SITEKNIK
Sistem Informasi, Teknik dan Teknologi Terapan
E-ISSN: 3032-3991
Vol. 2. No. 2 April 2025. Pages. 88-97

Performance Analysis of Support Vector Machine and Gradient Boosting Machine Algorithms for Heart Disease Prediction

Tegar Wirawan[1], Kusnawi[2]

[1,2] Informatics, Faculty of Computer Science, AMIKOM University Yogyakarta, Indonesia

Keywords: Vector,Machine, Gradient,Boosting, Heart

Submitted: 15/02/2025

Revised: 25/02/2025

Accepted: 20/03/2025

Abstrak

Cardiovascular disease ranks among the primary causes of mortality globally, with death rates rising each year. Assessing heart disease risk is crucial for enhancing the efficiency of prevention and treatment strategies. This study seeks to evaluate the effectiveness of two machine learning techniques, namely Support Vector Machine and Gradient Boosting Machine, in forecasting heart disease using a dataset obtained from Kaggle. The research process starts with gathering data, followed by exploratory analysis, preprocessing through label encoding, handling class imbalance with SMOTE, and normalizing data using Standard Scaler. Features were selected using the Correlation Thresholding method. Subsequently, the dataset was divided into training and testing sets to develop predictive models. The model performance was assessed using evaluation metrics, including accuracy, precision, recall, and F1-Score. The findings indicate that the *Gradient Boosting Machine* outperformed the Support Vector Machine, achieving an accuracy of 98% compared to SVM's accuracy of 93%. This research is expected to contribute to healthcare practices by enabling early detection of heart disease risks. Future research is recommended to explore other algorithms or employ more diverse datasets to achieve better results.

Corresponding Author:
Tegar Wirawan
Informatics Study Program, Faculty of Computer Science, Amikom University Yogyakarta
Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta Telp: (0274) 884201 – 207
Email: tegarwirawan@students.amikom.ac.id, khusnawi@amikom.ac.id

INTRODUCTION

The rapid development of information and communication technology has transformed many aspects of life, including the medical field. A major advancement is the use of technology to collect and analyze data to support better decision-making (Huda 2020). Information technology plays a crucial role in predicting health risks and enhancing efforts for disease prevention and treatment. For example, heart disease

prediction, influenced by clinical factors, can now be analyzed using a data-driven approach (Munawar 2021).

Heart disease is one of the deadliest diseases in the world. According to reports from various global health organizations such as WHO, heart disease causes approximately 17.9 million deaths annually, and this number is projected to increase to 23 million by 2030(Yudi Her Oktaviono 2024). Factors such as diabetes, hypertension, and high cholesterol significantly contribute to the development of heart disease risk. Therefore, it is essential to develop predictive models that can identify high-risk individuals, enabling early intervention (Amelia 2017).

In recent years, machine learning algorithms have become increasingly popular in medical data analysis, particularly for predictive tasks (Parikesit Dito; Putranto Arli Aditya; Anurogo 2018). Algorithms like Support Vector Machine are widely used in the medical field. SVM has the advantage of constructing an optimal hyperplane to maximize the margin between two different classes, making it effective for classifying complex datasets. However, SVM faces challenges in handling large datasets due to its computational complexity (Zhang 2001).

On the other hand, the Gradient Boosting Machine is an algorithm that builds predictive models gradually, where each model attempts to correct the errors of the previous one. GBM's main advantage lies in its ability to handle various types of data, especially in classification cases. However, GBM also has a drawback, as it requires higher computational time compared to simpler algorithms. GBM has been widely used for predicting other medical cases, such as diabetes and cancer (Corey Wade 2020).

A better understanding of heart disease factors from both clinical and lifestyle perspectives can provide benefits in improving public health and reducing heart disease mortality rates (Erdania, Faizal, and Anggraini 2023). This study utilizes the Support Vector Machine and Gradient Boosting Machine algorithms to analyze heart disease and provide data-driven recommendations for more effective and targeted prevention efforts.

Based on the explanation above, this study is titled "Performance Analysis of Support Vector Machine and Gradient Boosting Machine Algorithms for Heart Disease Prediction." This research is expected to contribute to understanding heart disease factors and strengthen prevention efforts through a data-driven approach.

