MSTS Methods in Science and Technology Studies

futuristech PT. Teknologi Futuristik Indonesia

Date of publication January 20, 2026, date of current version January 20, 2026.
Digital Object Identifier 10.64539/msts.v1i2.2025.44

Article

Machine Learning Approach for Heart Failure Patient Classification Using K-Nearest Neighbors Algorithm

Alya Masitha[1], Syahrani Lonang[2,*], Julia Mega Reski[1]

[1] Department of Software Engineering, Institut Teknologi Statistika dan Bisnis Muhammadiyah, Semarang 50185, Indonesia; alyamasitha@gmail.com; julia.mega@itesa.ac.id
[2] Department of Information Technology, Universitas Qamarul Huda Badaruddin Bagu, Central Lombok 83371, Indonesia; lonangsyahrani3@gmail.com
* Correspondence

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Abstract: Heart failure is a cardiovascular disease with a high mortality rate and tends to increase every year. Therefore, a method is needed that can help the process of classifying heart failure quickly and accurately. This study aims to design and implement a heart failure classification system using the K-Nearest Neighbor (K-NN) machine learning method. The dataset used consists of 918 patient data with eleven input variables and two output classes, namely patients diagnosed with heart failure and patients not diagnosed with heart failure. The research stages include data loading, dividing training data and test data, implementing the K-NN algorithm with various K values, and evaluating model performance using accuracy, precision, recall, and F1-score metrics. The test results show that variations in the K value have a significant effect on the performance of the classification model. The K value = 9 produces the best performance with an accuracy of 93.48%, a recall of 96.36%, and an F1-score of 94.64%, which indicates a good balance between precision and recall. Based on these results, the K-NN method with a value of K = 9 is recommended as the optimal configuration in the classification of heart failure disease in this study.

Keywords: Heart Failure; Health; Machine Learning; K-Nearest Neighbors; Prediction.

1. Introduction

Based on the WHO definition referred to by the Indonesian Ministry of Health's Data Center (2014), cardiovascular disease is a group of diseases that occur due to disorders of the heart and blood vessels, which include coronary heart disease, hypertension, stroke, and heart failure. The results of the Basic Health Research (Riskesdas) show a continuous increase in the group of cardiovascular diseases, especially hypertension, which has experienced significant growth in prevalence from year to year. Throughout 2013 alone, it increased by 25.8%, and even in 2018 it increased by 34.1% [1]. Coronary heart disease remained at 1.5% in 2013 to 2018. Kidney failure disease increased from 0.2% in 2013 to 0.38% in 2018. Heart disease is one of the diseases that causes death of around 17.9 million people every year worldwide [2], [3]. Patients who have been diagnosed by a doctor as having heart failure usually die within 1-2 years [4], [5].

Classification of heart failure can be done by utilizing machine learning algorithms, which play a role in identifying patterns and determining classes based on available data [6]-[8]. Machine learning algorithms are computational methods designed to solve specific tasks while extracting latent information from available data [8]. The primary characteristic of machine learning lies in its ability to build algorithms that accept data input and utilize statistical analysis to generate predicted outputs. Generally, learning algorithms fall into two categories: supervised learning, unsupervised learning, and reinforcement learning [9].

Supervised learning is a category of supervised machine learning because the machine learning algorithm used can correct the results of its predictions based on the output [10], [11]. This category of machine learning already uses classes or labels in its dataset. In the supervised learning category, there are four models: regression, decision tree, classification, and random forest [12].

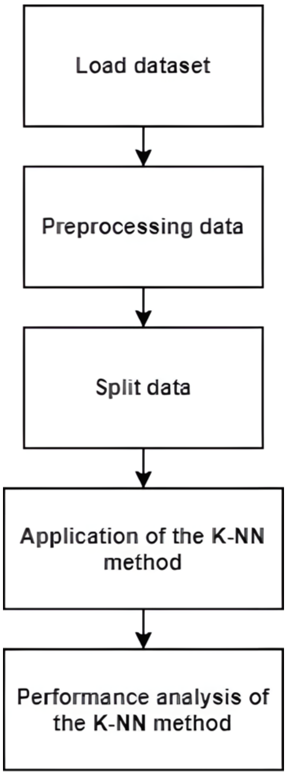

Figure 1. Research Stages.

