Research on match fluctuation prediction and strategy based on OFS algorithm

Yilei Wang[1, #], Chengyu Zhang[1, #], Gulinisha Yiming[2, #], Yening Yu[1, #, *]

[1]School of Civil and Transportation Engineering, Beijing University of Civil Engineering and Architecture Beijing, China, 102616

[2]School of Science, Beijing University of Civil Engineering and Architecture, Beijing, China, 102616

* Corresponding Author Email: yuyening@bucea.edu.cn

#These authors contributed equally.

Abstract. The aim of this paper is to construct a model for predicting on-field fluctuations in matches by using OFS algorithm, combined with random forest and lattice point search algorithms, and to analyse the key factors in order to improve the coach's decision-making ability in matches. Cross-validation and hyper-parameter tuning techniques were used during the model building process to find the best hyper-parameter combinations through lattice point search to optimise the model performance and generalisation ability. The study provides a detailed description of the model implementation process, including data processing, feature selection, parameter setting, model building, training and evaluation. Meanwhile, a series of targeted recommendations are proposed to address the variability of match fluctuations, including the analysis of past matches, the focus on recent performance, the assessment of psychological quality, the study of opponent's weaknesses, the flexible response to changes in matches and the maintenance of focus, with a view to helping athletes and coaches to formulate a more effective match strategy and to achieve better match results.

Keywords: OFS Algorithm, Random Forest, Grid Search.

1. Introduction

Predicting and understanding game fluctuations is an important task of common concern for both coaches and athletes in the dynamic field of sports competition. Accurately predicting changes in momentum and analysing the factors affecting the outcome of a match not only helps to improve the level of strategic decision-making during the match, but also significantly improves the overall competitive performance. Therefore, OFS (Outcome Fluctuations on the Field of Play) algorithms are particularly important in this context[1].

The OFS algorithm incorporates powerful methods such as random forest model and grid point search optimisation from machine learning techniques. Through in-depth research on feature selection, hyperparameter tuning and performance evaluation, the OFS algorithm provides us with a new perspective that enables us to understand the nature of match fluctuations more comprehensively. This understanding not only helps us to make informed decisions during the game, but also helps us to better cope with the various changes and challenges in the game.

In particular, in modern sports, the outcome of the game is often affected by many factors, including the state of the athletes, tactical arrangements, the performance of the opponent, etc. By analysing a large amount of historical data and real-time game information, OFS algorithms can help us quickly and accurately grasp the trend of the game, so that we can make timely adjustments and decisions. This is of great significance in improving the winning rate, optimising the tactical layout and improving the overall level of competition.

Overall, OFS algorithm is not only a tool, but also a way of thinking and methodology. It provides us with a new way to explore the fluctuation of the game and interpret the law of the game, which helps us to understand the nature of sports competition more comprehensively, and provides us with strong support to achieve more excellent results in the arena[2].

2. Match Swings Prediction: OFS Algorithm

We are required to build a model that can accurately predict fluctuations on the playing field. At the same time, it is important to analyse which factors have the greatest impact on predicting the outcome. Doing so will help coaches make more informed and rational decisions during the game. Ultimately, we will make relevant recommendations and strategies based on these fluctuations to improve the competitiveness and performance of the team during the game.

2.1. OFS algorithm modeling

2.1.1. Description of the OFS algorithm

The OFS algorithm comprises a combination of two key components: the random forest and the lattice search algorithm. The lattice search algorithm plays a crucial role in fine-tuning the hyper-parameters of machine learning models. Hyper-parameters are parameters that must be set prior to training the model, distinct from those learned from the training data. Grid search, a technique employed within the lattice search algorithm, involves defining a range of potential values within a specified hyper-parameter space. It then systematically explores all possible combinations within this space to identify the optimal set of hyper-parameters. This hyper-parameter space is often likened to a "lattice" due to its multidimensional nature, resembling a grid structure, as depicted in Figure 1 (left).

On the other hand, the random forest is an ensemble learning method that combines multiple decision trees to form a robust predictive model. It aggregates the predictions of individual decision trees through voting or averaging, resulting in a final prediction that enhances the overall model's performance and stability. This approach, known as Bagging (Bootstrap Aggregating), contributes significantly to improving predictive accuracy.

The OFS algorithm leverages grid search and cross-validation techniques to fine-tune the hyper-parameters of the model. This process leads to a substantial enhancement in the model's performance and its ability to generalize well to unseen data. Additionally, prior to modeling, feature normalization is conducted to ensure uniform scaling across features, facilitating model convergence and further boosting overall performance[3]. The workflow of the OFS algorithm is illustrated in Figure 1 (right), showcasing the systematic approach adopted to optimize model parameters and improve overall efficacy.

