Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Optimization of the Random Forest Algorithm Using Random Search for Potable Water Quality Classification

Adi Fajri Firmansyah[1]*, Basuki Rahmat[2], Muhammad Muharrom Al Haromainy[3]

[1,2,3]Program Studi Informatika, Fakultas Ilmu Komputer, UPN "Veteran" Jawa Timur
21081010175@student.upnjatim.ac.id[1], basukirahmat.if@upnjatim.ac.id[2], muhammad.muharrom.if@upnjatim.ac.id[3]*

Abstract

Potable water suitable for consumption is a fundamental human need that plays a crucial role in health. In Indonesia, challenges regarding water quality and accessibility remain significant due to infrastructure limitations and environmental pollution. This study aims to compare the effectiveness of Random Forest algorithm optimization using Random Search in clean water quality classification. The dataset used is the "Water Quality and Potability" dataset from Kaggle, consisting of 3,276 data, 9 water chemistry parameters, and 1 potability target. All processes were conducted using the Google Colab platform. Evaluation results show that the default Random Forest model achieved an accuracy of 84.12% with a computation time of 3 seconds. The Random Search-optimized model reached 84.38% accuracy in 2 hours 2 minutes and 54 seconds. Overall, hyperparameter optimization can improve classification performance.. However, this improvement was accompanied by a substantial increase in computation time, which should be considered when selecting optimization methods.

Keywords: *Classification, Machine Learning, Random Forest, Random Search, Water Quality.*

1. Introduction

Potable water is a vital necessity for the survival of humans and other living beings. The availability of sufficient and high-quality potable water significantly affects public health and environmental sustainability. Access to potable water has been recognized as a fundamental human right by various global institutions, given the serious consequences of its scarcity, such as increased risk of disease and mortality [1]. In Indonesia, issues regarding the quality and access to potable water remain major challenges, primarily due to geographical conditions, climate change, and environmental pollution. According to the 2021 Household Drinking Water Quality Surveillance, only 11.4% of the population uses piped water as their main source of drinking water, while the rest rely on non-piped sources such as refilled water, dug wells, and bottled water [2]. This preference is influenced by factors such as affordability, availability of drinking water facilities in the surrounding environment, and practicality.

The potability of drinking water is determined by several physical and chemical parameters established in water quality standards. These parameters include pH value, turbidity, and the presence of contaminants such as chloramines, sulfate, and trihalomethanes [3][4]. Although conventional laboratory testing provides accurate results, it typically requires substantial time and resources. With the advancement of technology, artificial intelligence (AI) and machine learning (ML)-based approaches have been adopted to enhance the efficiency and accuracy of water quality assessment. Machine learning algorithms such as Random Forest have proven capable of classifying the potability of water based on various measurable parameters [5][6]. Several studies have shown that Random Forest performs better in water quality classification compared to other algorithms such as Logistic Regression, SVM, and K-NN [7][8]. Moreover, the performance of the Random Forest algorithm can be further improved through hyperparameter optimization, such as tuning the number of trees (n_estimators), maximum tree depth (max_depth), and the number of features considered (max_features) [9].

Previous studies have demonstrated the effectiveness of Random Search in enhancing the performance of Random Forest models. For example, study [10] successfully improved the accuracy of heart failure detection from 77.93% to 85.63% after applying Random Search optimization. Another study on heart disease classification also reported that Random Search optimization led to better model accuracy compared to the default Random Forest configuration [11]. These findings indicate that hyperparameter tuning using Random Search can significantly improve a model's predictive capabilities, with better time efficiency compared to other optimization techniques.

Based on this background, this study aims to implement and evaluate the optimization of the Random Forest algorithm using Random Search in potable water quality classification. The main focus of this research is to analyze the improvement in model performance and its impact on computational time, in order to support data-driven decision-making in assessing water potability.

2. Research methods

This section presents the methodology used in the study, starting from dataset collection to model evaluation. The dataset used is "Water Quality and Potability" in .csv format, which includes various physical parameters to determine water potability. The preprocessing dataset involves data cleaning, transformation through normalization, handling class imbalance, and splitting the data into training and testing sets. A baseline model is first built using the Random Forest algorithm without optimization. Next, hyperparameter tuning is performed using the Random Search method to enhance model performance. Finally, both models are evaluated using accuracy, precision, recall, F1-score, and computation time, and the results are compared to assess the effect of optimization. The complete research workflow is illustrated in Fig. 1, which shows the sequential steps from dataset preparation, preprocessing data, modeling, optimization, to final evaluation.