The K-NN algorithm is a classification method that is widely applied in the field of machine learning because it has a simple concept, is easy to implement, and is flexible for development [13], [14]. This algorithm relies on the distance metric parameters and the k value to determine the proximity between data, with the main goal of finding the nearest neighbor of each data point in the dataset [15]. Many studies have been conducted using the K-NN algorithm, including one on predicting death from heart failure using the K-NN algorithm. This study used a k value of 7 with an accuracy of 94.92%. Testing using Python yielded an accuracy of 68% [15].

Another study applied K-NN algorithm to perform correlation analysis aimed at identifying the relationship between heart failure datasets and the developed prediction model [16], [17]. This study analyzed the performance of K-NN and obtained an accuracy value of 97.07 % in predicting heart failure [18]. Additionally, previous research using the K-NN algorithm for heart failure classification utilized 12 attributes from a total of 299 data sets. This study compared the use of a subset of 20 data sets with the full 299 sets, resulting in accuracy rates of 89.29% and 96.66%, respectively [19].

This study aims to design a heart failure disease classification system that can later be used to build a heart failure disease classification system using machine learning methods, one of which is the K-NN method. There are eleven input variables that can be seen in Figure 5. There are two outputs, namely: patients diagnosed with heart failure and patients not diagnosed with heart failure.

2. Research Methods

The dataset used in this study was obtained from the Kaggle platform and consists of 918 patient data which then went through a preprocessing stage to produce learning data suitable for predicting heart disease. The first attribute represents the patient's age with a range of 29 to 77 years, while the second attribute indicates gender, where a value of 0 represents female and a value of 1 represents male. The third attribute describes the type of chest pain experienced by the patient and is classified into four numeric categories, namely Typical Angina (TA = 0), Atypical Angina (ATA = 1), Non-Anginal Pain (NAP = 2), and Asymptomatic (ASY = 3). The fourth attribute represents resting blood pressure, the fifth attribute indicates cholesterol levels, and the sixth attribute indicates fasting blood sugar levels, with a value of 1 indicating blood sugar levels above 120 mg/dl and a value of 0 indicating the opposite condition. The seventh attribute reflects the results of an electrocardiogram (ECG) examination coded with values 0 to 2 to indicate different levels of the condition, the eighth attribute indicates the maximum heart rate achieved with a range of values between 71 and 202, and the ninth attribute indicates the presence of angina triggered by physical activity, where a value of 1 indicates the presence and a value of 0 indicates the absence. The tenth attribute describes the level of depression experienced by the patient, the eleventh attribute represents the ST segment slope at peak exercise conditions classified into upsloping, flat, and downsloping categories, while the last attribute serves as a class label that represents the target variable in the dataset. This study applies a binary classification approach, where a value of 0 indicates no indication of heart failure, while a value of 1 indicates a high probability of heart failure.

Figure 1 illustrates the research stages conducted in this study, which begins with the process of loading the dataset from the source that has been determined as the research object. The obtained dataset is then prepared to ensure its suitability in the next analysis stage. The next stage is data preprocessing aimed at improving data quality, including the process of cleaning for missing values, duplicate data, and inconsistencies, followed by data transformation if necessary to suit the needs of the K-Nearest Neighbors (K-NN) algorithm. After that, the data is divided into training data and test data, where the training data is used to build the K-NN model, while the test data is used to test the model's ability to classify data that has not been studied before. In the next stage, the K-NN method is implemented by determining the appropriate parameters, such as the k value and distance calculation method, to carry out the classification process on the test data. The final stage of the research is the analysis of the K-NN model performance, which is carried out using evaluation metrics such as accuracy, precision, recall, and f-measure to assess the level of accuracy and

