Bulletin of Informatics and Data Science
Vol. 4 No. 2, November 2025, Page 71−78
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v4i2.136
https://ejurnal.pdsi.or.id/index.php/bids/index

Classification Model Optimization using Grid Search and Random Search in Machine Learning Algorithms

Syawaluddin Kadafi Parinduri[1,*], Putrama Alkhairi[2], Irawan[2], Hendry Qurniawan[1]

[1] Information Systems, STIKOM Tunas Bangsa, Pematangsiantar, Indonesia
[2] Informatics Management, STIKOM Tunas Bangsa, Pematangsiantar, Indonesia
Email: [1,*]syawaluddin@amiktunasbangsa.ac.id, [2]putrama@amiktunasbangsa.ac.id,[3]irawaniwan56@gmail.com, [4]hendry@amiktunasbangsa.ac.id

ARTICLE INFORMATION

ARTICLE HISTORY:

Submited	: October 17, 2025
Revised	: November 22, 2025
Accept	: November 22, 2025
Publish	: November 29, 2025

KEYWORD

Random Forest;
Grid Search;
Random Search;
Hyperparameter Optimization;
Kidney Disease

CORRESPONDENCE AUTHOR

Email: syawaluddin@amiktunasbangsa.ac.id

ABSTRACT

The performance of a machine learning model is highly dependent on the selection and tuning of appropriate hyperparameters. The main problem in this study is how to improve the accuracy and stability of a classification model without sacrificing computational time efficiency, especially in the case of kidney disease classification that requires accurate and fast prediction results. This study aims to optimize the classification model by applying two hyperparameter search methods, namely Grid Search and Random Search, to the Random Forest algorithm. The kidney disease dataset is used as a case study with preprocessing processes including data cleaning, missing value imputation, categorical variable encoding, and normalization. Each model is tested using accuracy, precision, recall, and F1-Score metrics. The results show that the Grid Search_RF model produces the highest performance with perfect accuracy, precision, recall, and F1-Score values (1.0000), while Random Search_RF provides results close to (accuracy 0.9875 and F1-Score 0.9900) with more efficient training time. Meanwhile, the standard Random Forest without tuning still shows competitive performance (accuracy 0.9917 and F1-Score 0.9930). Based on these results, it can be concluded that hyperparameter optimization, using both Grid Search and Random Search, can significantly improve the performance of the classification model, with Random Search being the most efficient method for practical implementation in machine learning-based disease detection systems.

1. INTRODUCTION

The rapid development of information and computing technology over the past two decades has driven the increasing use of machine learning in various fields of life, such as healthcare, finance, industry, education, and government [1], [2]. One of the main capabilities of machine learning is classification, the process of grouping data into specific classes based on patterns learned from historical data. The success of a classification system is heavily influenced by the selection of the appropriate algorithm and parameter tuning (hyperparameter tuning) [3], [4], [5], [6].

In the context of supervised learning, models such as Random Forest (RF) [7], [8], Support Vector Machine (SVM) [9], [10], [11], and Logistic Regression (LR)[12], [13] require a series of hyperparameters that directly affect model performance. For example, in Random Forest, the number of trees (n_estimators), the maximum tree depth (max_depth), and the minimum number of samples in a split (min_samples_split) significantly determine the model's generalization ability. Selecting suboptimal values can cause the model to underfit (inability to capture complex patterns) or overfit (overfitting the training data). Therefore, hyperparameter optimization is a crucial step in improving predictive accuracy and model reliability [14], [15].

Various machine learning methods can be used to classify kidney disease data, including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) [16], [17], [18]. Logistic Regression has advantages in interpretability and computational speed, making it suitable for small to medium-sized datasets. However, its weakness lies in its inability to capture complex non-linear relationships between features. Decision Tree is easy to understand and capable of handling mixed data (numeric and categorical), but it is susceptible to overfitting if not pruned. Random Forest, as an extension of Decision Tree based on ensemble learning, has the advantage of reducing overfitting and increasing accuracy because it combines many decision trees, although the computation time and memory requirements are relatively higher [19], [20], [21]. Support Vector Machine (SVM) is effective for high-dimensional datasets and is able to produce optimal decision boundaries, but its performance can degrade on large datasets and is difficult to interpret directly. Meanwhile, K-Nearest Neighbors (KNN) is simple and does not require explicit model training, but its performance is highly dependent on the choice of k value and is sensitive to feature scale and high-dimensional data. Considering these advantages and disadvantages, selecting the most appropriate classification method needs to be tailored to the data characteristics, interpretation needs, and available computational resources [9], [22].