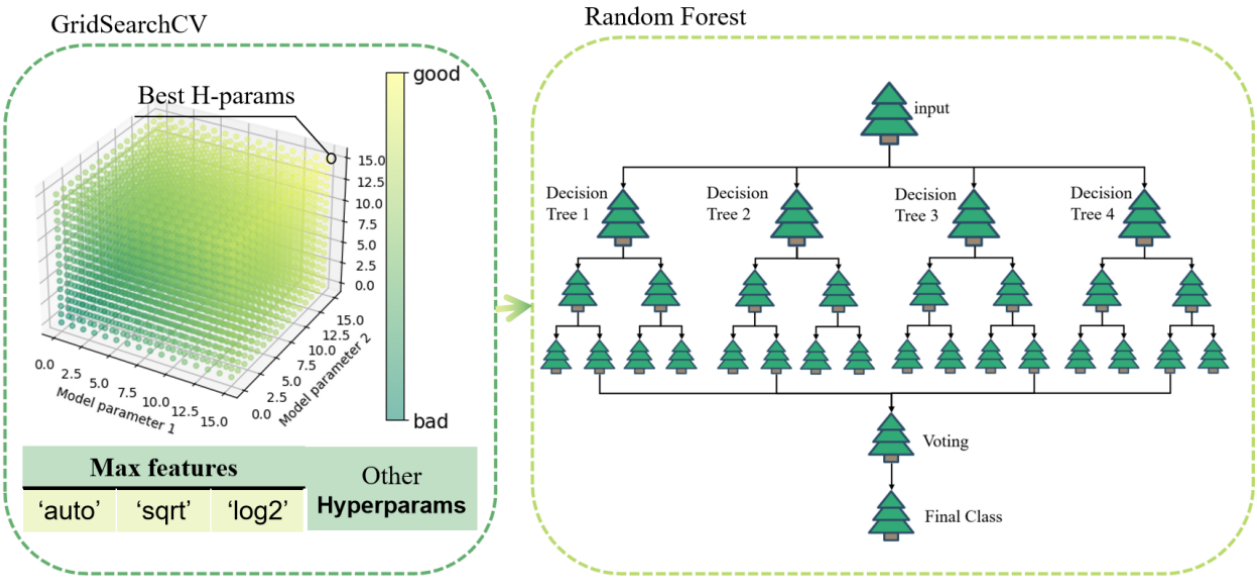


Figure 1. OFS Algorithm Flow

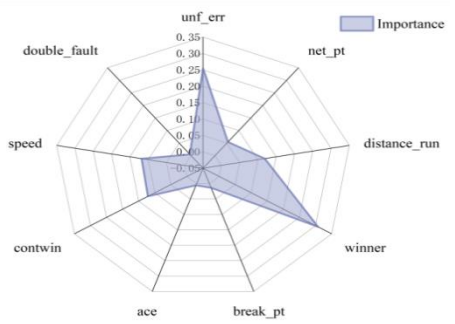
2.1.2. OFS algorithm implementation and result analysis

The specific steps realized in this paper are as follows:

Step 1: The first step is data reading. In this step, we first read the data from the file that will be used as the basis for our model training. To ensure the adequacy and representativeness of the training data, we choose 70% of the data in the attached file as the training set. This choice helps the model learn enough information and patterns to improve the accuracy and generalisation of the model.

To process and analyse this data, we used the Pandas library to create a data frame (Data Frame). A data frame is a structured data object that can easily store and process data and provides rich data manipulation and analysis capabilities. By using the Data Frame, we can easily perform data cleaning, feature engineering, data visualisation and other operations, laying a solid foundation for subsequent model training and analysis..

Step 2: Feature selection and data division: in machine learning, We create ace, $winner$, net_pt, $break_pt$, etc. As our characters as the eigenvalues of the training model in this paper, and the importance of each eigenvalue is shown in Fig.2. These indicators are generated by the difference between the indicators related to player1 and player2 according to the relationship between victory and defeat advantage. These indicators are generated by the difference between player1 and player2 according to the relationship between winning and losing, and the target variable of this paper is point_victor. in this paper, the data are divided into training set and testing set, so as to evaluate the performance of the model after training.



