Football Match Prediction using Random Forest Classifier

Aloysius Chua Jia Xing
School of Computing
Asia Pacific University of Technology and Innovations(APU)
Kuala Lumpur, Malaysia
tp068892@mail.apu.edu.my

Cheah Qiyang
School of Computing
Asia Pacific University of Technology and Innovations(APU)
Kuala Lumpur, Malaysia
tp068003@mail.apu.edu.my

Yahya Mohammed Abdullah Al-Fakih
School of Computing
Asia Pacific University of Technology and Innovations(APU)
Kuala Lumpur, Malaysia
tp064914@mail.apu.edu.my

Ebrahim Abdulrahman Hamdan Shaker
School of Computing
Asia Pacific University of Technology and Innovations(APU)
Kuala Lumpur, Malaysia
tp065022@mail.apu.edu.my

Zailan Arabee bin Abdul Salam
School of Computing
Asia Pacific University of Technology and Innovations(APU)
Kuala Lumpur, Malaysia
zailan@apu.edu.my

***Abstract*—This study explores the difficulty of applying the Random Forest Algorithm to predict football outcomes. The application and modification of the Random Forest method, with a focus on improving prediction accuracy and efficiency, is the aim of this study. Key algorithm parameters, such as min_sample_split and min_sample_leaf, are adjusted and contrasted throughout this study to determine their influence on the accuracy of predictions. The painstaking optimization of these variables led to the discovery of an ideal combination, significantly strengthening the algorithm's capacity for precise football match prediction.**

Keywords*—Random Forest, Machine Learning, parameter, classification

I. Introduction

Football is a sport that is very popular in the world, regardless of gender and age, a lot of people enjoy football. And there are a lot of leagues, including English Premier League, French League, German Bundesliga and Spanish La Liga. Among those League, the English Premier League is most favored by the public. Based on statistics from ESPN, a match in the 2022/2023 season had 3.1 million viewers per game, which is the highest viewership ever recorded. Therefore, the media and community often try to predict the outcome of a football match due to the high enthusiasm of the community towards football. Hence, there is a need for football match predictors. Previously, the prediction was done by humans, which is based on the data collected from previous matches.

However, sports analysis as well as outcome prediction conducted by humans is not always reliable. Therefore, we have introduced to use of Machine Learning, which is a subset of Artificial Intelligence, to use in this domain. Among different approaches that have been conducted by various researchers in their study, we have chosen to use Random Forest as the main algorithm used to predict the outcome due to its high accuracy (Alfredo and Isa, 2019; Eryarsoy and Delen, 2019). We believe that the football match outcome will be more accurate with the use of Artificial Intelligence.

In short, although there are various approaches that can be used to handle football match outcome prediction, but the algorithm choice can have a significant influence on the predictive performance. Hence, the objective of this study is to investigate the predictive performance as well as the usefulness of the Random Forest Algorithm in football outcome prediction.

II. Literature Reiview

The Random Forest Algorithm may be a directed learning method that makes a gathering of choice trees. It utilizes the stowing strategy to prepare these trees collectively, pointing to improve general execution unlike traditional decision trees that search for the most important feature when splitting a node, a random forest introduces an element of randomness by selecting the best feature from a random subset of features (Donges, 2023). This approach promotes diversity and typically leads to improved model performance. Consequently, in a random forest classifier, the algorithm considers only a random subset of features when making decisions about node splitting. Additionally, the algorithm can introduce further randomness by utilizing random thresholds for each feature instead of seeking the optimal thresholds, as done in standard decision trees.

Random Forest Algorithm, which is an ensemble method made up of a set of decision trees to solve some problems, includes classification as well as regression problems, by generating prediction (IBM, n.d.). However, we are more focused on solving the classification problem as it is the main algorithm we have used.

