VOL. 10. NO. 3 FEBRUARY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480 /jitk.v10i3.6214

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

OPTIMIZING THE KNN ALGORITHM FOR CLASSIFYING CHRONIC KIDNEY DISEASE USING GRIDSEARCHCV

Muhammad Rahmansyah Siregar[1*]; Dedy Hartama[2]; Solikhun[3]

Informatics Engineering Study Program[1, 3], Information Systems Study Program[2]
STIKOM Tunas Bangsa, Pematangsiantar, Indonesia[1, 2, 3]
https://stikomtunasbangsa.ac.id[1, 2, 3]
rahmansyahsiregar77@gmail.com[1*], dedyhartama@amiktunasbangsa.ac.id[2],
solikhun@amiktunasbangsa.ac.id[3]

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract— *Chronic Kidney Disease (CKD) is a progressive condition that impairs kidney function and cannot be cured. Early detection is crucial for effective management and therapy. However, diagnosing CKD is challenging as patients often have comorbidities such as diabetes, hypertension, or heart disease, which complicate diagnosis and treatment. Accurate classification methods are essential for early detection. K-Nearest Neighbor (KNN) is a classification algorithm that groups data based on feature similarity. K-NN is an algorithm that is resistant to outliers, easy to implement, and highly adaptable. It only requires distance calculations between data points and does not involve complex parameters. However, its performance depends on hyperparameters such as the number of neighbors (k), weighting, and distance metric. Incorrect hyperparameter selection can lead to overfitting, underfitting, or reduced accuracy. To address these issues, GridSearchCV is used to optimize KNN by systematically selecting the best hyperparameters, ensuring improved accuracy and reduced overfitting. This optimization enhances the model's reliability in early CKD detection compared to other methods. This study aims to determine the optimal KNN parameters for CKD classification using GridSearchCV. The results show 8.05% accuracy improvement and reduction in overfitting, with the prediction gap between training and testing decreasing from 6% to only 1.15%. These enhancements contribute to more reliable CKD diagnosis, enabling accurate early detection and better clinical decision-making.*

Keywords: *chronic kidney disease, classification, gridsearchCV, KNN, optimization.*

Intisari— *Penyakit Ginjal Kronis (PGK) adalah kondisi progresif yang merusak fungsi ginjal dan tidak dapat disembuhkan. Deteksi dini sangat penting untuk manajemen dan terapi yang efektif. Namun, diagnosis PGK seringkali menantang karena pasien sering mengalami komorbiditas seperti diabetes, hipertensi, atau penyakit jantung, yang memperumit diagnosis dan pengobatan. Oleh karena itu, metode klasifikasi yang akurat sangat diperlukan untuk deteksi dini. K-Nearest Neighbor (KNN) adalah algoritma klasifikasi yang mengelompokkan data berdasarkan kesamaan fitur. K-NN merupakan algoritma yang tahan terhadap outlier, mudah diimplementasikan, dan dapat beradaptasi dengan baik. Algoritma ini hanya memerlukan perhitungan jarak antar data dan tidak melibatkan parameter yang kompleks. Namun, kinerjanya sangat bergantung pada hyperparameter seperti jumlah tetangga terdekat (k), pembobotan, dan metrik jarak. Pemilihan hyperparameter yang tidak tepat dapat menyebabkan overfitting, underfitting, atau akurasi yang menurun. Untuk mengatasi masalah ini, GridSearchCV digunakan untuk mengoptimalkan KNN dengan cara memilih hyperparameter terbaik secara sistematis, sehingga meningkatkan akurasi dan mengurangi overfitting. Optimasi ini meningkatkan keandalan model dalam deteksi dini PGK dibandingkan dengan metode lainnya. Penelitian ini bertujuan untuk menentukan parameter optimal KNN untuk klasifikasi PGK menggunakan GridSearchCV. Hasil penelitian menunjukkan peningkatan akurasi sebesar 8.05% dan pengurangan overfitting, dengan selisih antara prediksi pelatihan dan pengujian yang menurun dari 6%*

VOL. 10. NO. 3 FEBRUARY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i3.6214

menjadi 1,15%. Peningkatan ini berkontribusi pada diagnosis PGK yang lebih andal, memungkinkan deteksi dini yang lebih akurat dan pengambilan keputusan klinis yang lebih baik.

***Kata Kunci**: penyakit ginjal kronis, klasifikasi, gridsearchCV, KNN, optimisasi.*

.

INTRODUCTION

The kidneys are essential for maintaining the body's balance by performing vital functions such as detoxifying the blood, filtering waste, regulating fluid levels, and eliminating toxins through urine [1]. The World Health Organization (WHO) ranks kidney disease as the 9th leading cause of death in high-income countries, with 20 deaths per 100,000 population in crude death rates [2].

Chronic kidney disease (CKD) is a condition marked by the gradual and irreversible decline of kidney function. This impairment prevents the body from maintaining proper metabolism, fluid, and electrolyte balance, leading to elevated levels of urea [3]. Patients with end-stage kidney disease must undergo either a kidney transplant or dialysis, which significantly impacts their quality of life [4]. The symptoms of this disease develop slowly and are often not noticeable at first [5]. CKD can be prevented and treated effectively, particularly if detected early [6]. Prompt action in chronic kidney disease patients can slow disease progression. This not only improves patients' quality of life but also reduces the risk of severe illness, death, and the high costs associated with kidney replacement therapy[7]. The challenge in diagnosing kidney disease is that patients frequently also have other health conditions, such as diabetes, hypertension, or heart disease [8]. These conditions complicate the diagnosis and treatment as they also impact kidney function. Therefore, an accurate diagnosis is essential for the early detection of chronic kidney disease. Classification is the process of categorizing objects based on similar features or characteristics [9]. Classification methods have been widely used to predict diseases [10]–[16]. The classification system estimates event likelihoods using existing data, helping to prevent losses and maximize potential gains [17]. By using classification, chronic kidney disease (CKD) patients can be easily detected based on several medical attributes that contribute to the disease. This study proposes using the K-NN algorithm optimized with GridSearchCV for classifying CKD.

