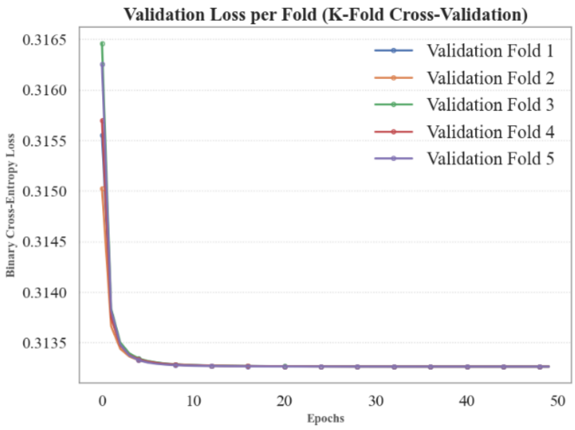
PART4： Furthermore, to enhance the model's performance, we employed k-fold cross-validation and a cross-entropy loss function. The formula for the cross-entropy loss is given as follows:

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -[y_i \cdot \log(p_i) + (1-y_i)\cdot \log(1-p_i)] \tag{6-3}$$

where, p and $1-p$ represent the predicted probabilities for each class, where p_i denotes the probability of sample i being classified as positive, and y_i represents the true label of sample i (1 for the positive class and 0 for the negative class). L signifies the loss function being computed.

6.1.2 Solution of theGWO-BP Mode

The resulting cross-entropy loss and k-fold cross-validation results for the obtained model are depicted in the following figure.



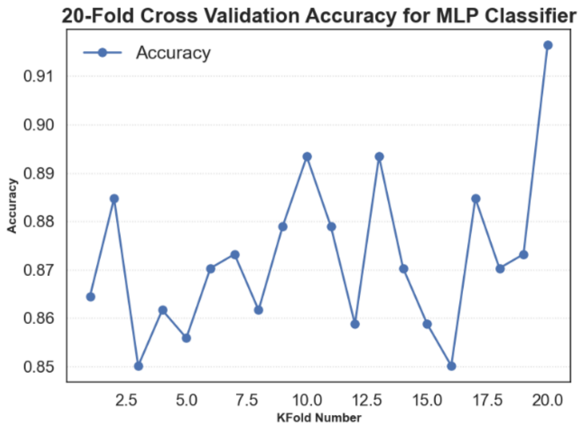


Fig. 9 Validation Loss per Fold **Fig. 10 RO20-Fold Cross ValidationC**

As shown in the above figure, we achieved an average accuracy rate of 89.7% across k-fold cross-validation, with a peak accuracy reaching 92.6%. The model performance is notably strong. Upon incorporating cross-entropy loss, the overall accuracy increased significantly to 94%. Consequently, we obtained the output weights as displayed in the following table.

Table3 Table of Influencing Factor Weights

Label	Weight	Label	Weight
p1_ace	3.811	p2_fault	0.674
p2_score	2.218	p2_ace	0.355
victor_value	2.063	p2_games	-0.501
p1_err	1.781	P2server	-0.641
p2-1_score	1.733	p2_distance	-0.809
p1_distance	1.707	p2_distance_sum	-1.082
p1_distance_sum	1.633	p1_fault	-1.183
p1_score	1.607	P1server	-1.368
game_victor	1.563	p2-1_games	-2.072
p1-2_games	1.454	p1-2_score	-2.401
p1_games	0.941	p2_err	-3.974

To investigate and visualize the progression of match scores, we employ a Markov chain model to compute transition probabilities based on in-game scenarios. Developing a Performance Model that quantifies actual performance, elucidating how athletes improve over time.

6.2 Momentum Modeling and Validation

Step1: To better observe and evaluate athletes' Momentum e in this match, drawing upon literature [3][4][6], we aim to establish a quantitative standard for processing the athletes' game situations. We define an athlete's Momentum score as being related to factors such as the number of games won, points scored, errors committed, serve quality, and running distance covered. The calculation formula for the Momentum score is as follows:

$$Momentum = \sum_{i=1}^{n} w_i x_i \tag{6-4}$$

where, w represents the optimized output weights from the backpropagation neural network, while x denotes the corresponding influencing factors.

Step2: The test validation using the above equation (5-4)'s yields the following results

Multivariate regression model results show that p1_ momentum has a statistically significant positive relationship **(coefficient: 0.2132, p-value ≈ 0)** with point_victor, while p2_ momentum exhibits a significant negative correlation **(coefficient: -0.1898, p-value ≈ 0)**. Therefore, both variables have a substantial impact on predicting point_victor.Additionally, the classification evaluation metrics are computed as shown in Table 3 below.

