**Bulletin of Informatics and Data Science**
Vol. 4 No. 1, May 2025, Page 1−9
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v4i1.94
https://ejurnal.pdsi.or.id/index.php/bids/index

# Enhancing Support Vector Machine Performance for Heart Attack Prediction using RobustScaler-Based Outlier Handling

**M Munawir Lasiyono[1,*], Nurhayati[2], Teotino Gomes Soares[3], Mulyadi[4]**

[1] Informatics Management Study Program, Politeknik Mitra Karya Mandiri, Brebes, Indonesia
[2] Informatics Engineering Study Program, Faculty of Engineering, Universitas Muhammadiyah Tangerang, Tangerang, Indonesia
[3] Computer Science Department, School of Engineering and Science, Dili Institute of Technology, Dili, Timor-Leste
[4] Information Systems Study Program, Faculty of Computer Science, Universitas Nurdin Hamzah, Jambi, Indonesia
Email: [1,*]mmunawirlasiyono@gmail.com, [2]nurhayati09011@ft-umt.ac.id, [3]tyosoares@gmail.com, [4]mulyadiroesly@gmail.com
Correspondence Author Email: mmunawirlasiyono@gmail.com

**Abstract**

Cardiovascular disease remains the leading cause of death worldwide, with most cases attributed to heart attacks and strokes. Early detection is crucial, yet conventional diagnostic methods are often constrained by time, cost, and uneven distribution of clinical expertise. Consequently, machine learning-based approaches offer a promising alternative for efficiently supporting heart attack prediction. This study employs the Support Vector Machine (SVM) algorithm, focusing on enhancing its performance through RobustScaler as a preprocessing technique to address outliers common in medical datasets. The objective of this study is to evaluate the impact of RobustScaler on SVM performance in heart attack classification. The model was developed using a dataset of 303 patient records, consisting of eight numerical features and one binary target label. Experiments were conducted under two preprocessing scenarios: without scaling (baseline) and with RobustScaler. Model performance was assessed using accuracy, precision, recall, F1-score, and ROC-AUC. The results show that applying RobustScaler significantly improves model performance, with accuracy increasing from 64.77% to 85.23%, representing a 20.46% improvement, and ROC-AUC rising from 73.65% to 93.36%, indicating a 26.78% increase in discriminatory ability. Additionally, recall for the negative class improved dramatically from 26.47% to 99.02%, reflecting better sensitivity in identifying non-heart attack cases. These findings demonstrate that proper preprocessing, particularly using RobustScaler, plays a vital role in optimizing SVM performance, especially when handling clinical data with extreme values.

**Keywords**: Support Vector Machine; RobustScaler; Heart Attack Prediction; Outlier Handling; Medical Data Classification

## 1. INTRODUCTION

Cardiovascular disease remains the leading cause of death worldwide. According to the World Health Organization (WHO), approximately 17.9 million deaths occur each year due to cardiovascular conditions, accounting for 32% of global mortality [1]. Of this figure, about 85% are caused by heart attacks and strokes, with a steadily increasing prevalence, particularly in developing countries [2]. Early detection is one of the most critical strategies to reduce this mortality rate. However, traditional diagnostic processes often require significant time, incur high costs, and heavily rely on clinical expertise, which is not always readily accessible. Therefore, data-driven prediction methods based on machine learning have emerged as promising alternatives for supporting the early identification of high-risk patients in a more efficient and accurate manner.

In practice, classical machine learning algorithms such as Logistic Regression, K-Nearest Neighbor (KNN), and Support Vector Machine (SVM) are still widely favored due to their interpretability, computational efficiency, and ability to operate effectively on small datasets [3]. Unlike ensemble or deep learning models-which often require large volumes of data, high computational resources, and tend to function as black-box systems-classical models are more transparent and easier to interpret, especially in applications that demand explainability and accountability [4]. Nevertheless, these classical approaches have several limitations, particularly when dealing with problematic data such as outliers and class imbalance, both of which can impair learning performance and reduce predictive accuracy [5].

Numerous studies have developed heart disease prediction models using classical machine learning methods. Ibrahima and Yu (2021) applied KNN and achieved 72.37% accuracy, but reported an imbalance in recall values, indicating the need for better attention to data distribution [6]. Barus et al. (2023) utilized Naive Bayes and achieved 74.58% accuracy, but showed a significant discrepancy between precision (97.67%) and recall (75%), suggesting a lack of proper preprocessing [7]. Febriani et al. (2023) proposed Fuzzy Logistic Regression, obtaining 80% accuracy but with low specificity and no consideration of outliers [8]. Azis (2024) employed Logistic Regression and reported accuracy ranging from 80% to 88%, but without addressing preprocessing techniques or the impact of extreme data values [9]. Akhdan et al. (2025) compared Decision Tree and Artificial Neural Network (ANN), reaching 87% accuracy, though low precision and F1-score pointed to the potential influence of outliers and class imbalance [10].

Based on prior research, most studies have not explicitly addressed the issue of outliers, which can reduce both accuracy and model generalization, particularly for algorithms like SVM that are highly sensitive to extreme values. Moreover, there is a lack of comparative studies that directly evaluate the impact of preprocessing techniques such as RobustScaler on SVM performance in the context of heart disease prediction. SVM is selected in this study as it is a robust and widely used classification algorithm, especially for binary classification problems [11]. SVM excels in constructing an optimal hyperplane that separates classes with a maximum margin and performs well on high-dimensional data [12]. However, SVM is also known to be sensitive to data scaling and outliers, which may affect the optimality of the decision boundary and decrease prediction accuracy [13]. To address this challenge, an appropriate preprocessing method is required to minimize the influence of outliers. RobustScaler is a data scaling technique designed to be resistant

**Bulletin of Informatics and Data Science**
Vol. 4 No. 1, May 2025, Page 1−9
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v4i1.94
https://ejurnal.pdsi.or.id/index.php/bids/index

to outliers by utilizing the interquartile range (IQR) rather than the mean and standard deviation used in StandardScaler [14]. This approach preserves the central distribution of the data while reducing the impact of extreme values, thus improving the model's stability and accuracy [15].