K-Nearest Neighbor (KNN) is an algorithm used for classification based on the similarity of features or attributes [18], [19]. The KNN algorithm has also been widely used to detect diseases [20]–[22]. KNN algorithm has several important hyperparameters, such as the number of nearest neighbors (k), weighting, and distance metric. An inappropriate choice of the k value can lead to overfitting, where the model fits the training data too closely, making it difficult to predict new data effectively, or underfitting, which occurs when the model is too simple to effectively capture the patterns in the data [23]–[25]. Additionally, improper weighting and distance calculations can lead to less accurate predictions [26]. To address the issues of underfitting, overfitting, and less accurate predictions in KNN, researchers use GridSearchCV. GridSearchCV, or Grid Search Cross-Validation, is a technique that helps find the set of parameters that yields the best performance for a particular model [27]. The use of GridSearchCV in K-NN aims to optimize hyperparameters to find the best combination that provides optimal model performance. The parameters that can be tested with GridSearchCV for the K-NN algorithm include parameter k, the weighting parameter, and the distance calculation parameter. By utilizing GridSearchCV, the K-NN model can be tuned to achieve more accurate and optimal predictions without the need to manually experiment with different hyperparameters. The aim of this research is to identify the optimal parameters for classifying chronic kidney disease using the K-NN algorithm combined with GridSearchCV.

Previous research on CKD is study [28], The researchers utilized the Naïve Bayes Classifier and K-Nearest Neighbor algorithms for classifying CKD. A 10-fold cross-validation was used to determine optimal results, with evaluation model such as accuracy, recall, precision, and ROC. The results indicated that the Naïve Bayes Classifier, using K-Fold Cross Validation, achieved an accuracy of 94.25%, a precision of 98.40%, a recall of 94.23%, and an AUC of 0.961. In comparison, the K-NN algorithm with K-Fold Cross Validation showed an accuracy of 77.79%, a precision of 80.20%, a recall of 95.06%, and an AUC of 0.627. These results indicate that classification using the Naïve Bayes Classifier outperformed the K-NN algorithm.

The primary reference for this research is study [29], which used the K-Nearest Neighbor method to classify chronic kidney disease. This study implemented a 75:25 data split and evaluated the model with a confusion matrix, measuring accuracy, F1-score, recall, and precision. The results showed that the system was able to classify the data

with an accuracy of 92.59%, a precision of 89.85%, a recall of 87.32%, and an F1-score of 88.57%.

In the studies referenced as [28] and [29], the K-NN algorithm utilized default parameters, resulting in suboptimal outcomes. Specifically, [28] reported K-NN performance with an AUC value close to 0.5, indicating predictive ability close to random guessing. Study [28] only optimized the model using K-Fold Validation without modifying the parameters of both algorithms. Additionally, the preprocessing step, which only involved data cleaning, caused the K-NN algorithm—sensitive to data distance—to suffer from underfitting. Meanwhile, study [29] conducted only a single experiment using the default K-NN parameters, leaving room for potential improvements through further parameter exploration. In this research, the GridSearchCV method is implemented to facilitate the selection of optimal parameters efficiently without requiring extensive time. Furthermore, proper preprocessing plays a crucial role in improving the model's accuracy. Therefore, this study applies preprocessing techniques, including data cleaning, normalization, class balancing, and feature selection.

The objective of this research is to achieve optimal performance for the K-NN algorithm in CKD classification. The GAP of this study from the referenced studies lies in the use of GridSearchCV for K-NN parameter optimization. The novelty of this research is the enhancement of the K-NN algorithm's performance in CKD classification. This study will implement the K-NN algorithm with parameter optimization using GridSearchCV for CKD classification.

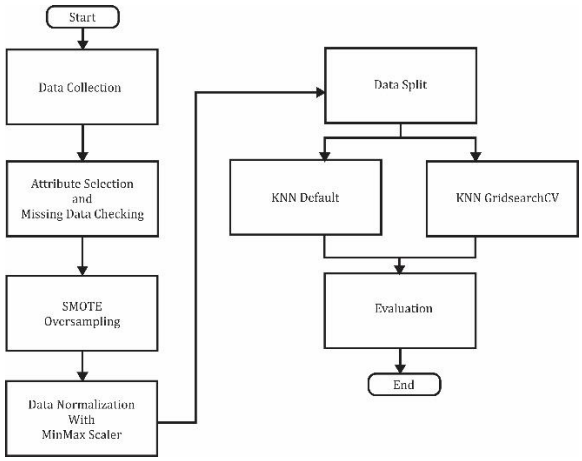
This research contributes to the medical field, particularly in the diagnosis CKD, by providing a robust method for detecting this condition. The findings of this study can contribute to clinical applications by improving patient classification accuracy, aiding in early disease detection, and optimizing treatment decisions based on data-driven insights like clinical decision support systems. Additionally, this study advances the field of data mining, specifically regarding the application of the K-NN algorithm for optimal parameter determination in health data analysis.

This research consists of four sections: introduction, research methods, results & discussion, and conclusion. In the research methods section, the researcher explains the steps taken and reviews relevant literature related to the study. In the results section, the researcher presents the findings of the study, and in the conclusion section, the researcher analyzes and summarizes the results obtained.

MATERIALS AND METHODS

Research Method

This research consists of three stages: the data preprocessing stage, the testing stage with KNN using default parameters and KNN with GridSearchCV, and the evaluation stage. The researcher uses Python programming with Jupyter Notebook to classify the data. The flowcharts of the research in Figures 1.