Table4 Multiple regression modeling results for Momentum

	Accuracy	**Recall**	**F1 Score**	**Precision**
Value	0.778	0.785	0.798	0.782

As shown in the above table, we observed that the values for Accuracy, Recall, F1 Score, and Precision are all notably high, exceeding 0.77, indicating a strong modeling effect when coupled with the significance analysis.

6.3 Evaluation Momentum Of RF Model

6.3.1 Preparation of the model

In our quest to identify the most suitable prediction model, we compared and analyzed five commonly used mathematical models: Random Forest, Support Vector Machines (SVM), Multilayer Perceptron (MLP), Backpropagation (BP), and Logistic Regression. Based on the existing data, we computed their Accuracy, F1-Score, Recall, Precision, Sensitivity, and Specificity to evaluate their performance, as presented in Table 5 below.

Table5 Comparison of Multiple Models for Momentum Analysis

Items	R F	SVM	BP	MLP	LR
Accuracy	0.867	0.856	0.853	0.821	0.827
F1-Score	0.868	0.857	0..856	0.819	0.831
Recall	0.869	0.853	0.854	0.818	0.825
Precision	0.865	0.840	0.851	0.817	0.821
Sensitivity	0.867	0.857	0.848	0.824	0.854
Specificity	0.859	0.822	0.833	0.818	0.801

Upon examining the multi-model comparison in Table 5, we discovered that the Random Forest model demonstrated the best performance in predicting point_victor outcomes, achieving an Accuracy of 0.867 and an equally impressive F1-Score of 0.868. Furthermore, other metrics were also favorable. Consequently, we selected Random Forest as our primary model due to its superior predictive capabilities.

6.3.2 Establishment of the RF Model

PART1: To study the impact of player momentum on tennis match scores, a Random Forest (RF) model will be optimized using grid search. This involves understanding the mathematical representation of the Random Forest model and its optimization.

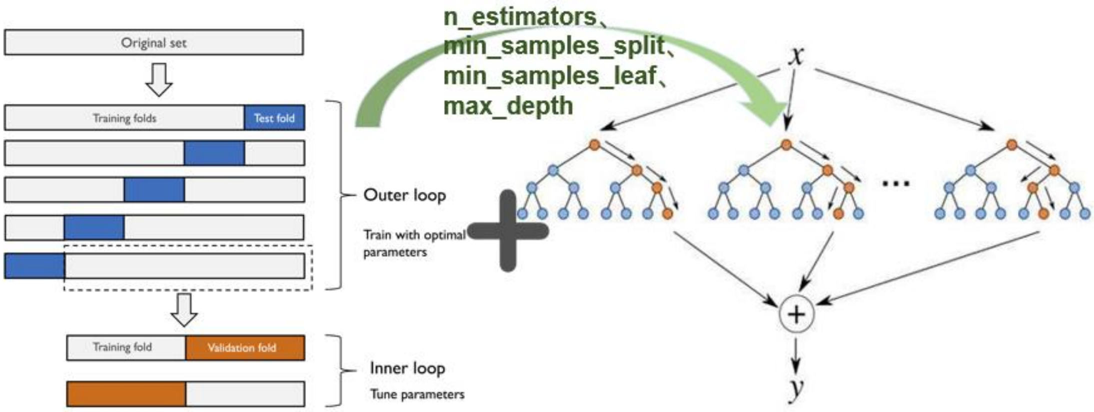


Fig. 11 Grid search facilitates optimization of RF

Step1: Ensemble of Decision Trees: A Random Forest is an ensemble learning method that combines multiple decision trees to enhance predictive performance.

$$\mathrm{RF}(X)=\frac{1}{N}\sum_{i=1}^{N}f_i(X) \tag{6-5}$$

where, $\mathrm{RF}(X)$: Random Forest prediction for input X. N : Number of decision trees in the forest. $f_i(X)$: Prediction of the i-th decision tree.

Step2: Grid Search Optimization: Define a hyperparameter grid for the Random Forest model, denoted as G, Perform grid search to find the optimal hyperparameters that maximize model performance.

$$\hat{\theta}=\arg\max_{\theta\in G}\mathrm{Performance}(\mathrm{RF}(X;\theta)) \tag{6-6}$$

where, G is Number of Trees, Tree Depth, …, Other Hyperparameters. $\mathrm{RF}(X)$: Random Forest prediction for input X. N : Number of decision trees in the forest. $f_i(X)$: Prediction of the i-th decision tree.