This study aims to enhance the performance of the SVM algorithm for heart attack prediction through outlier handling using RobustScaler and to conduct a comprehensive evaluation of model performance using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. The main contribution of this study lies in presenting a systematic approach for outlier handling to optimize SVM performance, along with an empirical comparison between models using no scaling and those using RobustScaler.

# 2. RESEARCH METHODOLOGY

## 2.1 Research Stages

The development of a heart attack prediction model using the Support Vector Machine (SVM) algorithm and the RobustScaler scaling technique was conducted through a series of systematic and integrated stages. Each step in the research process was methodologically designed to ensure that the proposed approach could be implemented in a structured manner and replicated in similar contexts [16]. The main stages of this research are illustrated in Figure 1.

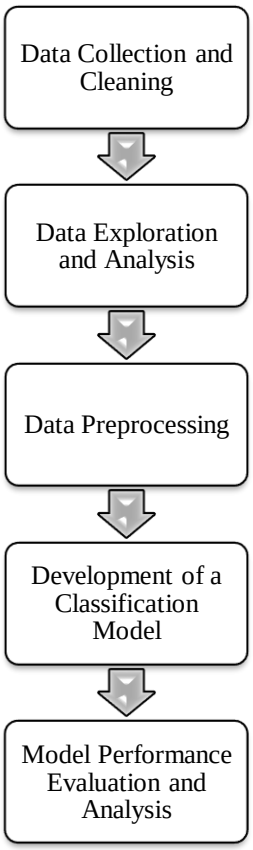


**Figure 1.** Research Pipeline

Figure 1 presents the overall flow of the research. A detailed explanation of each stage is provided below.

1) Data Collection and Cleaning
   The dataset used in this study was obtained from the Kaggle platform, titled “Heart Attack Dataset”, which is publicly available [17]. This dataset contains medical records of 303 patients, with a total of 9 attributes (8 features and 1 label). The features include: Age, Heart Rate, Systolic Blood Pressure, Diastolic Blood Pressure, Blood Sugar, CK-MB, Troponin, and Gender. The target label is provided in the *Result* column, indicating whether a patient experienced a heart attack (*positive*) or not (*negative*). Prior to modeling, the data underwent a cleaning process, including the removal of missing values, conversion of all features into appropriate numeric types, and binarization of the target label (0 for negative, 1 for positive). This step ensured that the data were of sufficient quality and consistency for preprocessing and model training.
2) Data Exploration and Analysis
   This stage aimed to understand the general characteristics of the dataset. Descriptive analysis was conducted to assess the distribution of each feature, relationships between variables, and the class proportions of the target label (positive and negative). Visualizations such as histograms, heatmaps, and boxplots were used to detect outliers and identify

relevant features. The results of this exploration informed the selection of preprocessing techniques and justified the use of RobustScaler.

3) Data Preprocessing
Data preprocessing was performed to ensure optimal conditions before training the model. The selected scaling technique was RobustScaler, which transforms features based on the median and interquartile range (IQR), making it more resistant to outliers [18]. Categorical features, such as gender, were numerically encoded. The data were then split into training and testing sets using an 80:20 stratified split to maintain balanced class proportions. This ratio is commonly used in classification modeling to allow the model to generalize well from 80% of the data while testing on the remaining 20% [19] . This stage resulted in two datasets ready for training and testing under two scenarios: without scaling (baseline) and with RobustScaler.
4) Development of a Classification Model
In this stage, the Support Vector Machine (SVM) algorithm was used as the primary classification model. The model was trained using the training set prepared under two scenarios: baseline (without scaling) and with RobustScaler. SVM works by finding the optimal hyperplane that separates two classes with the maximum margin. For non-linear data, SVM utilizes a kernel function to transform the data into a higher-dimensional space where a linear separation becomes possible [20]. This study applied the Radial Basis Function (RBF) kernel due to its effectiveness in capturing non-linear patterns among features. All experiments were conducted with consistent parameters to ensure that any observed performance differences were solely due to the preprocessing techniques used.
5) Model Evaluation
Model evaluation was conducted by assessing classification performance on the test set using several metrics. This stage began with the generation of a confusion matrix and ROC (Receiver Operating Characteristic) curve. The confusion matrix was used to calculate metrics such as accuracy, precision, recall, and F1-score, reflecting the model's correctness, sensitivity, and class-wise balance [21]. The ROC curve illustrates the relationship between the true positive rate (recall) and the false positive rate, and was used to compute the AUC (Area Under the Curve) as an indicator of the model's overall discriminatory capability [22]. The performance of the two models (with and without RobustScaler) was compared to evaluate the extent to which preprocessing affected accuracy and sensitivity, particularly in detecting positive (heart attack) cases.

### 2.2 Scaling Techniques Using RobustScaler

RobustScaler is a data normalization technique designed to reduce the impact of outliers. Unlike StandardScaler, which transforms data based on the mean and standard deviation, RobustScaler uses the median and interquartile range (IQR), making it more robust to skewed distributions and extreme values [23]. The transformation is defined by Equation (1).

$$x_{scaled} = \frac{x - Q_2}{Q_3 - Q_1} \tag{1}$$

where $Q_2$ is the median, and $Q_1$ and $Q_3$ are the first and third quartiles, respectively.