Source: (Research Result's, 2024)

Figure 1. Research Flow Diagram

In Figure 1, a research flowchart is presented. The first step involves finding suitable data for this study, which is sourced from Kaggle and consists of 491 data points. This data has not undergone preprocessing, so applying it directly to the algorithm may lead to issues. Preprocessing addresses these problems by cleaning and enriching the data, as well as normalizing values to ensure they fall within a consistent range. With cleaner and more uniform data, machine learning models can learn patterns and relationships within data and make predictions more accurately and efficiently.

The second step is the application of feature selection and the removal of missing data. Choosing the most relevant features allows the model to focus on significant information, reducing the risk of overfitting, improving performance, and speeding up the training process. Removing missing data is crucial to ensure that the analysis is conducted on clean and complete data, leading to more valid conclusions.

The third step involves applying the Synthetic Minority Over-sampling Technique (SMOTE). When working with imbalanced data in classification tasks, issues like majority class bias can occur. This bias happens when the model tends to predict the more frequently occurring class, leading to low sensitivity for the minority class.

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 3 FEBRUARY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i3.6214

Consequently, the model may miss crucial patterns within underrepresented data, leading to poor performance on the minority class, despite appearing to have high overall accuracy. To address this class imbalance in the dataset, the researcher employs SMOTE. SMOTE was chosen for oversampling the minority class as it generates synthetic samples through interpolation between existing minority class instances, effectively addressing the class imbalance issue. Other methods, such as ADASYN, also aim to improve minority class representation, but ADASYN focuses on generating samples around difficult-to-classify instances. While ADASYN can be beneficial in some cases, SMOTE was selected here for its simplicity and general effectiveness in balancing the class distribution.

The fourth step is to perform data normalization using MinMaxScaler. Data normalization is a preprocessing step aimed at standardizing the scale of features within the dataset, ensuring that all variables have a uniform range of values. Without normalization, features with larger scales can dominate the influence on the model, resulting in the model being less responsive to features that may also be important but have smaller value ranges. Normalization ensures that all features contribute equally to decision-making. MinMaxScaler is a normalization technique used to transform features in the dataset so that their values fall within a specified range, typically [0, 1]. This process involves subtracting the minimum value of each feature and then dividing the result by the range (the difference between the maximum and minimum values).

The fifth step is to split the preprocessed data using stratified sampling for classification. Splitting the data helps ensure that the model can be effectively evaluated and optimized. By splitting the dataset into training and testing sets, the model can be trained on one subset of data and tested on its ability to generalize to unseen data. In this stage, the researcher performs a data split of 80:20, allocating 80% of the data for training and 20% for testing. The researcher uses stratified splitting, which maintains the proportion of class distribution in each resulting subset, ensuring that both the training and testing sets have the same class distribution. The data splitting stage completes the preprocessing phase before testing the model using KNN Default and KNN with GridSearchCV.

The sixth step is to implement the K-Nearest Neighbors (KNN) algorithm, along with KNN using GridSearchCV. In the GridSearchCV for KNN, researcher will test several parameters, including the value of k, which will be selected from odd values ranging from 3 to 30. The researcher also evaluated two weighting options: "uniform" and "distance." The cross-validation will use 10 folds. Additionally, we will consider two distance calculation methods: Euclidean Distance and Manhattan Distance. The optimal result from the KNN GridSearchCV will be determined as the average accuracy of the training data with the specified parameters. The model's performance will be assessed using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

The final step is the evaluation stage. At this point, the researcher will compare the performance of KNN with GridSearchCV to that of the default KNN, aiming to determine if using GridSearchCV optimization leads to an improvement.

Data Collection

The data used in this research consists of medical records of chronic kidney disease patients from Tawam Hospital, Al-Ain City, Abu Dhabi, United Arab Emirates, obtained from the Kaggle website (https://www.kaggle.com/datasets/davidechicco/chronic-kidney-disease-ehrs-abu-dhabi).

Table 1. Sample of Chronic Kidney Disease Patient Record

No	Sex	AgeBaseline	...	EventCKD35	TIME_YEAR
1	0	64	...	0	8
2	0	52	...	0	9
...	...	...	...	...	...
491	0	0	...	0	0

Source: (Research Result's, 2024)

Table 1 presents an example of patient records with Chronic Kidney Disease. The dataset contains 491 records with 22 attributes. Each attribute includes Sex, AgeBaseline, HistoryDiabetes, HistoryCHD, HistoryVascular, HistorySmoking, HistoryHTN, HistoryDLD, HistoryObesity, DLDmeds, DMmeds, HTNmeds, ACEIARB, CholesterolBaseline, CreatinineBaseline, eGFRBaseline, sBPBaseline, dBPBaseline, BMIBaseline, TimeToEventMonths, EventCKD35, and TIME_YEAR, all of which can indicate whether a person has CKD. This data will be used for evaluation and classification using the KNN algorithm.

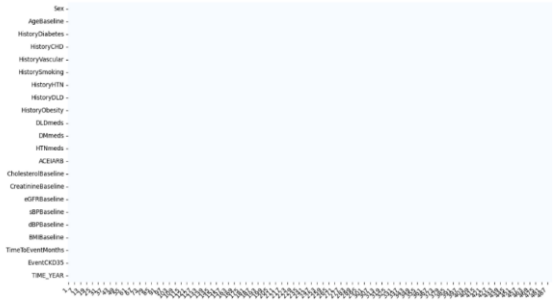
RESULTS AND DISCUSSION

In this section, the researcher will apply the methods outlined in the flowchart to conduct the study. The results obtained from this research indicate a significant improvement in classifying

chronic kidney disease using GridSearchCV for the KNN algorithm.