Several studies are conducted on generating predictions in football outcomes, using various algorithms, including random forest algorithms. Among those studies, one of the studies was conducted to develop an accurate model for outcome prediction (Eryarsoy and Delen, 2019). To achieve this, they have compared several models developed with different approaches, namely Naïve Bayes, Decision Trees, Gradient Boosting Trees, and Random Forests by using the dataset from the Turkish Super League in the years 2007-2017. As compared, the Random Forest and Gradient Boosting Trees have about 74.50% and 74.60% of accuracy in "Win/Lost/Draw" problems respectively. Meanwhile, they have 86.30% and 86.40% accuracy correspondingly in "Point/NoPoint" types of problems, which is significantly higher than others that only have accuracies within 50 to 59 percent. Hence, both Random Forest and Gradient Boosting

Trees are suggested by Eryarsoy and Delen (2019). However, researchers also believe that higher accuracy can be yielded as a larger and more comprehensive dataset has been used (Eryarsoy and Delen, 2019).

Another study that generated an opposite conclusion was also conducted on a similar domain by Alfredo and Isa (2019), which aimed to investigate the usefulness of the Tree-based model algorithm in football outcomes prediction. Within their study, they have compared several approaches, namely Random Forest, Extreme Gradient Boosting, and C5.0. Among these approaches, Random Forest has the highest accuracy, which is 68.55%. While the accuracy of Extreme Gradient Boosting and C5.0 is 67.89% and 64.87% respectively (Alfredo and Isa, 2019). This result depicted that Random Forest is the superior approach that has the highest accuracy among other models. Nonetheless, the difference between those models is not significant, as it is due to insufficient data collection (Alfredo and Isa, 2019). Therefore, the researchers considered that the Tree-based model is not reliable in the relevant domain. On the other hand, they have suggested a richer dataset with more relevant features to be used in order to get a model with better performance.

Apart from those studies in relevant domains, there are a few studies that use the same approach, namely Random Forest, but in different fields. The study conducted by those researchers is aimed at unraveling the face recognition issues by utilizing the Random Forest algorithm (Chee Chiew et al., 2022). Within their study, they tried to test the performance of their trained model by hyperparameters tuning process. According to the result experimented by Chee Chiew et al. (2022), the model can have an accuracy of 94% after the parameters tuning process. The 94% accuracy can be achieved by 2 sets of parameters, which are *"Epsilon" = 0.01, "NumberTree" = 1000* and *"Epsilon" = 0.01, "NumberTree" = 300.*

Another study contributed by Junn Fai et al. (2023), set out to investigate the performance of the Random Forest classifier in digit classification. To achieve their objective, they have developed a model for digit classification by using a digit dataset chosen from scikit-learn, whereby proceed with the hyperparameter tuning process (Junn Fai et al., 2023). Besides, the digits dataset is a collection of 1,797 handwritten digits images which is widely used as a benchmark for machine learning algorithms. Based on the study conducted by Junn Fai et al. (2023), the Random Forest Classifier has achieved the highest accuracy of 94.16% within the parameter tuning process, whether individual or cross hyperparameters tuning process. In the individual parameter tuning process, the accuracy can be achieved by setting the value *"n-estimators" = 125* or *"max_features" = 4.* Meanwhile, in the cross-tuning process of three hyperparameters, the accuracy can be achieved by 2 sets of values, which are *"max_depth" = "None", "n-estimators" = 125 or "max_features" = 8* and *"max_depth" = "None", "n-estimators" = 100 or "max_features" = 4.* Due to its high accuracy, Junn Fai et al. (2023) suggest using Random Forest Classifier with a parameters tuning process to solve digit classification problems.

III. Materials and Methods

This part will explain the steps that were taken to get the materials and the methods that were used to address the issue.

A. Selection of materials

1) Source Code: The Python source code was obtained from an open-source cloud storage site called GitHub *(Aziztitu, 2020)*.

2) Machine: The Python source code is run on Google Collaboratory, which is a place where Python codes can be run. It uses the Python 3 core of Google Compute Engine and has 12.7GB of RAM and 107.7GB of disk storage.

3) Dataset: The football match dataset is obtained from DataHub, a website for peoples to share or find quality datasets. 5 folders of dataset were used in the model training to ensure sufficient data. These datasets are basically made up of statistical data such as home team goals at halftime, historical match-up results and others. Each dataset consists of around 380 rows of data and each folder includes 10 of these datasets which means around 19,000 rows of data are used to train and test the model.