In medical datasets, outliers often arise due to clinical variations, recording errors, or rare conditions. If not properly addressed, outliers can negatively impact model performance, especially for algorithms sensitive to data scale, such as SVM and KNN. Therefore, RobustScaler is considered a suitable approach for improving the stability of predictive models. It is particularly recommended when datasets contain significant outliers or extreme values [24]. This scaling technique transforms features based on their interquartile range, minimizing the distortion caused by extreme observations.

It is important to note that in this study, RobustScaler was not used as a separate outlier detection or removal technique. Instead, it served purely as a preprocessing method to mitigate the influence of outliers through scaling, by transforming features relative to their interquartile range. This approach ensured that extreme values did not disproportionately affect the SVM's margin-based decision boundary.

### 2.3 Support Vector Machine (SVM) Method

Support Vector Machine (SVM) is a supervised learning algorithm commonly used for classification and regression tasks [25]. SVM works by identifying a hyperplane that optimally separates two classes with the maximum margin [26]. When data are not linearly separable, SVM employs kernel functions to map the data into a higher-dimensional space, enabling linear separation. In this study, the Radial Basis Function (RBF) kernel was chosen due to its ability to effectively capture non-linear relationships among features. Mathematically, SVM solves optimization through Equation (2).

$$\min_{w,b} \frac{1}{2}\|w\|^2 \text{ with conditions } y_i(w^T x_i + b) \geq 1, \forall_i \tag{2}$$

To handle non-perfect separability, slack variables and a penalty parameter *C* are introduced, allowing a balance between maximizing the margin and minimizing classification error. SVM is known for its strength in handling high-dimensional data and producing strong generalization on test data. However, a known limitation of SVM is its sensitivity to feature scaling and outliers, which can shift the hyperplane and degrade model performance [13]. Therefore, selecting

**Bulletin of Informatics and Data Science**
Vol. 4 No. 1, May 2025, Page 1−9
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v4i1.94
https://ejurnal.pdsi.or.id/index.php/bids/index

an appropriate preprocessing method such as RobustScaler is essential to ensure optimal model behavior when dealing with complex and varied medical datasets.

In this study, the Support Vector Machine model was implemented using the Scikit-learn library in Python. The classifier was instantiated using the SVC class from sklearn.svm, with the kernel set to 'rbf' to support non-linear separation. The regularization parameter C was set to 1.0 and the kernel coefficient gamma was set to 'scale', which is the default configuration in Scikit-learn and has shown good empirical performance on small to medium datasets. These parameters were kept constant across both scenarios (with and without scaling) to isolate the impact of the preprocessing technique on model performance. No cross-validation or hyperparameter optimization was performed, as the main objective was to evaluate the effectiveness of RobustScaler in enhancing model robustness under identical modeling conditions.

## 3. RESULT AND DISCUSSION

The development of a heart attack classification model using the Support Vector Machine (SVM) approach began with the preparation of an appropriate dataset. The dataset used in this study was sourced from the public platform Kaggle, titled “Heart Attack Dataset” [17]. It contains medical records of 303 patients, comprising nine attributes, which include eight input features and one target label. The available features are Age, Heart Rate, Systolic Blood Pressure, Diastolic Blood Pressure, Blood Sugar, CK-MB enzyme, Troponin, and Gender. The target label is represented by the Result attribute, which indicates whether a patient has experienced a heart attack (positive) or not (negative). Before the modeling process, the dataset underwent a cleaning stage involving the removal of missing values, conversion of all features into appropriate numeric formats, and binarization of the target label into 0 for negative and 1 for positive. These steps were essential to ensure data quality and consistency for the preprocessing and training stages.

The subsequent stage involved exploratory data analysis, which encompassed a detailed descriptive assessment of the distribution of each feature, investigation of inter-variable relationships, and evaluation of the proportion of instances across target classes. This step aimed to gain initial insights into the structure and characteristics of the dataset. As a starting point, the analysis focused on visualizing the distribution of the target variable to assess the degree of class balance. Understanding class distribution is essential in binary classification tasks, as imbalanced datasets can significantly influence model performance, particularly in terms of bias toward the majority class and reduced sensitivity in detecting the minority class. The visualization of class distribution is presented in Figure 2.


**Figure 2.** Visualization of Target Result Class Distribution

Figure 2 illustrates the distribution of the target class, consisting of patients who did not experience a heart attack (class 0) and those who did (class 1). The class distribution shows a moderate imbalance, with 61.4% of the data belonging to the negative class and 38.6% to the positive class. Although the class proportion differs, this imbalance is considered tolerable for training purposes. Therefore, oversampling or any other balancing techniques were not applied. This decision was made to preserve the natural structure of the data, although the possibility of slight bias toward the majority class was considered during model evaluation.

The next exploration step focused on analyzing the relationships between variables. This aimed to identify correlation patterns among numeric features and assess how strongly each feature relates to the target label. Such analysis provides preliminary insights into the strength and direction of these relationships, supporting decisions in feature selection and the use of appropriate predictive models. A heatmap was used to visualize the correlation values, with color intensity indicating the strength of the relationship. The correlation heatmap is presented in Figure 3.

