Institut Riset dan Publikasi Indonesia (IRPI)
IJATIS: Indonesian Journal of Applied Technology and Innovation Science
Journal Homepage: https://journal.irpi.or.id/index.php/ijatis
Vol. 2 Iss. 2 August 2025, pp: 80-93
ISSN(P): 3032-7466 | ISSN(E): 3032-7474

Implementation of Machine Learning Algorithm for Heart Attack Disease Prediction

Febbi Ardiani[1], Irma Fitriani[2], Nabil Gustian[3] , Meliani Putri Diamon Chandra[4], Hasna Uzakiyah[5]

[1,2]Department of Information System, Faculty of Science and Technology,
Universitas Islam Negeri Sultan Syarif Kasim Riau, Indonesia
[3]Department of Islamic Studies, Faculty of Adab and Humaniora,
Sidi Mohamed Ben Abdellah University, Morocco
[4]Departement of Management, Faculty of Economy and Business,
Kütahya Dumlupınar University, Turkey
[5]Departement of E-commerce, Faculty of Science and Technology,
Nantong University, China

E-Mail: [1]12250323179@students.uin-suska.ac.id, [2]12250325541@students.uin-suska.ac.id,
[3]gustiannabil@gmail.com, [4]melainiputridiamon@gmail.com, [5]hasnauzakiyah21@gmail.com

Received Jan 01st 2025; Revised Aug 18th 2025; Accepted Aug 30th 2025; Available Online Aug 31th 2025
Corresponding Author: Febbi Ardiani

Abstract

Heart attack disease is one of the leading causes of death worldwide, making early detection a critical factor in reducing mortality. However, manual prediction is often inaccurate due to the complexity of medical data. To address this issue, this study evaluates five machine learning algorithms K-Nearest Neighbors (KNN), Decision Tree, Naïve Bayes, Random Forest, and Support Vector Machine (SVM) for predicting heart attack risk. The dataset, obtained from Kaggle, was preprocessed and divided into training and testing sets using 70:30 and 80:20 ratios. Algorithm performance was assessed using accuracy, precision, recall, and F1-score. The results showed that Decision Tree and Random Forest achieved the best performance with accuracy up to 97.98%, while KNN recorded the lowest accuracy at around 61.36%. This study not only demonstrates the comparative effectiveness of these algorithms on the same dataset, contributing to the growing body of research on AI in healthcare, but also highlights their potential clinical utility. In particular, Decision Tree and Random Forest can support the development of AI-based clinical decision support systems to assist healthcare professionals in early diagnosis and risk management.

Keywords: Decision Tree, Heart Attack Prediction, K-Nearest Neighbors, Random Forest, Support Vector Machine

1. INTRODUCTION

Myocardial infarction, another name for heart attack, occurs when the heart muscle does not receive sufficient blood and oxygen to function properly [1] [2]. Blockage of the arteries that supply blood to the heart is often the main cause, making a heart attack a life-threatening medical emergency [3] [4]. Globally, this disease is one of the leading causes of death, responsible for an estimated 17 million deaths each year. The impact is also evident at the national level: in Pakistan, heart attacks accounted for 19% of total deaths in 2020 and increased to 29% in the following year, while in Indonesia they were recorded as the leading cause of death across all age groups at 12.9% [5]. These figures highlight that heart attacks are not only a global health concern but also a serious national challenge, emphasizing the urgency of research on more accurate early detection methods to help reduce mortality rates.

Heart attacks can be caused by various lifestyle factors as well as genetic predisposition [4]. The most common symptoms include chest pain, shortness of breath, and unusual fatigue. In addition, patients may experience dizziness, nausea, excessive sweating, or pain that radiates to the neck, jaw, back, shoulders, and arms [6]. These diverse symptoms reflect the disruption of blood supply to the heart muscle and highlight the importance of early recognition.

Prevention of heart attacks can be achieved through lifestyle modifications and risk factor management, such as controlling blood pressure, cholesterol, and blood sugar. Several medical procedures are also available to treat heart attack patients, including stenting, angioplasty, coronary artery bypass

surgery, heart valve surgery, and even heart transplantation [1]. Correct detection and timely treatment have been shown to significantly reduce mortality and the risk of recurrence [7].

Early prediction of heart attack disease is critical to saving a patient's life [8]. Direct prediction by doctors is sometimes inaccurate, so in predicting heart attacks, artificial intelligence technology can be used [9]. Machine Learning can be used to find invisible patterns to predict heart attack disease and provide medical insights that will help doctors determine whether or not someone is having a heart attack [9], [10]. Machine Learning helps discover hidden features by analysing data [11]. Some frequently used algorithms include K Nearest Neighbors (K-NN), Naive Bayes Classifier (NBC), Decision Tree, Random Forest, and Support Vector Machine (SVM) [10]. Each algorithm has a unique approach to dealing with data patterns, such as the ability of K-NN to classify based on the closeness of k values, Naïve Bayes' flexibility in handling probabilities, and Random Forest's high accuracy in utilizing ensemble methods [12], [13].

