I.J. Intelligent Systems and Applications, **2019, 7, 20-28**
Published Online July 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2019.07.03

Football Match Prediction with Tree Based Model Classification

Yoel F. Alfredo
Computer Science Department, BINUS Graduate Program-Master in Computer Science,
Bina Nusantara University Jakarta, 11480, Indonesia
E-mail: yoel.alfredo001@binus.ac.id

Sani M. Isa
Computer Science Department, BINUS Graduate Program-Master in Computer Science,
Bina Nusantara University Jakarta, 11480, Indonesia
E-mail: sani.m.isa@binus.ac.id

Received: 26 January 2019; Accepted: 19 March 2019; Published: 08 July 2019

***Abstract*—**This paper presents the football match prediction using a tree-based model algorithm (C5.0, Random Forest, and Extreme Gradient Boosting). Backward wrapper model was applied as a feature selection methodology to help select the best feature that will improve the accuracy of the model. This study used 10 seasons of football data match history (2007/2008 – 2016/2017) in the English Premier League with 15 initial features to predict the match results. With the tuning process, each model showed improvement in accuracy. Random Forest algorithm generated the best accuracy with 68,55% while the C5.0 algorithm had the lowest accuracy at 64,87% and Extreme Gradient Boosting algorithm produced accuracy of 67,89%. With the output produced in this study, the Decision Tree based algorithm is concluded as not good enough in predicting a football match history.

***Index Terms*—**Football match prediction, supervised machine learning, decision tree, feature selection, classification.

I. Introduction

Football is a very popular sport among people of various ages and genders. This sport has been enjoyed since the times of ancient Egypt, where it was played by kicking a ball formed from a collection of linen fabrics [1]. The English Premier League is one of the leagues most favored by the public. Based on statistics from ESPN on season 2017/2018, there are as many as 14,56 million viewers who attend these football matches. With the high enthusiasm from the community towards football, the media and the community often try to predict of the outcome of a football matches.

Presently, football is not only a means of a sport but also a form of entertainment and investment. In investing, an investor needs help from an expert to determine the right decision to invest. Even so with a football coach, they need need guidance in developing a strategy to face the opposing team. The involvement of an expert is the common solution, however that does not mean it is unflawed. Human assistance as a tool in making decisions is not always reliable. Some psyche factors can influence the analysis of the outcome and cause the decisions that are made to be inappropriate.

In addition to relying on experts as a solution in helping decision making, several studies have been carried out in predicting the results of football matches [2,3,4,5,6]. Some studies have output limitation of football matches [2,4], the algorithm used is logistic regression which only gives 2 output 'home win' or 'away win' while football match has 3 result possibilities, 'home win', 'away win', or 'draw'. Another study has been conducted using the random forest algorithm [3], but the prediction accuracy is only 63,4%.

Based on several studies conducted in making predictions in other fields, in this study the authors proposed a decision tree as the method of choice in predicting the end result of a football match. A sample research that uses a decision tree is the initial prediction of a heart disease. The algorithms used in this study are CART, ID3 and decision table, with accuracy values of 83.49% using CART algorithm, 72.93% using ID3 algorithm, and 82.50% using decision table algorithm [7]. Other studies were conducted to detect bad data using a decision tree with an accuracy rate of 94% [8]. The use of the XGBoost method carried out in generating predictions of bank failures in America can produce up to 94.74% accuracy [9].

This study will use the C5.0, Random Forest and XGBoost algorithms from the decision tree method to predict the outcome of a football match in the English Premier League. With larger datasets and feature selection method using backward wrapper method, it is expected to improve the prediction accuracy from previous studies that have been done.

This paper will be divided into five sections, including the introduction section. Section II describes the related works. Section III describes the research methodology

used in this research. Result and analysis will be described in section IV. Conclusions of this research drawn in section V.

II. Related Works

Since football match prediction have become commonplace for people of all ages, several studies were conducted to find out the criteria associated with football match result and the model that can generate the highest accuracy in predicting football match result. These are following studies that have been conducted in order to find the optimal model for football match predictions.