**Bulletin of Informatics and Data Science**
Vol. 4 No. 1, May 2025, Page 1−9
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v4i1.94
https://ejurnal.pdsi.or.id/index.php/bids/index

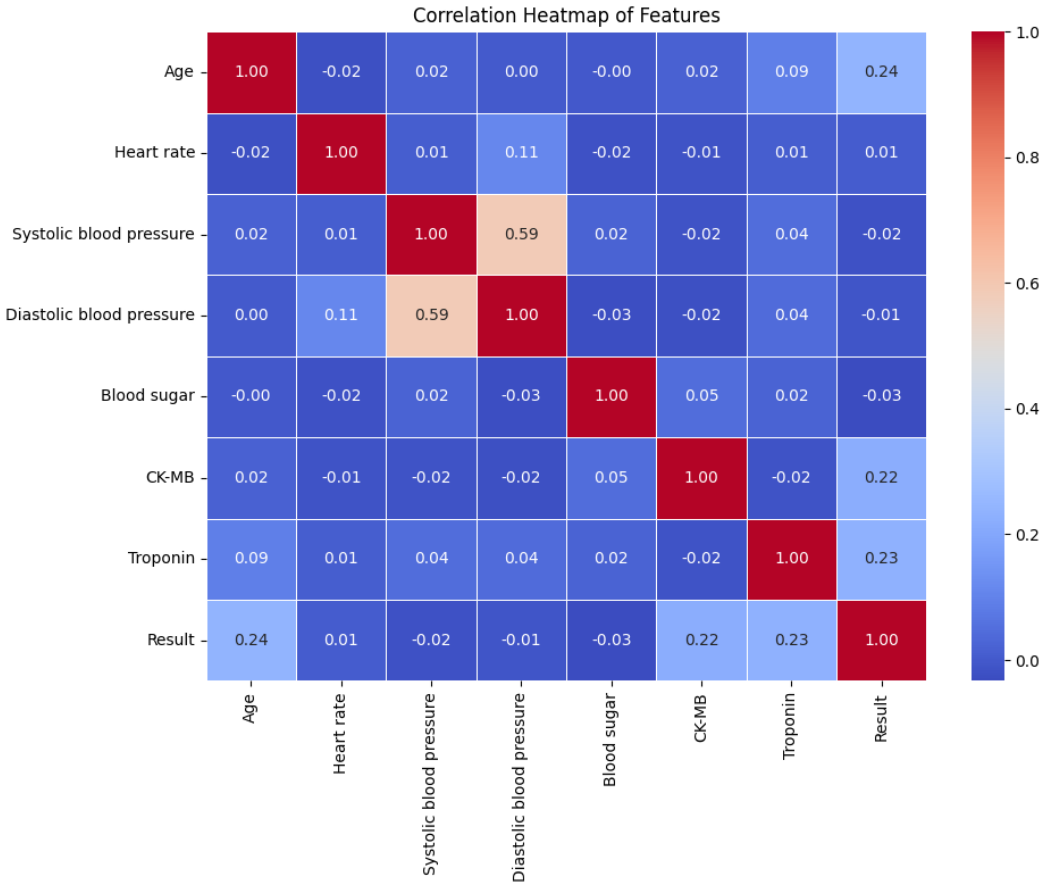


**Figure 3.** Heatmap of Correlation Between Features

Figure 3 displays the correlations among the numeric features in the dataset. The results indicate that most features have weak correlations with each other and with the target label. The strongest correlations with the *Result* label were found in Age (0.24), Troponin (0.23), and CK-MB (0.22). A moderate correlation of 0.59 was found between Systolic and Diastolic Blood Pressure. These findings imply that no individual feature dominates the prediction, which supports the use of classification models based on feature interaction, such as SVM with non-linear kernels.

Further exploration was conducted to observe the characteristics of the numeric features. This step aimed to examine the distribution of values, detect the presence of outliers, and identify potential impacts on model training. Such information is useful for determining suitable preprocessing strategies, including the selection of scaling techniques. Boxplots for the numeric features prior to scaling are shown in Figure 4.

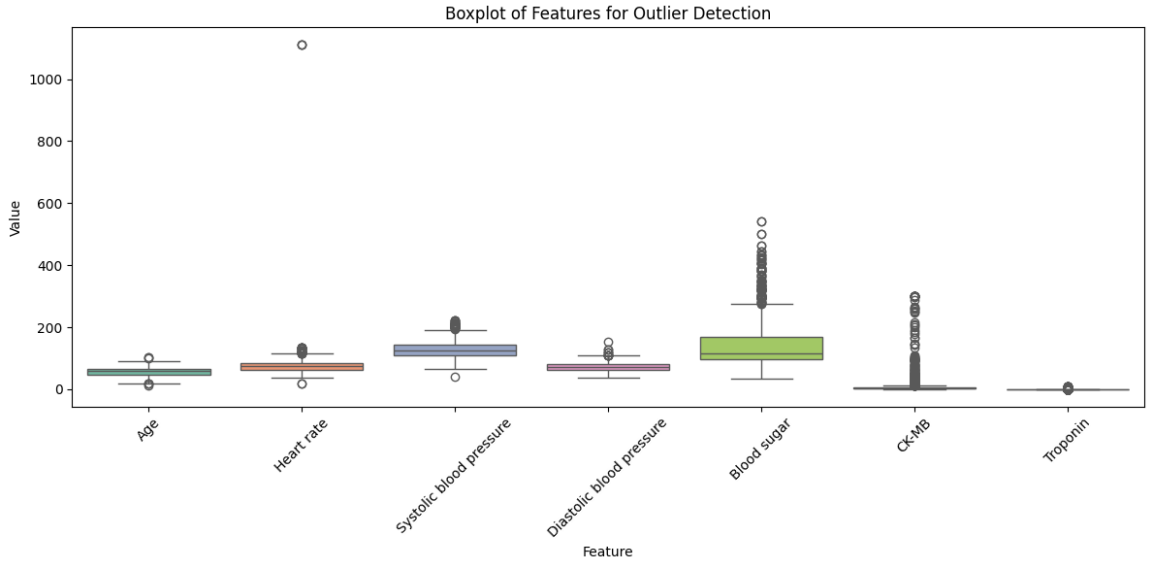


**Figure 4.** Initial Numerical Features Boxplot

**Bulletin of Informatics and Data Science**
Vol. 4 No. 1, May 2025, Page 1−9
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v4i1.94
https://ejurnal.pdsi.or.id/index.php/bids/index