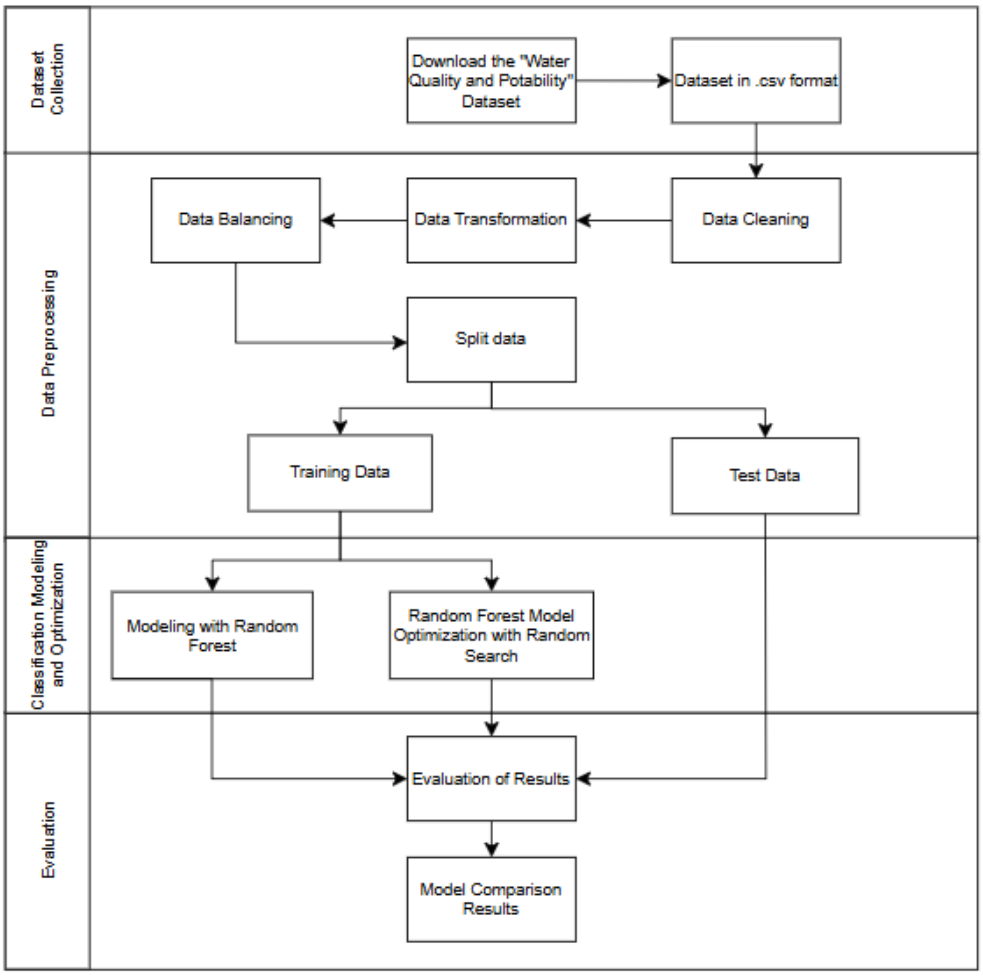

Fig. 1: Research flow diagram

2.1. Data collection

The dataset was obtained from Kaggle, a public platform for data science and machine learning. The dataset, titled "Water Quality and Potability", contains 3,276 records with 9 water quality indicators and 1 target variable. It includes features such as pH, hardness, solids, chloramines, sulfate, conductivity, organic carbon, trihalomethanes, turbidity, and a target label potability, which indicates whether the water is safe to drink (1) or not (0).