RESEARCH METHODS

Support Vector Machine is a classification algorithm that is often used to separate two classes by finding the optimal hyperplane to maximize the margin between the two classes (Ingo Steinwart 2008).

Classification Function:

$$f(x) = sign\left(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b\right) \quad (1)$$

Where:

$f(x)$ is the predicted output.

α_i is the multiplier of the optimal solution.

y_i is the class label.

$K(x_i, x)$ is a kernel function for non-linear data.

b is the bias.

Gradient Boosting Machine is a machine learning algorithm that is often used for classification and regression cases. In the context of classification, the Gradient Boosting Machine algorithm is frequently utilized. Gradient Boosting Machine works by building models iteratively, where each new model improves the errors of the previous model (Corey Wade 2020).

With the formula:

$$F(x) = \sum_{m=1}^{m} h_m(x) \quad (2)$$

Where:

$F(x)$ represents the final prediction.

M is the number of trees.

$h_m(x)$ is the prediction from the m-th tree.

In this study, the research workflow is as follows:

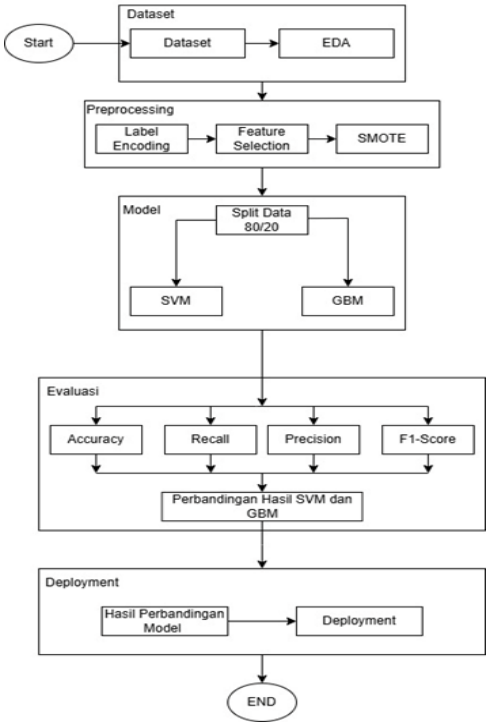

Figure 1. Research Flow

In this study, several research flow stages were carried out, namely:

Dataset Collection: The dataset was obtained from Kaggle, named the Heart Disease Dataset UCI.

Exploratory Data Analysis (EDA): In this stage, several EDA processes were conducted to examine general information about the Heart Disease Dataset UCI and perform data visualization to understand the dataset's contents.

Preprocessing Data: This stage involves processing the data before feeding it into the model by applying encoding to non-numeric columns so that the model can process the data. Once all data has been converted into numerical format, SMOTE is applied to

balance the classes (0 and 1), followed by the application of Standard Scaler to the selected features obtained through the Correlation Thresholding method.

Data Splitting: After preprocessing and feature selection, the next step is to split the data into training and testing sets with an 80/20 ratio and a random state of 42.

Model Development: Once the data is split, models are built. This study utilizes two machine learning models, namely Support Vector Machine and Gradient Boosting Machine, to predict heart disease.

Evaluation: After successfully creating the SVM and GBM models, an evaluation is conducted by measuring accuracy, precision, recall, and F1-score to compare the performance of both models.

Deployment: Once the desired evaluation results are achieved, the final step is deployment, where a web-based application is developed to implement the research findings.

RESULTS AND DISCUSSION

This study aims to compare the performance of Support Vector Machine and Gradient Boosting Machine in predicting heart disease using a dataset from Kaggle. The steps involved include data preprocessing with label encoding for categorical variables, SMOTE to handle class imbalance, and Correlation Thresholding for feature selection. The models are then trained and tested with an 80:20 data split and evaluated using accuracy, precision, recall, and F1-score.

At the initial stage of this study, the dataset was obtained from Kaggle, an open platform that provides various types of datasets. This research utilizes the dataset titled "Heart Disease Dataset UCI", uploaded by Ketan Gangal in 2022. The dataset contains 1,026 samples and 14 columns, stored in a CSV file.