Preprocessing Data

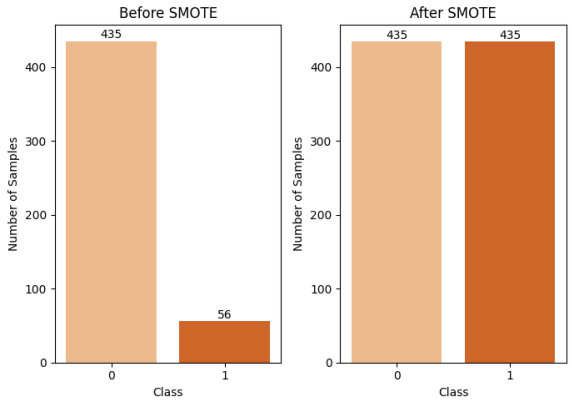
At this stage, the researcher performs data preprocessing to ensure that the results of the study are optimal. The researcher utilizes SMOTE, MinMaxScaler, feature selection, and checks for missing data to obtain a quality dataset that can yield optimal testing. First, a check for missing values in the chronic kidney disease data is conducted. In this process, the identification of columns or rows with empty or missing values is performed:



Source: (Research Result's, 2024)

Figure 2. Plot Missing Value

In Figure 2 above, the data used does not have any missing values, so there is no need for adding or removing data. The next step is feature selection for the classification. In this stage, the researcher removes the attributes TimeToEventMonths and TIME_YEAR from the data, as these attributes do not influence the classification or the chronic kidney disease.



Source: (Research Result's, 2024)

Figure 3. Classes CKD35 Before and After SMOTE

Next is the step of balancing classes within the data. The attribute being balanced in this stage is the EventCKD35 attribute (the class attribute). The class imbalance between class 0 (435 samples) and class 1 (56 samples) may cause the model to underfit. Therefore, oversampling is applied to class 1 to balance the data distribution between classes. The balancing of this attribute uses the SMOTE oversampling technique. The results before and after applying SMOTE can be seen in Figure 3.

Next, the researcher applies MinMaxScaler for data normalization, which aims to change the range of values for each feature to the interval [0, 1]. This process helps improve the model's performance by ensuring that all features have the same scale. The attributes that undergo normalization are AgeBaseline, CholesterolBaseline, CreatinineBaseline, eGFRBaseline, sBPBaseline, dBPBaseline, and BMIBaseline. The results of the data normalization can be seen in Table 1 below.

Table 2. Data Attributes after Normalization

No	Age Baseline	CholesterolBaseline	CreatinineBaseline	eGFRBaseline	sBPBaseline	dBPBaseline	BMIBaseline
1	0.62	0.36	0.45	0.18	0.59	0.64	0.61
2	0.43	0.58	0.39	0.25	0.63	0.7	0.72
3	0.5	0.58	0.43	0.21	0.64	0.63	0.63
...	...	...	...	...	...	...	...
870	0.69	0.16	0.61	0.16	0.46	0.32	0.29

Source: (Research Result's, 2024)

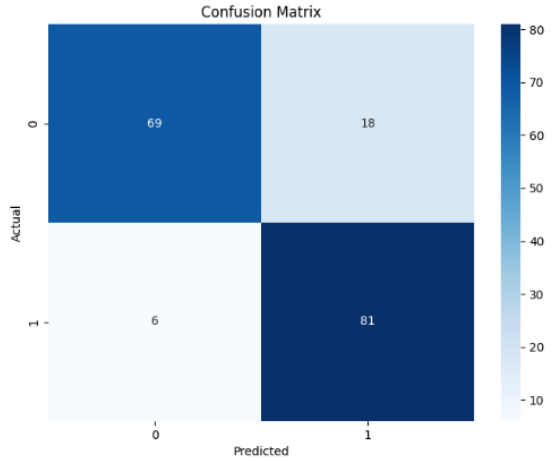
As seen in Table 2, the normalized data has a value range from 0 to 1. This normalized data is now ready for splitting. In the final stage, the researcher divides the data using the Stratified K-Fold method to ensure a balanced class distribution in each fold. The data is divided into 80% for the training set and 20% for the testing set. The Stratified K-Fold method is chosen because it maintains the class proportions in both the training and testing datasets, making the model evaluation results more representative of the overall data.

KNN Implementation

After the data preprocessing is completed, the next step is to test the data using K-NN, starting with the KNN Default test. The KNN with default parameters uses a metric of Minkowski, n=5, and weight = 'uniform'. The model was evaluated only once using the test data. The results from the KNN with default parameters yield a high F1 score of 86%, accuracy of 86%, recall of 86%, and precision of 87%. This model shows a good balance between precision and recall, reflecting the effectiveness of KNN in classification. The confusion matrix for the default KNN algorithm can be seen in Figure 4.

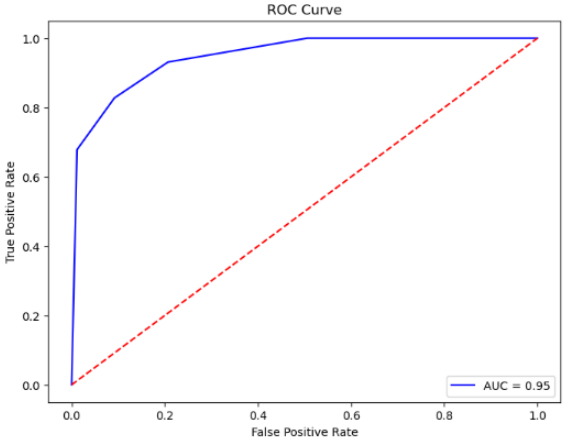
Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 3 FEBRUARY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i3.6214



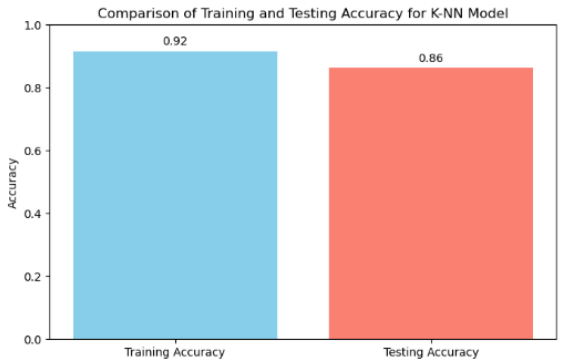
Source: (Research Result's, 2024)
Figure 4. Confusion Matrix of Default KNN.