Figure 4 presents boxplots of numeric features that highlight the presence of outliers. Most features, especially Blood Sugar, CK-MB, and Heart Rate, show extreme values beyond the normal range. This confirms that outliers exist in the data and may disrupt model training, which justifies the use of RobustScaler to reduce their influence. To better understand how RobustScaler operates, a simple manual calculation is presented using a sample of five Blood Sugar values: [100, 110, 120, 400, 115]. The first step is to compute the median (Q2):

$Sorted\ values$: $[100, 110, 120, 400, 115]$

$Q_2(Median) = 115$

Next, the second step is to calculate the lower quartile (Q1) and the upper quartile (Q3). So, the calculation is as follows:

$$[100,110] \rightarrow Q_1 = \frac{(100 + 110)}{2} = 105$$

$$[120,400] \rightarrow Q_3 = \frac{(120 + 400)}{2} = 260$$

The third step is to find the Interquartile Range (IQR) value, where IQR is a statistical measure that shows the middle dispersion of a dataset, namely the distance between the third quartile (Q3) and the first quartile (Q1). So the IQR value is as follows:

$$IQR = Q_3 - Q_1 = 260 - 105 = 155$$

After obtaining the median and IQR, each value is transformed using the scaling formula. The results are presented in Table 1.

**Table 1.** Data Transformation Results

| Original Data | Calculation | Scaled Result |
|---|---|---|
| 100 | (100 - 115) / 155 = -15 / 155 | ≈ -0.097 |
| 110 | (110 - 115) / 155 = -5 / 155 | ≈ -0.032 |
| 120 | (120 - 115) / 155 = 5 / 155 | ≈ 0.032 |
| 400 | (400 - 115) / 155 = 285 / 155 | ≈ 1.839 |
| 115 | (115 - 115) / 155 = 0 | 0 |

Table 1 illustrates that the median becomes the center of distribution with a value of zero, while extreme values such as 400 retain high magnitudes but are no longer dominant. RobustScaler reduces the influence of outliers by transforming data based on the interquartile range rather than mean and standard deviation. The result of applying RobustScaler to the dataset is visualized in Figure 5.

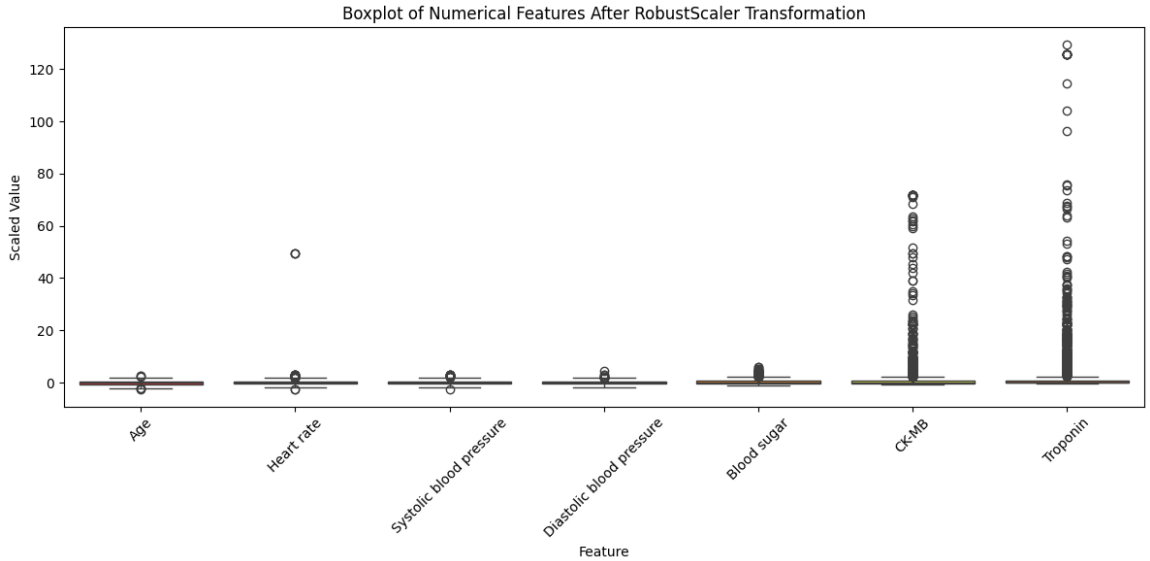


**Figure 5.** Boxplot of Numeric Features After Scaling Using RobustScaler

Figure 5 illustrates the boxplots of numerical features after transformation using RobustScaler. It can be observed that extreme values (outliers) have been significantly suppressed, and the distribution of each feature is now more concentrated around the zero median. This indicates that RobustScaler effectively reduces the influence of outliers and prepares the data more appropriately for classification algorithms such as SVM.

**Bulletin of Informatics and Data Science**
Vol. 4 No. 1, May 2025, Page 1−9
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v4i1.94
https://ejurnal.pdsi.or.id/index.php/bids/index

The next step involved the construction of the classification model using the Support Vector Machine (SVM) algorithm, which was implemented for both training and testing processes. The dataset was split into 80% training and 20% testing using stratified sampling to maintain the proportion of target classes in both subsets. In this study, the SVM algorithm was implemented using the *scikit-learn* (sklearn) library, a widely used Python library for machine learning. The model was instantiated using the SVC (Support Vector Classifier) class from the sklearn.svm module. The parameter kernel='rbf' was used, as the Radial Basis Function (RBF) kernel is known for its effectiveness in capturing nonlinear relationships between features in high-dimensional space.