B. Algorithm Implementation

The Random Forest algorithm is a "ensemble learning" method. It makes a unique classification system by putting together a lot of different classifiers, which are called "weak learners." This method is part of a group that lets you combine different kinds of decision trees. Each tree in the forest takes its numbers from a vector that was chosen at random. Each tree picks its own numbers for these things, but they all come from the same range. Random Forest is like a mix of the boosting ensemble method and the bagging ensemble method. It is more or less a mix of boosting and bagging. It chooses a subset of features for each decision tree and a subset of training data for each tree using a random sampling method.

But before the method can be used to solve problems, there are a few important steps that must be taken to make sure it is fully used. The first step is data preparation, which is where data preprocessing happens. The major jobs in this step include data cleaning, dealing with missing data, and any other data transformations that are needed to get the data ready for model training. Next is model training, where we use our ready information to teach the random forest-based model what to do. Then there is cross-validation, where we test how well the model works and make sure it does not fit too well by testing it with different groups of the data. Feature importance analysis is a way to figure out which traits are most important for making a guess. Lastly, parameter tuning is when we change the settings to improve the performance of the random forest.

C. Parameter

1. *min_sample_leaf*: min_sample_leaf is the number of samples that must be at a leaf node for it to be considered a leaf node. The complexity of the tree in the random forest classifier is controlled by the value of this number. The tree will be more complicated if the value is high, and less complicated if the value is low. After a few tweaks, we set this parameter's value to 40 because we found that this value gave the best accuracy while training the model.

2. *min_sample_split*: Depending on what value you put in this option, the trees in the random forest will split more or less quickly. When the value is low, trees will be more aggressive. When the value is high, trees will be less aggressive. This number is important to stop overfitting and make the computer work better. In our model, this parameter's number is set to 20, which we found to be the best.

IV. Results and Discussion

The Random Forest Algorithm may be a directed learning method that makes a gathering of choice trees. It utilizes the stowing strategy to prepare these trees collectively, pointing to improve general execution unlike traditional decision trees that search for the most important feature when splitting a node, a random forest introduces an element of randomness by selecting the best feature from a random subset of features (Donges, 2023). This approach promotes diversity and typically leads to improved model performance. Consequently, in a random forest classifier, the algorithm considers only a random subset of features when making decisions about node splitting. Additionally, the algorithm can introduce further randomness by utilizing random thresholds for each feature instead of seeking the optimal thresholds, as done in standard decision trees.

A. Discussion on Implementation

In this study, a computing system will be employed, featuring the Windows 11 operating system and an Intel Core i7 central processing unit running at 2.30GHz, accompanied by 16GB of installed RAM. Additionally, the Random Forest algorithm will be executed using Google Collab as the software platform for processing.

The primary dataset utilized for this experiment consists of information from the English Premier League, French Ligue 1, German Bundesliga, Italian Serie A, and Spanish La Liga.

B. Result

Table 1 presents the results of adjusting two parameters, "min_sample_leaf" and "min_sample_split," and their effect on the exactness of a machine learning show on both preparing and testing information.

The table presents the results of adjusting two parameters, "min_sample_leaf" and "min_sample_split," and their effect on the exactness of a machine learning show on both preparing and testing information.

TABLE 1. ACCURACY FOR TUNING 2 PARAMETERS

min_sample_leaf	**min_sample_split**					
	10		***60***		***110***	
25	69.583	66.779	68.848	66.779	67.774	66.471
150	65.702	65.548	65.492	65.213	65.806	64.430
275	64.926	63.591	64.800	63.786	64.863	63.898

In the initial random setting, with "min_sample_leaf" set at 150 and "min_sample_split" at 60, the model achieved an accuracy of approximately 65.492% on the training data and 65.213% on the testing data. Subsequent modifications to these parameters were made, resulting in various combinations and corresponding accuracy values. For instance, when "min_sample_leaf" was set to 275 while keeping "min_sample_split" at 60, the training accuracy decreased to around 64.800%, and the testing accuracy dropped to approximately 63.786%. On the other hand, when "min_sample_leaf" was reduced to 25 while maintaining "min_sample_split" at 60, the training accuracy improved to about 68.848%, and the testing accuracy increased to approximately 66.779%.

The table also shows the impact of variations in "min_sample_split" values while keeping "min_sample_leaf" constant. For instance, with "min_sample_leaf" at 150 and "min_sample_split" at 110, the training accuracy increased to approximately 65.806%, but the testing accuracy decreased to around 64.430%. Conversely, with "min_sample_split" at 10, the training accuracy remained stable at approximately 65.702%, and the testing accuracy also remained consistent at about 65.548%.