One study was done by Prasetio and Harlili [2] using the logistic regression method and home offense, home defense, away offense and away defense as their model features with prediction numbers reaching 69.5%. A weakness in predicting the results of a soccer match using the logistic regression method is that the results obtained only have 2 values, where in reality, the results of a soccer match can produce 3 final scores, namely win, draw or lose. Yezus *et.al.* [3] produces predictions of football matches using machine learning with K-nearest neighbors and random forest algorithms, which use form, concentration, motivation, goal difference, score difference, and history as key features to develop classifier and had accuracy values of 55.8% and 63.4%, respectively. Igiri and Nwachukwu [4] also conducted another research on football match predictions by using artificial neural network and logistic regression with knowledge discovery in database framework with 18 features which consists of goals, shots, corner, odds, attack strength, player's performance index, manager performance index, managers win and streak for each home and away team. Artificial neural network method has a prediction percentage of 75.04% and it improved to 85% by giving weight to the features. On another hand, the logistic regression method has a high accuracy percentage, 93%, but limited due to the results only obtain 2 values. Igiri [5] have another research for football match prediction using the SVM method with 15 features and reaching prediction percentage of 53.3%. Bayesian network method also used by Razali *et.al.* [6] to predict football match. They use shots, shots on target, corners, fouls committed, yellow cards, red cards, half-time goals, and full-time goals as features to build the model. In this research, the model prediction accuracy is 75.09%.

III. Research Method

Fig. 1 will display a design experiment for this research. To predict football match results, this research will have two major steps: data preprocessing and classification, with each step broken down into more detailed steps. This research will use R as a programming language to develop the model.

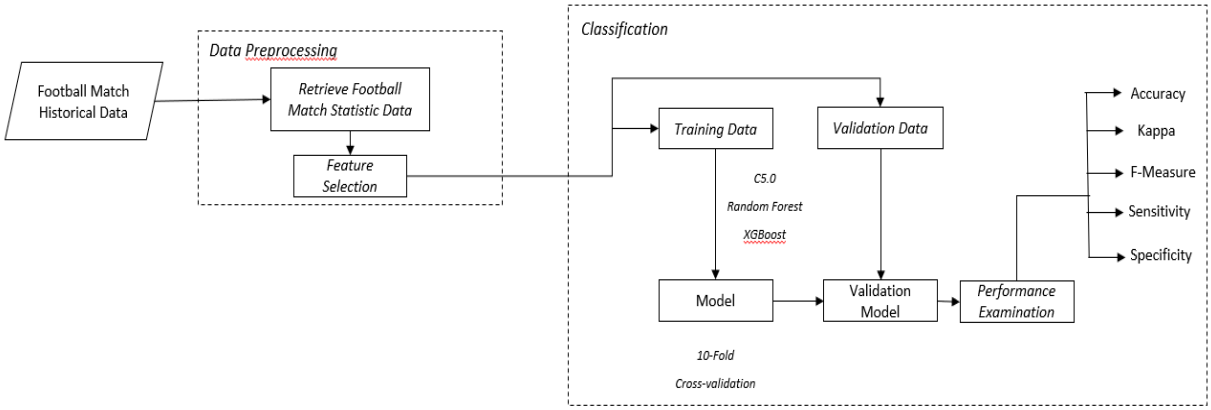

Fig.1. Design experiment.

A. Data Collection

Dataset used in this research comes from football-data.co.uk which is a common dataset to be used in conducting research in football match predictions. The data used comes from 10 seasons of English Premier League matches from the 2007/2008 season to the 2016/2017 season. The total number data used for these entire study is 3800 historical match data consisting of 380 matches per season. Each season, 20 football team participated in the English Premier League and each team acted as a host or a guest.

The dataset has 71 attributes that will be cleaned in the preprocessing step. From 71 attributes, we can divide it into two categories: football match statistics and bookmaker odds prediction. This research will only use football match statistics to develop a model to predict football match results. In addition, the feature selection process will be done by using the backward wrapper model to determine which attributes will be used for creating model.

B. Preprocessing

Dataset used in this research still needs to be cleaned. There are some attributes which have no value for several seasons. In this process, the bookmaker odds and irrelevant data which have no impact in the model development will be removed such as match date, referee

name, and football team name. Attribute FTR (Full Time Result) will be used as the label of the model.

Table 1. Initial features

Features	Data Scale	Description
FTR	Nominal	Full-time result. Used as the label of the model
HTHG	Ratio	Number of goal for the home team in half time
HTAG	Ratio	Number of goal for away team in half time
HS	Ratio	Number of shot done by the home team
AS	Ratio	Number of shot done by away team
HST	Ratio	Number of shot on target is done by the home team
AST	Ratio	Number of shot on target is done by the home team
HF	Ratio	Number of fouls done by the home team
AF	Ratio	Number of fouls done by away team
HC	Ratio	Number of corners obtained by the home team
AC	Ratio	Number of corners obtained by away team
HY	Ratio	Number of yellow cards obtained by the home team
AY	Ratio	Number of yellow cards obtained by away team
HR	Ratio	Number of red cards obtained by the home team
AR	Ratio	Number of red cards obtained by away team

From a total 14 attributes in the initial feature, feature selection will be done to only select the best attributes. Only those that have the potential to have a good impact on prediction and result accuracy, will be included in the model development.