Model performance was evaluated using several metrics, including the confusion matrix, classification report, and ROC-AUC score, which collectively measure the model's ability to distinguish between classes in a binary classification task. These metrics provide comprehensive insights into the model's accuracy, sensitivity, and overall performance, particularly in the presence of class imbalance. The confusion matrices and ROC curves for both the SVM model without scaling and the SVM model with RobustScaler preprocessing are presented in Figure 6.

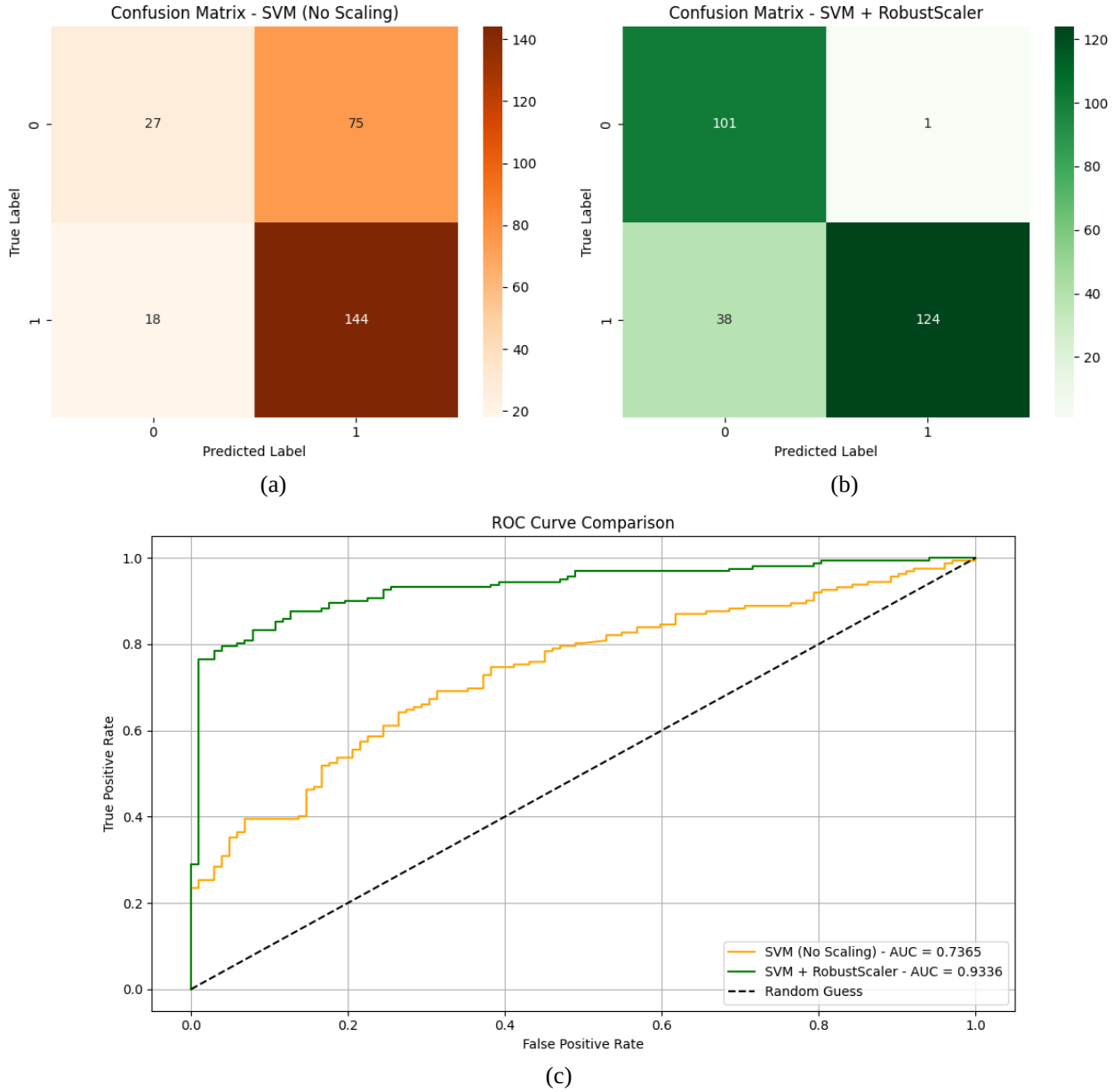


**Figure 6.** (a) Confusion Matrix for the SVM Model Without Scaling, (b) Confusion Matrix for the SVM Model With RobustScaler, (c) ROC Curve of Both Models

Figure 6 compares the confusion matrices and ROC curves of the two SVM models. The results indicate that the model using RobustScaler achieved a higher AUC score (0.9336), signifying better classification performance compared to the model without scaling (AUC 0.7365). Based on the confusion matrix and ROC curve results, further evaluation was conducted using classification reports and ROC-AUC scores for both models. A complete comparison of the performance metrics is shown in Table 2.