Previous studies show varying results in heart attack prediction. Kamila et al. (2023) reported that Random Forest achieved an accuracy of 81.82%, higher than Decision Tree which reached 77.44% [14]. Likewise, Shashikant R. and Chetankumar P. (2023) found Random Forest to be the most accurate model with 93.61%, outperforming Logistic Regression with 88.50% and Decision Tree with 92.59% [14]. In contrast, Lite et al. (2024) highlighted the advantages of Naïve Bayes, which demonstrated high speed and reliable accuracy in both binary and multiclass prediction, surpassing algorithms such as SVM, K-NN, and Logistic Regression. These findings indicate that each algorithm has distinct strengths depending on data characteristics. Therefore, this study evaluates five algorithms K-Nearest Neighbors (K-NN), Decision Tree, Naïve Bayes, Random Forest, and SVM selected because they represent different categories of machine learning methods, to determine the most effective model for heart attack risk prediction [15],[16].

Previous reviews on heart attack datasets show that Random Forest and Decision Tree algorithms are superior to the SVM, and K-NN algorithms, while research conducted by Lite, et al shows the Naïve Bayes algorithm has high speed and accuracy [9], [12]. Based on the review of previous research, the novelty of this research is to test the prediction of a person's risk of heart attack with five algorithms, namely K-NN, Decision Tree, Naïve Bayes Classifier, Random Forest, and SVM. This research aims to test the effectiveness of the five algorithms on the heart attack dataset to find out which algorithms provide the most effective results.

2. MATERIAL AND METHOD

The methodology of this research can be seen in Figure 1.

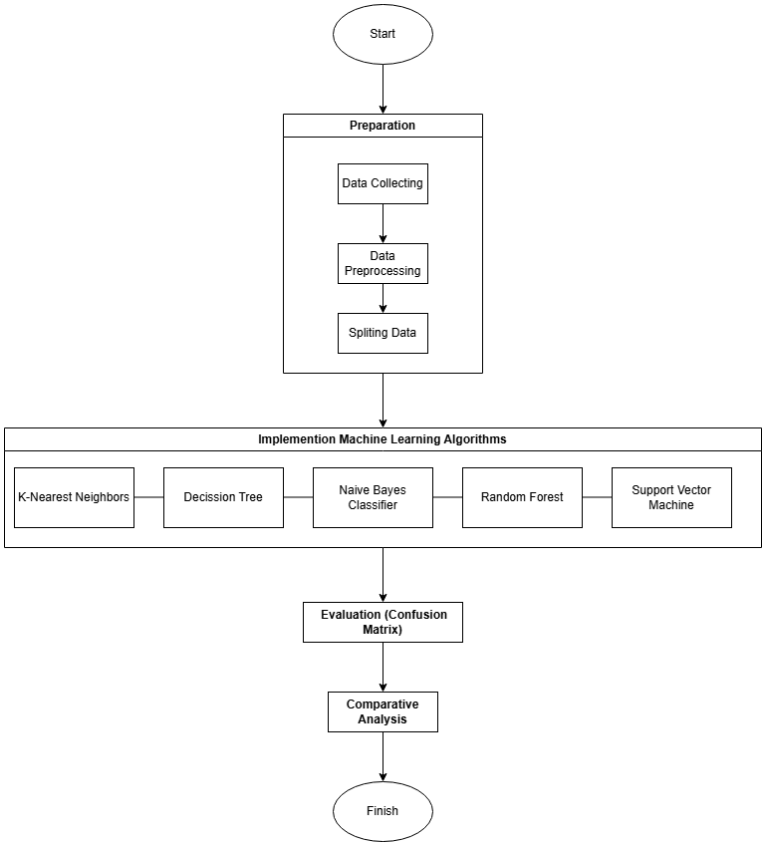

Figure 1. Research Methodology

This research began with a literature review, which examined heart attack risk, relevant health indicators, and machine learning algorithms such as K-NN, Decision Tree, Naïve Bayes Classifier, Random Forest, and SVM. This study helps determine the appropriate approach and dataset. The next stage was data collection, where a dataset from Kaggle containing 1,319 rows and 9 features, such as age, blood pressure, glucose level, and troponin level, was used as the research base. In the Data Preprocessing stage, the dataset was analyzed and no empty values were found. The data was then separated into features (x) and labels (y), and divided into training data (70%,80%) and test data (30%,20%). This stage ensures data is ready for Next, in the Implementation Machine Learning Algorithms stage, five algorithms are applied to predict heart attack risk based on health features. The results of each algorithm are evaluated using metrics such as accuracy, precision, recall, and F1-score in the Evaluation stage. In the Comparative Analysis step, the performance of the algorithms is compared to determine the most accurate model.