PART2: After training the Random Forest model, the feature importance can be computed to understand the contribution of each factor, including player momentum. The feature importance is typically derived from the decrease in impurity or information gain associated with each feature. Here is the formula and explanation:

6.3.3 Solution of the RF Model

The model is prepared by calculating the importance of its features by means of a random forest, In this study, we performed classification of the point-victor in the current match based on the available features. The performance of the Random Forest classifier, following grid search and cross-validation, is presented in the table below.

Table6 RF for Momentum Classification evaluation indicators

	Accuracy	Recall	F1 Score	Precision
Value	0.921	0.897	0.912	0.906

We observe that after grid search optimization and cross-validation, the Random Forest model exhibits excellent results, further substantiating that **Momentum is not random but rather observable and can be effectively utilized for predictive purposes.**

7 Momentum Fluctuation and Transition Detection

To investigate the impact dynamics and transition patterns of the indicators, we initially predict the fluctuating data using the Random Forest model previously discussed in this paper. Alongside, we utilize a correlation model and perform D'Agostino's K-squared Test for normality and significance to discern the most influential factors. Subsequently, we establish a ChangeFinder-based framework to identify points of significant change.

7.1 Establishment of Prediction and Correlation Model

7.1.1 Establishment and Solution of Prediction Model

PART1: Establishment Prediction Model

For the predictive model, we employ the Random Forest model optimized by grid search and cross-validation from the previous section to predict Momentum for both athletes across 1301 matches. We also calculate feature importance using the following formula:

$$GainImportance\ (X_i) = \sum_{(t=1)}^{N=trees} (\frac{N_t}{N} \bullet variancedecreaset_t) \tag{7-1}$$

where: variance decrease $_t$: The decrease in variance achieved by splitting based on X_i in the t-th tree.

PART2: Solution Prediction Model

The obtained results include radar plots of predicted outcomes and feature importances, as well as a table for evaluating the prediction performance.

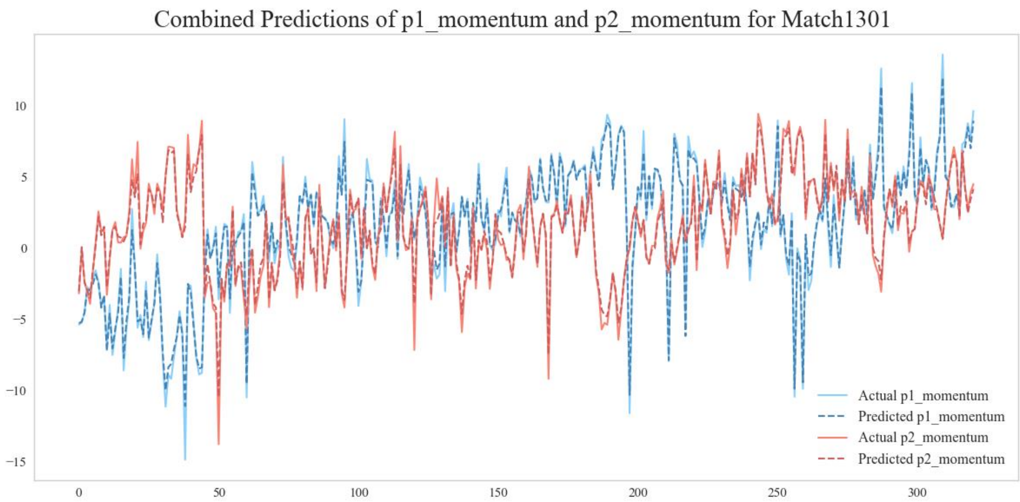
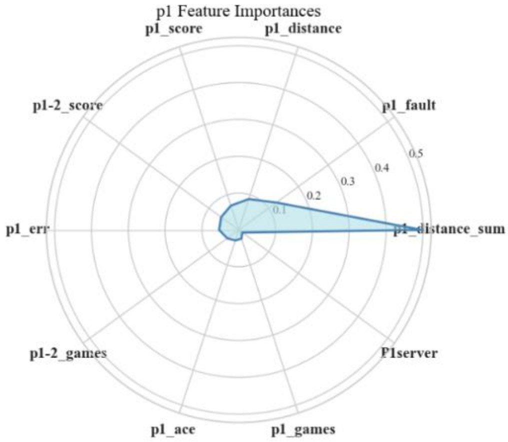


Fig. 12 RF for Momentum Prediction results

Table7 RF for Momentum Projected evaluation indicators

	R^2	RMSE	MAE	Explained Variance
p1_momentum	0.951	0.767	0.675	0.957
P2_momentum	0.937	0.797	0.724	0.944

In the figure above, our Random Forest model's prediction evaluation metrics demonstrate a good fit quality. Further, examining the feature importance data in the subsequent section, we identify that the primary influencing factors are the athletes' running distance, lead score, and number of match victories.