Table 1. Data Collection

Code	Function
df = pd.read_csv('/content/drive/MyDrive/SKRIPSI /Untitled folder/HeartDiseaseTrain-Test.csv') df.head()	Read a CSV file named HeartDiseaseTrain-Test located in the Untitled folder within the SKRIPSI directory and convert it into a DataFrame. Store the DataFrame in the variable df.
df.head()	Display the first 5 rows of the DataFrame.

	age	sex	chest_pain_type	resting_blood_pressure	cholestoral	fasting_blood_sugar	rest_ecg	Max_heart_rate	exercise_induced_angina	oldpeak	slope	vessels_colored_by_flourosopy	thalassemia	target
0	52	Male	Typical angina	125	212	Lower than 120 mg/ml	ST-T wave abnormality	168	No	1.0	Downsloping	Two	Reversable Defect	0
1	53	Male	Typical angina	140	203	Greater than 120 mg/ml	Normal	155	Yes	3.1	Upsloping	Zero	Reversable Defect	0
2	70	Male	Typical angina	145	174	Lower than 120 mg/ml	ST-T wave abnormality	125	Yes	2.6	Upsloping	Zero	Reversable Defect	0
3	61	Male	Typical angina	148	203	Lower than 120 mg/ml	ST-T wave abnormality	161	No	0.0	Downsloping	One	Reversable Defect	0
4	62	Female	Typical angina	138	294	Greater than 120 mg/ml	ST-T wave abnormality	106	No	1.9	Flat	Three	Fixed Defect	0

Figure 2. Import Data

Figure 2 shows the result of the data import process, where the dataset was originally in a CSV file format. After being imported into Google Colab, it was converted into a DataFrame and displays the top five rows of the dataset to provide an initial understanding of the number of columns and the data types of each column in the Heart Disease Dataset UCI.

Table 2 Label Encoding

Code	Function

label_encoders = {}
columns_to_encode = df.select_dtypes(include=['object', 'float64']).columns
for col in columns_to_encode:
if df[col].dtype == 'object':
# Melakukan Label encoding untuk kolom bertipe object 	
le = LabelEncoder()
df[col] = le.fit_transform(df[col])
label_encoders[col] = le
elif df[col].dtype == 'float64':
# Mengubah tipe data float menjadi integer 	
df[col] = df[col].astype(int)
	To convert the data type from object to numeric using label encoders and convert the data type from float to integer

df.head()	then display the first five rows after applying the label encoder.

	age	sex	chest_pain_type	resting_blood_pressure	cholestoral	fasting_blood_sugar	rest_ecg	Max_heart_rate	exercise_induced_angina	oldpeak	slope	vessels_colored_by_flourosopy	thalassemia	target
0	52	1	3	125	212	1	2	168	0	1	0	3	3	0
1	53	1	3	140	203	0	1	155	1	3	2	4	3	0
2	70	1	3	145	174	1	2	125	1	2	2	4	3	0
3	61	1	3	148	203	1	2	161	0	0	0	1	3	0
4	62	0	3	138	294	0	2	106	0	1	1	2	0	0

Figure 3. Label Encoding

Figure 3 is the result of the Label Encoding process in the initial preprocessing stage. Label Encoding is used to convert previously non-numeric columns into numeric data types so that they can be processed by both algorithms, namely Support Vector Machine and Gradient Boosting Machine.

Table 3. Correlation Thresholding

Code	Function
correlation_threshold = 0.1 features = correlation_matrix['target'][abs(correlation_matrix['target']) > correlation_threshold].index.tolist() features.remove('target')	Perform feature selection to choose features that have a strong linear relationship between variables and the target.
print(f"Selected Features: {features}")	Display the results of the feature selection.

Table 4. Correlation Thresholding Results

Selected Features: ['age', 'sex', 'chest_pain_type', 'resting_blood_pressure', 'rest_ecg', 'Max_heart_rate', 'exercise_induced_angina', 'oldpeak', 'slope', 'vessels_colored_by_flourosopy', 'thalassemia']

Table 4 presents the results after performing Correlation Thresholding, displaying features that have a strong and significant linear relationship between the variables in the dataset and the predetermined target. These selected features will then be used for model development.

