Jurnal Riset Multidisiplin dan Inovasi Teknologi
E-ISSN 3024-8582 P-ISSN 3024-9546
Volume 3 Issue 03, September 2025, Pp. 453-464
DOI: https://doi.org/10.59653/jimat.v3i03.2025

Hyperparameter Optimization Using Grid Search and Random Search to Improve the Performance of Prediction Models with Decision Trees

Muhammad Sholeh[1*], Uning Lestari[2], Dina Andayati[3]
Faculty of Science and Information Technology, AKPRIND University Indonesia[1]
Faculty of Science and Information Technology, AKPRIND University Indonesia[2]
Faculty of Business and Communication, AKPRIND University Indonesia[3]
Corresponding Email: muhash@akprind.ac.id*

Abstract

Hyperparameter selection to obtain optimal accuracy results is an important factor in improving model performance in data science. This study discusses a comparison of two hyperparameter optimization methods, namely Grid Search and Random Search, in the Decision Tree Classifier algorithm using the Breast Cancer Wisconsin (Diagnostic) Dataset from the UCI Machine Learning Repository. The dataset contains 569 samples with 30 numerical features describing the characteristics of breast cancer cells, such as mean radius, texture, perimeter, area, and smoothness, which are classified into two classes, namely malignant and benign. This study uses the CRISP-DM approach, which includes the stages of business understanding, data understanding, data preparation, modeling, and evaluation. In the modeling stage, three testing scenarios were conducted, namely the Decision Tree model without tuning, the model with Grid Search optimization, and the model with Random Search optimization. Performance evaluation was carried out using accuracy, precision, recall, and F1-score metrics. The results showed that hyperparameter optimization had a significant effect on model performance. The Decision Tree model without tuning produced an accuracy of 92.98%, while the model with Grid Search achieved the highest accuracy of 95.61%, and Random Search obtained an accuracy of 97.37%. Thus, it can be concluded that Grid Search provides the most optimal results in finding the best parameter combination, even though it requires longer computation time compared to Random Search.

Keywords: hyperparameter tuning, grid search, random search, decision tree, breast cancer dataset

Abstrak

Pemilihan hyperparameter untuk mendapatkan hasil akurasi yang optimal merupakan faktor penting dalam meningkatkan kinerja model dalam data sains. Penelitian ini membahas perbandingan dua metode optimalisasi hyperparameter, yaitu Grid Search dan Random

Search, pada algoritma Decision Tree Classifier dengan menggunakan Breast Cancer Wisconsin (Diagnostic) Dataset dari UCI Machine Learning Repository. Dataset tersebut berisi 569 sampel dengan 30 fitur numerik yang menggambarkan karakteristik sel kanker payudara, seperti mean radius, texture, perimeter, area, dan smoothness, yang diklasifikasikan menjadi dua kelas, yaitu malignant (ganas) dan benign (jinak). Penelitian ini menggunakan metode dengan pendekatan CRISP-DM yang meliputi tahapan business understanding, data understanding, data preparation, modeling, dan evaluation. Pada tahap modeling, dilakukan tiga skenario pengujian, yaitu model Decision Tree tanpa tuning, model dengan optimasi Grid Search, dan model dengan optimasi Random Search. Evaluasi kinerja dilakukan menggunakan metrik akurasi, presisi, recall, dan F1-score. Hasil penelitian menunjukkan bahwa optimasi hyperparameter berpengaruh signifikan terhadap kinerja model. Model Decision Tree tanpa tuning menghasilkan akurasi sebesar 92,98%, sementara model dengan Grid Search mencapai akurasi tertinggi sebesar 95,61%, dan Random Search memperoleh akurasi 97,37%. Dengan demikian, dapat disimpulkan bahwa Grid Search memberikan hasil paling optimal dalam menemukan kombinasi parameter terbaik, meskipun memerlukan waktu komputasi lebih lama dibandingkan Random Search