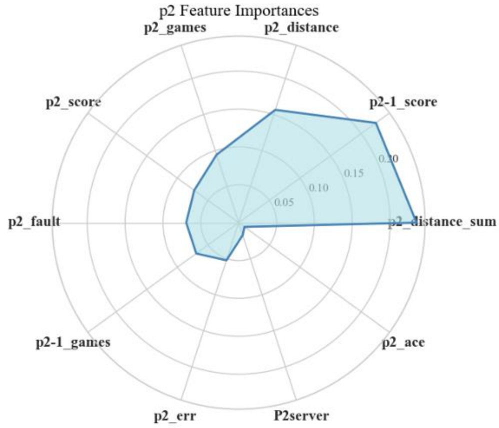


Fig. 13 Radar chart of importance of features

7.1.2 Establishment and Solution of Correlation Model

We directly examine correlations by applying correlation analysis to the model, and to discern the model's data distribution characteristics, we initially utilize the Kolmogorov-Smirnov Test for assessing normality. The Kolmogorov-Smirnov (K-S) test is employed to assess whether metrics such as a player's running distance, score, and momentum in a tennis match follow a normal distribution.

PART1: Kolmogorov-Smirnov Test for Normality

The K-S test involves comparing the Empirical Cumulative Distribution Function (ECDF) of the data ($F_n(x)$) with the Cumulative Distribution Function (CDF) of the standard normal distribution ($F(x)$).

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \le x) \tag{7-2}$$

where, X_i is the i-th observed data point. $I(\cdot)$ is the indicator function. Test Statistic (D) : The test statistic D represents the maximum vertical deviation between the ECDF and the CDF.

The test statistic D represents the maximum vertical deviation between the ECDF and the CDF.

$$D = \max\left(\sup_x |F_n(x) - F(x)|\right) \tag{7-3}$$

where, $supx$ denotes the supremum over all possible values of x.

Ultimately, our examination revealed that the p-value is close to zero, indicating significant differences among the feature data. Consequently, the dataset does **not conform to a normal distribution.**

PART2: Spearman Rank Correlation Coefficient Model

Spearman's rank correlation coefficient is as follows

$$r_s = 1 - \frac{6\sum_{i=1}^{n} D_i^2}{n(n^2-1)} \tag{7-4}$$

where, ρ : Spearman rank correlation coefficient. d_i : The difference between the ranks of corresponding observations. n : The number of observations.

Through our calculations, we obtained the following correlation coefficient heatmap:

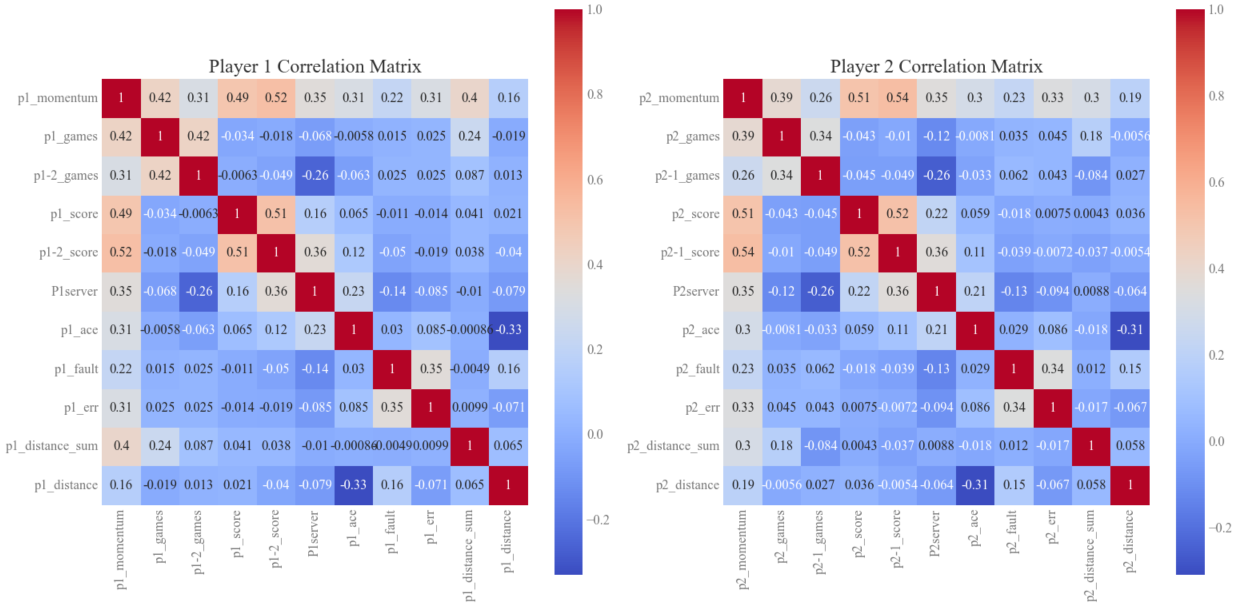


Fig. 14 Heatmap for Correlation Analysis

The heatmap clearly shows that the most impactful factors on the Momentum of both athletes share similarities. Within the Spearman correlation framework, an athlete's personal score and the score differential vis-à-vis their opponent display the strongest correlations. Moreover, substantial associations are found between the athletes' running distances and the number of games they win.

7.2 ChangeFinder Modeling and Solution

7.2.1 Establishment of ChangeFinder Model

The ChangeFinder model is designed to detect abrupt changes or fluctuations in time series data. The mathematical formula for the ChangeFinder algorithm involves comparing the observed sequence with its expected value based on a certain window size. The ChangeFinder score (CF) at each time point is calculated using the following formula:

$$CF_t = \frac{1}{2}\left(\frac{(x_t - m_t)^2}{s_t^2} + \log\left(\frac{s_t^2}{\sigma^2}\right)\right) \tag{7-5}$$

Where, CF_t : ChangeFinder score at time t. x_t : Observed value at time t. m_t : Expected value at time t (typically the moving average). s_t^2 :Variance of the observed values within a certain window. σ^2: Threshold parameter.

7.2.2 Solution of ChangeFinder Model

We employed the ChangeFinder Model to identify fluctuations in Momentum for the two athletes, generating the subsequent graphical representation of the results.

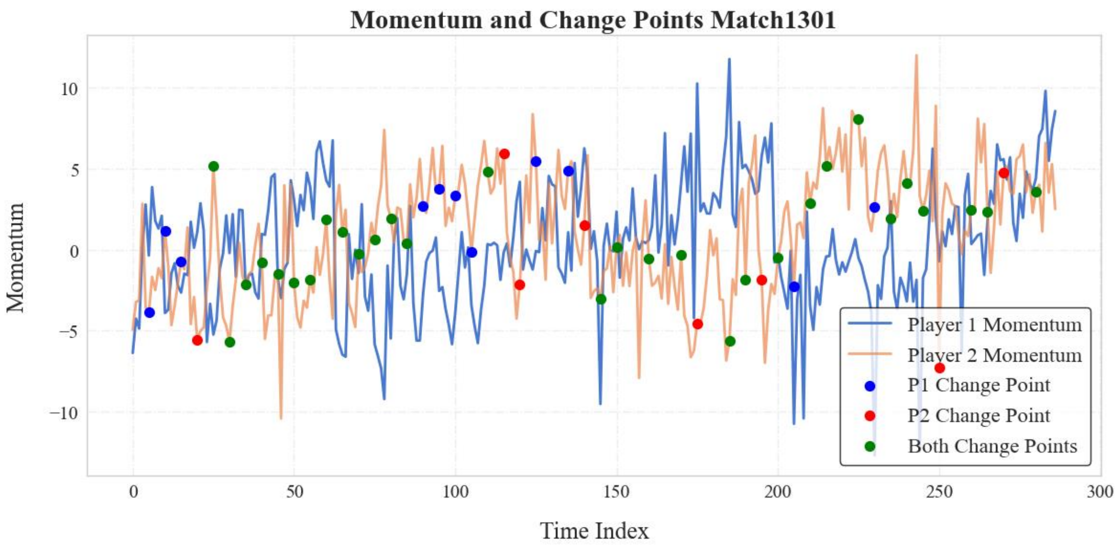


Fig. 15 "Fluctuation Detection Results Graph" Translation:

From the figures obtained above, particularly in reference to Fig. 4, Fig. 5, we observe that the fluctuations in Momentum are quite similar. Our findings from earlier sections suggest that often both athletes transition together from one Momentum state to a higher or lower one; however, there are instances where one athlete's status changes while the other experiences an opposing shift.