The previous study conducted by [23] in this article had the advantage of comprehensively comparing various machine learning models and two hyperparameter tuning techniques (Grid Search and Random Search) for heart disease prediction. Evaluation using various metrics (such as accuracy, precision, recall, F1-score, AUC) showed Random Forest to be the best model (94.95% accuracy), with Random Search being more computationally efficient. However, its weakness is that it does not address model interpretability, which is crucial in a medical context, and it does not explore techniques for handling imbalanced data for optimal performance. The generalizability of the findings may also be limited due to the use of only a single dataset. The advantage of another study conducted by [24] in this article is the systematic application of Grid Search to optimize seven classification algorithms on a diabetes dataset, with a process that included preprocessing, EDA, and validation using Mean Cross Validation, where XGBoost achieved the highest score (0.772). However, its main drawback is the high computational burden of Grid Search, where the tuning time increases exponentially with the addition of hyperparameters, as well as the lack of comparison with other tuning methods such as Random Search which may be more efficient.

Various approaches have been developed for hyperparameter optimization, including Grid Search, Random Search, and more advanced methods such as Bayesian Optimization or Genetic Algorithms. Among these methods, Grid Search is the most conventional approach, evaluating all parameter combinations within a predetermined search space. This method guarantees that the best configuration within the defined parameter space will be found, but it has the major drawback of being very computationally expensive, especially when the number of parameters and values tested is large. Alternatively, Random Search evaluates parameter combinations randomly with a specified number of iterations. Research by [14], [25] shows that Random Search often achieves comparable or even better results than Grid Search with a significantly lower number of evaluations, especially for models with many parameters, not all of which significantly impact performance. Thus, Random Search is an attractive method because it can save computational time without significantly sacrificing performance.

This efficiency issue becomes particularly relevant when applied to the prediction of chronic kidney disease (CKD), a disease with a high mortality rate worldwide. According to World Health Organization (WHO) data, CKD is among the top ten causes of global death. Early detection of kidney disease is crucial because most sufferers only discover their condition when it has reached an advanced stage. The use of machine learning algorithms allows for rapid and accurate identification of high-risk patients based on clinical data such as creatinine levels, blood pressure, hemoglobin, blood glucose, and other attributes [23], [26].

However, for this detection system to work optimally, a classification model with the best parameter configuration is required. In practice, researchers and practitioners often face limited time and computing resources to conduct a comprehensive parameter search. Therefore, it is important to conduct empirical studies on the effectiveness and efficiency of Grid Search and Random Search methods in optimizing machine learning algorithms for the task of kidney disease classification. Based on this description, this study focuses on optimizing classification models using Grid Search and Random Search in machine learning algorithms. The main objective of the study is to compare the performance and efficiency of the two methods in finding the best hyperparameter configuration in the Random Forest model (and other comparison models), by measuring performance using accuracy, precision, recall, and F1-Score metrics.

This research aims to provide a comprehensive understanding of the advantages and limitations of each optimization method, as well as practical recommendations for researchers and practitioners in selecting the appropriate tuning strategy based on data complexity and available computational resources. Thus, this research not only contributes to improving the accuracy of classification models but also highlights the importance of efficient machine learning model development in real-world contexts, particularly in healthcare. It is hoped that the results of this research can serve as a basis for the development of faster, more accurate, and more reliable artificial intelligence-based decision support systems for detecting kidney disease and other medical classification cases.

2. RESEARCH METHODOLOGY

This chapter comprehensively explains the methods and stages used in conducting the research entitled "Classification Model Optimization Using Grid Search and Random Search in Machine Learning Algorithms." This chapter outlines the research design, data sources and characteristics, pre-processing steps, the classification algorithms used, and the hyperparameter optimization procedure using two main approaches: Grid Search and Random Search. Furthermore, this chapter describes the model performance evaluation techniques, the parameters used in the experiments, and the computing environment in which the research was conducted. With this explanation of the methodology, readers are expected to systematically understand how the research process was conducted, from data processing to analysis of the results, thereby ensuring the validity and replicability of the study. The following subchapter discusses the research design.

2.1 Research Data

Research data is a key component in the process of developing a machine learning model, as the quality and characteristics of the data significantly determine the model's accuracy and generalizability. This study used the Kidney Disease Dataset, which contains clinical data from patients with chronic kidney disease (CKD) and those without chronic kidney disease (not-CKD). This dataset is often used as a benchmark in various medical classification studies because it has a complex combination of numerical and categorical features and describes the patient's health condition quite completely.

Tabel 1. Data Penelitian

id	age	bp	sg	dm	…etc	cad	appet	pe	ane	classification
0	0	48.0	80.0	yes	…	yes	no	good	no	no
1	1	7.0	50.0	no	…	no	no	good	no	no
2	2	62.0	80.0	no	…	yes	no	poor	no	yes
3	3	48.0	70.0	yes	…	no	no	poor	yes	yes
4	4	51.0	80.0	no	…	no	no	good	no	no
5	5	60.0	90.0	yes	…	yes	no	good	yes	no
6	6	68.0	70.0	no	…	no	no	good	no	no
7	7	24.0	NaN	no	…	yes	no	good	yes	no
8	8	52.0	100.0	yes	…	yes	no	good	no	yes
9	9	53.0	90.0	yes	…	yes	no	poor	no	yes
...etc	…	…	…	…	…	…	…	…	…	…

In general, this dataset consists of 400 rows of data (records) and 24 attributes (features), including a target attribute indicating the patient's kidney disease status. These attributes include: patient age, blood pressure, random blood glucose, serum creatinine, sodium, hemoglobin, packed cell volume, and several other clinical indicators such as the presence of protein, glucose, or bacteria in the urine. The target attribute (classification) is coded as "ckd" for patients with chronic kidney disease and "notckd" for healthy patients. Therefore, the problem being solved is a binary classification problem, where a model is trained to predict whether a patient has CKD or not based on their clinical attributes.