**Table 2.** Comparison of SVM Model Performance Without Scaling and With RobustScaler

| Method | Class | Precision | Recall | F1-Score | Accuracy | ROC-AUC Score |
|---|---|---|---|---|---|---|
| SVM (No Scaling) | Negative | 60.00% | 26.47% | 36.73% | 64.77% | 73.65% |
| | Positive | 65.75% | 88.89% | 75.59% | | |
| SVM + RobustScaler | Negative | 72.66% | 99.02% | 83.82% | 85.23% | 93.36% |
| | Positive | 99.20% | 76.54% | 86.41% | | |

The evaluation results in Table 2 demonstrate that the SVM model with RobustScaler consistently outperformed the model without scaling across all major performance metrics. The accuracy improved from 64.77% to 85.23%, representing an increase of 20.46%. This improvement highlights the significant impact of RobustScaler preprocessing on the model's predictive performance.

To further contextualize the results, an accuracy comparison with prior studies was conducted. It is important to note that the datasets used in those studies may differ from the one employed in this research. The comparison is summarized in Table 3.

**Table 3.** Accuracy Comparison with Prior Studies

| Study | Method | Reported Accuracy |
|---|---|---|
| Ibrahima & Yu (2021) [6] | K-Nearest Neighbor (KNN) | 72.37% |
| Barus et al. (2023) [7] | Naive Bayes | 74.58% |
| Febriani et al. (2023) [8] | Fuzzy Logistic Regression | 80.00% |
| Azis (2024) [9] | Logistic Regression | 80–88% |
| Akhdan et al. (2025) [10] | Decision Tree, ANN | 87.00% |
| This study | SVM + RobustScaler | 85.23% |

Table 3 summarizes the accuracy achieved in this study compared to previous classical machine learning approaches. Among the referenced studies, Akhdan et al. [10] obtained the highest accuracy of 87.00% using a combination of Decision Tree and Artificial Neural Network. Ibrahima and Yu [6] reported 72.37% using K-Nearest Neighbor, Barus et al. [7] achieved 74.58% with Naive Bayes, and Febriani et al. [8] obtained 80.00% with Fuzzy Logistic Regression. Azis [9] reported an accuracy ranging from 80% to 88% using Logistic Regression, although details regarding preprocessing were not explicitly mentioned. In comparison, the proposed model achieved 85.23% accuracy using a single algorithm, Support Vector Machine, demonstrating competitive performance.

This study focuses specifically on addressing the outlier problem, which was not explicitly handled in previous studies. By applying RobustScaler as a preprocessing strategy, the research aims to demonstrate that outlier handling can significantly enhance model accuracy. The performance improvements shown in Table 2 reinforce the importance of robust preprocessing, particularly when dealing with clinical datasets that often contain extreme values.

Despite the strong results, one limitation was a slight reduction in recall for the positive class, indicating that some heart attack cases remained undetected. This issue requires careful attention in medical contexts, as it may impact clinical decision-making. Future research may focus on SVM hyperparameter optimization, class imbalance handling (such as class weighting), and combining preprocessing with feature selection techniques to further improve model performance.

## 4. CONCLUSION

This study demonstrated that applying RobustScaler as a preprocessing technique significantly improved the performance of the Support Vector Machine (SVM) algorithm in predicting heart attack cases. Without preprocessing, the baseline SVM model achieved an accuracy of 64.77% and showed poor sensitivity toward the negative class, with a recall of only 26.47%. After using RobustScaler, the model's accuracy increased to 85.23%, and the ROC-AUC score rose from 73.65% to 93.36%, indicating a 26.78% improvement in classification capability. These findings confirm that selecting the appropriate preprocessing strategy, particularly in handling outliers, plays an essential role in enhancing model performance on clinical datasets. However, this study has several limitations. The dataset used was relatively small, consisting of only 303 patient records, and the results were not validated on external datasets. In addition, the study focused exclusively on RobustScaler without comparing it to other scaling or outlier-handling techniques. The model also showed a slight decline in recall for the positive class, indicating that a number of heart attack cases were still misclassified. Future research is encouraged to expand the dataset, perform parameter optimization, evaluate other preprocessing methods, and test the model across different populations or clinical settings to improve robustness and generalizability.

## REFERENCES

[1] T. A. Gaziano, *Cardiovascular Diseases Worldwide*. Boca Raton: CRC Press, 2022. doi: 10.1201/b23266-2.
[2] J. Lin, Y. Chen, N. Jiang, Z. Li, and S. Xu, "Burden of Peripheral Artery Disease and Its Attributable Risk Factors in 204 Countries and Territories From 1990 to 2019," *Front. Cardiovasc. Med.*, vol. 9, pp. 420–431, 2022, doi: 10.3389/fcvm.2022.868370.

