Computer Science and Information Technologies
Vol. 6, No. 2, July 2025, pp. 159~168
ISSN: 2722-3221, DOI: 10.11591/csit.v6i2.p159-168

Effects of hyperparameter tuning on random forest regressor in the beef quality prediction model

Ridwan Raafi'udin[1,4], Yohanes Aris Purwanto[2], Imas Sukaesih Sitanggang[1], Dewi Apri Astuti[3]
[1]Department of Computer Science, IPB University, Bogor, Indonesia
[2]Department of Mechanical and Biosystem Engineering, IPB University, Bogor, Indonesia
[3]Department of Nutrition and Feed Technology, IPB University, Bogor, Indonesia
[4]Department of Informatics, UPN Veteran Jakarta, Jakarta, Indonesia

Article Info

Article history:

Received Aug 12, 2024
Revised Mar 11, 2025
Accepted May 23, 2025

Keywords:

Beef quality prediction
Hyperparameter tuning
Random forest regressor
RandomizedSearchCV
Spectroscopy

ABSTRACT

Prediction models for beef meat quality are necessary because production and consumption were significant and increasing yearly. This study aims to create a prediction model for beef freshness quality using the random forest regressor (RFR) algorithm and to improve the accuracy of the predictions using hyperparameter tuning. The use of near-infrared spectroscopy (NIRS) in predicting beef quality is an easy, cheap, and fast technique. This study used six meat quality parameters as prediction target variables for the test. The R^2 metric was used to evaluate the prediction results and compare the performance of the RFR with default parameters versus the RFR with hyperparameter tuning (RandomSearchCV). Using default parameters, the R-squared (R^2) values for color (L*), drip loss (%), pH, storage time (hour), total plate colony (TPC in cfu/g), and water moisture (%) were 0.789, 0.839, 0.734, 0.909, 0.845, and 0.544, respectively. After applying hyperparameter tuning, these R^2 scores increased to 0.885, 0.931, 0.843, 0.957, 0.903, and 0.739, indicating an overall improvement in the model's performance. The average performance increase for prediction results for all beef quality parameters is 0.0997 or 14% higher than the default parameters.

Corresponding Author:

Yohanes Aris Purwanto
Department of Mechanical and Biosystem Engineering, IPB University
Bogor, Indonesia
Email: arispurwanto@apps.ipb.ac.id

1. INTRODUCTION

Meat as a source of protein has been widely consumed by humans in recent decades, and its consumption is increasing [1]. Beef is the choice of food product to meet the need for animal protein because it has a high protein content [2]. However, beef is also a food product that is easily perishable and spoiled [3]. Under specific temperature and environmental conditions, microbial growth accelerates, leading to faster beef spoilage [4], [5]. Pathogenic bacteria present in meat, and their number exceeding safe thresholds can cause consumers to become sick [6], [7]. Near-infrared spectroscopy (NIRS) technology can be used to detect the composition contained in beef [8], [9]. NIRS can detect several molecular contents in beef, such as chemical components, technological parameters or electronic equipment, mineral contents, quality traits, fatty acids, and many more [10]–[12].

The freshness quality of beef can change so quickly that a tool or method is needed to predict the freshness quality quickly and accurately. A method that can be used to do modeling is using machine learning [13]. Predictive modeling of meat quality attributes from human sensing like tenderness, juiciness, and flavor has been done with machine learning [14]. The random forest algorithm [15] was used as the main model in this study. There have been many implementations of machine learning in predicting or classifying the type and quality of beef [16], [17]. The random forest regressor (RFR) effectively performs the real-time prediction of pH values in beef within a beef freshness monitoring system [18].

In this study, beef quality prediction uses six quality parameters: color with lightness value (L*), drip loss in percentage, pH value, storage time in the hour unit, total plate colony (TPC) in the colony forming unit per gram (cfu/g), and water moisture value in the percentage. For the experimental scenario, the RFR algorithm was used with a default configuration [19], and then it tried to improve the algorithm's performance by performing hyperparameter tuning [20], [21]. The hyperparameter tuning method used is RandomizedSearchCV [22].

2. METHOD

2.1. Dataset

This study used a dataset of beef quality parameters from previous research [23]. The object examined in the study was fresh beef. The data acquisition process was carried out in two ways: a destructive method using laboratory tools and a non-destructive method using NIRS sensors. The destructive method produced data on the value of meat quality parameters, which became the target variables of the modeling. The non-destructive method generated spectroscopy data that served as training data in modeling. An example of meat quality parameter data can be observed in Table 1. These data have 6 data columns according to the parameters to be predicted and 80 rows of data.

Spectroscopic data are similar to signal data, but in this study the spectroscopy data is already in the form of a spreadsheet file. The data consists of 136 columns, with the column name being the wavelength value of the sensor in nanometers (nm) and consists of 720 rows data. This spectroscopy data was obtained from 80 samples that were scanned nine times, and an example of spectroscopic data can be seen in Table 2 and visualized as shown in Figure 1.