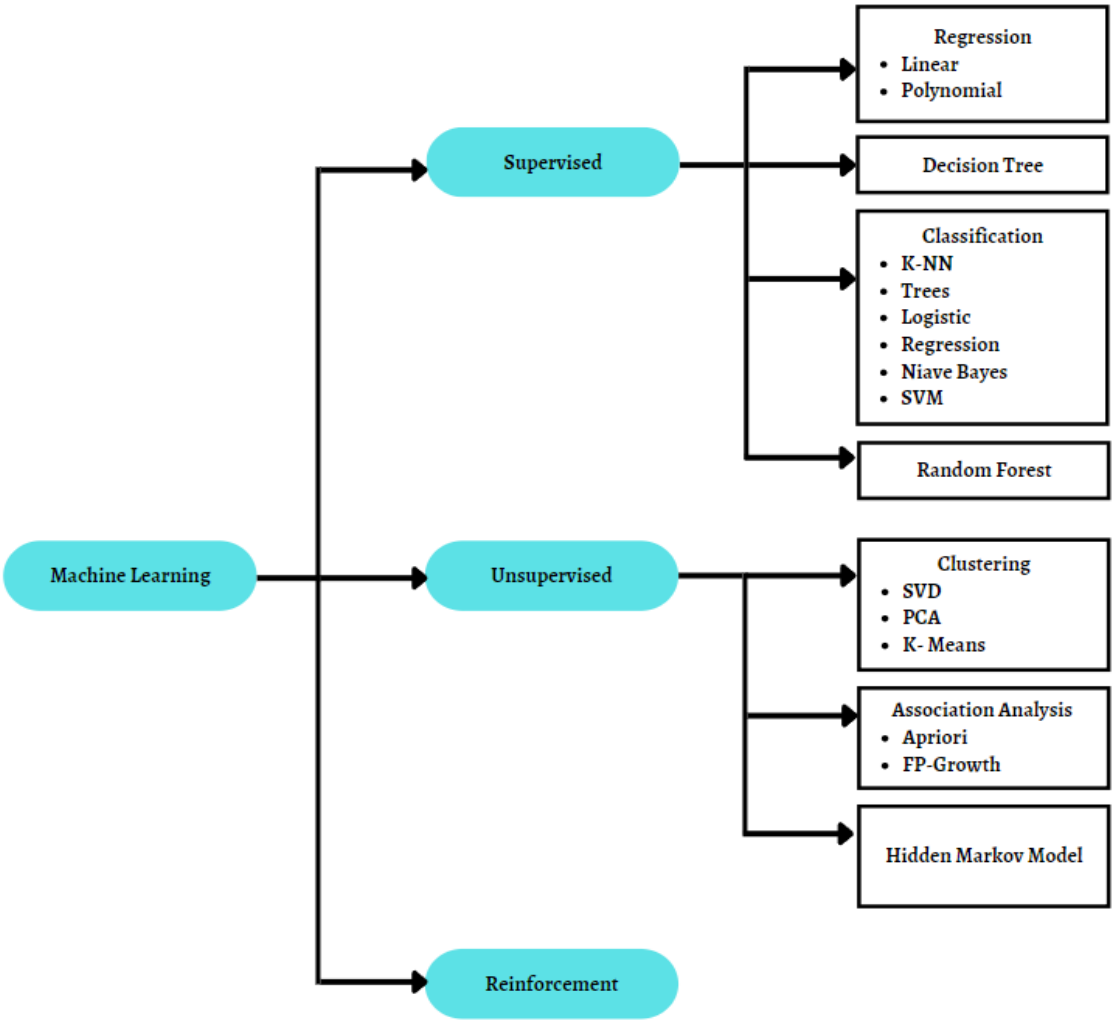

Figure 2. Categories of Machine Learning Techniques.

effectiveness of the K-NN method in solving the problems studied.

2.3. Machine Learning

Machine learning is a collection of statistical algorithms designed to solve specific tasks and extract hidden patterns or information from available data. In general, the main characteristic of machine learning lies in its ability to build computational models that accept data as input and utilize various statistical approaches to produce specific predictions or outputs. Based on their learning approach, machine learning algorithms can be classified into three main categories: supervised learning, unsupervised learning, and reinforcement learning [9].

Supervised learning algorithms use pairs of input and output data as the basis for the model training process, and evaluate model performance using various statistical measures [9]. This approach integrates historical knowledge with new data through labeled examples to generate predictions on unknown data. Based on the analysis of identified training data, the learning algorithm constructs an inference function that is used to predict specific output values. Furthermore, an evaluation mechanism allows for comparison between predicted results and actual values, thus identifying errors and using them as a basis for continuous model improvement and optimization [20].

In Figure 2, the categories of machine learning techniques, K-NN is one of the algorithms included in the supervised learning category which is used for the classification process.