From Figure 4, it is evident that the default parameters of the KNN algorithm successfully classified at least 150 data points, with around 24 data points resulting in errors. This indicates that even without utilizing GridSearchCV, the KNN algorithm is already quite effective for classification tasks.



Source: (Research Result's, 2024)
Figure 5. ROC Curve of Default KNN.

The ROC-AUC graph in Figure 5 on the right shows a high ROC AUC score of 0.948 (94.8%), which is close to 1. A ROC AUC value approaching 1 signifies very good pattern recognition in classification. Figure 5 shows high TPR and low FPR at most thresholds, indicating the model's ability to detect most positive cases while minimizing false positives. the overall performance of the K-NN model is very good. Additionally, the comparison between the testing data and training data is illustrated in Figure 6.



Source: (Research Result's, 2024)
Figure 6. KNN Default Comparison Results.

In Figure 6, that the accuracy on the training set is 0.92(92%) while the accuracy on the testing set is 0.86(86%). This model shows indications of overfitting, as indicated by the 0.6(6%) accuracy difference between training and testing. This accuracy gap suggests that the model may have captured specific details from the training data that do not necessarily apply to unseen data, thereby risking a lack of generalization.

Tuning With GridSearchCV

Next, model tuning will be performed using GridSearchCV. In this study, the parameters selected for testing on the K-Nearest Neighbors (KNN) model include the number of neighbors (n_neighbors), weighting method (weights), type of distance calculation (metric), and the number of folds in cross-validation (K-Fold). These parameters are summarized in Table 3, which aims to optimize the performance of the KNN model by finding the best parameter combinations.

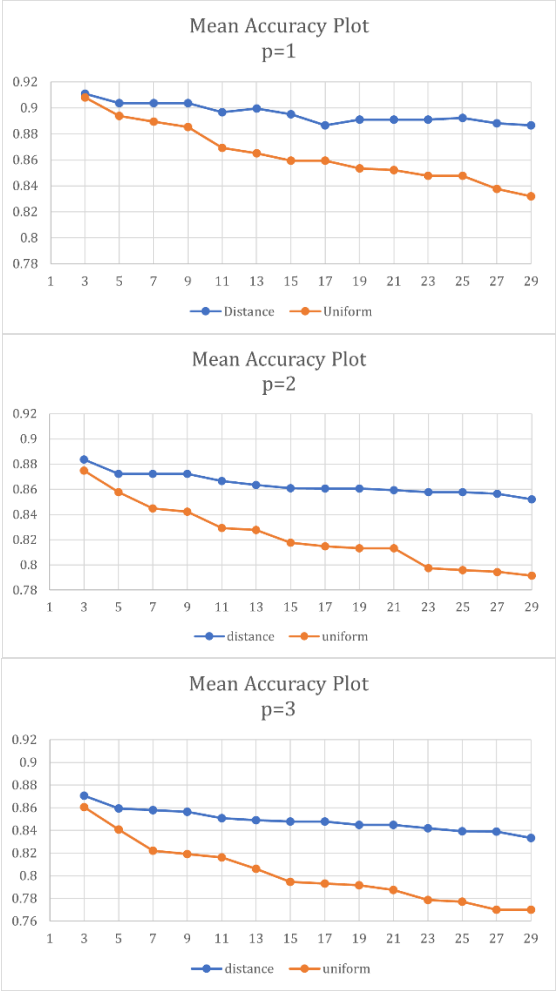
Table 3. Parameters to Be Tested With GridsearchCV.

Parameters Name	Value
K (n_neighbors)	Odd value 3 - 30
weights	"uniform","distance"
distance metric (p)	'euclidean'(p=1), 'manhattan'(p=2), 'minkowski'(p=3)
K-Fold	10

Source: (Research Result's, 2024)

In Table 3 the parameter range of 3-30 for the number of neighbors in K-NN was chosen to explore a variety of values that could balance the model's bias and variance. Odd values were specifically selected to avoid ties in classification, as even values of k could lead to ambiguous results in cases of equal class distribution among neighbors. This range provides a comprehensive search for the optimal k-value to improve model performance. The tested weighting methods included both uniform and distance-based approaches. Three different

distance metrics were utilized for calculating the distance between neighbors: Euclidean distance, Manhattan distance, and Minkowski distance. The data were split using the K-Fold method, with the parameter set to 5 folds. A comparison of the accuracy results based on these specified parameters can be found in Figure 7 below.



Source: (Research Result's, 2024)

Figure 7. Comparison of Training Accuracy for Each Weight Using GridSearchCV Parameters

In Figure 7, it can be observed that the variation in the value of k significantly affects the performance of the K-Nearest Neighbors (KNN) algorithm. Higher values of k, or a larger number of neighbors, tend to decrease the model's accuracy. This is due to the nature of the KNN algorithm, where an increasing value of k means that more neighbors are considered in the classification process. Consequently, predictions become more influenced by points that may not be fully relevant, thus reducing the model's ability to capture specific patterns in the data. On the other hand, smaller values of k allow the model to consider only a few closest neighbors in determining the class of the data. This increases sensitivity to local patterns around the tested data point, resulting in higher accuracy. However, with a low value of k, the model risks overfitting, as classification decisions are overly influenced by the data surrounding that point and are less capable of generalizing to new data. The researcher has summarized the best results from each specified distance metric, which can be seen in Table 4 below.