1. *Feature Selection*

Feature selection is a process that is commonly used by researchers to get a smaller part of the entire data. This smaller set of data is chosen based on relevance, to improve the performance of the model created in the training process [10]. This feature selection process will be very influential in the model training stage. From initial feature candidates, the feature selection process will select some features that have relevance and process data with the selected features to form the model.

In this research, backward wrapper model will be used as a feature selection methodology. The classifier will act as a black box and will use random forest algorithm as the algorithm for the classifier. Features that have been evaluated will be assessed for their performance by utilizing the classifier. This will be done until all the features have been tested. Feature with the highest estimated value will be used in the development of the classifier model. On this method, Random Forest will be used as a classifier method. Fig. 2 display the framework of wrapper models [11].

C. *Classification*

Classification is one of the predictive methods of data mining. The purpose of classification is to get a prediction of future value based on other variables contained in the dataset. The classification has four fundamental components: class, predictors, training datasets and testing datasets [12]. This research will use supervised machine learning (classification) technique as a prediction method.

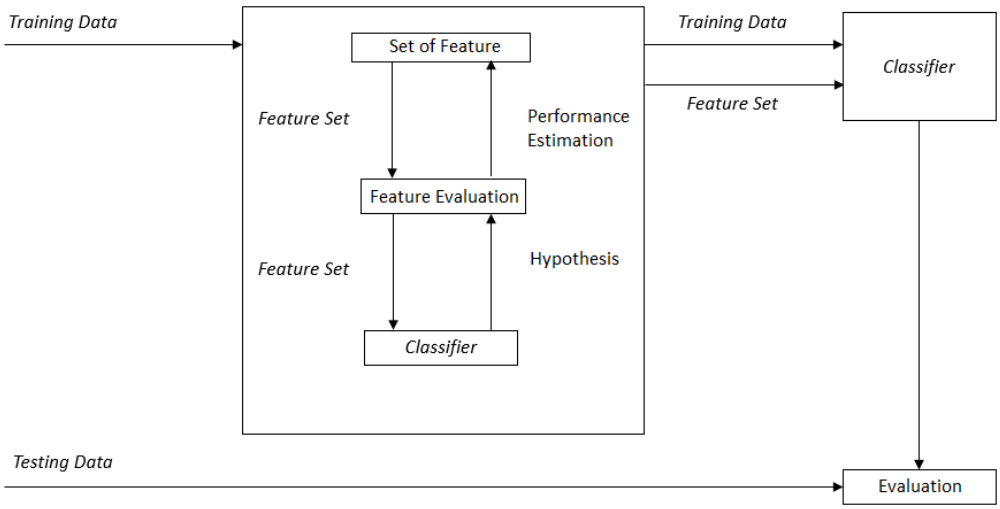

Fig.2. Wrapper model framework.

1. *Data Partition*

Data that have been processed through feature selection will be divided into training data and testing data. Training data will be used to build the model and testing data will be used to test the performance of the model. Data set will be divided with composition 80:20. 80% of the data will be used as training data and the other 20% will be used as testing data.

2. *Cross-Validation*

Cross-validation is a statistical method to evaluate and

compare algorithms in the learning process by dividing the data into two parts. The first part is used for the learning process of the model and the second part is used to validate the model [13].

This research uses K-fold cross-validation method, with 10 as the number of K variable. This method will be applied to model development using only training data. Fig. 3 explains how K-fold cross-validation work.

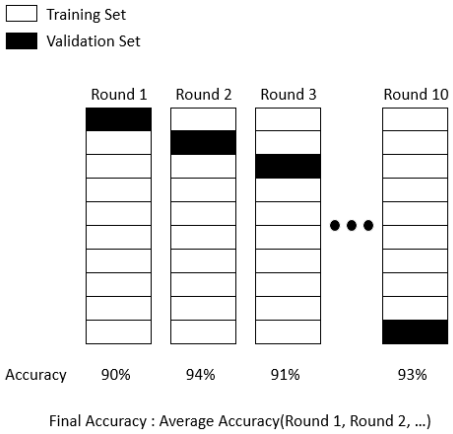

Fig.3. K-fold cross-validation illustration.