2.2 Research Design

This study uses a quantitative experimental approach by applying several machine learning algorithms to classify kidney disease data, then optimizing model performance using two hyperparameter search methods: Grid Search and Random Search. The main objective of this study was to compare the effectiveness and efficiency of the two optimization methods in producing the best-performing classification model. This study was conducted systematically, starting with data collection, data pre-processing, modeling, optimization, and model performance evaluation.

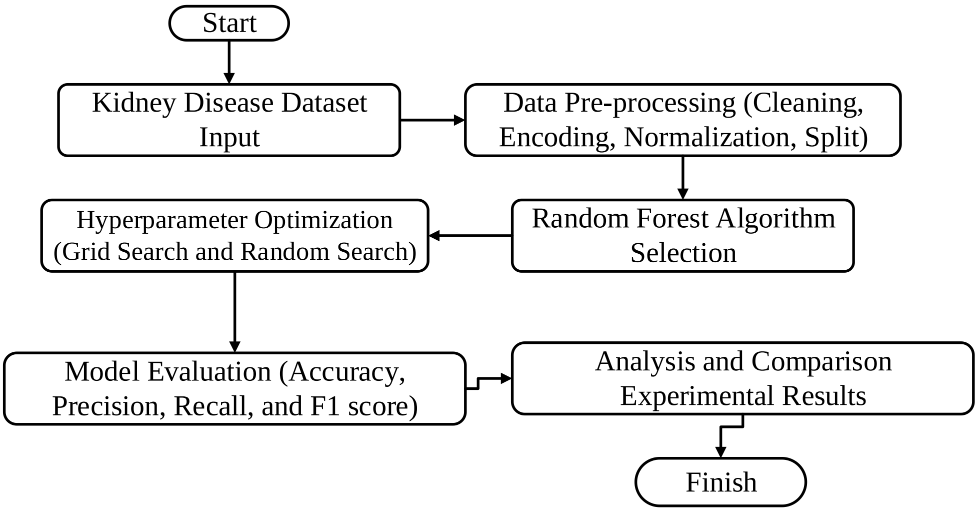

Start
Kidney Disease Dataset Input
Data Pre-processing (Cleaning, Encoding, Normalization, Split)
Random Forest Algorithm Selection
Hyperparameter Optimization (Grid Search and Random Search)
Model Evaluation (Accuracy, Precision, Recall, and F1 score)
Analysis and Comparison Experimental Results
Finish

Figure 1. Rancangan Penelitian

This research design systematically describes the process flow of developing a kidney disease classification model using machine learning algorithms, starting from the data input stage to the analysis of experimental results. The study began with input of a kidney disease dataset (Kidney Disease Dataset) containing patient clinical data such as blood pressure, creatinine levels, hemoglobin, and blood glucose, which was then used to predict kidney disease status (chronic kidney disease or not-CKD). Next, data pre-processing was carried out including data cleaning, imputation of missing values, transformation of categorical data into numeric using One-Hot Encoding, feature normalization using StandardScaler, and dividing the dataset into training data and test data using the Stratified Train-Test Split method to maintain a balanced class distribution. Once the data was ready, the Random Forest algorithm was chosen as the main model due to its ability to handle mixed data and reduce the risk of overfitting through an ensemble learning approach. The next process was hyperparameter optimization using two methods, namely Grid Search and Random Search, to find the best parameter combination with cross-validation (5-fold cross-validation) using the F1-score metric as the main reference. The optimized model was then evaluated based on accuracy, precision, recall, and F1-score metrics using test data to assess the model's performance and generalization ability. The evaluation results of the two optimization methods were compared to analyze differences in performance and execution time efficiency, and statistically tested using a paired t-test to assess the significance of the differences. The final stage of the research was analysis and conclusion drawing, aimed at determining the most effective optimization method for improving the performance of the kidney disease classification model and providing recommendations for implementing similar optimization techniques in other medical classification research.

2.3 Comparison of Random Forest Models Without and With Optimization

The next stage of this research was to conduct a comparative analysis between the Random Forest (RF) model without optimization and models optimized using two different approaches: Grid Search_RF and Random Search_RF. The purpose of this comparison was to determine the extent to which the hyperparameter optimization method could improve the performance of the kidney disease classification model, both in terms of prediction accuracy and computational efficiency.