2.4 K-NN

The K-NN algorithm is a classification method in supervised learning that works using labeled datasets. This algorithm determines the class of a test dataset by measuring the similarity between its features and features from training datasets that already have class labels [21]. K-NN is widely used in various classification problems because it has a simple and flexible concept to be implemented and developed further. The working principle of the K-NN algorithm is based on the use of distance metrics and the determination of the number of nearest neighbors represented by the parameter k. The main goal of this algorithm is to identify a number of data closest to a target data in the feature space. The classification process is carried out by assigning a class to a sample based on the most dominant class among its k

```
import numpy as np
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix, classification_report
```

Figure 3. Import Library.

Table 1. Results of Transformation data.

	Age	Sex	Chest Pain Type	Resting BP	Cholesterol	Fasting BS	Resting ECG	Max HR	Exercise Angina	Old peak	ST Slope	Heart Disease
0	40	1	1	140	289	0	1	172	0	0.0	2	0
1	49	0	2	160	180	0	1	156	0	1.0	1	1
2	37	1	1	130	283	0	2	98	0	0.0	2	0
3	48	0	0	138	214	0	1	108	1	1.5	1	1
4	54	1	2	150	195	0	1	122	0	0.0	2	0

```
i = 2 #increment
accuracy_rate = []
for x in range (1, 10, i):
    KNN_model = KNeighborsClassifier(n_neighbors=x,metric='euclidean')
    KNN_model.fit(X_train, y_train)
    y_pred = KNN_model.predict(X_test)
    acc_rate = print('K',x,'=',round(accuracy_score(y_test, y_pred)*100 , 2))
    accuracy_rate.append(acc_rate)
```

Figure 4. implementation of the K-NN method.

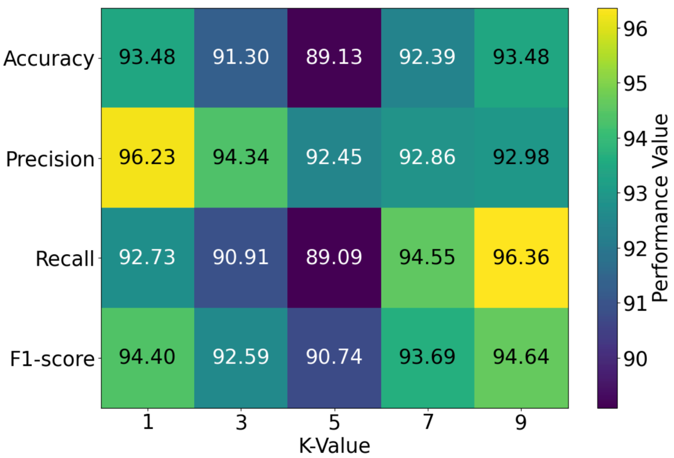

Figure 5. K-Nearest Neighbor method test results.

nearest neighbors. By utilizing labeled datasets, K-NN groups data into certain classes to predict the class of new, unknown data. Operationally, the K-NN algorithm calculates the distance between the test data and all training data, then selects the k data points with the shortest distance as the nearest neighbors. The final result is determined based on the class that appears most frequently in those neighbors for classification cases, or the average value of the nearest neighbors for regression cases [15]. In the process of measuring distance, K-NN can use various distance functions, where Euclidean distance is one of the most commonly used metrics. The mathematical formulation of Euclidean distance is shown in Equation 1 [22]:

$$\begin{aligned} &\mathit{Euclidean\ Distance} \\ &= \sqrt{(a_1 - b_1) + (a_2 - b_2) + \cdots (a_n - b_n)^2} \\ &= \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \end{aligned} \tag{1}$$

Where Euclidean distance is a method of measuring distance used for calculations of real number type.

2.5 Confusion Matrix

Measuring the performance of a classification is important because it can determine the quality of a model. This can be measured using the confusion matrix method. There are four parameters used to represent classification results: True Positive (TP), which is data predicted to be correct and is correct; True Negative (TF), which is data predicted to be negative and is correct; and False Positive (FP), which is data predicted to be correct and is incorrect. False Negative (FN) is the opposite of TP [23].