3. C5.0 Algorithm

C5.0 is a decision tree algorithm that has been developed based on C4.5 algorithm [14]. It has all C4.5 algorithm functionality with improvements on the technology. Techniques introduced in C5.0 as an improvement from it ancestor is [15] boosting, variable misclassification, new attributes, values can be marked as missing or not applicable on a particular case, support sampling and cross-validation. To get entropy value, we can use (1) formula.

$$Entropy(k) = \sum_i - p_i \log p_i \quad (1)$$

where p_i is the probability of class i within node k used to split node. To find splitting criterion from a root that has been defined before can be expressed with formula (2).

$$Gain\ Ratio(k) = \frac{Gain(k)}{SplitInformation(k)} \quad (2)$$

4. Random Forest Algorithm

Random forest is a classification of tree that can be used to make a prediction. [16] proposed this methodology which changes how the classification tree and regression tree are constructed. In this method, each node is split based on the best among subsets of predictors randomly chosen at that node. Ref. [16] stated this technique will make the tree constructed robustly against overfitting. The algorithm for this random forest is explained below [17].

- Draw n_{tree} bootstrap sample from original data
- For each of the bootstrap samples, make an unpruned classification tree with modification at each node randomly sample m_{try} of the predictors and pick the best split among other variables.
- Predict new data by aggregating the predictions of the n_{tree} tree.

With an ensemble of classifiers $h_1(X)$, $h_2(X)$, … , $h_K(X)$ and training dataset is drawn in the random forest from the distribution of vector X, Y, the margin function can be expressed as formula (3).

$$mg(X,Y) = av_K I(h_K(X) = Y) - \max_{j \neq Y} av_K I(h_K(X) = j) \quad (3)$$

Estimation of the error rate on the training data can be done by predicting data not in the bootstrap sample, using tree grown with the bootstrap sample on each bootstrap iteration. The next step is to aggregate the OBB (out-of-bag) and calculate the error rate. We can call it the OOB estimate of error rate.

5. Extreme Gradient Boosting Algorithm

Extreme Gradient Boosting or known as XGBoost is a decision tree algorithm introduced by Tianqi Chen and Tong He. This method was introduced to solve problems in the Higgs boson machine learning competition. This method is a development of the gradient boosting approach by studying ensembles from boosted trees. This method is able to offer speed in the training process and good accuracy values [18].

XGBoost has been used in many competitions in the machine learning field. One of them was in the competition held by Kaggle in 2015, where 17 solutions used XGBoost from a total of 29 who entered and succeeded as the competition's winning candidate [19]. In the 2015 KKDCup competition, the top 10 winning team all used XGBoost method [19]. The XGBoost methodology works by combining all predictions of a set of weak learners to develop a strong learning with additive training strategy. Equation (4) is the general formula to make predictions:

$$f_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = f_i^{(t-1)} + f_t(x_i) \quad (4)$$

where $f_t(x_i)$ is a learner at step t, $f_i^{(t)}$ and $f_i^{(t-1)}$ is the prediction in step t and t-1 and x_i as the input variable.

To improve the performance of the model, this method optimize computation resources. In the XGBoost method, combining predictive and regularization is used to simplify the objective function and to maintain the optimal computational speed expressed in the formula (5).

$$Obj^t = \sum_{k=1}^{n} l(\tilde{y}_i, y_i) + \sum_{k=1}^{t} \Omega(f_i) \quad (5)$$

where l is the loss function, n is the number of observations and Ω is the regularization terms. The regularization terms can be expressed in (6) formula.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda ||\omega||^2 \quad (6)$$

where ω is the vector score in the leaves, λ is the regularization parameters, and γ is the minimum loss needed to expand partition of the leaf node.

6. Evaluation Methodology

The evaluation will use a confusion matrix to calculate each model's prediction accuracy, kappa, F1, sensitivity, and specificity. Each model will have one confusion matrix that consists of each class from the chosen label. The class will have 3 nominal value, H for home win, D for draw and A for away win.

IV. Result and Analysis

A. Feature Selection

Feature selection method used in this research is the wrapper method with a backward approach. From a total of 14 features available, features combination will be tested against the training model. Each combination will be measured by the accuracy percentage of the training model. Data partition in this feature selection will be 80:20 which 80% of the available data used as training data. The model will use a random forest algorithm to check the prediction accuracy of the model with the parameter n_{tree} is 500 and m_{try} is 2. n_{tree} parameter used in the Random Forest algorithm is the number of trees to grow while the model developed and parameter m_{try} is the number of variables available for splitting at each tree node while the model developed.