Table 1. Example data from laboratory [23]

No.	StorageTime (hour)	DripLoss (%)	Color (L*)	pH	WaterMoisture (%)	TPC (cfu/g)
1	0	0.00%	29.09	5.45	76.78%	21,802.18
2	1	3.97%	32.48	5.52	76.92%	72,895.38
3	2	6.46%	35.68	5.32	75.51%	110,204.67
…	…	…	…	…	…	…
80	7	17.04%	31.27	5.59	74.86%	2,459,801.15

Table 2. NIRS data example

Hour	Wavelength (nm)				
	2556.24	2539.35	…	1351.35	1346.61
0	2.27	2.37	…	1.41	1.35
1	1.96	2.08	…	1.72	1.95
2	1.78	1.86	…	1.62	1.53
3	2.12	2.22	…	1.89	1.88
4	2.11	2.28	…	1.85	1.77
5	1.71	1.86	…	2.01	2.11
6	2.13	2.30	…	2.86	2.55
7	2.81	2.91	…	3.79	4.01

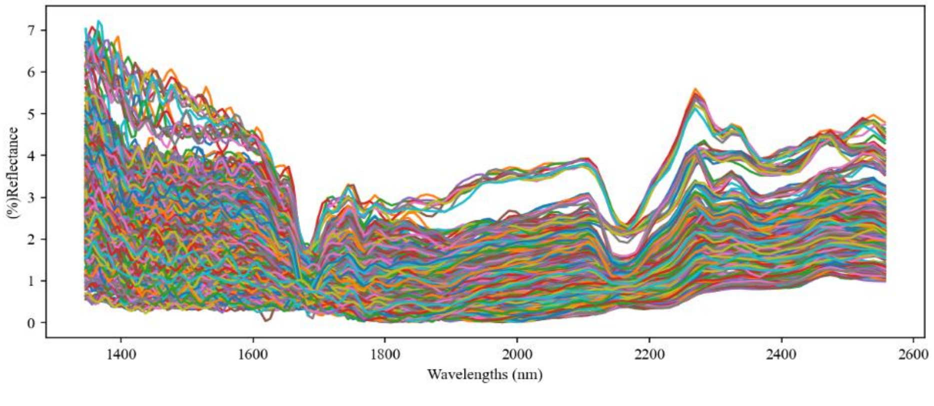


Figure 1. NIRS data plotting

2.2. Experimental scenario

In this study, two scenarios were proposed involving RFR with default parameters and the application of hyperparameters, as shown in Figure 2. The modeling process was carried out alternately on each meat quality parameter. A comparative distribution of training and testing data of 7 to 2, as shown in Table 3.

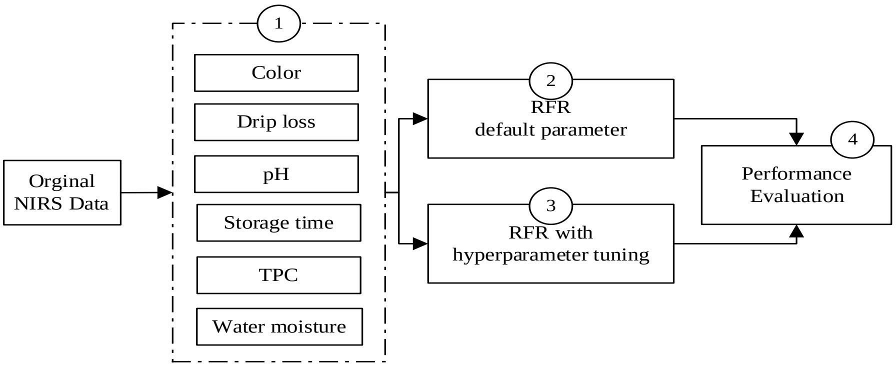


Figure 2. Experimental scenario

Table 3. Dataset distribution

Dataset	Amount of data	Percentage (%)
Training data	560	77.8
Testing data	160	22.2
Sum	720	100.0

2.3. Hyperparameter tuning configuration

In this study, the RandomizedSearchCV method was used [19]. RandomizedSearchCV provides both 'fit' and 'score' methods. Additionally, it supports 'score_samples', 'predict', 'predict_proba', 'decision_function', 'transform', and 'inverse_transform', provided these methods are available in the estimator utilized. The estimator's parameters utilized for implementing these techniques are fine-tuned through cross-validated exploration across various parameter configurations. Unlike GridSearchCV, which tests every parameter value, RandomizedSearchCV selects a predetermined number of parameter configurations randomly from specified distributions. The quantity of configurations tested is determined by n_iter. When all parameters are listed, sampling without repetition occurs. Conversely, if any parameter is defined as a distribution, sampling with replacement is employed. It's advisable to utilize continuous distributions for continuous parameters [22].