Four evaluation metrics were used in this study: accuracy, precision, recall, and F1-Score. Accuracy represents the percentage of correctly classified data, representing the ratio of the number of correct predictions to the total test data. Precision measures the accuracy of predictions in the positive class, representing the ratio of the number of true positives (TP) to the total number of positive predictions, consisting of true positives (TP) and false positives (FP). Meanwhile, recall measures the model's ability to identify all data that truly fall into the positive class, calculated as the ratio of true positives (TP) to the number of true positives (TP) and false negatives

(FN). F1-Score, also known as F-measure, is used to assess the balance between precision and recall, where the F1-Score is obtained by multiplying the precision and recall by the sum of the two [24].

3. Results and Discussion

3.1. Load Data

This research focuses on developing a machine learning-based classification model using the K-Nearest Neighbor (K-NN) method to predict heart failure. The dataset used in this study is a heart failure dataset obtained from the Kaggle platform, consisting of 918 patient data. The dataset includes two classes: patients with indications of heart failure and patients without indications of heart failure, with the heart disease class label representing the patient's cardiovascular condition. A total of 12 attributes are used as parameters in the classification process, reflecting the patient's clinical and demographic characteristics. The implementation of the K-NN algorithm is carried out through data preprocessing stages to ensure dataset quality and consistency, followed by dividing the data into training data and test data to build and evaluate the predictive model. The K-NN model is then used to classify the test data based on the proximity of the data in the feature space. Model performance is evaluated using accuracy, precision, recall, and f-measure metrics to assess the effectiveness of the K-NN method in producing accurate and reliable predictions in the case of heart failure classification. This research uses the Python programming language. This research begins by importing the libraries to be used. A snippet of the program code can be seen in Figure 3.

Figure 3 illustrates the initial stages of implementing the machine learning method using the K-Nearest Neighbors (K-NN) algorithm in this study. The process begins with calling various libraries that support data analysis and classification model development, including NumPy and Pandas, which are used for processing and manipulating numerical and tabular data. The KNeighborsClassifier module from the scikit-learn library is utilized as the core of the K-NN algorithm implementation, while train_test_split and cross_val_score are used for data splitting and cross-validation to evaluate model performance more objectively. In the data preprocessing stage, LabelEncoder is applied to convert categorical attributes into numerical form for processing by the K-NN algorithm. In addition, the Matplotlib and Seaborn libraries are used to support data visualization and presentation of model evaluation results. Model performance evaluation is conducted using several metrics available in scikit-learn, including accuracy, precision, recall, F1-score, confusion matrix, and classification report, which aim to comprehensively assess the model's ability to classify heart failure patient data. The final stage is marked by loading the heart failure dataset using the read_csv function, which contains patient clinical data with the heart disease label as the target variable, and is then used as a basis for the preprocessing, training, and testing of the K-NN model to build an accurate heart failure prediction system.

3.2. Data Transformation

Data preprocessing is performed before applying the K-Nearest Neighbors (K-NN) algorithm. At this stage, categorical attributes are transformed into numeric form using the Label Encoding method, considering that the K-NN algorithm can only process data in numeric format. Some of the categorical attributes encoded include the variables Sex, ChestPainType, RestingECG, ExerciseAngina, and ST_Slope, which represent patient clinical characteristics related to cardiovascular conditions. The encoding process is carried out by initializing a LabelEncoder object and applying the fit_transform function to map each category into a discrete numeric value. This transformation aims to ensure that all features in the dataset are in a format that can be processed computationally by the K-NN model without losing any categorical information contained therein. After the encoding process is complete, the data.head() function is used to display the first few rows of the dataset as a verification step against the results of the data transformation.

Table 1 displays the results of data transformation after going through the preprocessing stage, specifically the label encoding process for categorical attributes in the heart failure dataset. All attributes in the dataset are now in numeric format, ready for use in the modeling stage using the K-Nearest Neighbor (K-NN) algorithm. Continuous numeric attributes such as Age, RestingBP, Cholesterol, MaxHR, and Oldpeak are retained in their original values, while categorical attributes such as Sex, ChestPainType, RestingECG, ExerciseAngina, and ST_Slope have been successfully converted to discrete numeric values. Furthermore, the HeartDisease attribute is used as the target variable (class label) representing the patient's condition, where a value of 1 indicates heart disease, while a value of 0 indicates the opposite. This transformation ensures there are no non-numeric values in the dataset, allowing for optimal distance calculations in the K-NN algorithm. Thus, the transformed dataset is ready for use in the training and test data division stages, as well as in the development and evaluation of heart failure classification models.