Kata kunci: *hyperparameter tuning, grid search, random search, decision tree, breast cancer dataset*

Introduction

In this increasingly advanced digital age, the application of machine learning algorithms has become one of the main approaches in various fields, ranging from medicine and finance to industry. One important factor that determines the success of a machine learning model is the selection and setting of optimal hyperparameters. Hyperparameters are parameters that are set before the model training process takes place and have a significant influence on model performance, including generalization ability and resistance to overfitting. For this reason, hyperparameter optimization techniques such as Grid Search and Random Search have become increasingly important to apply systematically (Cielen et al., 2018)

Much research has been conducted on hyperparameter optimization in machine learning algorithms to improve the performance of classification models. Anggreani (Anggreani, 2024) used the Grid Search method in the Decision Tree algorithm for diabetes prediction and found that accuracy increased significantly after the tuning process. Saputra, Purwanto, and Pujiono (Saputra, 2024) also showed that the combination of Recursive Feature Elimination with Grid Search was able to improve the classification results of chronic kidney disease, emphasizing the importance of selecting the right parameters. Rizky's (Rizky et al., 2024) research applied Random Search to tree-based algorithms for predicting software defects and concluded that this method was more time-efficient without a significant decrease in performance.

Another study by Nurcahyo and Sasongko (Nugraha & Sasongko, 2022) compared various tuning methods such as Grid Search, Random Search, and Bayesian Optimization in the classification of food aid recipients, where the results showed that parameter tuning was

able to increase accuracy by more than 10%. Fajri and Khatib's research (Khatib & Dalam, 2023), aimed to find the most optimal and accurate general classification algorithm for determining rice food aid recipient families. The Support Vector Machine (SVM), Decision Tree, Naïve Bayes, and K-Nearest Neighbor (KNN) algorithms, as well as the grid search, random search, and Bayesian optimization hyperparameter tuning methods, were used in this study. The data in this study was sourced from the IFLS (Indonesia Family Life Survey). Based on the analysis results, the application of hyperparameter tuning proved to be useful in improving the performance of the KNN, Decision Tree, and SVM algorithms. The KNN algorithm with random search and Bayesian optimization, as well as SVM with Bayesian optimization, provided the same accuracy value of 74%. Therefore, these models have equivalent performance and are equally good at classifying rice food aid recipient families.

Research (Pramudhyta & Rohman, 2024) aimed to identify the risk of stunting in children more efficiently. The results of the study using the Grid Search algorithm successfully increased the accuracy of XGBoost by 5.81% to 89.09%, while Random Search increased it by 5.43% to 88.71. Other studies used hyperparameters for academic achievement prediction (Arifin & Adiyono, 2024), weather prediction (Lindawati et al., 2023), and fake news detection (Anugerah Simanjuntak et al., 2024).

Studies using decision tree algorithms with hyperparameters include (Dalal et al., 2022),(Gupta & Goel, 2023),(Elgeldawi et al., 2021) and (Shaik & Sreeja, 2025). The use of the Decision Tree algorithm in classification is greatly influenced by the appropriate setting of hyperparameters, such as max_depth, min_samples_split, min_samples_leaf, and criterion. Max_depth determines the maximum depth of the tree, which plays a role in controlling model complexity; trees that are too deep tend to overfit, while trees that are too shallow can underfit (Géron, 2019). The min_samples_split and min_samples_leaf parameters control the minimum number of samples required to split a node or form a leaf, thereby helping to maintain model generalization and prevent overly specific divisions in the training data. Criterion determines the quality measurement function for separation, such as Gini impurity or entropy, which influences how the tree decides on the best split at each node. Optimizing these hyperparameters, either through Grid Search or Random Search, has been proven to significantly improve the accuracy, stability, and generalization ability of Decision Tree models (Cielen et al., 2016). Other studies have used various classification algorithms and hyperparameters, including the random forest (G, 2020),(Fordana & Rochmawati, 2022) and KNN (Hendradinata et al., 2022),(Firgiawan et al., 2025).