For the use of RFR by default, there are still algorithm parameters set, while the default parameter settings can be seen in Table 4. Meanwhile, for hyperparameter setup, there is actually no standard reference for how many parameter combinations was used, but usually, the more parameter combinations are used, the longer the training process in machine learning. The setup for the combination of hyperparameter tuning in this study can be seen in Table 5.

Table 4. Default parameter of RFR [24]

Parameter	Data type	Default value
n_estimators		100
criterion		squared_error
max_depth		None
min_samples_split		2
min_samples_leaf		1
min_weight_fraction_leaf	float	0.0
max_features	int or float	1.0
max_leaf_nodes	int	None
min_impurity_decrease	float	0.0
bootstrap	bool	True
oob_score	bool or callable	False
n_jobs	int	None
random_state	int	None
verbose	int	0
warm_start	bool	False
ccp_alpha	non-negative float	0.0
max_samples	int or float	None
monotonic_cst	array-like of int of shape (n_features)	None

Table 5. Hyperparameter configuration

Parameter	Values
n_estimators	[200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]
max_depth	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None]
max_features	['log2', 'sqrt', None]
bootstrap	[True, False]
min_samples_split	[2, 5, 10]
min_samples_leaf	[1, 2, 4]

2.4. Model evaluation

This modeling was evaluated using root mean square error (RMSE) and R-squared (R^2) to measure its performance. RMSE and R^2 are used to assess how well the model predicts data. The selection of RMSE and R^2 as evaluation metrics is based on their ability to provide comprehensive understanding of the model's prediction accuracy. RMSE was chosen because it can show how far the predicted value deviates from actual value in same unit as data, making it easy to interpret. RMSE is more sensitive to large errors because it calculates the square mean of errors, which makes it suitable for detecting models with significant prediction errors [25].

Meanwhile, R^2 was chosen because it was able to show the proportion of variance from the data that the model could explain. R^2 is a commonly used evaluation metric in regression because it gives an idea of how well the model fits against the data [26]. The higher the R^2 value, the better the model is able to account for variations in the data. The RMSE formula can be seen in (1) and formula R^2 can be seen in (2).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{1}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{2}$$

Where y_i = actual data, $\hat{y}_i$ = predicted data, $\bar{y}$ = mean of actual data, and n = number of data.

3. RESULTS AND DISCUSSION

The model was evaluated using RMSE and R^2 to assess its predictive performance. Hyperparameter tuning resulted in the best set of configurations, as indicated by the smallest RMSE value and greater R^2 testing results for each meat quality parameter. As a comparison and to see the effect of hyperparameter tuning, the modeling accuracy results compare the results of RFR with the default configuration and the results of hyperparameter tuning. The results of the default configuration and using hyperparameter tuning can be seen in Table 6. Based on the data in Table 6, it can be seen that the performance of the modeling improved from the one using the default configuration to the result of hyperparameter tuning. The increase can be seen from the smaller RMSE value and the increasing R^2 value. The increase in performance in the increasing R^2 value with an average increase of 0.0997 or 14% can be seen in Table 7. R^2 default shows the results of the model evaluation for all meat quality parameters, while R^2 hyperparameter is the evaluation results of the random forest model that has implemented hyperparameter tuning.

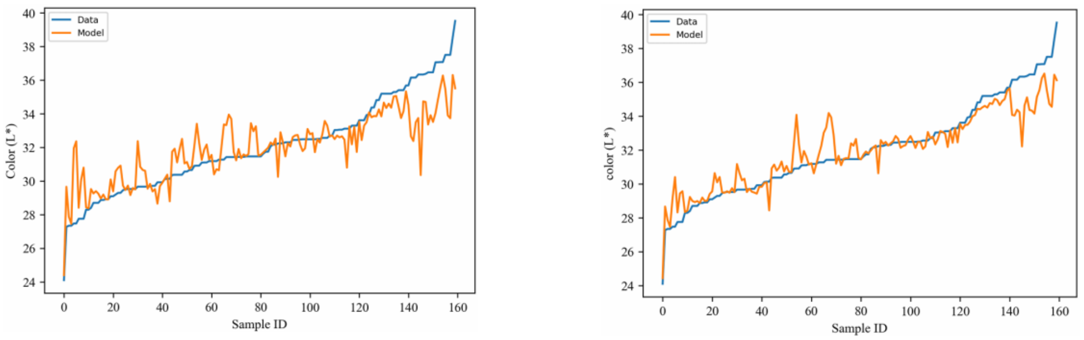
Table 6. Prediction results

Beef quality parameter	Default configuration		Hyperparameter tuning	
	RMSE	R^2	RMSE	R^2
color	2.030	0.789	1.110	0.885
drip loss	0.0007	0.839	0.0003	0.931
pH	0.010	0.734	0.006	0.843
storage time	0.480	0.909	0.227	0.957
TPC	68104242089.277	0.845	42401651948.646	0.903
water moisture	0.0008	0.544	0.0005	0.739