[3] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning," *Decis. Anal. J.*, vol. 3, pp. 1–21, 2022, doi: 10.1016/j.dajour.2022.100071.
[4] M. Awais, L. Chiari, E. A. F. Ihlen, J. L. Helbostad, and L. Palmerini, "Classical machine learning versus deep learning for the older adults free-living activity classification," *Sensors*, vol. 21, no. 14, pp. 1–13, 2021, doi: 10.3390/s21144669.
[5] F. Bozkurt, "A Comparative Study on Classifying Human Activities Using Classical Machine and Deep Learning Methods," *Arab. J. Sci. Eng.*, vol. 47, no. 2, pp. 1507–1521, 2022, doi: 10.1007/s13369-021-06008-5.
[6] I. Bah and X. Yu, "KNN Algorithm Used for Heart Attack Detection," *FES J. Eng. Sci.*, vol. 11, no. 1, pp. 7–19, 2021, doi: 10.52981/fjes.v11i1.758.
[7] O. P. Barus, K. Lauwren, J. J. Pangaribuan, and Romindo, "Implementation of the Naive Bayes Algorithm to Predict the Safety of Heart Failure Patients," *IAIC Int. Conf. Ser.*, vol. 4, no. 1, pp. 172–177, 2023, doi: 10.34306/conferenceseries.v4i1.651.
[8] V. Febriani, D. Lestari, S. Mardiyati, and O. Lilyasari, "Fuzzy Logistic Regression Application on Predictions Coronary Heart Disease," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 17, no. 1, pp. 0571–0580, 2023, doi: 10.30598/barekengvol17iss1pp0571-0580.
[9] H. Azis, "Assessing the Performance of Logistic Regression in Heart Disease Detection through 5-Fold Cross-Validation," *Int. J. Artif. Intell. Med. Issues*, vol. 2, no. 1, pp. 1–11, 2024, doi: 10.56705/ijaimi.v2i1.137.
[10] F. Muhammad, R. Akhdan, A. Ismail, I. A. Mashudi, and A. L. Maukar, "Comparative Analysis of Decision Tree and Artificial Neural Network Methods for Predicting Potential Heart Disease," *Inf. J. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 10, no. 1, pp. 29–33, 2025.
[11] N. Jasmin, R. K. Dinata, and I. Sahputra, "Implementation of Data Mining for Vertigo Disease Classification Using the Support Vector Machine (SVM) Method," *J. Adv. Comput. Knowl. Algorithms*, vol. 1, no. 4, pp. 103–108, 2024.
[12] R. Hoque, M. Billah, A. Debnath, S. M. S. Hossain, and N. Bin Sharif, "Heart Disease Prediction using SVM," *Int. J. Sci. Res. Arch.*, vol. 11, no. 2, pp. 412–420, 2024, doi: 10.30574/ijsra.2024.11.2.0435.
[13] R. Guido, S. Ferrisi, D. Lofaro, and D. Conforti, "An Overview on the Advancements of Support Vector Machine Models in Healthcare Applications: A Review," *Inf.*, vol. 15, no. 4, 2024, doi: 10.3390/info15040235.
[14] Z. L. Thakker and S. H. Buch, "Effect of Feature Scaling Pre-processing Techniques on Machine Learning Algorithms to Predict Particulate Matter Concentration for Gandhinagar, Gujarat, India," *Int. J. Sci. Res. Sci. Technol.*, vol. 11, no. 1, pp. 410–419, 2024, doi: 10.32628/ijsrst52411150.
[15] A. Khoirunnisa and N. G. Ramadhan, "Improving malaria prediction with ensemble learning and robust scaler: An integrated approach for enhanced accuracy," *J. Infotel*, vol. 15, no. 4, pp. 326–334, 2023, doi: 10.20895/infotel.v15i4.1056.
[16] R. I. Borman, F. Rossi, D. Alamsyah, R. Nuraini, and Y. Jusman, "Classification of Medicinal Wild Plants Using Radial Basis Function Neural Network with Least Mean Square," in *International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)*, IEEE, 2022.
[17] S. S. Brar, "Heart Attack Dataset," Kaggle. Accessed: Mar. 15, 2025. [Online]. Available: https://www.kaggle.com/datasets/sukhmandeepsinghbrar/heart-attack-dataset
[18] M. M. Ahsan, M. A. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance," *Technologies*, vol. 9, no. 52, pp. 1–17, 2021, doi: 10.3390/technologies9030052.
[19] I. O. Muraina, "Ideal Dataset Splitting Ratios in Machine Learning Algorithms: General Concerns for Data Scientists and Data Analysts," in *International Mardin Artuklu Scientific Researches Conference*, 2022, pp. 496–505.
[20] M. P. Sharma, U. Meena, and G. K. Sharma, "Intelligent Data Analysis using Optimized Support Vector Machine Based Data Mining Approach for Tourism Industry," *ACM Trans. Knowl. Discov. Data*, vol. 16, no. 5, 2022, doi: 10.1145/3494566.
[21] Parjito, I. Ahmad, R. I. Borman, A. D. Alexander, and Y. Jusman, "Combining Extreme Learning Machine and Linear Discriminant Analysis for Optimized Apple Leaf Disease Classification," in *International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)*, IEEE, 2024, pp. 138–143. doi: 10.1109/ICE3IS62977.2024.10775844.
[22] Y. Liu, Y. Li, and D. Xie, "Implications of imbalanced datasets for empirical ROC-AUC estimation in binary classification tasks," *J. Stat. Comput. Simul.*, vol. 94, no. 1, pp. 183–203, Jan. 2024, doi: 10.1080/00949655.2023.2238235.
[23] I. Izonin, B. Ilchyshyn, R. Tkachenko, M. Gregus, N. Shakhovska, and C. Strauss, "Towards Data Normalization Task for the Efficient Mining of Medical Data," in *International Conference on Advanced Computer Information Technologies (ACIT)*, 2022, pp. 480–484. doi: 10.1109/ACIT54803.2022.9913112.
[24] A. Ramsauer, P. M. Baumann, and C. Lex, "The Influence of Data Preparation on Outlier Detection in Driveability Data," *SN Comput. Sci.*, vol. 2, no. 3, pp. 1–16, 2021, doi: 10.1007/s42979-021-00607-7.
[25] L. K. Ramasamy, S. Kadry, Y. Nam, and M. N. Meqdad, "Performance analysis of sentiments in Twitter dataset using SVM models," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 3, pp. 2275–2284, 2021, doi: 10.11591/ijece.v11i3.pp2275-2284.
[26] A. Kurani, P. Doshi, A. Vakharia, and M. Shah, "A Comprehensive Comparative Study of Artificial Neural Network (ANN) and Support Vector Machines (SVM) on Stock Forecasting," *Ann. Data Sci.*, vol. 10, no. 1, pp. 183–208, 2023, doi: 10.1007/s40745-021-00344-x.