2.1. Data Collecting

The heart attack risk prediction dataset is taken from the Kaggle platform and contains 1,319 records with 9 main variables. These variables include various indicators of an individual's health, such as age, gender, heart rate, systolic and diastolic blood pressure, blood glucose levels, body mass, and troponin levels, which are markers of heart damage. The dataset also includes a category label in the class column, which identifies whether the individual is at risk of heart attack (positive) or not (negative). The data is collected from relevant clinical sources or medical surveys, to help predict heart attack risk based on an individual's physiological data.

2.2. Data Preprocessing

Preprocessing, an important stage in data processing, is performed to improve the quality of data used in machine learning models. Resolving missing values, removing noise, and normalizing and standardizing data are all part of this process [1] [2]. The dataset preprocessing stage involves checking and handling missing values to ensure the data is clean and complete. In this heart attack dataset from Kaggle, no missing values were found, so no imputation or deletion steps were required. The data was then separated into features (x) and targets (y), where targets in the form of ‘positive’ and ‘negative’ labels were converted into numeric format. After that, the dataset was divided into two scenarios, namely 70:30 and 80:20 for training and testing each algorithm.

2.3. Heart Attack

The World Health Organisation (WHO) states that, in the last 15 years, an estimated 17 million people have died each year from heart disease, particularly heart attacks [3] [4]. Heart attack is a term that refers to a collection of conditions that include the heart, blood vessels, muscles, valves, or the internal electrical pathways that control muscle contraction [5] [3]. Risk factors such as a patient's age and cholesterol levels can lead to heart attack disease [6]. There are two main types of heart attack. Type I heart attacks occur when the inner wall of an artery ruptures, releasing cholesterol and other chemicals into the circulation. Type II heart attacks occur when the artery is not completely blocked, but the heart still does not receive enough oxygen-rich blood [7].

2.4. Machine Learning

It is very important to predict heart attack disease early to save the patient's life [8]. Some academics have recently discovered a new method for selecting elements that play a role in the diagnosis of heart attack disease [9]. Machine learning is one way of modeling that can be used in the early detection of heart attack disease [10]. Machine learning methods are more comprehensive than traditional risk prediction approaches [10]. Many variables in a data set are often repetitive or irrelevant when building machine learning models. If all these features are included in the model, it may adversely affect the performance and accuracy of the model. Therefore, it is important to select the most relevant features and remove unnecessary features, which is done through the feature selection process in machine learning [6].

2.5. K-Nearest Neighbors (K-NN)

The k-nearest Neighbour approach is a commonly used classification method proposed by Fix and Hodges [11]. The k-NN algorithm works on the principle that data that have similar characteristics tend to be nearby in the feature space. It predicts the label or value of a new data point by referring to the majority of the labels or average values of its k nearest neighbors, which are determined using a distance metric such as Euclidean, Manhattan, or Minkowski [9]. The k-NN algorithm equation [9]:

$$D(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i} - y_i)^2 \quad (1)$$

D(X, Y) is the distance between objects a and b. Where x_i is the training data and yi is the testing data. While n is the number of independent variables.

2.6. Decision Tree

Decision tree is a method used to help make decisions by considering various factors that affect a problem [12]. This algorithm forms a tree-like structure, where the main attribute becomes the root, and each branch represents a specific value or condition [14]. Decision trees divide the entire data set into subgroups containing instances of the same class [16] [17]. This algorithm can be used to predict category class names, classify data based on the training set and class labels, and classify new data received [18]. The decision tree structure is depicted in Figure 2.

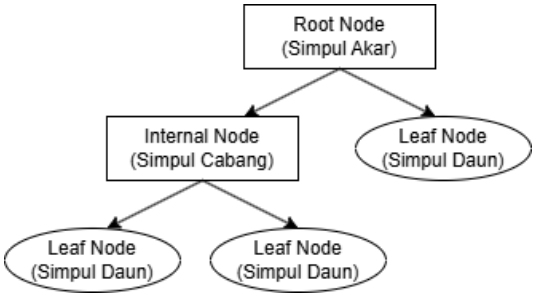

Figure 2. Decision Tree Structure

2.7. Naïve Bayes Clasifier

Naïve Bayes is a statistical classification technique based on Bayes' Theorem [19]. Naive Bayes is one of the simplest supervised learning algorithms. It calculates an initial probability for a particular class label as well as its likelihood probability, then generates a conditional probability for a particular target [20][21]. On the other hand, the NBC classifier is significantly faster than other algorithms because it only performs probability calculations and achieves a high performance rate on categorical data. NBC is based on the assumption of independent predictors, although it is very rare to find a set of independent predictors in the real world. Several measures can be used to evaluate this learning algorithm [22], [23]. Equation of the Naïve Bayes algorithm [24]:

$$P(A\backslash B) = \frac{P(B\backslash A)P(A)}{P(B)} \quad (2)$$

P(A|B) is the conditional probability from A to B. Where P(A) is the probability of event A and P(B) is the probability of event B.