2.2. Data Preprocessing

The data preprocessing phase in this study consisted of four key steps: data cleaning, transformation, balancing, and dataset splitting. The first step, data cleaning, addressed missing values that could negatively affect model performance. Missing data were identified using functions such as isnull().sum() and handled using Mean Imputation, where missing entries were replaced with the mean of the corresponding feature using SimpleImputer. This method was chosen for its simplicity and effectiveness on numerical data. The second step, data transformation, aimed to normalize feature scales using the Min-Max Scaling method via MinMaxScaler. Given the varying ranges of feature values, normalization was essential to ensure fair contribution of each feature during model training. This method transformed values into a standardized range between 0 and 1 based on each column's minimum and maximum values.

Next, data balancing was applied to address class imbalance between potable (1) and non-potable (0) samples. The dataset showed a disparity in class distribution, which could lead to biased model predictions. Oversampling was used to increase the number of minority class instances, ensuring a balanced distribution across classes. The data were then shuffled to prevent pattern bias in the training and testing sets. Finally, the dataset was split into training and testing subsets with an 80:20 ratio. The training set was used to build and train

the classification model, while the testing set was reserved for evaluating model performance on unseen data, allowing for an objective assessment.

2.3. Modeling with Random Forest

Random Forest is one of the most popular ensemble learning methods, which combines the predictions of multiple decision trees to improve overall model performance. In this study, Random Forest is used to classify water potability based on various measured parameters. The implementation utilizes the scikit-learn library, which provides default hyperparameter settings that can be used without manual tuning as an initial model configuration. The default configuration includes n_estimators = 100, which defines the number of decision trees in the ensemble, and max_depth = None, allowing each tree to grow without any depth restriction. The minimum number of samples required to split an internal node is set to min_samples_split = 2, while the minimum number of samples required to be at a leaf node is min_samples_leaf = 1. For feature selection during node splitting, max_features = 'sqrt' is used, meaning the number of features considered at each split is the square root of the total number of features. These hyperparameters provide a balance between model complexity and computational efficiency and serve as a baseline before further tuning. Overall, the use of default hyperparameters in Random Forest provides a reliable starting point for evaluating model performance prior to optimization..

2.4. Optimization of the Random Forest Model with Random Search

The optimization of the Random Forest algorithm was carried out through a hyperparameter tuning process. The method employed for tuning was Random Search, an optimization technique that randomly selects combinations of hyperparameters from a predefined search space. This approach is more efficient than exhaustive search methods such as Grid Search, as it evaluates only a subset of all possible combinations. Random Search is particularly well-suited for models like Random Forest, especially when dealing with large datasets or high-dimensional hyperparameter spaces.The search space defined in this study included the following: n_estimators ranging from 100 to 1000 in increments of 100; max_depth values of None, 5, 10, 15, and 20; min_samples_split ranging from 2 to 5; min_samples_leaf ranging from 1 to 4; and max_features set to 'sqrt'. These combinations were randomly evaluated to identify the optimal hyperparameter configuration that yielded the best model performance..

2.5. Evaluation Using Confusion Matrix

The evaluation stage was carried out using a confusion matrix along with performance metrics including accuracy, precision, recall, and F1-score. The results were compared between the Random Forest model with default hyperparameters and the model optimized using Random Search.

3. Results and discussion

In the initial implementation, the Random Forest model was configured using the default hyperparameters provided by the scikit-learn library, serving as a baseline for subsequent performance comparisons. To enhance the model's classification performance, hyperparameter optimization was conducted using the Random Search technique. This method involved the exploration of a predefined hyperparameter space in order to identify the most suitable configuration for the given dataset. The optimization process was carried out using the RandomizedSearchCV class, with 300 randomized iterations and 5-fold cross-validation, resulting in a total of 1,500 model fits. Each combination of hyperparameters was evaluated based on accuracy to determine the optimal setting. The best-performing configuration identified through this process consisted of n_estimators = 400, min_samples_split = 5, min_sample_leaf = 1, max_features = 'sqrt', and max_depth = None. This optimized model demonstrated improved performance compared to the default configuration, thereby confirming the effectiveness of Random Search in identifying a more suitable hyperparameter combination tailored to the characteristics of the dataset.

Each model that has been built is tested using test data to obtain its classification performance objectively. This test aims to determine the level of accuracy of each model in classifying data that has never been seen before. By utilizing test data, a picture is obtained of the extent to which the model is able to generalize to new data, which is an important indicator in assessing the quality and consistency of the resulting classification model. Table 1 shows the accuracy results for each model.