3.3. Splitting data

Data splitting is the process of dividing a dataset into three main parts: a training set, a validation set, and a testing set. This is done to ensure that the machine learning model is properly learned. The data split stage involves dividing the dataset into training and testing sets before applying the K-Nearest Neighbor (K-NN) algorithm. Data splitting is performed using the train_test_split function from the scikit-learn library, with 90% of the data used as training data and 10% as testing data (test_size = 0.1). This proportion is chosen to provide sufficient data for the model learning process while retaining a representative portion of the data to test the model's generalization ability on data that has not been previously trained. The random_state = 1 parameter is used to ensure that the data splitting process is consistent and can be replicated in subsequent experiments. The variable X represents a set of independent attributes or features, while y is the target variable that indicates the class of heart failure conditions. The splitting results in four data subsets: X_train and y_train as training data, and X_test and y_test as test data. The X_train.head() command is used to display the first few rows of training data as a verification step for the results of data division.

Figure 4 is the program code used to test the performance of the K-NN algorithm with varying values of the number of neighbors (K) and see its effect on the level of model accuracy. At the beginning of the program, the variable i is set to 2 which functions as the interval for increasing the value of K, so that the K values tested are 1, 3, 5, 7, and 9. In addition, an empty list accuracy_rate is created which aims to store the accuracy value of each trial value of K. Next, the program loops using for x in range(1, 10, i) to test several different values of K. At each iteration, a KNN model is created with the parameter n_neighbors valued at x and uses the Euclidean distance metric to measure the closeness between the data. The model is then trained using the training data (X_train and y_train) and used to predict labels on the test data (X_test). The prediction results are stored in the variable y_pred. After the prediction process, the model accuracy is calculated by comparing the predicted results (y_pred) and the actual labels (y_test) using the accuracy_score function. The accuracy values are then converted to percentages and rounded to two decimal places before being displayed for each tested K value. However, the printed accuracy values are not actually stored in the accuracy_rate list because the print() function does not return a value. Overall, this code aims to evaluate the performance of the KNN model based on varying K values to determine the number of neighbors that produces the best accuracy.

3.5. K-NN performance analysis

This section discusses the performance analysis of the K-NN method based on the variations in the tested K value. Performance evaluation was conducted using several metrics, namely accuracy, precision, recall, and F1-score, to determine the effect of changes in the K value on the model's classification ability. Detailed testing results for the K-NN method are presented in Figure 5.

Figure 5 shows the results of testing the K-NN method. It can be seen that variations in the K value affect model performance. At K = 1, the model produces an accuracy of 93.48%, a precision of 96.23%, a recall of 92.73%, and an F1-score of 94.44%, indicating excellent performance, especially in terms of precision. However, using a K value that is too small can potentially make the model sensitive to noise. At K = 3, all metrics decreased, with an accuracy of 91.3% and an F1-score of 92.59%, indicating that increasing the number of neighbors did not provide an increase in performance. The most significant decrease occurred at K = 5, where the accuracy, recall, and F1-score were 89.13%, 89.09%, and 90.74%, respectively, so this K value can be said to be less than optimal for the data used. Furthermore, at K = 7, the model performance increased again with an accuracy of 92.39% and an F1-score of 93.69%, indicating the model was starting to become more stable. The best overall performance was obtained at K = 9, with an accuracy of 93.48%, a recall of 96.36%, and an F1-score of 94.64%, indicating a good balance between precision and recall. Therefore, it can be concluded that K = 9 is the most optimal K value in this test because it provides the most balanced and stable performance in the KNN method.

4. Conclusion

Based on the test results, it can be concluded that the K-NN method is able to provide good classification performance on the data used in this study. The variation in the applied K value shows a significant effect on model performance, as seen from changes in accuracy, precision, recall, and F1-score values in each test scenario. A K value that is too small tends to produce high precision but has the potential to make the model sensitive to noise, while a larger K value can improve model stability and generalization ability. The evaluation results show that a value of K = 9 produces the best overall performance with an accuracy of 93.48%, a recall of 96.36%, and an F1-score of 94.64%. This value shows an optimal balance between precision and recall, so that the model is not only able to minimize prediction errors but is also effective in identifying all data in the positive class. Therefore, a value of K = 9 can be recommended as the optimal parameter in the application of the K-NN method in this study. For further development, future research can consider using cross-validation techniques, distance weighting, or combining the K-NN method with other algorithms to improve model accuracy and robustness.