Table 5. Smote

Code	Function
smote = *SMOTE*(random_state=42) X_train_resampled, y_train_resampled = *smote*.fit_resample(X_scaled,y)	Generate synthetic data to balance class distribution
plt.figure(figsize=(6, 6)) sns.set(font_scale=1.2) sns.countplot(x=pd.Series(y_train_resampled)) plt.title('Setelah *SMOTE*') plt.show()	Display the results after applying SMOTE using a bar chart

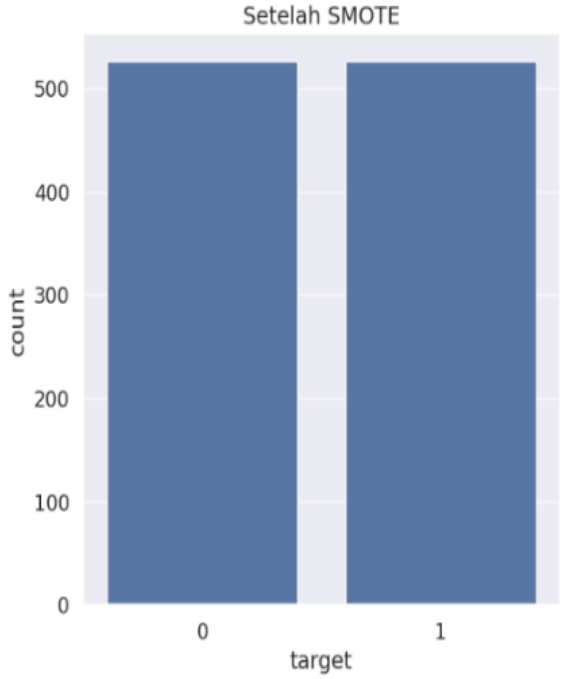

Figure 4. SMOTE

Figure 4 represents the result of the SMOTE (Synthetic Minority Oversampling Technique) process to address class imbalance between class 0 and class 1 in the dataset. In the two bar charts above, it can be seen that before applying SMOTE, class 1 had a data distribution of 526, while class 0 had only 499. In this study, SMOTE was applied to balance the class distribution in the dataset.

Table 6. Evaluation of Support Vector Machine

Code	Function
y_pred_svm = svm_model.predict(X_test)	Make predictions using the test data (X_test)

# Evaluasi model 	
print(classification_report(y_test, y_pred_svm))
print("Accuracy:", accuracy_score(y_test, y_pred_svm))
	Display the classification report results from the Support Vector Machine algorithm

```
              precision    recall  f1-score   support

           0       0.92      0.94      0.93        94
           1       0.95      0.93      0.94       117

    accuracy                           0.93       211
   macro avg       0.93      0.93      0.93       211
weighted avg       0.93      0.93      0.93       211

Accuracy: 0.933649289099526
```

Figure 5. Evaluation of Support Vector Machine

Figure 5 represents the evaluation results after the Support Vector Machine model was successfully built. The evaluation results of the Support Vector Machine model are as follows:

The precision evaluation in this study shows that class 0 has a precision of 0.92, meaning that 92% of class 0 predictions are correct, while class 1 has a precision of 0.95, meaning that 95% of class 1 predictions are correct.

The recall evaluation results indicate that class 0 has a recall of 0.94, meaning that 94% of actual class 0 data was correctly identified by the model. Meanwhile, class 1 has a recall of 0.93, meaning that 93% of actual class 1 data was correctly identified by the model.

The F1-score, which is a combination of precision and recall, shows that class 0 achieved 93%, while class 1 performed better with 94%.

The accuracy of the Support Vector Machine algorithm in this study reached 0.93, meaning that 93% of the model's total predictions were correct out of the total dataset.

Label encoding is a machine learning method used to convert text or non-numeric data into numeric data, allowing algorithms like Support Vector Machine and Gradient Boosting Machine to process input data efficiently.