Table 1: Comparison of accuracy results

Scenario	Algorithm	Optimization	Hyperparameter	Accuracy
1	Random Forest	-	default	84.12%
2	Random Forest	Random Search	n_estimators, min_samples_split, min_sample_leaf , max_features, max_depth	84.28%

Table 1 summarizes the accuracy comparison between the default Random Forest model and the one optimized using Random Search. The default model achieved an accuracy of 84.12%, serving as a baseline. After hyperparameter tuning with Random Search adjusting parameters such as n_estimators, min_samples_split, min_samples_leaf, max_features, and max_depth the optimized model reached an accuracy of 84.28%, indicating a slight performance improvement. A more detailed comparison of the classification performance between the Random Forest model with default hyperparameters and the one optimized using Random Search is presented in Fig. 2 and Fig. 3. These figures display the classification reports, including key evaluation metrics such as accuracy, precision, recall, and F1-score for each class. Through this visualization, it is possible to assess the extent to which hyperparameter optimization improves the model's ability to classify data accurately and consistently across different classes.

Akurasi Model Random Forest: 84.12%

Laporan Klasifikasi Random Forest:

	precision	recall	f1-score	support
0.0	0.85	0.84	0.85	419
1.0	0.83	0.84	0.83	381
accuracy			0.84	800
macro avg	0.84	0.84	0.84	800
weighted avg	0.84	0.84	0.84	800

Fig. 2: Random forest classification report

Akurasi Model Random Forest optimasi Random Search: 84.38%

Laporan Klasifikasi Random Search:

	precision	recall	f1-score	support
0.0	0.85	0.86	0.85	419
1.0	0.84	0.83	0.83	381
accuracy			0.84	800
macro avg	0.84	0.84	0.84	800
weighted avg	0.84	0.84	0.84	800

Fig. 3: Random Search classification report

Fig. 4 presents a bar chart illustrating the comparison of accuracy values for each classification model. This visual representation provides a clear overview of the performance differences between the baseline Random Forest model and the model optimized using Random Search. By displaying the accuracy scores side by side, the chart facilitates a direct interpretation of the effectiveness of hyperparameter tuning in improving model performance.

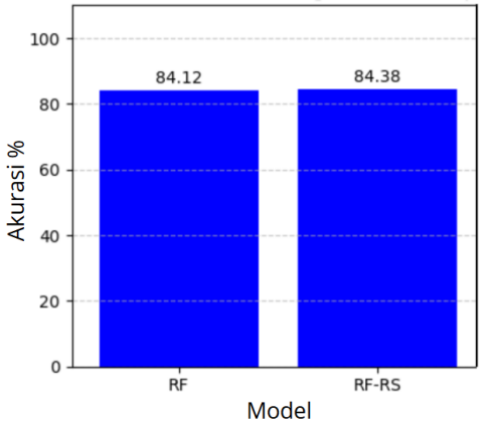

Fig. 4: Model accuracy diagram

The computational time results are presented in Fig. 5, showing a significant difference in execution time among the three classification models used. The Random Forest (RF) model without optimization required the shortest computation time, taking only 3 seconds. In contrast, the Random Forest model optimized using Random Search (RF-RS) required 2 hours, 2 minutes, and 54 seconds to complete the process. This notable discrepancy highlights a trade-off between model performance and computational efficiency, indicating that while hyperparameter optimization may improve classification accuracy, it also incurs a substantial increase in processing time.