Table 7. R-squared improvement

Beef quality parameters	R^2 default	R^2 hyperparameter	Improvement	
			ΔR2	(%)
Color	0.789	0.885	0.096	12
Drip loss	0.839	0.931	0.092	11
pH	0.734	0.843	0.109	15
Storage time	0.909	0.957	0.048	5
TPC	0.845	0.903	0.058	7
water moisture	0.544	0.739	0.195	36

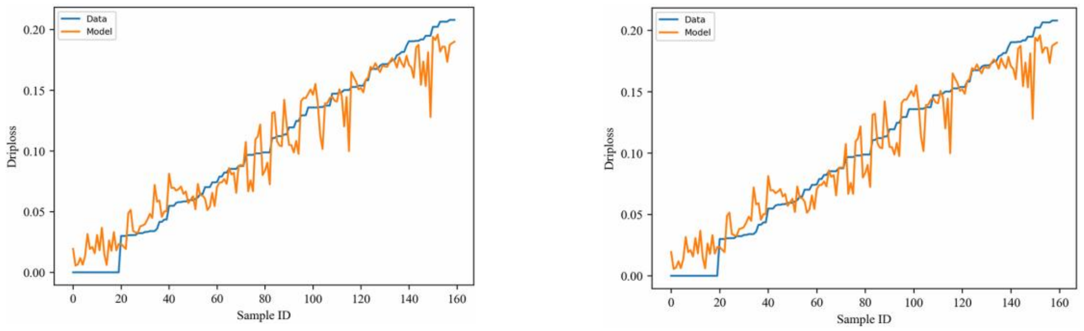
The results of the improvement using hyperparameter tuning on the RFR are visualized to compare the results by using the default parameters and by using hyperparameter tuning. For the meat color quality parameters, the prediction comparison results can be seen in Figures 3 to 8. In Figure 3, we can see a comparison of the prediction results for the color parameter, where in Figure 3(a), the yellow color, which represents the prediction data, appears quite far from the actual data line, which is blue. Meanwhile, the results of the application of hyperparameter tuning showed an increase in accuracy by seeing that the orange line in Figure 3(b) is closer to the blue line, which means that the prediction error value is smaller with a larger R^2 value. The value of R^2 using the default parameter is 0.789 and then increases to 0.885 after applying hyperparameter tuning. Overall, it can be concluded that hyperparameter tuning has a positive impact on the improvement of the model's accuracy in predicting the color (L*) value, as seen from the increase in the R^2 value and the shape of the prediction line that is closer to the data pattern. An increase in R^2 of 0.096 or an increase of 12%.



(a) (b)

Figure 3. Comparison of color (L*) prediction results: (a) default parameters and (b) hyperparameter tuning

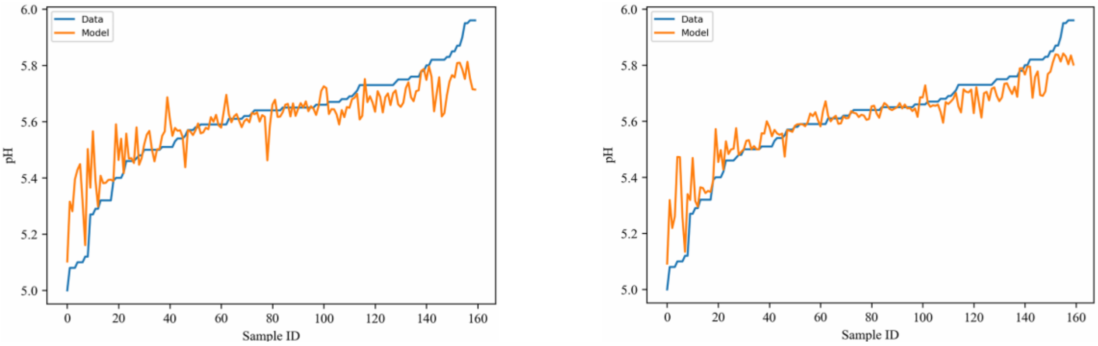
The prediction results for the drip loss quality parameter are shown in Figure 4, where Figure 4(a) shows a graph of the prediction results with an R^2 value of 0.839 with the orange line still looking fructuous, which is still far from the actual data on the blue line. While in Figure 4(b), which has applied hyperparameter tuning, it can be seen that there is a change where the orange line is closer to the blue color of the actual data with an R^2 value of 0.931 or 0.092 greater than before. In the prediction of drip loss quality, R^2 increased by 11%, and it was also seen that there was less deviation of the orange color line compared to Figure 4(a).