Table 7. Evaluation of Gradient Boosting Machine

Code	Function
y_pred_gb = gb_model.predict(X_test)	Make predictions using the test data (X_test)
print(classification_report(y_test, y_pred_gb)) print(confusion_matrix(y_test, y_pred_gb)) print("Accuracy:", accuracy_score(y_test, y_pred_gb))	Display the classification report results from the Gradient Boosting Machine algorithm

	precision	recall	f1-score	support
0	0.97	0.99	0.98	94
1	0.99	0.97	0.98	117
accuracy			0.98	211
macro avg	0.98	0.98	0.98	211
weighted avg	0.98	0.98	0.98	211

Accuracy: 0.981042654028436

Figure 6. Evaluation of Gradient Boosting Machine

Figure 6 shows the results of the evaluation after the model creation using the Gradient Boosting Machine algorithm, which includes:

The accuracy of the Gradient Boosting Machine model after testing achieved a result of 0.98, indicating that 98% of the model's predictions are correct.

The precision from the evaluation in this study for class 0 is 0.97, meaning 97% of class 0 predictions are correct, and for class 1, it is 0.99, meaning 99% of class 1 predictions are correct.

The recall evaluation of the Gradient Boosting Machine shows the recall results from this study, with class 0 having a result of 0.99, meaning 99% of the actual class 0 data was correctly identified by the model, while class 1 has a result of 0.97, indicating that 97% of the actual class 1 data was correctly identified by the model.

The F1-Score from this study's evaluation, which is the combination of precision and recall, shows 0.98 for both class 0 and class 1, indicating a balanced result between precision and recall.

Table 8. Comparison of Evaluation Results

Model	Precisi	%	Recall	%	F1-Score	%	Akurasi %
	Class 1	Class 0	Class 1	Class 0	Class 1	Class 0	
SVM	95%	92%	93%	94%	94%	93%	93%
GBM	99%	97%	97%	99%	98%	98%	98%

Table 8 presents the comparison results between the precision of the Support Vector Machine and Gradient Boosting Machine after model creation and testing on both models. The results indicate that:

The Support Vector Machine algorithm achieved a precision of 95% for class 1, meaning it was able to correctly predict class 1, while for class 0, it achieved 92% precision for correctly predicting class 0. The recall for class 1 is 93%, meaning 93% of the actual class 1 data was detected, while for class 0, it was 94%, meaning 94% of the actual class 0 data was detected. The F1-Score for class 1 is 94% and for class 0 is 93%, reflecting the combination of precision and recall, while the accuracy of the Support Vector Machine model is 93%, indicating that 93% of the total predictions were correct.

The Gradient Boosting Machine algorithm achieved a precision of 99% for class 1, meaning it was able to correctly predict class 1, while for class 0, it achieved 97% precision for class 0 predictions. The recall for class 1 is 97%, meaning 97% of the actual class 1 data was detected, while for class 0, it was 99%, meaning 99% of the actual class

0 data was detected. The F1-Score for class 1 is 98% and for class 0 is 98%, reflecting the combination of precision and recall, while the accuracy of the Gradient Boosting Machine model is 98%, meaning 98% of the total predictions were correct.

The conclusion drawn from Table 1.8 is that the Gradient Boosting Machine model is able to predict heart disease with very few errors, achieving a result of 98%, while the Support Vector Machine model is less effective at predicting heart disease, with a lower accuracy of 93%. This shows that Gradient Boosting Machine outperforms the Support Vector Machine model overall.

CONCLUSIONS AND SUGGESTIONS

Conclusion

The implementation of the Support Vector Machine and Gradient Boosting Machine algorithms in this study shows that both algorithms perform well in predicting heart disease and produce good results, especially the Gradient Boosting Machine algorithm, which outperforms the Support Vector Machine algorithm.

This study demonstrates that SMOTE, StandardScaler, and Correlation Thresholding enhance the performance of both the Support Vector Machine and Gradient Boosting Machine in predicting heart disease. SMOTE addresses class imbalance, StandardScaler normalizes numeric data, and Correlation Thresholding selects relevant features. As a result, Gradient Boosting Machine achieved 98% accuracy, outperforming the Support Vector Machine, which achieved only 93%.

Suggestion

Seek or add more data to further improve the accuracy of the research conducted. Conduct experiments comparing other algorithms, especially newer ones, to gain different experiences with similar cases.

REFERENCE

Amelia, Reni. 2017. “Faktor-Faktor Yang Mempengaruhi Status Kesehatan.” *Sosio Konsepsia* 2(2): 137–52. doi:10.33007/ska.v2i2.772.