All three models were trained and tested using the same dataset to ensure that performance differences were solely due to the parameter tuning process, not data variation. The main hyperparameters optimized include the number of trees (n_estimators), the maximum tree depth (max_depth), and the minimum number of samples in each split (min_samples_split). The optimization process was performed using 5-fold cross-validation (5-fold CV) to obtain stable results and avoid bias in the training data.

Table 2. Comparison of Standard Random Forest, Grid Search_RF, and Random Search_RF Models

Comparative Aspects	Random Forest (Standard)	Grid Search_RF	Random Search_RF
Parameter Determination Methods	Uses default values from the scikit-learn library.	Exhaustive search of all parameter combinations in the search space.	Randomly searches for a number of parameter combinations in the search space (random sampling).
Main Objectives	Builds a baseline model without tuning.	Finds the best parameter combination with maximum accuracy.	Finds the best parameter combination with high time efficiency.
Computational Complexity	Low overhead, fast training time.	High, because all combinations are tested one by one.	Moderate, depending on the number of search iterations.
Optimal Parameter Finding Ability	Limited by default parameters.	Very high, guarantees the best configuration in the search space.	High, but does not always guarantee optimal results.
Computational Time Efficiency	Very efficient, fast training.	Less efficient, takes a long time for large datasets.	More efficient than Grid Search, suitable for rapid experimentation.
Model Result Stability	Fairly stable, but performance can be lower.	Very stable, produces maximum accuracy.	Stable and efficient, results close to Grid Search.
Main Advantages	Simple, fast, suitable for baselines.	Provides the highest accuracy, systematically optimized.	Fast and efficient, results close to Grid Search.
Main Weaknesses	Suboptimal, sensitive to default configurations.	Requires a lot of time and resources.	Does not always find the globally best configuration.
Suitable for Use When	Initial experiments or small datasets.	The research objective emphasizes maximum accuracy without time constraints.	A balance between time and results is required.

Table 2 shows that the standard Random Forest model serves as a baseline model to assess initial performance without any parameter tuning. This model is very efficient in terms of training time, but may not produce the best prediction results because the parameters used are not tailored to the characteristics of the dataset. Grid Search_RF provides the most optimal results in terms of performance because it evaluates all possible parameter combinations. However, this method has a significant drawback in terms of execution time, especially when the parameter space is large or the dataset has many features. Nevertheless, Grid Search is an ideal choice if the research goal is to obtain a model with the highest accuracy without considering time or computational resource constraints. Meanwhile, Random Search_RF offers a more efficient approach because it only evaluates a random portion of the parameter space. While the results may not always be absolutely optimal, based on theory and empirical research, Random Search often achieves results very close to Grid Search at a much lower computational cost. Thus, Random Search is a practical choice in real-world scenarios, especially when speed and computational efficiency are priorities.

3. RESULT AND DISCUSSION

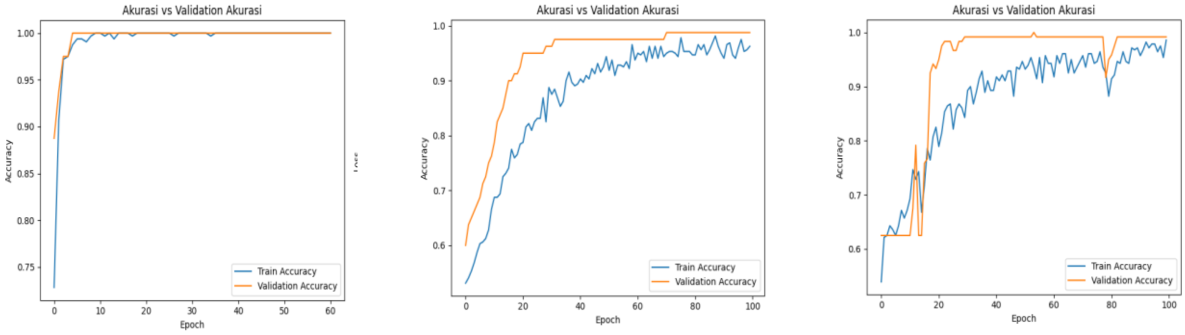
This chapter discusses in-depth the experimental results and analysis obtained from the application of the Grid Search and Random Search hyperparameter optimization methods to the Random Forest algorithm in the kidney disease classification task. The analysis was conducted to compare the performance of the unoptimized model (standard Random Forest) with the two optimized models, and to assess the effectiveness and efficiency of each method. The discussion includes model testing results, performance metric analysis, and interpretation of the findings based on theory and previous research.

3.1 Model Testing Results

The testing phase was carried out after preprocessing and dividing the dataset into training and test data.
The three main models tested were:

1. Standard Random Forest (RF) – a baseline model without parameter tuning.
2. Grid Search_RF – a Random Forest model with parameter optimization using GridSearchCV.
3. Random Search_RF – a Random Forest model with parameter optimization using RandomizedSearchCV.