(a) (b)

Figure 4. Comparison of drip loss (%) prediction results: (a) default parameters and (b) hyperparameter tuning

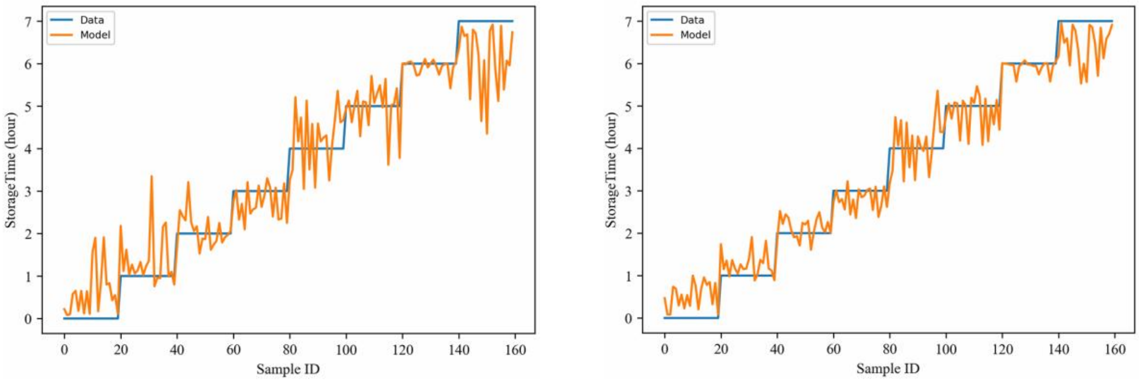
The comparison of prediction results in the pH quality parameter can be seen in Figure 5, where Figure 5(a) is a graph that shows the prediction results using the default parameter with an R^2 value of 0.734, which is considered small because it is below 0.8. While Figure 5(b) shows the results of pH quality prediction after applying hyperparameter tuning with an R^2 value of 0.843, with this value, it can be considered that hyperparameter tuning can have a significant impact. An increase in the R^2 value of 0.109 or 15%. This increase can smooth the orange line, which is the prediction data, and the values get closer to the actual data.



(a) (b)

Figure 5. Comparison of pH value prediction results: (a) default parameters and (b) hyperparameter tuning

The prediction of the storage time quality parameter is shown in Figure 6, specifically in the actual data time prediction data; it looks like a ladder because there is a group of data in the same time period, namely at the same hour as the difference of 1 hour to the data group before and after. Actual data is a data interval per 1 hour according to the data collection technique. In Figure 6(a), it can be seen that there are still many orange lines that are towering or too low down with an R^2 value of 0.909. The default result of this parameter can be considered very high because it has exceeded 0.9, but hyperparameter tuning is still applied to see the performance results. In Figure 6(b), the graph shows the prediction results after applying hyperparameter tuning, with the R^2 result being 0.957. With the R^2 value, this model can be said to be close to perfect in predicting quality with the storage time parameter. The difference in R^2 increases using the default parameter, and after applying the hyperparameter tuning, it is 0.048 or 5%.

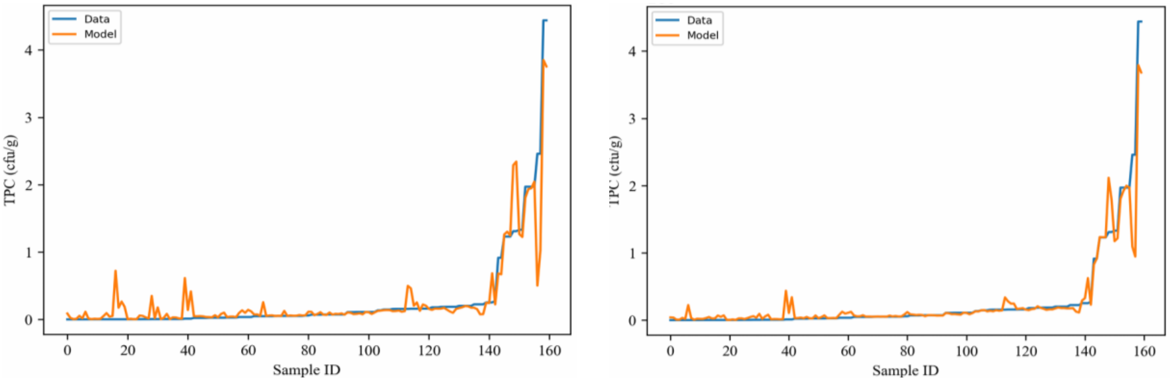


(a) (b)

Figure 6. Comparison of storage time (hour) prediction results: (a) default parameters and (b) hyperparameter tuning