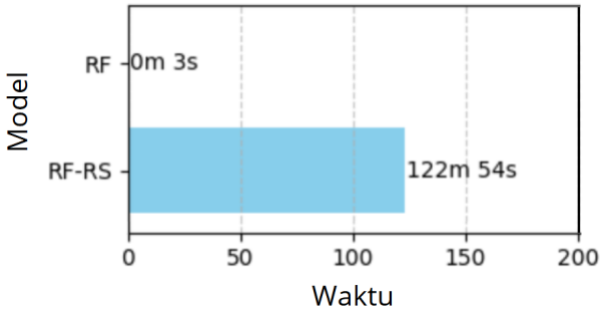

Fig. 5: Compute time comparison

4. Conclusion

The Random Forest model without hyperparameter optimization served as a baseline, achieving an accuracy of 84.12%, a precision of 83.07%, a recall of 83.78%, and an F1-score of 83.42%, with a computational time of only 3 seconds. In comparison, the model optimized using Random Search demonstrated improved performance, achieving an accuracy of 84.38%, a precision of 84.04%, a recall of 82.94%, and an F1-score of 83.49%. Although the hyperparameter search was conducted randomly, this method proved effective in enhancing model performance. However, it required significantly more computational time, approximately 2 hours, 2 minutes, and 54 seconds. Overall, the evaluation results indicate that hyperparameter optimization positively impacts the performance of the Random Forest classification model, albeit with a considerable increase in computational cost..

Acknowledgement

The author would like to express sincere gratitude to all parties who have supported the completion of this research. Special thanks go to the Department of Informatics, Faculty of Computer Science, Universitas Pembangunan Nasional “Veteran” Jawa Timur, for providing

academic guidance and facilities throughout the research process. The author is also grateful to the academic supervisors for their valuable insights, constructive feedback, and continuous encouragement during the development of this study. Furthermore, appreciation is extended to family and friends for their unwavering moral support and motivation.This research would not have been possible without the inspiration and collaboration of those who have shared their time and knowledge. All contributions are deeply appreciated.

References

[1] NICEF EAST ASIA AND PACIFIC REGIONAL OFFICE (EAPRO), "WATER, SANITATION AND HYGIENE (WASH) Prepared by the: UNICEF EAST ASIA AND PACIFIC REGIONAL OFFICE (EAPRO)," 2020, doi: www.unicef.org/eapro

[2] Utami. Widya dkk, "PENGAMANAN KUALITAS AIR MINUM," 2023.

[3] F. Yusa Rahman, I. Indah Purnomo, N. Hijriana, dan I. Kalimantan Muhammad Arsyad Al Banjari, "PENERAPAN ALGORITMA DATA MINING UNTUK KLASIFIKASI KUALITAS AIR," 2022.

[4] Kementerian Kesehatan Republik Indonesia, "4 Permenkes-No-492-Tahun-2010-tentang-Persyaratan-Kualitas-Air-Minum," 2010.

[5] M. Lowe, R. Qin, dan X. Mao, "A Review on Machine Learning, Artificial Intelligence, and Smart Technology in Water Treatment and Monitoring," *Water (Switzerland)*, vol. 14, no. 9, Mei 2022, doi: 10.3390/w14091384.

[6] M. M. Mutoffar *dkk.*, "KLASIFIKASI KUALITAS AIR SUMUR MENGGUNAKAN ALGORITMA RANDOM FOREST," vol. 04, 2022.

[7] K. Abdi, A. Warjaya, I. Muthmainnah, dan P. H. Pahutar, "Penerapan Algoritma Random Forest dalam Prediksi Kelayakan Air Minum," *Jurnal Ilmu Komputer dan Informatika*, vol. 3, no. 2, hlm. 81–88, Jan 2024, doi: 10.54082/jiki.81.

[8] L. Savitri dan R. Nursalim, "Klasifikasi Kualitas Air Minum menggunakan Penerapan Algoritma Machine Learning dengan Pendekatan Supervised Learning," Jun 2023. [Daring]. Tersedia pada: https://ejournal.unib.ac.id/diophantine,

[9] F. Firdiani, S. Mandala, Adiwijaya, dan A. H. Abdullah, "WaQuPs: A ROS-Integrated Ensemble Learning Model for Precise Water Quality Prediction," *Applied Sciences (Switzerland)*, vol. 14, no. 1, Jan 2024, doi: 10.3390/app14010262.

[10] U. Sunarya dan T. Haryanti, "Perbandingan Kinerja Algoritma Optimasi pada Metode Random Forest untuk Deteksi Kegagalan Jantung," *Jurnal Rekayasa Elektrika*, vol. 18, no. 4, Des 2022, doi: 10.17529/jre.v18i4.26981.

[11] P. R. Togatorop, M. Sianturi, D. Simamora, dan D. Silaen, "Optimizing Random Forest using Genetic Algorithm for Heart Disease Classification," *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, vol. 13, no. 1, hlm. 60, Agu 2022, doi: 10.24843/lkjiti.2022.v13.i01.p06.