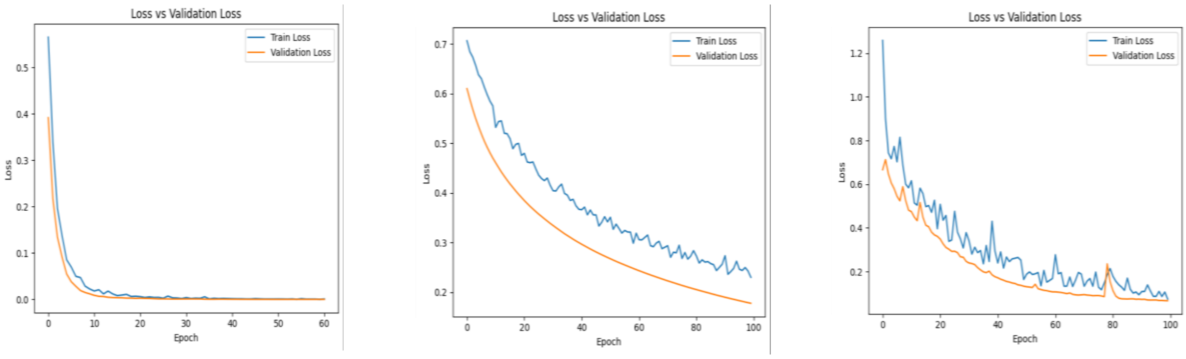
Each model was trained on the training data and tested on the test data in a 75:25 ratio. Evaluation was performed using Accuracy, Precision, Recall, and F1-Score metrics..



a) Grid Search_RF b) Random Search_RF c) Random Forest (basic)

Figure 2. Accuracy Comparison Results of the Three Models Tested

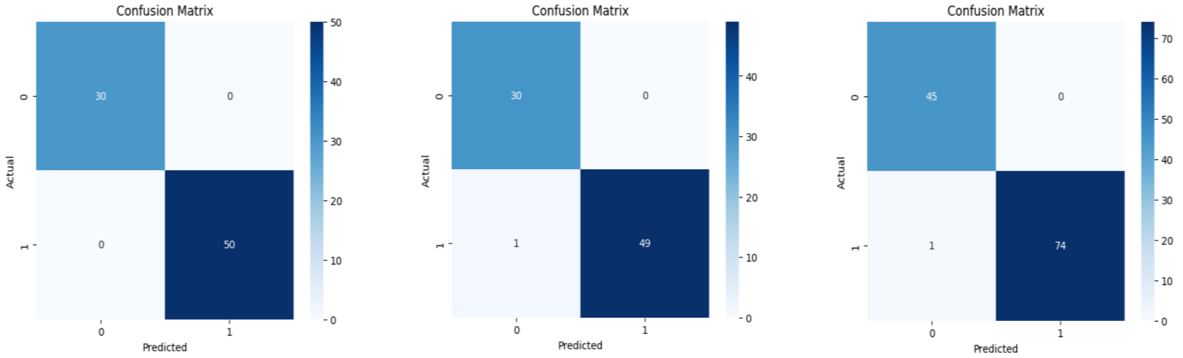
Figure 2 shows a graphical comparison of the training and validation accuracy between three Random Forest models: Grid Search_RF (a), Random Search_RF (b), and the standard or basic Random Forest (c). Graph (a) shows that the Grid Search_RF model achieved very high and stable training and validation accuracy from the beginning of the training process, with the lines of both metrics nearly overlapping around 1.0, indicating that the model had reached the optimal hyperparameter configuration and converged very quickly. Graph (b) shows the Random Search_RF model, which also showed progressive accuracy improvement. Although the validation value fluctuated slightly initially, the model later stabilized near 1.0, indicating that the tuning results through random sampling were quite effective in accelerating model convergence. Meanwhile, graph (c) shows a more gradual increase in accuracy and tends to require more epochs to reach stability, with a small gap between training and validation accuracy. This shows that although the basic Random Forest is already capable of providing good results, the hyperparameter optimization process with Grid Search and Random Search can significantly increase the convergence speed and stability of the model's performance.



a) Grid Search_RF b) Random Search_RF c) Random Forest (basic)

Figure 3. Loss Comparison Results of the Three Models Tested

Figure 3 shows a comparison graph of Loss vs. Validation Loss for three Random Forest models: Grid Search_RF (a), Random Search_RF (b), and the standard or basic Random Forest (c). Graph (a) shows that the Grid Search_RF model exhibits a very rapid and stable loss reduction in the first few epochs, where the training loss and validation loss values decrease sharply to near zero and overlap. This indicates that the Grid Search optimized model is able to find the optimal parameters that efficiently and stably minimize prediction errors and avoid overfitting. Graph (b) also shows a consistent loss reduction trend, although the rate of decrease is slower than Grid Search, and there is still a slight difference between the training loss and validation loss, indicating that the model is still converging toward optimal conditions. Meanwhile, graph (c) shows a relatively high initial loss value and greater fluctuations before gradually decreasing to a stable point around epoch 80. This pattern indicates that the baseline model takes longer to reach stability and is susceptible to variations in the training data. Overall, the three graphs show that hyperparameter optimization, using both Grid Search and Random Search, significantly accelerates model convergence, reduces error (loss), and improves model stability and generalization compared to Random Forest without tuning



a) Grid Search_RF
b) Random Search_RF
c) Random Forest (basic)