From all number of feature combination available, 10 features combination has the highest prediction accuracy which is 69.21% and an out-of-bag estimate of error rate is 31.71%. The confusion matrix of 10 features combination is shown in table 2.

Table 2. Confusion matrix of 10 features combination

	Class	Output Class		
		A	D	H
Target	A	153	50	22
	D	24	74	32
	H	36	70	299

Table 3 shows the results of each feature combination using the backward wrapper method. The process will have 14 iterations based on the number of features used in this research. Feature deletion is done based on the lowest feature importance on the generated training model. The table displaying accuracy, kappa and out-of-bag error rate as a measurement on each model.

Although not having the lowest out-of-bag error rate, the backward wrapper method concluded that 10 feature combinations give the highest accuracy and kappa number. To ensure the best feature combination, the training model re-generated the 10 feature combinations by reducing 4 feature combinations for each iteration. The experiment result of 10 feature combinations generated the following combinations of HTHG, HTAG, HS, AS, HST, AST, HF, AF, HC, and HY. It is the best feature combinations with prediction accuracy reaching 69.21%. Selected feature combinations will be used in further model predictions using a tree-based model algorithm.

Table 3. Features selection process

No. of Features	Accuracy	Kappa	OOB Error Rate
14	67,5%	0.4761	30,99%
13	68,16%	0.4882	31,84%
12	68,03%	0.4865	31,71%
11	66,97%	0.4708	31,48%
10	69,21%	0.5080	31,71%
9	67,37%	0.4808	32,04%
8	68,03%	0.4909	31,48%
7	67,63%	0.4845	32,27%
6	66,45%	0.4675	33,59%
5	64,61%	0.4375	36,02%
4	63,55%	0.4266	36,38%
3	63,68%	0.4208	39,38%
2	58,82%	0.3824	38,98%
1	55,26%	0.2694	44,54%

B. Analysis of Tree-Based Classification Model

The experiment process started by dividing the data into two partitions, training data, and testing data, with composition of 80:20. Training data consists of 3,040 data with 10 features and testing data consist of 760 data with 10 features.

Table 4. Evaluation of training data using C5.0 algorithm

Trial	Rules	
	Number	Errors
0	48	870(28,6%)
1	29	980(32,2%)
2	26	1054(34,7%)
3	34	1072(35,3%)
4	24	1075(35,4%)
5	26	1123(36,9%)
6	30	1045(34,4%)
7	33	1027(33,8%)
8	40	906(29,8%)
9	59	896(29,5%)
Boost		741(24,4%)

1. C5.0 Model

The first algorithm used in this experiment is C5.0. The model trained with 10 trials and a rule-based tree. Training data evaluation can be seen in table 4. The model was evaluated using a confusion matrix to see the classification performance, this can be seen in table 5 with accuracy reaching 75,534%.

The trained model is tested against testing data using a confusion matrix. The prediction accuracy result is 63, 29% with Kappa 0.4095. The confusion matrix result can be seen in table 6.

Table 5. Confusion matrix of C5.0 training model

	Class	Output Class		
		A	D	H
Target	A	691	53	130
	D	155	370	271
	H	81	51	1238

Table 6. Confusion matrix of C5.0 validation model

	Class	Output Class		
		A	D	H
Target	A	142	63	28
	D	26	48	34
	H	45	83	291

2. Random Forest Model

The next algorithm used in this experiment is Random Forest. The model trained with 10 fold cross-validation and m_{try} parameter with value 2, 6 and 10. With prediction accuracy, 67,89% and Kappa 0.493, the best Random Forest model used the value of 2 as m_{try} variable. Trained model tested using testing data to measure prediction accuracy. The accuracy of Random Forest model is 62,76% with Kappa 0.392. The confusion matrix and statistics of Random Forest model validation can be seen in table 7 and table 8.