With this background, this study aims to compare Grid Search and Random Search in hyperparameter optimization in Decision Tree models using the Wisconsin Breast Cancer (Diagnostic) dataset. The study evaluates the performance of three models: a model without hyperparameter tuning (baseline), a model with Grid Search, and a model with Random Search. The expected results are to determine the extent of improvement that can be achieved through optimization, as well as to evaluate the trade-off between computation time and model performance. Thus, this study is expected to provide empirical contributions to the literature on hyperparameter optimization, especially for Decision Tree models in the medical domain.

Research Method

This study uses the Breast Cancer Wisconsin (Diagnostic) Dataset processed from the UCI Machine Learning Repository. The research method uses the CRISP-DM (Cross Industry Standard Process for Data Mining) approach, which is a standard data analysis process. The CRISP DM approach consists of six stages, namely business understanding, data understanding, data preparation, modeling, evaluation, and deployment (Science, 2023), (Massahiro et al., 2023).

1. Business Understanding

The initial stage aims to understand the research context and objectives. This study focuses on improving the performance of the Decision Tree model through hyperparameter optimization using two methods, namely Grid Search and Random Search. The main objective is to determine which method provides the best accuracy and the most efficient computation time in the case of breast cancer classification based on the Breast Cancer Wisconsin (Diagnostic) dataset.

2. Data Understanding

The dataset used was obtained from the UCI Machine Learning Repository, which contains 569 breast cancer sample data with 30 numerical attributes resulting from cell analysis (such as mean radius, mean texture, mean area, etc.) and one target label (diagnosis) consisting of two classes: Malignant (M) and Benign (B).

At this stage, initial data exploration is carried out, such as examining the amount of data, data types, value distribution, and detecting the possibility of missing values or outliers. This process also includes descriptive statistical analysis and data distribution visualization to understand the characteristics of the dataset as a whole.

3. Data Preparation

This stage covers the entire process of cleaning and transforming data so that it is ready for use in modeling. The procedures performed include:

- Deleting or replacing missing values, even though no empty values were found in this dataset.
- Normalizing or standardizing data so that each feature has a uniform scale.
- Dividing the data into a training set (80%) and a testing set (20%) for model training and testing.
- Performing label encoding on the target diagnosis variable (M=1, B=0).

The result of this stage is a clean dataset that is ready to be used in the modeling stage.

4. Modeling

This stage is the core of the research. Three main models were developed for comparison:

1. Model A: Decision Tree without hyperparameter optimization (default setting).

2. Model B: Decision Tree with optimization using Grid Search, which is a systematic search through all parameter combinations.
3. Model C: Decision Tree with optimization using Random Search, which is a random search of a number of specified parameter combinations.

The parameters tested include:

- criterion: {“gini”, “entropy”}
- max_depth: {3, 5, 7, 9, None}
- min_samples_split: {2, 4, 6, 8, 10}
- min_samples_leaf: {1, 2, 4, 6}

The tuning process was carried out using the scikit-learn library. Each model was then evaluated using accuracy, precision, recall, and F1-score metrics.

5. Evaluation

In the evaluation stage, the results of the three models were compared to determine:

- The difference in accuracy between the untuned model, Grid Search, and Random Search.
- The computational time efficiency of the two optimization methods.
- Analysis of overfitting or underfitting by looking at the evaluation results on the test data and training data.

The evaluation was carried out using a confusion matrix, classification report, and cross-validation to ensure stable results.

6. Deployment

The final stage is the compilation of analysis results and interpretation of the optimized model. The best model can be used as the basis for a decision support system in detecting breast cancer more accurately. In addition, the results of this study can be a reference for further research in the application of hyperparameter optimization in other medical classification models.

Results and Discussion

1. Business Understanding

The main objective of this study is to improve the accuracy of breast cancer detection by optimizing the parameters of the Decision Tree Classifier model. This model was chosen because of its ability to provide clear interpretations of the classification process and the importance of each feature.