Figure 4. Confusion Matrix

A model comparison analysis was conducted to understand the differences between the three Random Forest model variants: standard RF, Grid Search_RF, and Random Search_RF, based on performance, computational efficiency, and optimization results. Based on the test results, Grid Search_RF demonstrated the best performance, with an F1-score reaching 1.0000, as the exhaustive parameter search method was able to find the most optimal hyperparameter configuration. Random Search_RF performed slightly lower (F1-score = 0.9900), but was more time-efficient because it only randomly evaluated a subset of parameter combinations. Meanwhile, the standard Random Forest maintained high performance (F1-score = 0.9930) even without tuning, making it a good baseline model for comparison. In general, the improvement in results due to optimization was not numerically significant, but it did improve the stability and consistency of model performance. In terms of efficiency, Grid Search_RF required the longest training time because it had to test all parameter combinations, while Random Search_RF achieved near-optimal results with much shorter computational time. The standard Random Forest is the fastest and simplest model, but it is not optimal for all data types. Therefore, Grid Search_RF is suitable for research focused on achieving maximum accuracy, while Random Search_RF is more suitable for practical implementations that require efficiency without significantly sacrificing performance.

3.2 Discussion of Research Results

Experimental results show that hyperparameter optimization has a positive effect on classification model performance. Although Random Forest without tuning produced high accuracy (0.9917), applying Grid Search successfully improved performance to perfection, while Random Search produced nearly identical results with shorter computation time. This aligns with previous research indicating that Random Search is able to find near-optimal parameters in a more time-efficient manner than Grid Search, especially when the parameter space is large and not all parameters significantly influence the final result..

Table 3. Random Forest Model Evaluation Results

Model	Accuracy	Precision	Recal	F1-Score
Grid Search_RF	1,0000	1,0000	1,0000	1,0000
Random Search _ RF	0,9875	1,0000	0,9800	0,9900
RF	0,9917	1,0000	0,9867	0,9930

The results show that all three Random Forest models provide excellent classification performance, with relatively small differences in metric values. The Grid Search_RF model achieved excellent results with accuracy, precision, recall, and F1-score values of 1.0000 each, indicating that the parameter optimization process through Grid Search successfully found the most optimal hyperparameter configuration and produced highly accurate predictions without classification errors. The Random Search_RF model showed slightly lower performance with an accuracy of 0.9875, a recall of 0.9800, and an F1-score of 0.9900, but still maintained a perfect precision value (1.0000). This indicates that the Random Search method is also capable of finding excellent parameter combinations with higher time efficiency than Grid Search.

Meanwhile, the standard Random Forest (without optimization) produced an accuracy of 0.9917, a recall of 0.9867, and an F1-score of 0.9930, indicating that even without tuning, this model still has high performance due to its strong characteristics in handling complex data. Overall, these results demonstrate that hyperparameter optimization, using both Grid Search and Random Search, can improve model stability and effectiveness, with Grid Search_RF providing the highest performance and Random Search_RF offering the best balance between accuracy and computational efficiency.

Overall, the small differences in F1-Score values between the three models indicate that the kidney disease dataset exhibits patterns that are relatively easy for the Random Forest algorithm to learn. This model is able to effectively capture relationships between clinical features even without in-depth optimization, due to the ensemble method's ability to reduce model variance.