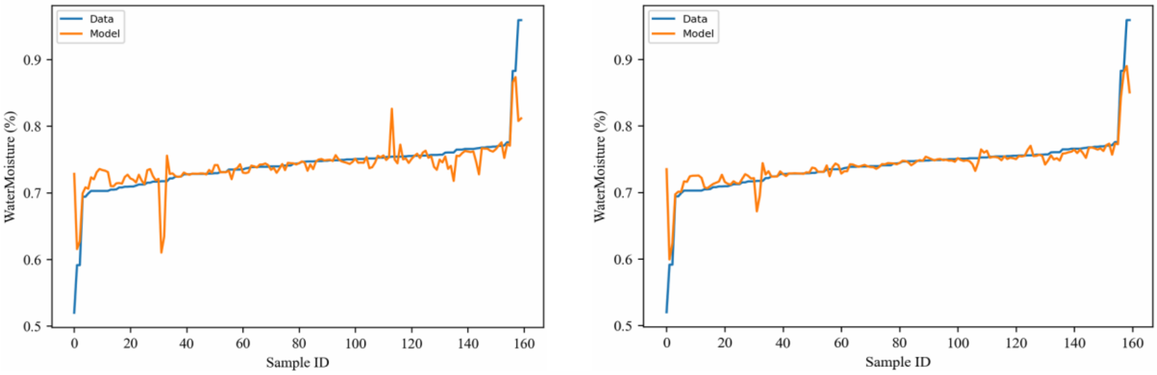
The next prediction result is about biological meat quality parameters, namely estimating the number of bacterial colonies using the TPC method with colony forming units per gram (cfu/g) without paying attention to the type of microbiology. In Figure 7(a), the results of the match between the actual data and the predicted data are shown. There, it can be seen that the prediction results are quite good for small values but seem to be much different in the actual data with high values. This is because, indeed, the modeling accuracy with an R^2 value is 0.845 for modeling with default parameters, and the distribution of actual data is more at small values. While the accuracy results after applying the R^2 value tuning hyperparameter is 0.903, as shown in Figure 7(b), this result gives an increase of 0.058 or 7%.

The next result is the prediction for the water moisture meat quality parameter shown in Figure 8. As for the prediction results that use the default parameter with the R^2 result of 0.544, which can be seen in Figure 8(a), this value can be said to be a poor prediction modeling result because the R^2 value is only around 0.5. The application of hyperparameter tuning is one of the techniques to improve modeling accuracy; the results of its application can be seen in Figure 8(b) with an R^2 value of 0.739 or an increase from the previous 0.195 or an increase of 36%. With the R^2 increase value, the water moisture prediction modeling after applying hyperparameter tuning has the highest increase among other quality parameter predictions. In Figure 8(b), which applies hyperparameter tuning, the orange line appears to be smoother and very close to the blue line, compared to Figure 8(a), which only uses the default parameters.



(a) (b)

Figure 7. Comparison of TPC (cfu/g) prediction results: (a) default parameters and (b) hyperparameter tuning



(a) (b)

Figure 8. Comparison of water moisture (%) prediction results: (a) default parameters and (b) hyperparameter tuning

Based on the data pairs (a) and (b) visualized in Figures 3 to 8, it can be seen that the orange line in each figure (b) can better follow the blue line pattern. The closer the orange line is to the blue line, the greater the R^2 value or the smaller the error value of the prediction results. The increase in the R^2 value means that the effect of hyperparameter tuning on the RFR can run well. The purpose of hyperparameter tuning is to select the best set or set of parameters in the RFR. The best parameter is also shown as the highest R^2 value from the random parameter selection process. The best parameters produced from the hyperparameter tuning process are shown in Table 8. The results of the hyperparameter tuning show several different parameters for each meat quality prediction target. The different parameters include n_estimators, max_features, max_depth, and min_samples_split. However, there is no difference between min_samples_leaf and bootstrap parameters, which indicates that they have no effect on the selection process in hyperparameter tuning.

Table 8. Best parameters from hyperparameter tuning

RFR parameter	Values for beef quality parameters					
	Color	Dripless	pH	Storage time	TPC	Water moisture
n_estimators	1800	1600	400	1800	2000	600
max_features	log2	sqrt	log2	log2	sqrt	log2
max_depth	80	30	30	None	80	60
min_samples_split	2	2	2	5	2	2
min_samples_leaf	1	1	1	1	1	1
bootstrap	False	False	False	False	False	False