At long last, the table outlines the comes about when both "min_sample_leaf" and "min_sample_split" were modified at the same time. Different combinations of these two parameters led to various accuracy values on both the training and testing data. For instance, when "min_sample_leaf" and "min_sample_split" were set at 275 and 110, respectively, the training accuracy decreased to around 64.863%, and the testing accuracy dropped to approximately 63.898%. Conversely, with "min_sample_leaf" at 25 and "min_sample_split" at 10, the training accuracy significantly improved to about 69.583%, but the testing accuracy decreased to approximately 63.675%.

These come about illustrate the affectability of the model's precision to the values of the "min_sample_leaf" and "min_sample_split" parameters. Particular combinations of these parameters can lead to changing levels of precision, emphasizing the importance of parameter turning in the optimizing illustrate execution for specific errands.

One notable configuration that yielded the highest accuracy in this experiment occurred when "min_sample_leaf" was set to 25 and "min_sample_split" was set to 10. In this case, the training accuracy reached an impressive approximately 69.583% which is the highest recorded accuracy. However, it's essential to note that while achieving high training accuracy is desirable, it must be balanced with the model's ability to generalize to unseen data, which is represented by the testing accuracy, In this configuration, the testing accuracy was approximately 66.779% which is relatively lower than the training accuracy. This discrepancy between the training and testing accuracy suggests that the model might be overfitting the training data, meaning it has learned to perform exceptionally well on the data it was trained on but might not generalize effectively to new, unseen data.

Conversely, the minimum accuracy values were observed when " min_sample_leaf" was set to 275 and "

min_sample_split" was set to 110. In this case, the training accuracy dropped to approximately 64.863%, and the testing accuracy was approximately 63.898%. This configuration resulted in the lowest accuracy observed in the experiment.

V. CONCLUSION

The endeavor to predict football match outcomes through the Random Forest Algorithm has been meticulously explored in this study. Through rigorous adaptation and fine-tuning of key parameters, notably min_sample_split and min_sample_leaf, the research underscores the profound impact of these parameters on predictive accuracy. The systematic approach undertaken led to the identification of an optimal parameter combination, offering a significant leap in the algorithm's predictive capability. This achievement not only underscores the potential of the Random Forest Algorithm in sports prediction but also paves the way for further refinements in this domain, ensuring even greater precision in future endeavors.

ACKNOWLEDGMENT

The authors want to thank everyone from the School of Computing who took part in this work. The goal of this study is to use the random forest algorithm to predict result of football matches.

REFERENCES

Alfredo, Y. F., & Isa, S. M. (2019). Football Match Prediction with Tree Based Model Classification. *International Journal of Intelligent Systems and Applications*, *11*(7), 20–28. https://doi.org/10.5815/ijisa.2019.07.03

Yeong, C.C., Siew, Y.H., Wong, Y.W., Chai, C.T., Abdul Salam, Z.A. (2022) Random Forests for Face Recognition. Journal of Applied Technology and Innovation vol. 6, no. 4, e -ISSN: 2600-7304. https://dif7uuh3zqcps.cloudfront.net/wp-content/uploads/sites/11/2022/10/24130455/Volume6_Issue4_Paper9_2022.pdf

Eryarsoy, E., & Delen, D. (2019). Predicting the outcome of a football Game: A comparative analysis of single and ensemble analytics methods. *Proceedings of the . . . Annual Hawaii International Conference on System Sciences*. https://doi.org/10.24251/hicss.2019.136

Ngan, J.F., Keong, Y.Q., Jee, M.C., Wong, K.W., Gan, J.X., Abdul Salam, Z.A. (2023) Digits Classification Using Random Forest Classifier. Journal of Applied Technology and Innovation vol. 7, no. 3. e -ISSN: 2600-7304. http://jati.sites.apiit.edu.my/files/2023/07/Volume7_Issue3_Paper11_2023.pdf

What is Random Forest? | IBM. (n.d.). https://www.ibm.com/topics/random-forest

Donges, N. (2021, July 22). Random Forest: A Complete Guide for Machine Learning. Built In. https://builtin.com/data-science/random-forest-algorithm