Table 7. Confusion matrix of Random Forest training model

	Class	Output Class		
		A	D	H
Target	A	144	63	25
	D	20	30	25
	H	49	101	303

Table 8. Random Forest validation statistics

Statistics	Output Class		
	A	D	H
Sensitivity	0.6761	0.1546	0.8584
Specificity	0.8391	0.9205	0.6314
Kappa	0.3920		
Accuracy	0.6276		

With accuracy prediction lower than the C5.0 algorithm, the Random Forest algorithm needs to be improved. Accuracy improvement will be done in this experiment by testing on several parameters such as n_{tree} and m_{try}. Using the brute force method, the model was tested using several values which are 250, 300, 350, 400, 450, 500, 550, 600, 800, 1000 and 2000. The result of the experiment displayed n_{tree} = 500 with the highest accuracy. The experiment results to find the best n_{tree} can be seen in table 9.

Table 9. Experiment result on best n_{tree} parameter

n_{tree}	Accuracy		
	Min	Mean	Max
250	0.6053	0.6273	0.6447
300	0.6086	0.6280	0.6480
350	0.6086	0.6293	0.6513
400	0.6066	0.6290	0.6546
450	0.6151	0.6309	0.6546
500	0.6131	0.6303	0.6579
550	0.6066	0.6309	0.6546
600	0.6086	0.6303	0.6579
800	0.6066	0.6283	0.6579
1000	0.6033	0.6290	0.6546
2000	0.6066	0.6286	0.6513

The experiment to improve Random Forest model prediction continues by testing on m_{try} parameter. The tuning experiment using a grid search method with number 1 to 10 on a m_{try} parameter in the training model. The experiment result is m_{try} = 3 gives the highest prediction accuracy 65,36%. The experiment result on m_{try} parameter can be seen in table 10 and fig. 4.

Table 10. Experiment result on best mtry parameter

m_{try}	Accuracy	Kappa
1	63,91%	0.4100
2	65,06%	0.4469
3	65,36%	0.4543
4	65,33%	0.4556
5	65,03%	0.4507
6	64,77%	0.4477
7	64,54%	0.4440
8	64,80%	0.4483
9	64,80%	0.4485
10	64,83%	0.4493

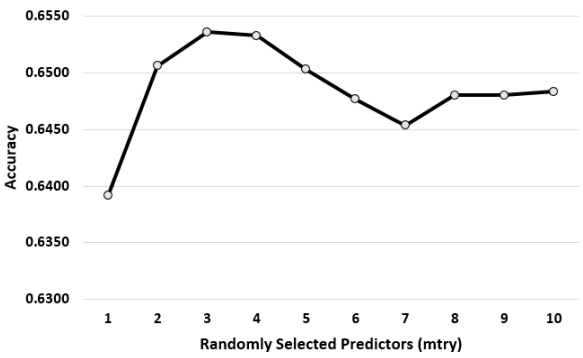

Fig.4. m_{try} parameter plot on Random Forest.

Based on the tuned parameters experiment, the Random Forest model was re-trained and tested with testing data again. The prediction accuracy with tuned parameters using Random Forest model is 68,55% with Kappa value 0.5005.

3. Extreme Gradient Boosting Model

The last algorithm used in this experiment is Extreme Gradient Boosting. With 10 fold cross-validation, the first step in this experiment is to get the best iteration with the lowest m-error. Based on the experiment run, 28 iteration is the best with the lowest testing m-error with value 0,3526. The m-error plot on each iteration of training and testing data can be seen on fig. 5 and fig. 6.

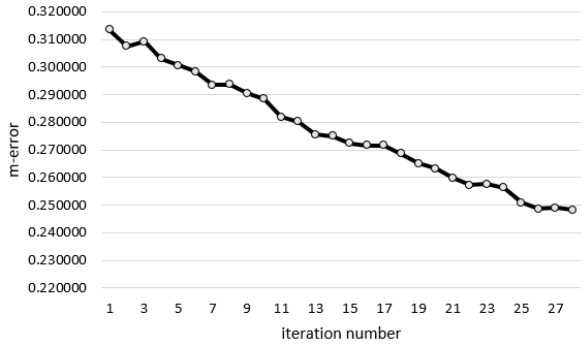

Fig.5. Training m-error plot on Extreme Gradient Boosting.

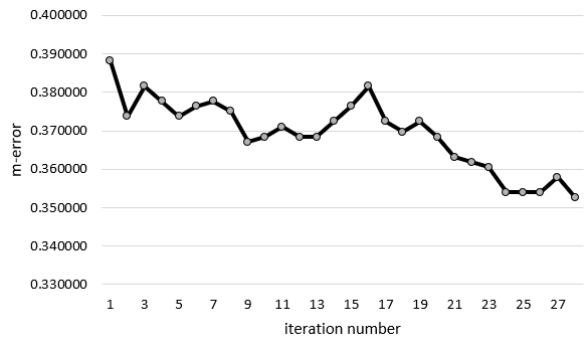

Fig.6. Testing m-error plot on Extreme Gradient Boosting.