4. CONCLUSION

The results of this study prove that the use of hyperparameter tuning can improve the performance of the RFR algorithm. The performance improvement was measured by evaluating an increase in R^2 values for all beef freshness quality prediction targets. The average increase in R^2 from all prediction results of meat quality parameters is 0.0997, or an increase of 14% from the R^2 value with the default parameter. The study's results on the application of hyperparameter tuning show that not all parameter configurations affect meat quality prediction modeling. This study showed that the "min_samples_leaf" or the minimum number of samples that must be present in each leaf node and bootstrap parameters did not show any difference in the results of meat quality prediction, which means that the RFR parameter did not affect the hyperparameter tuning process. So, in the application of the prediction model, RFR and hyperparameter tuning must be adjusted to the parameters generated by hyperparameter tuning to be re-trained in modeling. Each of the highest prediction results on all meat quality parameters was affected by different RFR parameters according to the results of the best parameter output from the hyperparameter tuning iteration process. For future work, hyperparameter tuning methods such as GridCV can also be used to explore further the ability of hyperparameter tuning to improve algorithm performance. The results of this study can also be continued by combining hyperparameter tuning with several preprocess methods, such as feature selection and NIRS data transformation. The combination of several methods, each of which has been proven to improve the performance of the algorithm, is expected to be able to improve more. An example of applying combinations is combining starting from feature selection followed by the transformation data method, then applying hyperparameter tuning.

FUNDING INFORMATION

This research used the author's personal funds.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Ridwan Raafi'udin	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Yohanes Aris Purwanto		✓			✓	✓		✓	✓	✓	✓	✓	✓	✓
Imas Sukaesih Sitanggang	✓		✓	✓	✓				✓	✓	✓	✓		
Dewi Apri Astuti				✓	✓	✓	✓	✓		✓		✓		✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available on request from the corresponding author, YAP.

REFERENCES

[1] M. Lima, R. Costa, I. Rodrigues, J. Lameiras, and G. Botelho, "A narrative review of alternative protein sources: highlights on meat, fish, egg and dairy analogues," *Foods*, vol. 11, no. 14, 2022, doi: 10.3390/foods11142053.

[2] M. Molfetta *et al.*, "Protein sources alternative to meat: state of the art and involvement of fermentation," *Foods*, vol. 11, no. 14, 2022, doi: 10.3390/foods11142065.

[3] B. Fletcher *et al.*, "Advances in meat spoilage detection: a short focus on rapid methods and technologies," *CYTA-Journal of Food*, vol. 16, no. 1, pp. 1037–1044, 2018, doi: 10.1080/19476337.2018.1525432.

[4] V. Tesson, M. Federighi, E. Cummins, J. de O. Mota, S. Guillou, and G. Boué, “A systematic review of beef meat quantitative microbial risk assessment models,” *International Journal of Environmental Research and Public Health*, vol. 17, no. 3, 2020, doi: 10.3390/ijerph17030688.

[5] W. Barragán-Hernández, L. Mahecha-Ledesma, J. Angulo-Arizala, and M. Olivera-Angel, “Near-infrared spectroscopy as a beef quality tool to predict consumer acceptance,” *Foods*, vol. 9, no. 8, 2020, doi: 10.3390/foods9080984.

[6] G. Ripoll *et al.*, “Near-infrared reflectance spectroscopy for predicting the phospholipid fraction and the total fatty acid composition of freeze-dried beef,” *Sensors*, vol. 21, no. 12, 2021, doi: 10.3390/s21124230.

[7] A. Sahar *et al.*, “Online prediction of physico-chemical quality attributes of beef using visible-near-infrared spectroscopy and chemometrics,” *Foods*, vol. 8, no. 11, 2019, doi: 10.3390/foods8110525.

[8] N. Patel, H. Toledo-Alvarado, A. Cecchinato, and G. Bittante, “Predicting the content of 20 minerals in beef by different portable near-infrared (NIR) spectrometers,” *Foods*, vol. 9, no. 10, 2020, doi: 10.3390/foods9101389.

[9] S. Savoia *et al.*, “Prediction of meat quality traits in the abattoir using portable and hand-held near-infrared spectrometers,” *Meat Science*, vol. 161, 2020, doi: 10.1016/j.meatsci.2019.108017.

[10] S. Savoia, A. Albera, A. Brugiapaglia, L. Di Stasio, A. Cecchinato, and G. Bittante, “Prediction of meat quality traits in the abattoir using portable near-infrared spectrometers: heritability of predicted traits and genetic correlations with laboratory-measured traits,” *Journal of Animal Science and Biotechnology*, vol. 12, no. 1, 2021, doi: 10.1186/s40104-021-00555-5.

[11] I. M. N. Perez, L. J. P. Cruz-Tirado, A. T. Badaró, M. M. de Oliveira, and D. F. Barbin, “Present and future of portable/handheld near-infrared spectroscopy in chicken meat industry,” *NIR news*, vol. 30, no. 5–6, pp. 26–29, 2019, doi: 10.1177/0960336019861476.

[12] M. Simoni, A. Goi, M. De Marchi, and F. Righi, “The use of visible/near-infrared spectroscopy to predict fibre fractions, fibre-bound nitrogen and total-tract apparent nutrients digestibility in beef cattle diets and faeces,” *Italian Journal of Animal Science*, vol. 20, no. 1, pp. 814–825, 2021, doi: 10.1080/1828051X.2021.1924884.