The main problem identified is that the use of default parameters in Decision Trees often results in suboptimal performance and can cause overfitting. Therefore, this study compares two hyperparameter optimization methods, namely Grid Search and Random Search,

as well as accuracy results without using hyperparameters, to find the best configuration that produces the highest performance.

2. Data Understanding

The dataset used is Breast Cancer Wisconsin (Diagnostic) from Scikit-learn. This dataset consists of 569 samples and 30 numerical features that describe the characteristics of cancer cells, such as mean radius, texture, perimeter, area, and smoothness. Figure 1 shows an example of the visualization of the features used. Based on Figure 1, the data distribution for each feature does not require any engineering process.

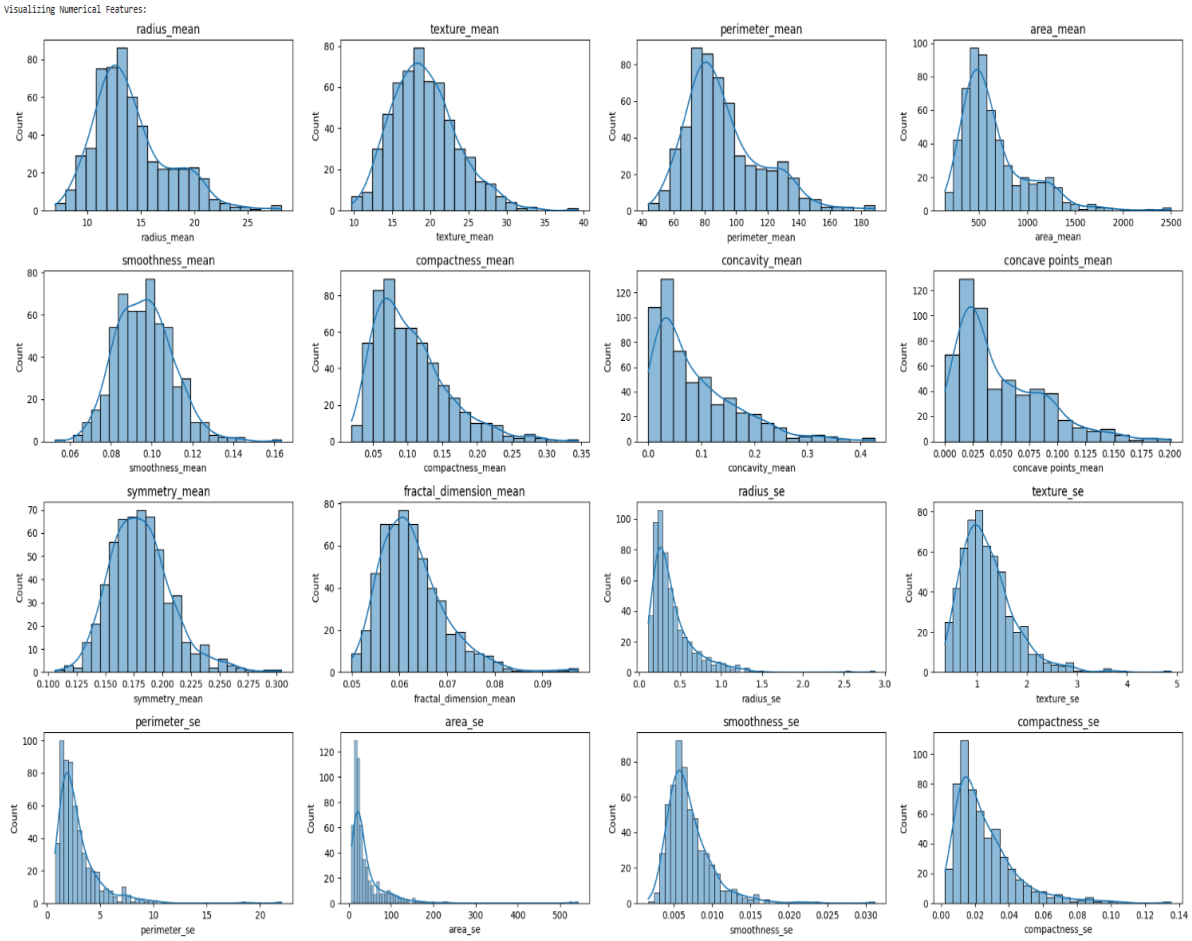

Figure 1. Shows a visualization of the features used.