Notation	Description
ace	$p_2_ace - p_1_ace$
$winner$	$p_2_winner - p_1_winner$
net_pt	$p_2_net_pt - p_1_net_pt$
$break_pt$	$p_2_break_pt - p_1_break_pt$
$distance_run$	$p_2_distance_run - p_1_distance_run$
$double_fault$	$p_1_double_fault - p_2_double_fault$
unf_err	$p_1_unf_err - p_2_unf_err$
$speed$	$speed_mph$*1(sever=2) or $speed_mph$*(−1)($sever = 1$)
countwin	winning streak

Figure 2. Characteristic Importance Schematic

Step 3: Define the parameter grid: this paper needs to determine the possible range of values for the hyper-parameters of the model. This paper uses a random forest classifier and defines the possible values of the hyper-parameters such as the number of trees and the maximum depth[4].

Step 4: Build the model: this paper uses a machine learning random forest classifier.

Step 5: Grid search: the grid search technique is utilized to search on the defined grid of hyper-parameters to find the best combination of hyper-parameters. The performance of each hyper-parameter combination is evaluated by cross-validation to find the best combination.

Step 6: Output the best parameters: After the grid search is completed, the paper can output the best hyper-parameter combinations found, which make the model perform optimally on the given problem.

Step 7: Use the best parameters to build the model: Using the best hyper-parameters found, this paper builds the final machine learning model.

Step 8: Model Training: The model is trained using a training set to learn patterns and relationships in the data.

Step 9: Prediction: The trained model is used to make predictions on a test set to evaluate the performance of the model on new data.

Step 10: Performance Evaluation, Output Report Results: Calculate the performance metrics of the model on the test set.Grid optimization search result is shown in table 1.

Table 1. Grid optimization search result

Max_depth	Max_features	Min_sample	Min_samples_split	N_estimators
10	**'sqrt'**	**1**	**2**	**100**

max_depth: setting the maximum depth of the tree to 10 means that each decision tree is limited to a maximum depth of 10. this helps to prevent overgrowth of the decision tree. Reduces the risk of over-fitting our model. At the same time, a certain model complexity is maintained, allowing the model to generalize better to new data.

max_features: the parameter is set to 'sqrt' to indicate that only the square root of the total number of features is considered when splitting nodes for each decision tree. This setting helps to reduce the randomness of feature selection and improves the stability and generalization of our model.

min_samples_leaf and *min_samples_split*: set to 1 and 2 respectively indicate that the minimum number of samples for leaf nodes is 1. The minimum number of samples for internal nodes is 2. These settings help to control the way the decision tree grows to avoid over-fitting. and improve the robustness of the model.

n_estimators: a setting of 100 indicates that 100 decision trees are constructed to form the random forest model. Increasing the number of decision trees improves the stability and predictive performance of the model, but also increases the computational cost.[5]

In selecting this set of optimal parameters, we consider several factors with the aim of improving the predictive performance and stability of the model, while avoiding the problem of overfitting. Overfitting is when a model overfits the training data, resulting in a reduced ability to generalise on new data. To address this challenge, we take a number of measures.

First, by limiting the depth and number of features per tree, we effectively control the complexity of the model. Too much depth or too many features may lead to an overly complex model that overfits the training data, whereas by setting a reasonable depth and number of features, we can ensure that the model avoids overfitting while maintaining its effectiveness.Second, we set the minimum number of samples for leaf nodes and internal nodes. This step helps to prevent the model from overfitting in the training data because it limits the way the tree grows, making the model more robust.In addition, increasing the number of decision trees was also done to improve the robustness and generalisation of the model. By constructing multiple decision trees and combining their predictions, we can reduce the variance of the model and improve its adaptability to new data.In summary, the purpose of choosing this set of optimal parameters is not only to improve the performance of the model on the training data, but also to ensure that the model is stable and has good generalisation ability when facing new data, which is crucial for the practical application of the model[6-7].

2.2. Recommendations for differences in match fluctuations

Given the differential in past match "momentum" swings we suggest:

1. Firstly, it is important to analyse past matches. It is important to look at the history of meetings between the two teams as well as how they have performed in their most recent matches. Find out the results of the historical games between them, as well as the wins and losses of the most recent games. This step can help athletes and coaches identify any momentum or trends that may exist.The purpose of analysing past matches is to gain insight into the athleticism of the two teams, their tactical tendencies, and the wins and losses between them. By examining the history of meetings, it is possible to gain insight into a team's level of competitiveness and mentality at different stages of the game, which in turn provides an important reference point for formulating the next step in the game strategy. At the same time, focusing on the win/loss situation of recent matches can help to grasp the team's strength and status in time, and provide guidance for subsequent training and match arrangements.This kind of comprehensive analysis not only helps to identify the key factors and potential trends in the game, but also helps coaches to adjust their tactics and strategies more accurately and improve the competitiveness of their teams in the game. Therefore, in-depth analysis of past matches is one of the most important steps in improving overall performance and achieving excellent results.[8]

2. Pay attention to recent performances: Pay attention to how the athlete has performed in the last few matches. If an athlete has performed well in the last few games, he or she may be in a good

position to gain momentum. Conversely, if it has performed poorly, it may need to consider adjusting tactics or regaining form[9-10].