[13] C. N. Sánchez, M. T. Orvañanos-Guerrero, J. Domínguez-Soberanes, and Y. M. Álvarez-Cisneros, “Analysis of beef quality according to color changes using computer vision and white-box machine learning techniques,” *Heliyon*, vol. 9, no. 7, 2023, doi: 10.1016/j.heliyon.2023.e17976.

[14] T. Qiao, J. Ren, C. Craigie, J. Zabalza, C. Maltin, and S. Marshall, “Quantitative prediction of beef quality using visible and NIR spectroscopy with large data samples under industry conditions,” *Journal of Applied Spectroscopy*, vol. 82, no. 1, pp. 137–144, 2015, doi: 10.1007/s10812-015-0076-1.

[15] G. Biau and E. Scornet, “A random forest guided tour,” *Test*, vol. 25, no. 2, pp. 197–227, 2016, doi: 10.1007/s11749-016-0481-7.

[16] H. Pu, J. Yu, D. W. Sun, Q. Wei, X. Shen, and Z. Wang, “Distinguishing fresh and frozen-thawed beef using hyperspectral imaging technology combined with convolutional neural networks,” *Microchemical Journal*, vol. 189, 2023, doi: 10.1016/j.microc.2023.108559.

[17] R. Kasarda, N. Moravčíková, G. Mészáros, M. Simčič, and D. Zaborski, “Classification of cattle breeds based on the random forest approach,” *Livestock Science*, vol. 267, 2023, doi: 10.1016/j.livsci.2022.105143.

[18] Y. Lin, J. Ma, D. W. Sun, J. H. Cheng, and Q. Wang, “A pH-Responsive colourimetric sensor array based on machine learning for real-time monitoring of beef freshness,” *Food Control*, vol. 150, 2023, doi: 10.1016/j.foodcont.2023.109729.

[19] F. Pedregosa *et al.*, “Scikit-learn: machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[20] P. Probst, M. N. Wright, and A. L. Boulesteix, “Hyperparameters and tuning strategies for random forest,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 3, 2019, doi: 10.1002/widm.1301.

[21] S. Bernard, L. Heutte, and S. Adam, “Influence of hyperparameters on random forest accuracy,” in *Multiple Classifier Systems*, Berlin, Heidelberg: Springer, 2009, pp. 171–180, doi: 10.1007/978-3-642-02326-2_18.

[22] *Scikit Learn*, “RandomizedsearchCV,” *Scikit Learn*. Accessed: May 17, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html.

[23] R. Raafi’udin, Y. A. Purwanto, I. S. Sitanggang, and D. A. Astuti, “Feature selection model development on near-infrared spectroscopy data,” *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 1, 2024, doi: 10.14569/ijacsa.2024.0150163.

[24] *Scikit Learn*, “RandomForestRegressor,” *Scikit Learn*. Accessed: May 17, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html.

[25] D. Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” *PeerJ Computer Science*, vol. 7, pp. 1–24, 2021, doi: 10.7717/PEERJ-CS.623.

[26] A. Chugh, “MAE, MSE, RMSE, coefficient of determination, adjusted R squared—which metric is better,” *Medium*, 2020. Accessed: Nov 11, 2024. [Online]. Available: https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e.

BIOGRAPHIES OF AUTHORS

Ridwan Raafi'udin received a B.Sc. degree from Universitas Pembangunan Nasional Veteran Jakarta in 2009 and an M.Com. degree from Universitas Budi Luhur in 2014, and now a Ph.D. student at IPB University. Currently, he is a lecturer in computer science at Universitas Pembangunan Nasional Veteran Jakarta. His research interests include artificial intelligence, machine learning, deep learning, data mining, and embedded systems. He can be contacted at email: ridwanraafiudin@apps.ipb.ac.id.

Yohanes Aris Purwanto received a B.Eng. from IPB University in 1986, M.Sc. and Ph.D. degrees from the University of Tokyo in 1997 and 2000. Currently, he is a professor at IPB University. His research interests include agricultural engineering, biodiesel production, NIR, partial least squares, spectroscopy bioprocess engineering, PLS biofuels infrared near-infrared spectroscopy, biosystem engineering, non-destructive measurement, fine bubble technology, and postharvest engineering. He can be contacted at email: arispurwanto@apps.ipb.ac.id.

Imas Sukaesih Sitanggang received a Ph.D. degree in computer science from the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, in 2013. She is a lecturer in Department of Computer Science, IPB University, Indonesia. Her main research interests include spatial data mining and smart agriculture. She can be contacted at email: imas.sitanggang@apps.ipb.ac.id.

Dewi Apri Astuti received an animal science degree from Universitas Gadjah Mada in 1984, a master in aminal nutrition degree from IPB University in 1988, and a Ph.D. in nutrition from IPB University in 1995. Currently, she is a professor at IPB University. Her research interests include animal nutrition. She can be contacted at email: dewiaa@apps.ipb.ac.id.