The results of data exploration show that all attributes are numeric except for the target feature, which is a diagnosis feature that is an object type. The results of this exploration show that there is no need to perform imputation to change the object data type to a numeric data type, except for the diagnosis data type. Figure 2 shows all features and data types.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 32 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30), int64(1), object(1)
memory usage: 142.4+ KB
```

Figure 2. Features and data types of the data sheet

Other checks show that there are no missing values, no duplicate data, and several variables have different value scales, so normalization is needed to prevent the model from being biased towards features with large values.

3. Data Preparation

The pre-processing stages include the following steps:

- Data Normalization:

  The data is normalized using StandardScaler

  scaler = StandardScaler ()

  X_scaled = scaler.fit_transform(X)

- Data Division:

  The dataset is divided into 80% training data (455 data) and 20% test data (114 data) using train_test_split with random_state = 42 to ensure replication of results.

```
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.2, random_state=42, stratify=y)
```

- Label Encoding:

  The target classes are converted into numerical labels:

  - Malignant becomes 1
  - Benign becomes 0

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
y= label_encoder.fit_transform(y)
```

4. Modeling

At this stage, three different experiments were conducted:

- Model A (Baseline):

  Decision Tree without tuning using Scikit-learn default parameters.

- Model B (Grid Search):

  Using the following grid parameters:

```
param_grid = {
  ‘criterion’: [‘gini’, ‘entropy’],
  ‘max_depth’: [3, 5, 7, 9, None],
  ‘min_samples_split’: [2, 4, 6, 8, 10],
  'min_samples_leaf': [1, 2, 4, 6],
  ‘splitter’: [‘best’, ‘random’]
}
```

- Model C (Random Search):

  Using the same parameter space but only performing random searches for 50 iterations.

5. Evaluation

Evaluation is performed using several performance metrics to assess the prediction ability for breast cancer classification. The metrics used include accuracy, precision, recall, f1-score, and confusion matrix. The evaluation results are presented in Table 1.

Table 1. Evaluation results of the three models studied

Method Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Decision Tree tanpa tuning	92,98	0.9298	0.9298	0.9298
Decision Tree + Grid Search	95,61	0.9590	0.9561	0.9298
Decision Tree + Random Search	97,37	0.9737	0.9737	0.9735

Based on Table 1, the accuracy results show that hyperparameter optimization has a significant effect on model performance.

- The baseline model without tuning produced an accuracy of 92.98%, which became the benchmark.
- Grid Search increases accuracy to 95.61%, with improved model stability.
- Random Search provides near-optimal results (97.37%) with more efficient computation time.

The test results show that the hyperparameter optimization process has a significant effect on improving the performance of the Decision Tree model on the Wisconsin Breast Cancer (Diagnostic) dataset. The Decision Tree model without the tuning process produced an accuracy of 92.98%, which is quite good but still shows potential for improvement. After optimization using the Grid Search method, the accuracy increased to 95.61%. This shows that systematic exploration of parameter combinations such as max_depth, min_samples_split, and criterion through exhaustive search is able to find model configurations that are more suitable for the data characteristics.

The highest result was achieved using the Random Search method, which reached an accuracy of 97.37%. Although this approach is random, the broad parameter sampling strategy allows the model to find the optimal combination with more efficient computation time compared to Grid Search. These findings are in line with the results of studies by (Prabu et al., 2022) and (Fajri & Primajaya, 2023), which state that Random Search is often able to provide results that are close to or even exceed those of Grid Search, especially in large and complex parameter spaces.