3. Mental Fitness: Consider the athlete's mental fitness and ability to cope with pressure. Sometimes, even in unfavorable situations, strong mental qualities can help an athlete turn the situation around.

4. Opponent's Weaknesses: Study your opponent's weaknesses and try to use them to your advantage. If your opponent has not been performing well in a certain area recently, try to focus on attacking that point.

5. Adapt to the course of the match: Be flexible and adapt to the situation during the match. If your athlete senses that the momentum is shifting, consider adjusting tactics or strategies, calling time-outs, etc. to maintain an advantage or turn the tide.

6. Stay Focused: Momentum can change quickly during a match, so it is critical to stay focused and concentrated. Don't let a short disadvantage affect your confidence, but focus on each point of the game.

3. Conclusions

In sports competition, effective prediction and analysis of match fluctuations is crucial for improving team performance and achieving victory. In this paper, we demonstrate the potential and application of machine learning techniques in sports through the study and implementation of OFS (Outcome Fluctuations on the Field of Play) algorithm. By integrating the Random Forest model and lattice search optimisation, the OFS algorithm is able to accurately capture key fluctuations in the game, providing valuable tactical advice and decision support to coaches and athletes. Meanwhile, this paper also proposes practical suggestions for match fluctuations, including in-depth analysis of historical data, focusing on athletes' psychological quality, making full use of opponents' weaknesses, and flexibly adjusting tactical strategies. These suggestions provide effective guidance and assistance for teams to achieve better results in the arena.

It is worth noting that although machine learning algorithms and data analytics tools have brought us unprecedented insights and predictive capabilities, they still need to be combined with professional knowledge and practical experience in real-world applications to ensure the accuracy and reliability of the models. Therefore, we call on practitioners in the sports field, especially coaches and athletes, to focus on developing their professionalism and tactical awareness while using technology to enhance their competitive level, and to continuously improve their training methods and game strategies, with a view to achieving even more outstanding results and performance.

Overall, OFS algorithms and similar machine learning technologies bring new development opportunities and challenges to the field of sports competition, and we look forward to further exploring their applications in different sports and competition scenarios in future research and practice, providing more comprehensive and effective support and guidance for athletes and teams on their way to success.

References

[1] Li Shipeng, Wei Gaoqi, Zhao Ziyan. A multi-model fusion-based method for predicting students' comprehensive grades in colleges and universities [J/OL]. Control Engineering, 1-8 [2024-05-10].

[2] Fei Teng, Fu Kangyu, Bian Meng, et al. Prediction of GIS Master's Interview Scores Based on Sentiment Calculation [J]. Geospatial Information, 2024, 22 (04): 128-132.

[3] Zheng Ni. Research on the application effect of GA-PCA model in education management of colleges and universities [J]. Journal of Tonghua Normal College, 2024, 45 (04): 73-79.

[4] Li Xinxin. Improving the application of group cooperative learning mode in animation professional courses in higher vocational colleges [J]. Shaanxi Education (Higher Education), 2024, (04): 76-78.

[5] Ye Yi, Xu Xinhua, Chen Suna, et al. Research on academic performance analysis and prediction based on data mining [J]. Computer and Information Technology, 2024, 32 (02): 78-81.

[6] Guanghai Zhang, Xuan Zu, Yang Wang. A three-classification prediction model for student achievement based on DBO-SVM [J]. Journal of Lanzhou College of Arts and Sciences (Natural Science Edition), 2024, 38 (02): 56-61.

[7] Fan W, Yang Y, Dong Yanqiu, et al. The effect of reward motivation on self-deception in a competitive situation:eye movement evidence [J]. Psychological Science, 2024, 47 (02): 411-423.

[8] S.C. Wei, B.L. Wang. Research on undergraduate student performance prediction and tiered teaching method based on machine learning [J]. Computer Knowledge and Technology, 2024, 20 (07): 158-161.

[9] Yang Xianqing. Neural network-based prediction of academic performance and empirical investigation [J]. Information and Computer (Theoretical Edition), 2024, 36 (04): 254-256.

[10] LIU Jianwei, DONG Zhengyu. Predicting physical test scores based on different neural network models [J]. Information Technology, 2024, (01): 65-70+76.