7.3 Countermeasures and Recommendations

Recommendation 1: Recognize your own Momentum patterns and capitalize on high-Momentum periods by securing more points, while minimizing losses during low-Momentum phases.

Recommendation 2: Minimize your overall running distance during gameplay to conserve energy and strategically induce greater movement from your opponent to potentially exhaust their stamina.

Recommendation 3: Maintain a stable mindset throughout the match, avoiding complacency when leading and despondency when trailing, akin to how Carlos Alcaraz may secure victory with a consistent approach.

Recommendation 4: Seize the advantage provided by serving first, leveraging this opportunity to set the pace and control the game.

8 Assessment of Model Transfer Capability

We tested the robustness of our model through several comparative trials using diverse datasets: Dataset1 and Dataset2, consisting of three and five games; Dataset3 from tennis data on GitHub[1]; volleyball matches[2] (Dataset4), and table tennis[3] statistics (Dataset5). Although there was a reduction in performance across these tests, our model retained relatively strong results, suggesting its potential for broader applicability.

Table 8 Comparison of different data sources

Items	Dataset1	Dataset2	Dataset3	Dataset4	Dataset5
R^2	0.947	0.932	0.853	0.821	0.874
RMSE	0.675	0.685	0.726	0.839	0.931
MAE	0.472	0.486	0.567	0.858	1.541

9 Sensitivity Analysis

In this paper, we mainly use the random forest model optimized with cross-loss entropy function, for this purpose we use to change the number of trees to test the sensitivity of our model, i.e., we use RMSE, MAE, and R^2, set the initial trees to 100, and change them up and down by 3% and 5%, respectively, and compute their regression performance metrics, respectively. The bp neural network was also used, setting the initial number of hidden layer neurons to 10, which was set to 9, 11, 7, and 13, respectively.

Table 9 Random forest and bp neural network sensitivity analysis

	RF Model					BP Model					
	95	97	100	103	105		7	9	10	11	13
RMES	0.69	0.64	0.65	0.67	0.71	F1	0.90	0.91	0.92	0.91	0.89
MAE	0.51	0.49	0.47	0.50	0.52	Acc	0.89	0.91	0.91	0.90	0.89
R2	0.92	0.93	0.93	0.92	0.91	Recall	0.89	0.90	0.91	0.90	0.90

Through the table we find that the model will be changed by changing its parameters, but the magnitude of the change is small, and the sensitivity of the model in this paper is high.

10 Model Evaluation and Further Discussion

10.1 Strengths

✧ **The model in this paper has a strong universality,** the use of the model in a variety of tasks to deal with that achieved good results, and are in line with the actual problem of the

[1] https://github.com/JeffSackmann/ This link uses women's tennis competition data
[2] https://www.sofascore.com/zh/volleyball This link uses volleyball competition data
[3] https://www.sofascore.com/table-tennis This link uses table tennis competition data

use of algorithmic models, such as the use of the random forest model is not only to achieve the prediction of the classification of the use of multi-effects and better, but also in the correlation of the relationship between the part of the importance of the study through the characteristics of the examination of the importance of the study.

- ✧ **This paper model has a certain degree of innovation**, in solving the problem of fluctuations in this paper using the mutation point detection algorithm, in solving the problem of the definition of Momentum, combined with the bp neural network model of the output weight to be calculated through the Momentum evaluation system.
- ✧ **The model in this paper has good robustness**. The choice of parameters is crucial for the performance of machine learning, this paper carries out sensitivity analysis in the analysis and testing of the model, and passes the sensitivity analysis and achieves good results on other datasets.

10.2 Weaknesses

- ✧ **Gray wolf optimization algorithms are prone to local optimal solutions**, especially in high dimensional and complex problems. The news of gray wolf hunting is guided by local and global optimal solutions, and when particles are concentrated in a local region, they may miss better global optimal solutions.
- ✧ **Computationally expensive.** Since this paper builds multiple models and compares their performance, and uses an exhaustive type of algorithm such as grid search, the computation time is relatively long, which increases the computational cost.。

10.3 Further Discussion

In the future our model will be in the interpretability to change the machine, such as asking what to use to establish the performance Performance system model, why go to establish this Momentum evaluation system, and then we will further study the more efficient model, such as XGBoost and other new integrated models, the study has not been affected by our model.