Table 4. Optimal Results of Each Distance Calculation

No	p	K	weight	Train	Test	F-1	AUC
1	1	3	distance	1	0.942	0.945	0.990
2	1	3	uniform	0.954	0.942	0.945	0.987
3	2	3	distance	0.939	0.908	0.911	0.962
4	2	3	uniform	1	0.925	0.928	0.973
5	3	3	distance	1	0.919	0.923	0.967
6	3	3	uniform	93.53	0.902	0.907	0.955

Source: (Research Result's, 2024)

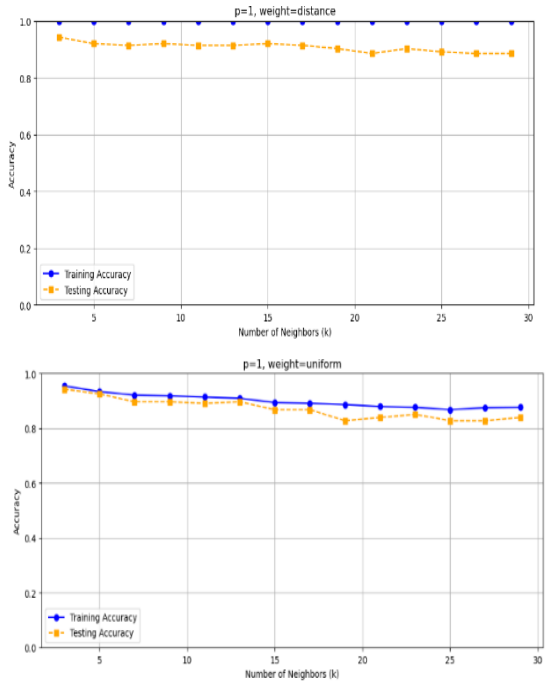
In Table 4, it is shown that the best results from GridSearch using 10-fold cross-validation were achieved with parameters p = 1, k = 3, and weights = distance. This combination of parameters resulted in 100% accuracy on the training data, an F1 score of 0.945, and a ROC-AUC value of 0.990. However, these results indicate signs of overfitting, with a difference in accuracy between the training and testing data of 0.58(5.8%), as same as the results obtained using the default KNN settings.

For more stable results without overfitting, the parameter combination of p = 1, k = 3, and weights = uniform showed better performance. This combination resulted in a difference in accuracy between the training and testing data of only 0.115 (1.15%). Compared to other parameters, this one had the smallest overfitting value while maintaining high accuracy. The small difference in accuracy indicates that the model successfully learned patterns from the data without overfitting. With minimal disparity between the training and testing data, the model demonstrates stable and consistent performance on unseen data, making it reliable for predictions beyond the training data.

Based Figure 8, it can be observed that the KNN algorithm using weights = distance tends to experience overfitting, while KNN with weights = uniform tends to be more stable. Based on this indication, it is determined that the algorithm using uniform weights is the optimal parameter among all the parameter combinations used. Therefore, the most optimal results are achieved with parameters p = 1, weights = uniform, and k = 3, resulting in an accuracy of 94.25%, precision of 95%, recall of 94%, F1-score of 94%, and an AUC value of 0.9865.

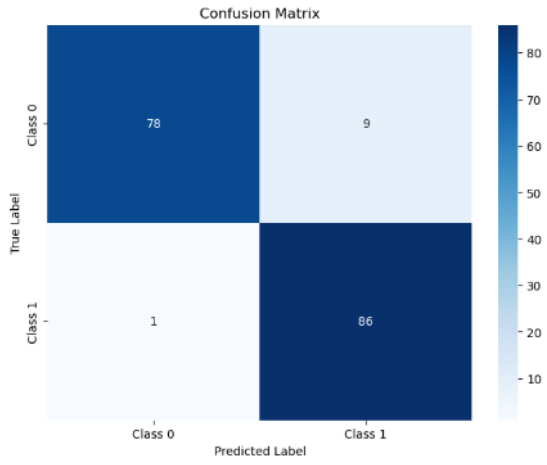
Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 3 FEBRUARY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i3.6214



Source: (Research Result's, 2024)

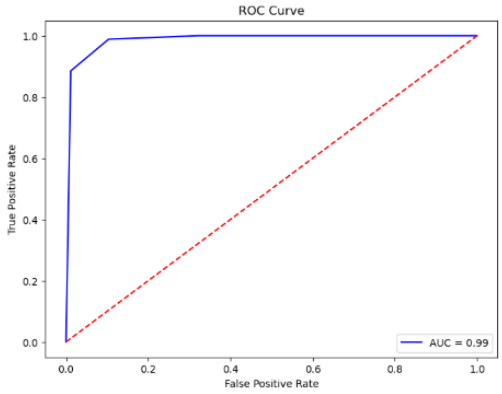
Figure 8. Comparison of Training Testing Accuracy for P=1 and P=2



Source: (Research Result's, 2024)

Figure 9. Confusion Matrix Best Parameters KNN Algorithm.

From Figure 9, the best parameters from GridSearchCV for KNN were able to accurately test at least 164 data points, with about 10 data points resulting in errors. This leads to the conclusion that the algorithm with the best parameters outperforms the default parameters.



Source: (Research Result's, 2024)

Figure 10. ROC Best Parameters KNN Algorithm.

The ROC results for the optimal parameters with GridSearchCV KNN indicate an optimal value of 0.98652, meaning this model is very effective at recognizing patterns in classification with the dataset provided (Figure 10). ROC = 0.98652 shows that the KNN model is nearly perfect in distinguishing between different classes. The use of GridSearchCV has helped optimize the hyperparameters of the KNN model, maximizing its performance for this data. A comparison between KNN with default parameters and KNN with GridSearchCV can be seen in Table 5.

Table 5. KNN Comparison

Algoritma	Testing	F-1 Uji	AUC
KNN Gridsearchcv	94.25%	0.945	0.987
KNN Default	86.2%	0.861	0.948

Source: (Research Result's, 2024)

In Table 5, it is evident that optimizing parameters using GridSearchCV significantly improves model performance. By using the optimized parameters, the model's accuracy increased by 0.805(8.05%), indicating that the model is better able to classify the data correctly. Additionally, there was an increase of 0.084 in the F1 score, and the AUC value rose by 0.038. This increase in AUC signifies that the model's ability to distinguish between classes has improved after optimization. Overall, the use of optimized parameters allows the algorithm to operate more efficiently and accurately compared to the default parameter settings.