5. Declarations

5.1. Author Contributions

Alya Masitha: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Writing - Original Draft; **Syahrani Lonang:** Formal analysis, Investigation, Conceptualization, Methodology; **Julia Mega Reski:** Writing - Original Draft, Project administration, Visualization, Resources.

5.2. Institutional Review Board Statement

Not applicable.

5.3. Informed Consent Statement

Not applicable.

5.4. Data Availability Statement

The dataset analyzed in this study is publicly available on Kaggle and can be accessed through the dataset link provided in the references section of this article.

5.5. Acknowledgment

Not applicable.

5.6. Conflicts of Interest

The authors declare no conflicts of interest.

6. References

[1] M. A. L. Suratri, "Pengaruh Hipertensi Terhadap Kejadian Penyakit Jaringan Periodontal (Periodontitis) pada Masyarakat Indonesia (Data Riskesdas 2018)," *Bul. Penelit. Kesehat.*, vol. 48, no. 4, pp. 227–234, 2020, https://doi.org/10.22435/bpk.v48i4.3516.

[2] P. Ghosh et al., "Efficient Prediction of Cardiovascular Disease Using Machine Learning Algorithms With Relief and LASSO Feature Selection Techniques," *IEEE Access*, vol. 9, pp. 19304–19326, 2021, https://doi.org/10.1109/ACCESS.2021.3053759.

[3] I. Dilalah, H. Christiandari, J. Y. Hernawan, dan E. Suprasetya, "Optimizing Cardiovascular Disease Medication Awareness: A Community Engagement Initiative at Posbindu Kenanga 3 Manggulan", *Jurnal Pengabdian Masyarakat Permata Indonesia*, vol. 3, no. 2, pp. 75–81, 2023, https://jurnal.permataindonesia.ac.id/index.php/JPMPI/article/view/243.

[4] A. U. Haq, J. P. Li, M. H. Memon, S. Nazir, R. Sun, and I. Garciá-Magarinõ, "A Hybrid Intelligent System Framework for the Prediction of Heart Disease using Machine Learning Algorithms," *Mob. Inf. Syst.*, vol. 2018, p. 3860146, 2018, https://doi.org/10.1155/2018/3860146.

[5] C. J. Taylor, R. Ryan, L. Nichols, N. Gale, F. D. Richard Hobbs, and T. Marshall, "Survival following a diagnosis of heart failure in primary care," *Fam. Pract.*, vol. 34, no. 2, pp. 161–168, 2017, https://doi.org/10.1093/fampra/cmw145.

[6] B. Chulde-Fernández et al., "Classification of Heart Failure Using Machine Learning: A Comparative Study," *Life*, vol. 15, no. 3, pp. 1–18, 2025, https://doi.org/10.3390/life15030496.

[7] L. N. Farida and S. Bahri, "Klasifikasi Gagal Jantung menggunakan Metode SVM (Support Vector Machine)," *Komputika J. Sist. Komput.*, vol. 13, no. 2, pp. 149–156, 2024, https://doi.org/10.34010/komputika.v13i2.11330.

[8] R. S. Nurhalizah, R. Ardianto, and P. Purwono, "Analisis Supervised dan Unsupervised Learning pada Machine Learning: Systematic Literature Review," *J. Ilmu Komput. dan Inform.*, vol. 4, no. 1, pp. 61–72, 2024, https://doi.org/10.54082/jiki.168.

[9] S. L. Karri, L. C. De Silva, D. T. C. Lai, and S. Y. Yong, "Classification and Prediction of Driving Behaviour at a Traffic Intersection Using SVM and KNN," *SN Comput. Sci.*, vol. 2, no. 3, 2021, https://doi.org/10.1007/s42979-021-00588-7.

[10] S. Tiwari, "Supervised Machine Learning: A Brief Introduction," *Proc. Int. Conf. Virtual Learn.*, vol. 17, no. 5, pp. 219–230, 2022, https://doi.org/10.58503/icvl-v17y202218.

[11] L. Puig, "On the reduction of Alperin's Conjecture to the quasi-simple groups," *J. Algebr.*, vol. 328, no. 1, pp. 372–398, 2011, https://doi.org/10.1016/j.jalgebra.2010.11.004.