And we will promote our model to other areas of competition research, not only ball games, such as some recent e-sports games, other similar chess and other confrontation games, and this is only a singles game, later will also study whether it can be multiplayer team competition whether there is a generalization.。

11 Conclusion

In this model building process, data preprocessing is very effective and makes our later model processing simplified. In this paper, the model uses Markov chain to calculate the transfer state to get the race process in the study of the transfer state of the race process, through Performance and Momentum model through some machine learning algorithms to classify the prediction, and get a good evaluation of the indicators of the data, to get a good performance, and optimize the use of each machine learning model to make the model Effective performance is better, and finally in the fluctuation analysis innovative use of changefinder model to find out the mutation point for corresponding research.。

12 Memorandum

MEMO

To: Tennis Coach **From:** Team# 2428897

Subject: Research Results **Date:** February 6th, 2024

Dear Coach:

I consider it a huge honor to be able to offer you this letter to write! We'll then provide you with some advice based on the forecasts from our model.

Using Markov Chain for matches

We use the Markov model, the purpose of this model on top of machine learning is to be able to calculate the transfer state of the process, then we can calculate the process of our game athletes to draw said player's process of the game the state of the game to determine whether he can win the game, this model is not amazing? Maybe you don't believe

Machine Learning Predictive Judgment and Evaluation

We next use a variety of advanced machine learning algorithms, which in order to make you believe that the work we do, the ball player's state performance and Momentum can be predicted, is useful, for this reason we first use, a professor's research, the impact of the factors and his weights cumulative construction of Momentum evaluation system, through artificial neural network algorithms to calculate the weights and then establish its system and verified it by multivariate logistic regression. We found that our model is reliable, after that we analyzed and predicted the prediction of the game winners and losers, the importance of their influencing factors and the relationship between the influencing factors analysis， We've had great results, so let's use some of our metrics to show how well we're doing。

Table Regression and Classification Evaluation indicators

Regression		Classification	
RMES	0.65	F1	0.92
MAE	0.47	Acc	0.91
R2	0.93	Recall	0.91

In the above table we find that the modeling is really good and it is possible to trust the results of our modeling， For this reason I will also continue to give some of our suggestions

① A good mindset for player, teaching participating athletes not to be anxious by the present loss, nor to lose vigilance because of the present lead

② Teach players not to run too much to exhaust themselves, teach players to know that every athlete has their own cycle of Momentum, and sometimes they need to wait for their chance.。

③ 当 Beware of being beaten when someone else serves, and take advantage of your serve.

References

[1]. Richardson P A, Adler W, Hankes D. Game, set, match: Psychological momentum in tennis[J]. The Sport Psychologist, 1988, 2(1): 69-76.

[2]. Moss B, O'Donoghue P. Momentum in US Open men's singles tennis[J]. International Journal of Performance Analysis in Sport, 2015, 15(3): 884-896.

[3]. Štrumbelj E, Vračar P. Simulating a basketball match with a homogeneous Markov model and forecasting the outcome[J]. International Journal of Forecasting, 2012, 28(2): 532-542.

[4]. Lock D, Nettleton D. Using random forests to estimate win probability before each play of an NFL game[J]. Journal of Quantitative Analysis in Sports, 2014, 10(2): 197-205.

[5]. Zhang J, Zhao X, Wu Y, et al. Analysis and modeling of football team's collaboration mode and performance evaluation using network science and BP neural network[J]. Mathematical Problems in Engineering, 2020, 2020.

[6]. Eberhart R C. The role of genetic algorithms in neural network query-based learning and explanation facilities[C]//[Proceedings] COGANN-92: International Workshop on Combinations of Genetic Algorithms and Neural Networks. IEEE, 1992: 169-183.