The model using KNN also has a small difference between training and testing results of only 0.115(1.15%), compared to the default KNN, which has a difference of 0.6(6%), where the training data has a higher accuracy than the testing data used. Furthermore, from the confusion matrices for KNN GridSearchCV (Figure 9) and KNN default (Figure 4), that the number of incorrect predictions for each class in KNN GridSearchCV is

10, while KNN default has 24 incorrect predictions. This indicates that KNN with GridSearchCV optimization can enhance the results of the KNN model.

CONCLUSION

The results of this study indicate an improvement in the model of 8.05% and a reduction to just 1.15% in the difference between training and testing predictions. The evaluation results demonstrate consistent performance across both training and test datasets, suggesting that the model generalizes well and is not overly sensitive to the specific data it was trained on. The best model for classifying CKD using KNN is with K=3, p=1 (Manhattan), and Weights = "Uniform." The best results from this algorithm include an accuracy of 94.25%, an F1 Score of 0.945, and an AUC of 0.987. It can be concluded that optimization with GridSearchCV enhances model performance, especially in the classification of CKD. For future research, it may be beneficial to use the KNN algorithm to predict CKD with other optimization techniques or to compare this model with other optimized algorithms.

REFERENCE

[1] I. Wisnuadji Gamadarenda and I. Waspada, "Implementasi Data Mining Untuk Deteksi Penyakit Ginjal Kronis (Pgk) Menggunakan K-Nearest Neighbor (Knn) Dengan Backward Elimination," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 7, no. 2, pp. 417–426, 2020, doi: 10.25126/jtiik.202071896.

[2] D. Anggraini, "Aspek Klinis Dan Pemeriksaan Laboratorium Penyakit Ginjal Kronik," *An-Nadaa Jurnal Kesehatan Masyarakat*, vol. 9, no. 2, p. 236, 2022, doi: 10.31602/ann.v9i2.9229.

[3] R. Dewi and A. Mustofa, "Penurunan Intensitas Rasa Haus Pasien Penyakit Ginjal Kronik Yang Menjalani Hemodialisa Dengan Menghisap Es Batu," *Ners Muda*, vol. 2, no. 2, p. 17, Aug. 2021, doi: 10.26714/nm.v2i2.7154.

[4] R. Simorangkir, T. M. Andayani, and C. Wiedyaningsih, "Faktor-Faktor yang Berhubungan dengan Kualitas Hidup Pasien Penyakit Ginjal Kronis yang Menjalani Hemodialisis," *Jurnal Farmasi Dan Ilmu Kefarmasian Indonesia*, vol. 8, no. 1, p. 83, 2021, doi: 10.20473/jfiki.v8i12021.83-90.

[5] N. P. Nugraha, R. Azim, S. Z. Daffa, and P. S. Ningayu, "Perbandingan Akurasi Metode Naïve Bayes dan Metode KNN untuk Memprediksi Gagal Ginjal Kronis," *Jurnal Rekayasa Elektro Sriwijaya*, vol. 5, no. 1, pp. 1–10, Nov. 2023, doi: 10.36706/jres.v5i1.63.

[6] E. Y. Chrisanto, R. P. Rahmawati, P. S. Azahra, and W. Amelia, "Penyuluhan kesehatan tentang prilaku hidup sehat pasien dengan gangguan ginjal kronik," *JOURNAL OF Public Health Concerns*, vol. 2, no. 1, pp. 34–40, Feb. 2022, doi: 10.56922/phc.v2i1.184.

[7] H. D. Siswaja and Y. Ramdhani, "Pendekatan Algoritma Neural Network dan Genetic Algorithm Untuk Prediksi Penyakit Ginjal Kronis," *Jurnal Responsif*, vol. 6, no. 2, pp. 232–239, 2024, [Online]. Available: https://ejurnal.ars.ac.id/index.php/jti

[8] U. Hasanah, N. R. Dewi, L. Ludiana, A. T. Pakarti, and A. Inayati, "Analisis Faktor-Faktor Risiko Terjadinya Penyakit Ginjal Kronik Pada Pasien Hemodialisis," *Jurnal Wacana Kesehatan*, vol. 8, no. 2, p. 96, Nov. 2023, doi: 10.52822/jwk.v8i2.531.

[9] M. Rizal, M. Z. Syahaf, S. R. Priyambodo, and Y. Rhamdani, "Optimasi Algoritma Naïve Bayes Menggunakan Forward Selection Untuk Klasifikasi Penyakit Ginjal Kronis," *Naratif : Jurnal Nasional Riset, Aplikasi dan Teknik Informatika*, vol. 5, no. 1, pp. 71–80, Jun. 2023, doi: 10.53580/naratif.v5i1.200.

[10] Z. Zuriati and N. Qomariyah, "Klasifikasi Penyakit Stroke Menggunakan Algoritma K-Nearest Neighbor (KNN)," *ROUTERS: Jurnal Sistem dan Teknologi Informasi*, vol. 1, no. 1, pp. 1–8, Nov. 2022, doi: 10.25181/rt.v1i1.2665.

[11] E. Saputro and D. Rosiyadi, "Penerapan Metode Random Over-Under Sampling Pada Algoritma Klasifikasi Penentuan Penyakit Diabetes," *Bianglala Informatika*, vol. 10, no. 1, pp. 42–47, Mar. 2022, doi: 10.31294/bi.v10i1.11739.

[12] G. Abdurrahman, "Klasifikasi Penyakit Diabetes Melitus Menggunakan Adaboost Classifier," *JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia)*, vol. 7, no. 1, pp. 59–66, Mar. 2022, doi: 10.32528/justindo.v7i1.4949.