[12] P. Singh, *Learn PySpark: Build Python-based machine learning and deep learning models.* Apress, Berkeley, CA, pp. 1-210, 2019, https://books.google.co.id/books?id=3-GtDwAAQBAJ.

[13] R. S. Daulay, "Analisis Kritis dan Pengembangan Algoritma K-Nearest Neighbor (KNN):

Sebuah Tinjauan Literatur," *J. Pendidik. Sains dan Komput.*, vol. 4, no. 02, pp. 131–141, 2024, https://doi.org/10.47709/jpsk.v4i02.5055.

[14] N. T. Ujianto, Gunawan, H. Fadillah, A. P. Fanti, A. D. Saputra, and I. G. Ramadhan, "Penerapan algoritma K-Nearest Neighbors (KNN) untuk klasifikasi citra medis," *IT-Explore J. Penerapan Teknol. Inf. dan Komun.*, vol. 4, no. 1, pp. 33–43, 2025, https://doi.org/10.24246/itexplore.v4i1.2025.pp33-43.

[15] A. Upadhyay, S. Nadar, and R. Jadhav, "Comparative Study of SVM & KNN for Signature Verification," *J. Stat. Manag. Syst.*, vol. 23, no. 2, pp. 191–198, 2020, https://doi.org/10.1080/09720510.2020.1724619.

[16] V. S. Souza and D. A. Lima, "Cardiac Disease Diagnosis Using K-Nearest Neighbor Algorithm: A Study on Heart Failure Clinical Records Dataset," *Artif. Intell. Appl.*, vol. 3, no. 1, pp. 56–71, 2025, https://doi.org/10.47852/bonviewAIA42022045.

[17] K. L. Kohsasih, D. S. Sunario, A. Alvin and F. Laurendio, "Enhancing Early Heart Disease Detection Through Comparative Analysis of Random Forest , Decision Tree , and K-NN Models," *IT J. Res. Dev.*, vol. 10, no. 2, pp. 66–77, 2025. https://doi.org/10.25299/itjrd.2025.24703.

[18] T. A. Assegie, S. J. Sushma, B. G. Bhavya, and S. Padmashree, "Correlation Analysis for Determining Effective Data in Machine Learning: Detection of Heart Failure," *SN Comput. Sci.*, vol. 2, no. 3, pp. 1–5, 2021, https://doi.org/10.1007/s42979-021-00617-5.

[19] R. Yunus, U. Ulfa, and M. D. Safitri, "Application of the K-Nearest Neighbors (K-NN) Algorithm for Classification of Heart Failure," *J. Appl. Intell. Syst.*, vol. 6, no. 1, pp. 1–9, 2021, https://doi.org/10.33633/jais.v6i1.4513.

[20] A. D. Kumari, J. P. Kumar, V. S. Prakash, and K. S. Divya, "Supervised Learning Algorithms : A Comparison," *Kristu Jayanti J. of Comput. Sci.*, vol. 1, no. 1, pp. 1–12, 2020. https://doi.org/10.59176/kjcs.v1i1.1259.

[21] U. S. Reddy, A. V. Thota, and A. Dharun, "Machine Learning Techniques for Stress Prediction in Working Employees," *2018 IEEE Int. Conf. Comput. Intell. Comput. Res. ICCIC 2018*, 2018, https://doi.org/10.1109/ICCIC.2018.8782395.

[22] Z. Mushtaq, A. Yaqub, S. Sani, and A. Khalid, "Effective K-Nearest Neighbor Classifications for Wisconsin Breast Cancer Data Sets," *J. Chinese Inst. Eng.*, vol. 43, no. 1, pp. 80–92, 2020, https://doi.org/10.1080/02533839.2019.1676658.

[23] N. Hadianto, H. B. Novitasari, and A. Rahmawati, "Klasifikasi Peminjaman Nasabah Bank Menggunakan Metode Neural Network," *J. Pilar Nusa Mandiri*, vol. 15, no. 2, pp. 163–170, 2019, https://doi.org/10.33480/pilar.v15i2.658.

[24] S. Ketu and P. K. Mishra, "Scalable kernel-based SVM classification algorithm on imbalance air quality data for proficient healthcare," *Complex Intell. Syst.*, vol. 7, no. 5, pp. 2597–2615, 2021, https://doi.org/10.1007/s40747-021-00435-5.