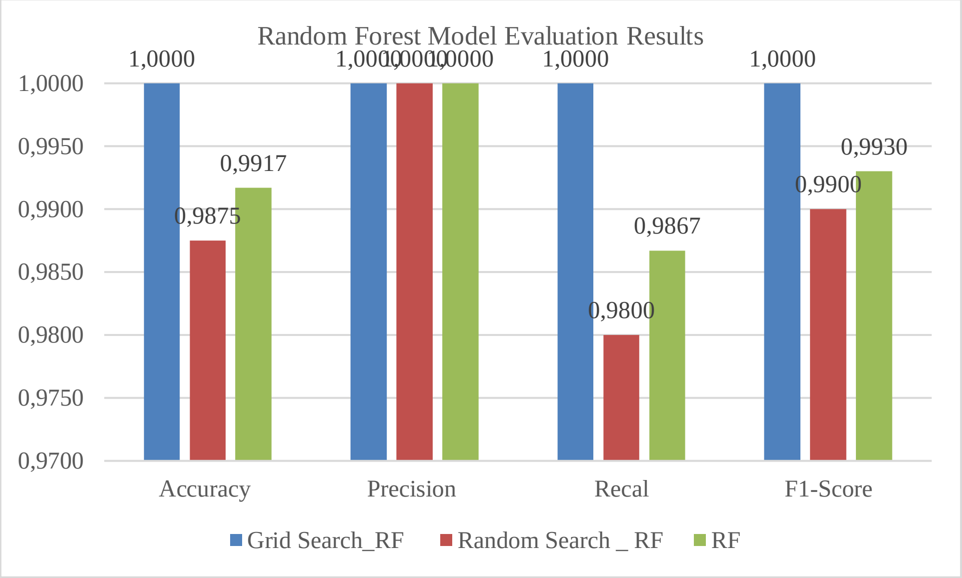


Figure 5. Random Forest Model Evaluation Results

However, it should be noted that excellent results on Grid Search_RF may indicate potential overfitting. Therefore, cross-validation and testing on different datasets are recommended to ensure model generalization to new data. In terms of computational efficiency, Random Search is superior because it only tries a subset of parameter combinations while still approximating Grid Search results. This makes it an ideal choice in real-world scenarios such as medical data-driven diagnosis, where training time is often a practical constraint. The findings of this study offer several important implications. Academically, the results reinforce the theory that hyperparameter optimization methods can significantly improve classification model performance, although the improvement is sometimes marginal for datasets with clear patterns. Practically, Random Search can be an efficient alternative for model tuning in the development of medical decision support systems, as it offers a balance between accuracy and computational time. Methodologically, this study demonstrates the importance of comparing different optimization approaches to find the best model for your needs—whether prioritizing highest performance or execution efficiency.

4. CONCLUSION

This study discusses the optimization of a kidney disease classification model using the Random Forest algorithm with two hyperparameter tuning methods, namely Grid Search and Random Search. Based on the experimental results, the three tested models showed high performance with F1-Score values above 0.99. The Grid Search_RF model produced the best performance with perfect accuracy, precision, recall, and F1-Score (1.0000), indicating that a thorough parameter search was able to find the most optimal configuration. The Random Search_RF model obtained an accuracy of 0.9875 and an F1-Score of 0.9900, results close to Grid Search but with more efficient computational time. Meanwhile, the standard Random Forest also showed competitive results (accuracy of 0.9917 and F1-Score of 0.9930), proving that this model is robust even without tuning. Overall, Grid Search excels in maximum accuracy, while Random Search is more efficient and practical to apply to conditions with limited time and computational resources. These results reinforce that hyperparameter optimization plays a crucial role in improving the stability, accuracy, and generalization of machine learning models.

REFERENCES

[1] A. Morchid, "High-technology agriculture system to enhance food security: A concept of smart irrigation system using Internet of Things and cloud computing," *J. Saudi Soc. Agric. Sci.*, 2024, doi: 10.1016/j.jssas.2024.02.001.

[2] J. Yin, "Conversation Technology With Micro-Learning: The Impact of Chatbot-Based Learning on Students' Learning Motivation and Performance," *J. Educ. Comput. Res.*, vol. 59, no. 1, pp. 154–177, 2021, doi: 10.1177/0735633120952067.

[3] A. Ahmadi, "Automatic tuning of PID controllers using deep recurrent neural networks with pruning based on tracking error," *J. Instrum.*, vol. 19, no. 2, 2024, doi: 10.1088/1748-0221/19/02/P02028.

[4] Q. Hou, R. Xia, J. Zhang, Y. Feng, Z. Zhan, and X. Wang, "Learning visual overlapping image pairs for SfM via CNN fine-tuning with photogrammetric geometry information," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 116, no. October 2022, p. 103162, 2023, doi: 10.1016/j.jag.2022.103162.

[5] M. K. Suryadi, "A Comparative Study of Various Hyperparameter Tuning on Random Forest Classification with SMOTE and Feature Selection Using Genetic Algorithm in Software Defect Prediction," *J. Electron. Electromed. Eng. Med. Informatics*, vol. 6, no. 2, pp. 137–147, 2024, doi: 10.35882/jeeemi.v6i2.375.

[6] N. Becherer, J. Pecarina, S. Nykl, and K. Hopkinson, "Improving optimization of convolutional neural networks through parameter fine-tuning," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 3469–3479, 2019, doi: 10.1007/s00521-017-3285-0.

[7] A. Kurniawan, "Hybrid PSO-ACO Optimization for Rice Leaf Disease Classification Using Random Forest and Support Vector Machines," *Int. J. Adv. Comput. Sci. Appl.*, vol. 16, no. 6, pp. 388–397, 2025, doi: 10.14569/IJACSA.2025.0160638.

[8] T. Sugihartono, "Optimizing Stunting Detection through SMOTE and Machine Learning: a Comparative Study of XGBoost, Random Forest, SVM, and k-NN," *J. Appl. Data Sci.*, vol. 6, no. 1, pp. 667–682, 2025, doi: 10.47738/jads.v6i1.494.

[9] H. Wang *et al.*, "Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non-small cell lung cancer from 18F-FDG PET/CT images," *EJNMMI Res.*, vol. 7, no. 1, 2017, doi: 10.1186/s13550-017-0260-9.