[13] K. Widya Kayohana, "KLASIFIKASI Penyakit Hati Menggunakan Random Forest Dan Knn," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 4, pp. 7924–7929, Aug. 2024, doi: 10.36040/jati.v8i4.10457.

[14] R. Rozaq, "Klasifikasi Penyakit Dengue Menggunakan Algoritma K-Nearest Neighbors Berbasis Flask," *Remik*, vol. 6, no. 3, pp. 359–369, Aug. 2022, doi:

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 3 FEBRUARY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i3.6214

10.33395/remik.v6i3.11501.

[15] Q. A'yuniyah *et al.*, "Implementasi Algoritma Naïve Bayes Classifier (NBC) untuk Klasifikasi Penyakit Ginjal Kronik," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 4, no. 1, p. 72, Sep. 2022, doi: 10.30865/json.v4i1.4781.

[16] Z. Maisat, E. Darmawan, and A. Fauzan, "Implementasi Optimasi Hyperparameter GridSearchCV Pada Sistem Prediksi Serangan Jantung Menggunakan SVM Implementation of GridSearchCV Hyperparameter Optimization in Heart Attack Prediction System Using SVM," *unipdu*, vol. 13, no. 1, pp. 8–15, 2023, doi: https://doi.org/10.26594/teknologi.v13i1.3098.

[17] A. Sepharni, I. E. Hendrawan, and C. Rozikin, "Klasifikasi Penyakit Jantung dengan Menggunakan Algoritma C4.5," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 7, no. 2, p. 117, Dec. 2022, doi: 10.30998/string.v7i2.12012.

[18] Y. Pratama, A. Prayitno, D. Azrian, N. Aini, Y. Rizki, and E. Rasywir, "Klasifikasi Penyakit Gagal Jantung Menggunakan Algoritma K-Nearest Neighbor," *Bulletin of Computer Science Research*, vol. 3, no. 1, pp. 52–56, Dec. 2022, doi: 10.47065/bulletincsr.v3i1.203.

[19] E. Laksono, A. Basuki, and F. Bachtiar, "Optimization of K Value in KNN Algorithm for Spam and Ham Email Classification," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 2, pp. 377–383, Apr. 2020, doi: 10.29207/resti.v4i2.1845.

[20] N. M. Putry, "Komparasi Algoritma Knn Dan Naïve Bayes Untuk Klasifikasi Diagnosis Penyakit Diabetes Mellitus," *EVOLUSI : Jurnal Sains dan Manajemen*, vol. 10, no. 1, pp. 45–57, Apr. 2022, doi: 10.31294/evolusi.v10i1.12514.

[21] S. T. Kusuma and T. B. Sasongko, "Optimasi K-Nearest Neighbor dengan Grid Search CV pada Prediksi Kanker Paru-Paru," *The Indonesian Journal of Computer Science*, vol. 12, no. 4, pp. 2162–2171, Aug. 2023, doi: 10.33022/ijcs.v12i4.3267.

[22] I. P. Putri, "Analisis Performa Metode K-Nearest Neighbor (KNN) dan Crossvalidation pada Data Penyakit Cardiovascular," *Indonesian Journal of Data and Science*, vol. 2, no. 1, pp. 21–28, Mar. 2021, doi: 10.33096/ijodas.v2i1.25.

[23] Rima Dias Ramadhani, A. Nur Aziz Thohari, C. Kartiko, A. Junaidi, T. Ginanjar Laksana, and N. Alim Setya Nugraha, "Optimasi Akurasi Metode Convolutional Neural Network untuk Identifikasi Jenis Sampah," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 2, pp. 312–318, Apr. 2021, doi: 10.29207/resti.v5i2.2754.

[24] N. Rikatsih, M. Anshori, R. Siwi Pradini, and F. Faurika, "K-Nearest Neighbor Method for Early Detection of Diabetes Patients Based on Symptoms and Clinical Data," *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 9, no. 2, pp. 187–193, Aug. 2024, doi: 10.25139/inform.v9i2.8582.

[25] Y. Putra Dinata, M. Fikry, F. Yanto, and E. Pandu Cynthia, "Analisis Sentimen Terhadap Sebuah Figur Publik di Twitter Menggunakan Metode K-Nearest Neighbor," *Media Online)*, vol. 4, no. 6, pp. 2822–2829, 2024, doi: 10.30865/klik.v4i6.1904.

[26] R. Fadilah, Y. H. Chrisnanto, and G. Abdillah, "Pengaruh metode pengukuran jarak dan smote pada klasifikasi penilaian kredit," *JIRE (Jurnal Informatika & Rekayasa Elektronika*, vol. 7, no. 2, pp. 193–202, 2024, doi: https://doi.org/10.36595/jire.v7i2.1213.

[27] G. N. Ahmad, H. Fatima, S. Ullah, A. Salah Saidi, and Imdadullah, "Efficient Medical Diagnosis of Human Heart Diseases Using Machine Learning Techniques With and Without GridSearchCV," *IEEE Access*, vol. 10, no. March, pp. 80151–80173, 2022, doi: 10.1109/ACCESS.2022.3165792.

[28] V. Wulandari, W. J. Sari, Z. Alfian, L. Legito, and T. Arifianto, "Implementasi Algoritma Naïve Bayes Classifier dan K-Nearest Neighbor untuk Klasifikasi Penyakit Ginjal Kronik," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 2, pp. 710–718, Apr. 2024, doi: 10.57152/malcom.v4i2.1229.

[29] N. Fatimah Indrianti, A. Kania Ningsih, and R. Ilyas, "Implementasi Data Mining Untuk Klasifikasi Penyakit Gagal Ginjal Kronis Menggunakan Metode K-Nearest Neighbor," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 2, pp. 2255–2260, Apr. 2024, doi: 10.36040/jati.v8i2.9464.