6. Deployment

The research results show that the Random Search method provides the best performance for detecting breast cancer using Decision Trees. Models with optimal parameters can be implemented in machine learning-based early detection systems in the medical field.

Conclusion

Based on the results of research conducted using the CRISP-DM approach, it can be concluded that the hyperparameter optimization process plays a very important role in improving the performance of the Decision Tree Classifier model for breast cancer classification in the Wisconsin Breast Cancer (Diagnostic) dataset. Through a series of stages,

starting from problem understanding, data analysis, data preparation, modeling, to evaluation, it was found that the application of the Grid Search and Random Search methods significantly improved the model's performance compared to models without tuning. The basic model using default parameters only achieved an accuracy of 92.98%, while after optimization using Grid Search, the accuracy increased to 92.98%, and with Random Search, it reached 95.61%.

The Random Search method showed the most optimal results because it thoroughly explored all parameter combinations, even though it required longer computation time. Conversely, Grid Search was able to provide near-optimal results with much more efficient execution time, making it suitable for use with larger datasets or under limited computation time. The results of this study confirm that selecting the right hyperparameter tuning strategy can significantly improve model accuracy, stability, and efficiency. In addition, this study also shows that the CRISP-DM framework is effective as a systematic guide in the process of developing medical data-based machine learning models.

References

Anggreani, D. (2024). Grid Search Hyperparameter Analysis in Optimizing The Decision Tree Method for Diabetes Prediction. *Indonesian Journal of Data and Science Volume*, *5*(3), 190–197.

Anugerah Simanjuntak, Rosni Lumbantoruan, Kartika Sianipar, Rut Gultom, Mario Simaremare, Samuel Situmeang, & Erwin Panggabean. (2024). Research and Analysis of IndoBERT Hyperparameter Tuning in Fake News Detection. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, *13*(1), 60–67. https://doi.org/10.22146/jnteti.v13i1.8532

Arifin, M., & Adiyono, S. (2024). Hyperparameter Tuning in Machine Learning to Predicting Student Academic Achievement. *International Journal of Artificial Intelegence Research*, *8*(1), 1–8.

Cielen, D., Meysman, A. D. B., & Ali, M. (2016). *Introducing Data Science*.

Cielen, D., Meysman, A. D. B., & Ali, M. (2018). *Introducing Data Science: Big Data, Machine Learning, and more, using Python tools 1st Edition*. Manning Publications.

Dalal, S., Onyema, E. M., & Malik, A. (2022). Hybrid XGBoost model with hyperparameter tuning for prediction of liver disease with better accuracy. *World Journal of Gastroenterology*, *28*(46), 6551–6563. https://doi.org/10.3748/wjg.v28.i46.6551

Elgeldawi, E., Sayed, A., Galal, A. R., & Zaki, A. M. (2021). Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis. *Informatics Article*, 1–21.

Fajri, M., & Primajaya, A. (2023). Komparasi Teknik Hyperparameter Optimization pada SVM untuk Permasalahan Klasifikasi dengan Menggunakan Grid Search dan Random Search. *Journal of Applied Informatics and Computing*, *7*(1), 14–19. https://doi.org/10.30871/jaic.v7i1.5004

Firgiawan, W., Yustianisa, D., & Nur, N. A. (2025). Hyperparameter Tuning for Optimizing Stunting Classification with KNN , SVM , and Naïve Bayes Algorithms. *Jurnal*

TEKNO KOMPAK, *19*(1), 92–104.

Fordana, M. D. Y., & Rochmawati, N. (2022). Optimisasi Hyperparameter CNN Menggunakan Random Search Untuk Deteksi COVID-19 Dari Citra X-Ray Dada. *Journal of Informatics and Computer Science (JINACS)*, *4*(01), 10–18. https://doi.org/10.26740/jinacs.v4n01.p10-18

G, S. G. C. (2020). *Grid Search Tuning of Hyperparameters in Random Forest Classifier for Customer Feedback Sentiment Prediction*. *11*(9), 173–178.

Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc.

Gupta, S. C., & Goel, N. (2023). ScienceDirect ScienceDirect Predictive Modeling and Analytics for Diabetes using Predictive Modeling and Analytics for Diabetes using Hyperparameter tuned Machine Learning Techniques Hyperparameter tuned Machine Learning Techniques. *Procedia Computer Science*, *218*(2022), 1257–1269. https://doi.org/10.1016/j.procs.2023.01.104

Hendradinata, N., Gede, I., & Astawa, S. (2022). Hyperparameter Tuning Algoritma KNN Untuk Klasifikasi Kanker Payudara Dengan Grid Search CV. *JNATIA Volume*, *1*(November), 397–402.

Khatib, J., & Dalam, S. (2023). Indonesian Journal of Computer Science. *Indonesian Journal of Computer Science*, *12*(1), 1351–1365.

Lindawati, L., Fadhli, M., & Wardana, A. S. (2023). Optimasi Gaussian Naïve Bayes dengan Hyperparameter Tuning dan Univariate Feature Selection dalam Prediksi Cuaca. *Edumatic: Jurnal Pendidikan Informatika*, *7*(2), 237–246. https://doi.org/10.29408/edumatic.v7i2.21179

Massahiro, A., Instituto, S., Tecnológicas, D. P., Paulo, D. S., Univer-, F., Paulo, S., Cordeiro, R., & Paulo, S. (2023). The evolution of CRISP-DM for Data Science : Methods , Processes and Frameworks. *SBC Reviews on Computer Science*, *4*(1). https://doi.org/10.5753/reviews.2024.3757

Nugraha, W., & Sasongko, A. (2022). Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search Hyperparameter Tuning on Classification Algorithm with. *SISTEMASI: Jurnal Sistem Informasi Volume*, *11*, 391–401.

Prabu, S., Thiyaneswaran, B., Sujatha, M., Nalini, C., & Rajkumar, S. (2022). Grid Search for Predicting Coronary Heart Disease by Tuning. *Computer Systems Science & Engineering*, *43*(2). https://doi.org/10.32604/csse.2022.022739

Pramudhyta, N. A., & Rohman, M. S. (2024). Perbandingan Optimasi Metode Grid Search dan Random Search dalam Algoritma XGBoost untuk Klasifikasi Stunting. *Jurnal Media Informatika Budidarma*, *8*(1), 19. https://doi.org/10.30865/mib.v8i1.6965

Rizky, M. H., Faisal, M. R., Budiman, I., & Kartini, D. (2024). Effect of Hyperparameter Tuning Using Random Search on Tree-Based Classification Algorithm for Software Defect Prediction. *Rizky, M. H., Faisal, M. R., Budiman, I., & Kartini, D. (2024). Effect of Hyperparameter Tuning Using Random Search on Tree-Based Classification Algorithm for Software Defect Prediction. 18(1), 95–106.*, *18*(1), 95–106.

Saputra, A. G. (2024). Hyperparameter Tuning Decision Tree and Recursive Feature Elimination Technique for Improved Chronic Kidney Disease Classification. *Scientific*

Journal of Informatics, *11*(3), 821–830. https://doi.org/10.15294/sji.v11i3.12990

Science, A. C. (2023). DATA ENGINEERING IN CRISP-DM PROCESS PRODUCTION DATA – CASE STUDY. *Applied Computer Science*, *19*(3), 83–95. https://doi.org/10.35784/acs-2023-26

Shaik, S., & Sreeja, D. (2025). Ensemble Machine Learning-based Heart Disease Prediction with Hyper- tuning Parameters. *Asian Journal of Research in Computer Science Volume*, *18*(5). https://doi.org/10.9734/ajrcos/2025/v18i5642