The trained model then tested using testing data to generate the prediction accuracy of Extreme Gradient Boosting algorithm. Using a confusion matrix, the accuracy of Extreme Gradient Boosting model is 64,74% with Kappa value 0.4375. The confusion matrix result can be seen in table 11.

Table 11. Confusion matrix of Extreme Gradient Boosting validation model

	Class	Output Class		
		A	D	H
Target	A	149	58	27
	D	23	60	43
	H	41	76	283

Extreme Gradient Boosting algorithm has the lowest accuracy between the three algorithms used in this research. Therefore the algorithm needs to be tuned. The tuning process will use mlr package from R library by creating a learner on each model training. Initial parameter defined in the tuning process is objective, evaluation metric, and nrounds. Table 12 displays the description of Extreme Gradient Boosting algorithm tuning.

Table 12. Extreme Gradient Boosting Tuning Parameters

Learner Initial Parameters		
objective	multi:softprob	
eval_metric	merror	
nrounds	250	
Parameters for Tuning		
Parameters	Lower	Upper
nrounds	200	600
max_depth	3	20
eta	0.001	0.5
lambda	0.55	0.6
subsample	0.1	0.8
min_child_weight	1	5
colsample_bytree	0.2	0.8

Tuning process will use 10 fold cross-validation and iterated as much as 100 times. The result of this tuning process displayed in table 13.

Table 13. Extreme Gradient Boosting Tuning Result

Parameters	
nrounds	204
max_depth	17
Eta	0.0178
lambda	0.5750
subsample	0.6610
min_child_weight	1,05
colsample_bytree	0.7580
test_mean_accuracy	67,0395%

Tuned model then tested and generated prediction accuracy as high as 67,89% with Kappa value 0.4867. The confusion matrix of tuned Extreme Gradient Boosting model can be seen in table 14.

Table 14. Confusion matrix of Extreme Gradient Boosting tuned model

	Class	Output Class		
		A	D	H
Target	A	145	50	23
	D	28	74	33
	H	40	70	297

Table 15. Model performance evaluation

Evaluation Metrics	**C5.0**	**Random Forest**	**XGBoost**
Sensitivity	0.5837	0.6545	0.6345
Specificity	0.8078	0.8381	0.8295
Precision	0.6053	0.6652	0.6477
Recall	0.5837	0.6545	0.6345
F1	0.5506	0.6546	0.6348
Kappa	0.4258	0.5123	0.4867
Accuracy	0.6487	0.6855	0.6789

To summarize the performance of each final model algorithm, table 15 and fig. 7 display the comparison with detail performance evaluation.

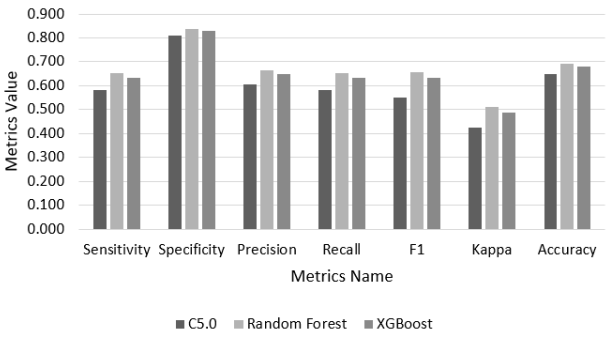

Fig.7. Evaluation Metrics Chart.

V. Conclusion

In this research, we developed a prediction model using the Decision Tree based machine learning to predict the output of the English Premier League using historical match statistic data. Feature HTHG, HTAG, HS, AS, HST, AST, HF, AF, HC, and HY is the best feature combinations to optimize the model prediction accuracy. The accuracy of C5.0, Random Forest, and Extreme Gradient Boosting consecutively is 64,87%, 68,55%, and 67,89%. Although the accuracy of the model using decision tree based is not good enough compared to the model prediction developed by Bayesian network algorithm or artificial neural network algorithm, it still has a better performance compared to SVM with accuracy 53,3% or logistic regression which only generated 2 output class with accuracy 69,5%. We concluded that the tuning method can be used on the model to improve the accuracy of prediction.