[10] Nabil Ibrahim El-Sawalhi, "Support Vector Machine Cost Estimation Model for Road Projects," *J. Civ. Eng. Archit.*, vol. 9, no. 9, pp. 1115–1125, 2015, doi: 10.17265/1934-7359/2015.09.012.

[11] E. Purwaningsih, S. Informasi, U. Bina, and S. Informatika, "Improving The Performance Of Support Vector Machine With Forward Selection For Prediction Of," vol. 8, no. 1, pp. 18–24, 2022, doi: 10.33480/jitk.v8i1.3327.From.

[12] P. Golpour *et al.*, "Comparison of support vector machine, naïve bayes and logistic regression for assessing the necessity for coronary angiography," *Int. J. Environ. Res. Public Health*, vol. 17, no. 18, pp. 1–9, 2020, doi: 10.3390/ijerph17186449.

[13] A. K. S. Yadav, "Distributed denial of service (DDOS) attacks and mitigation method using logistic regression-based GoogLeNet for real time in security games," *Int. J. Model. Simulation, Sci. Comput.*, 2024, doi: 10.1142/S1793962324410204.

[14] K. S. Chong, "Comparison of Naive Bayes and SVM Classification in Grid-Search Hyperparameter Tuned and Non-Hyperparameter Tuned Healthcare Stock Market Sentiment Analysis," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 12, pp. 90–94, 2022, doi: 10.14569/IJACSA.2022.0131213.

[15] M. Bellaj, "Educational Data Mining: Employing Machine Learning Techniques and Hyperparameter Optimization to Improve Students' Academic Performance," *Int. J. online Biomed. Eng.*, vol. 20, no. 3, pp. 55–74, 2024, doi: 10.3991/ijoe.v20i03.46287.

[16] N. Somching, "Using machine learning algorithm and landsat time series to identify establishment year of para rubber plantations: a case study in Thalang district, Phuket Island, Thailand," *Int. J. Remote Sens.*, vol. 41, no. 23, pp. 9075–9100, 2020, doi: 10.1080/01431161.2020.1799450.

[17] S. Aboukadri, A. Ouaddah, and A. Mezrioui, "Major Role of Artificial Intelligence, Machine Learning, and Deep Learning in Identity and Access Management Field: Challenges and State of the Art BT - Proceedings of the 8th International Conference on Advanced Intelligent Systems and Informatics 2022," A. E. Hassanien, V. Snášel, M. Tang, T.-W. Sung, and K.-C. Chang, Eds., Cham: Springer International Publishing, 2023, pp. 50–64.

[18] M. M. Moein, "Predictive models for concrete properties using machine learning and deep learning approaches: A review," *J. Build. Eng.*, vol. 63, 2023, doi: 10.1016/j.jobe.2022.105444.

[19] M. Said, Y. Omar, S. Safwat, and A. Salem, "Explainable Artificial Intelligence Powered Model for Explainable Detection of Stroke Disease BT - Proceedings of the 8th International Conference on Advanced Intelligent Systems and Informatics 2022," A. E. Hassanien, V. Snášel, M. Tang, T.-W. Sung, and K.-C. Chang, Eds., Cham: Springer International Publishing, 2023, pp. 211–223.

[20] A. A. Aburomman and M. Bin Ibne Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Appl. Soft Comput. J.*, vol. 38, pp. 360–372, 2016, doi: 10.1016/j.asoc.2015.10.011.

[21] L. Chaves and G. Marques, "Data mining techniques for early diagnosis of diabetes: A comparative study," *Appl. Sci.*, vol. 11, no. 5, pp. 1–12, 2021, doi: 10.3390/app11052218.

[22] E. Ismail, W. Gad, and M. Hashem, "A hybrid Stacking-SMOTE model for optimizing the prediction of autistic genes," *BMC Bioinformatics*, vol. 24, no. 1, pp. 1–18, 2023, doi: 10.1186/s12859-023-05501-y.

[23] S. Arshad, S. M. J. Zaidi, M. Ali, M. U. Hashmi, A. Manan, and ..., "A Comparative Study of Machine Learning Models for Heart Disease Prediction Using Grid Search and Random Search for Hyperparameter Tuning," *J. Comput. ...*, vol. 08, no. 01, 2024, [Online]. Available: https://jcbi.org/index.php/Main/article/view/697

[24] W. Nugraha and A. Sasongko, "Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search Hyperparameter Tuning on Classification Algorithm with Grid Search," *Sist. J. Sist. Inf.*, vol. 11, no. 2, pp. 2540–9719, 2022, [Online]. Available: https://doi.org/10.32520/stmsi.v11i2.1750

[25] D. M. Belete, "Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results," *Int. J. Comput. Appl.*, vol. 44, no. 9, pp. 875–886, 2022, doi: 10.1080/1206212X.2021.1974663.

[26] Y. Kurniawati and M. Muhajir, "Optimization of Backpropagation Using Harmony Search for Gold Price Forecasting," *Pakistan J. Stat. Oper. Res.*, vol. 18, no. 3, pp. 589–599, 2022, doi: 10.18187/pjsor.v18i3.3915.