Corey Wade, Kevin Glynn. 2020. *Gradient Boosting Machines*. Germany: Packt Publishing Ltd.

Erdania, Erdania, M. Faizal, and Rima Berti Anggraini. 2023. “FAKTOR – FAKTOR YANG BERHUBUNGAN DENGAN KEJADIAN PENYAKIT JANTUNG KORONER (PJK) Di RSUD Dr. (H.C.) Ir. SOEKARNO PROVINSI BANGKA BELITUNG TAHUN 2022.” *Jurnal Keperawatan* 12(1): 17–25. doi:10.47560/kep.v12i1.472.

Huda, Irkham Abdaul. 2020. “Perkembangan Teknologi Informasi Dan Komunikasi (Tik).” *Jurnal Pendidikan dan Konseling (JPDK)* 2(1): 121–25. doi:10.31004/jpdk.v1i2.622.

Ingo Steinwart, Andreas Christmann. 2008. *Support Vertor Machines*. Germany: Springer International Publishing.

Munawar, Zen. 2021. “Manfaat Teknologi Informasi Di Masa Pandemi Covid-19.” *Jurnal Sistem Informasi* 03(02): 9. https://ejournal.unibba.ac.id/index.php/j-sika/article/view/692.

Parikesit Dito; Putranto Arli Aditya; Anurogo, Riza Arief. 2018. “Kontribusi Aplikasi Medis Dari Perkembangan Pembelajaran Mesin (Machine Learning) Terbaru.”

Cermin Dunia Kedokteran 45(9): 700–703. http://www.kalbemed.com/DesktopModules/EasyDNNNews/DocumentDownload.ashx?portalid=0&moduleid=471&articleid=225&documentid=65.

Yudi Her Oktaviono. 2024. *PENYAKIT JANTUNG*. Jawa Timur: Airlangga University Press.

Zhang, Xian-Da. 2001. "Support Vector Machines (SVM) Support Vector Machines (SVM)." *Gesture* 23(6): 349–61.

Amelia, Reni. 2017. "Faktor-Faktor Yang Mempengaruhi Status Kesehatan." *Sosio Konsepsia* 2(2): 137–52. doi:10.33007/ska.v2i2.772.

Corey Wade, Kevin Glynn. 2020. *Gradient Boosting Machines*. Germany: Packt Publishing Ltd.

Erdania, Erdania, M. Faizal, and Rima Berti Anggraini. 2023. "FAKTOR – FAKTOR YANG BERHUBUNGAN DENGAN KEJADIAN PENYAKIT JANTUNG KORONER (PJK) Di RSUD Dr. (H.C.) Ir. SOEKARNO PROVINSI BANGKA BELITUNG TAHUN 2022." *Jurnal Keperawatan* 12(1): 17–25. doi:10.47560/kep.v12i1.472.

Huda, Irkham Abdaul. 2020. "Perkembangan Teknologi Informasi Dan Komunikasi (Tik)." *Jurnal Pendidikan dan Konseling (JPDK)* 2(1): 121–25. doi:10.31004/jpdk.v1i2.622.

Ingo Steinwart, Andreas Christmann. 2008. *Support Vertor Machines*. Germany: Springer International Publishing.

Munawar, Zen. 2021. "Manfaat Teknologi Informasi Di Masa Pandemi Covid-19." *Jurnal Sistem Informasi* 03(02): 9. https://ejournal.unibba.ac.id/index.php/j-sika/article/view/692.

Parikesit Dito; Putranto Arli Aditya; Anurogo, Riza Arief. 2018. "Kontribusi Aplikasi Medis Dari Perkembangan Pembelajaran Mesin (Machine Learning) Terbaru." *Cermin Dunia Kedokteran* 45(9): 700–703. http://www.kalbemed.com/DesktopModules/EasyDNNNews/DocumentDownload.ashx?portalid=0&moduleid=471&articleid=225&documentid=65.

Yudi Her Oktaviono. 2024. *PENYAKIT JANTUNG*. Jawa Timur: Airlangga University Press.

Zhang, Xian-Da. 2001. "Support Vector Machines (SVM) Support Vector Machines (SVM)." *Gesture* 23(6): 349–61.