In this research, the random forest algorithm and C5.0 performed better than the extreme gradient boosting algorithm before the parameters were tuned. It happened due to the algorithm of extreme gradient boosting being overfit and the trained model only had a small dataset with few features to learn. In the future, we could further develop this research by combining datasets to improve the accuracy with more features that have significant relevance, also by improving the feature selection method to maximize the feature selection process.

References

[1] W. J. Murray and B. Murray, *The worlds game: a history of soccer*, vol. 14. Urbana: University of Illinois Press, 1998.

[2] D. Prasetio and Harlili, "Predicting football match results with logistic regression," in *4th IGNITE Conference and 2016 International Conference on Advanced Informatics: Concepts, Theory, and Application, ICAICTA 2016*, 2016.

[3] M. Faculty, A. Yezus, and A. Igoshkin, "Predicting outcome of soccer matches using machine learning," *Saint-petersbg. Univ.*, 2014.

[4] C. P. Igiri and E. O. Nwachukwu, "An Improved Prediction System for Football a Match Result," *IOSR J. Eng.*, vol. 04, no. 12, pp. 12–20, 2014.

[5] C. P. Igiri, "Support Vector Machine–Based Prediction System for a Football Match Result," *IOSR J. Comput. Eng. Ver. III*, vol. 17, no. 3, pp. 2278–661, 2015.

[6] N. Razali, A. Mustapha, F. A. Yatim, and R. Ab Aziz, "Predicting Football Matches Results using Bayesian Networks for English Premier League (EPL)," in *IOP Conference Series: Materials Science and Engineering*, 2017, vol. 226, no. 1.

[7] V. Chaurasia, "Early Prediction of Heart Diseases Using Data Mining," *Caribbean. J. Sci. Technol.*, vol. 1, pp. 208–217, 2013.

[8] A. Zakerian, A. Maleki, Y. Mohammadnian, and T. Amraee, "Bad data detection in state estimation using Decision Tree technique," in *2017 25th Iranian Conference on Electrical Engineering, ICEE 2017*, 2017.

[9] P. Carmona, F. Climent, and A. Momparler, "Predicting failure in the U.S. banking sector: An extreme gradient boosting approach," *International Review of Economics and Finance*, 2018.

[10] J. Tang, S. Alelyani, and H. Liu, "Feature Selection for Classification: A Review," *Data Classif. Algorithms Appl.*, 2014.

[11] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, 1997.

[12] F. Gorunescu, "Data mining: Concepts, models and techniques," *Intell. Syst. Ref. Libr.*, 2011.

[13] P. Refaeilzadeh, L. Tang, and H. Liu., "'Cross-Validation.,'" in *Encyclopedia of database systems*, 2009.

[14] S. PANG and J. GONG, "C5.0 Classification Algorithm and Application on Individual Credit Evaluation of Banks," *Syst. Eng. - Theory Pract.*, 2009.

[15] T. Bujlow, T. Riaz, and J. M. Pedersen, "A method for classification of network traffic based on C5.0 machine learning algorithm," in *2012 International Conference on Computing, Networking, and Communications, ICNC'12*, 2012.

[16] L. Breiman, "Random Forrest," *Mach. Learn.*, 2001.

[17] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R news*, 2002.

[18] T. Chen, G. Cowan, C. Germain, I. Guyon, B. Azs Kégl, and D. Rousseau, "Higgs Boson Discovery with Boosted Trees," in *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, 2014.

[19] T. Chen and C. Guestrin, "XGBoost: Reliable Large-scale Tree Boosting System," in *LearningSys*, 2016.

Authors' Profiles

Yoel F. Alfredo is a student of the graduate program for Master in Computer Science BINUS University. He received his bachelor degree from BINUS University in 2016 with System Information major. He has worked as a Product Manager in the startup industry for 3 years. His research interests are machine learning, big data, and artificial intelligence.

Sani M. Isa is a lecturer and researcher in the Computer Science Department, BINUS Graduate Program - Master of Computer Science. He has numerous experience in teaching and research in remote sensing and biomedical engineering areas. He got his doctoral degree in Computer Science from the University of Indonesia. He is also received his master degree in Computer Science from the

University of Indonesia as well as a bachelor degree from Padjadjaran University, Bandung, Indonesia.

How to cite this paper: Yoel F. Alfredo, Sani M. Isa, "Football Match Prediction with Tree Based Model Classification", International Journal of Intelligent Systems and Applications(IJISA), Vol.11, No.7, pp.20-28, 2019. DOI: 10.5815/ijisa.2019.07.03