13 Report of use AI

Name and version	Open AI GPT4Turbo(Jun 15,2023 version, GPT-4)
Query 1	< If there is a curve and he is fluctuating, what are the ways to detect his fluctuating state or his changing situation, how to detect his mutation situation, what are the algorithms >
Output 1	<Detecting fluctuations and mutations in a curve, which typically refers to analyzing changes or shifts in patterns within time-series data, can be approached using various algorithms. Here are some common methods: Moving Average (MA): A simple way to identify fluctuations is by calculating the moving average over a window of data points. Deviations from this average can signal changes Exponentially Weighted Moving Average (EWMA): This method assigns more weight to recent observations, making it more sensitive to detecting recent fluctuations. Autoregressive Integrated Moving Average (ARIMA): A statistical model that captures both trend and seasonality, as well as irregular fluctuations. It's used for time series forecasting and anomaly detection. Change Point Detection Algorithms: Kernel Change Point Detection: Detects abrupt changes in mean or variance. CUSUM (Cumulative Sum Control Chart): Monitors process stability by accumulating deviations from a target value. Page-Hinkley Test: Identifies when a stable system starts to change its behavior. Bayesian Online Changepoint Detection (BOCPD): Detects changes continuously as new data arrives. Wavelet Transform: This technique decomposes the signal into different frequency components, enabling the identification of localized fluctuations and transient events. Hidden Markov Models (HMM): Can detect underlying state changes that generate observable fluctuations. Long Short-Term Memory (LSTM) Networks: Deep learning models like LSTMs can learn complex patterns and can be trained to predict anomalies or changes in a sequence. DBSCAN (Density-Based Spatial Clustering of Applications with Noise): Although not specifically designed for time-series, DBSCAN can cluster similar data points together and potentially detect clusters representing different states or regimes.

	Seasonal and Trend decomposition using Loess (STL): Decomposes a time series into seasonal, trend, and residual components, allowing you to isolate and analyze fluctuations. Each algorithm has its strengths and weaknesses, and the choice depends on the specific nature of the fluctuation or mutation being sought and the characteristics of the data.>

Name and version	Open AI GPT4Turbo(Jun 15,2023 version, GPT-4)
Query 2	<---> 23 plt.xticks(angles[:-1], top_importances.index, color='grey', size=12, fontname='Times New Roman') ValueError: The number of FixedLocator locations (10), usually from a call to set_ticks, does not match the number of ticklabels (11).>
Output 2	<import matplotlib.pyplot as plt import pandas as pd import numpy as np from math import pi # importances = pd.Series([0.14, 0.13, 0.12, 0.11, 0.10, 0.09, 0.08, 0.07, 0.06, 0.05], # index=['Feature1', 'Feature2', 'Feature3', 'Feature4', 'Feature5', # 'Feature6', 'Feature7', 'Feature8', 'Feature9', 'Feature10']) num_vars = len(importances) angles = np.linspace(0, 2 * np.pi, num_vars, endpoint=False).tolist() angles += angles[:1] importances = pd.concat([importances, importances[:1]]) fig, ax = plt.subplots(figsize=(8, 8), subplot_kw=dict(polar=True)) ax.plot(angles, importances, color='blue', linewidth=2) ax.fill(angles, importances, color='blue', alpha=0.25) ax.set_xticks(angles[:-1]) ax.set_xticklabels(importances.index[:-1], fontname='Times New Roman', size=14) plt.title('Feature Importances in Random Forest Model', size=20, color='blue', fontname='Times New Roman') plt.show()>

Name and version	Open AI GPT4Turbo(Jun 15,2023 version, GPT-4)
Query 3	<--Can you help me learn a sample python code for a Bayesian mutation point detection model?>

Output 3	<import numpy as np
	import pymc3 as pm
	import matplotlib.pyplot as plt
	
	# Generate some artificial data with a mutation at position 50
	np.random.seed(42)
	data = np.concatenate((np.random.normal(0, 1, 50), np.random.normal(2, 1, 50)))
	
	# Define the model
	with pm.Model() as model:
	    # Priors
	    mutation_point = pm.DiscreteUniform('mutation_point', lower=0, upper=len(data) - 1)
	
	    # Mutation occurs before and after the mutation point
	    early_mean = pm.Normal('early_mean', mu=0, sd=10)
	    late_mean = pm.Normal('late_mean', mu=0, sd=10)
	
	    # Likelihood
	    early_data = pm.Normal('early_data', mu=early_mean, sd=1, observed=data[:mutation_point.value])
	    late_data = pm.Normal('late_data', mu=late_mean, sd=1, observed=data[mutation_point.value:])
	
	    # Sample from the posterior distribution
	    trace = pm.sample(10000, tune=2000, cores=1, return_inferencedata=True)
	
	# Get the mean posterior estimate for the mutation point
	mutation_point_posterior_mean = trace.posterior['mutation_point'].mean().item()

```
# Plot the results
plt.figure(figsize=(12, 6))
plt.plot(data, 'o', label='Data')
plt.axvline(x=mutation_point_posterior_mean, color='r', linestyle='--', linewidth=2, label='Estimated Mutation Point')
plt.xlabel('Position')
plt.ylabel('Measurement Value')
plt.legend()
plt.show()>
```