2.8. Random Forest

Random Forest is an ensemble tree technique developed by Breiman [25]. This technique predicts the dependent variable by averaging the predictions of the decision tree [26]. Each tree is trained with a randomly sampled subset of the predictor variables and bootstrap samples from the training set [27], [28]. By using an ensemble of various decision trees, RF can reduce bias and variance, which helps improve prediction accuracy [29]. The downside is the high computational requirements needed to train the model well, especially if the dataset is large and complex [30]. Random forest algorithm equation [26]:

$$\text{Entropy (Y)} = -\sum \text{p(c\Y) i log2 p (c\Y)} \quad (3)$$

Where Y is the set of cases, and p(c|Y) is the proportion of Y values to class c.

$$\text{(Y, a)} = \text{Entropy (Y)} - \sum \text{|Yv| |Ya| values Entropy)Yv)} \quad (4)$$

Where the values of (a) are all possible values in the set of cases of a. Yv is a subclass of Y with class v, which corresponds to class a. Yes, there are all values corresponding to a

2.9. Support Vector Machine (SVM)

SVM is used to categorize image feature vectors into two groups based on class labels [31]. This method uses a kernel function to transform the input data into a large space, allowing the separation of non-linear data [31], [32]. Some of the kernel functions considered are Gaussian, linear, quadratic, and cubic, and kernel selection is done based on the best performance. SVM excels in optimizing data separation using a

hyperplane that maximally separates different classes. SVM is combined with the Recursive Feature Elimination (RFE) method to select the most relevant features among the 104 quantitative features extracted from the image. RFE gradually removes features that contribute the least to the classification thereby improving the efficiency of SVM in distinguishing abruption [33]. SVM requires accurate motor parameters and additional PI control to generate optimal torque change reference [34]. Super Vector Machine algorithm equation [35].

$$w \cdot x + b \tag{5}$$

SVM Hyperplane Equation that separates two classes w is the vector normal to the hyperplane that determines the direction of separation, x is the input feature vector, and b is the bias that determines the position of the hyperplane.

3. RESULTS AND DISCUSSION

3.1. Initial Data

This heart attack dataset is of good quality as there are no missing values, and all data types are suitable for analysis. The clean data structure allows its use in various machine learning algorithms to model heart disease risk prediction. Heart attack disease is affected by several conditions shown in Table 1.

Table 1. Heart Attack Prediction Datasets

Age	Gender	Impulse	Pressurehight	…..	Class
64	1	66	160	…..	Negative
21	1	94	98	…..	Positive
55	1	64	160	…..	Negative
64	1	70	120	…..	Positive
55	1	64	112	…..	Negative
58	0	61	179	…..	Negative
32	0	40	214	…..	Negative
…..	…..	…..	…..	…..	…..
…..	…..	…..	…..	…..	…..
…..	…..	…..	…..	…..	…..
45	1	85	168	…..	Positive
54	1	58	117	…..	Positive
51	1	94	157	…..	Positive

3.2. Data Preprocessing

The preprocessing stage of machine learning classification datasets aims to prepare the data for use in modeling algorithms. It starts with data cleaning, such as removing missing values, duplicates, or handling outliers. In this heart attack dataset, there are no missing values or the data is clean and normal, it's just that the class part which was originally categorical is changed to numeric, negative is changed to 0, and positive is changed to 1. Data that has passed the preprocessing stage can be seen in Table 2.

Table 2. Data Preprocessing Results

Age	Gender	Impulse	Pressurehight	…..	Class
64	1	66	160	…..	0
21	1	94	98	…..	1
55	1	64	160	…..	0
64	1	70	120	…..	1
55	1	64	112	…..	0
58	0	61	179	…..	0
32	0	40	214	…..	0

3.3. K-Nearest Neighbors (K-NN)

The test evaluation results of the K-NN algorithm with a confusion matrix can be seen in Figures 3 and 4. Based on the confusion matrix in Figure 3, the results of testing the K-NN algorithm with a 70:30 data split on the heart attack disease dataset correctly predict 169 positive cases (people at risk of developing the disease) and 74 negative cases (people who are not at risk). Howeveer, 81 cases were wrongly predicted positive, when in fact the person was not at risk (False Positive), and 72 cases that were wrongly predicted negative, when in fact the person was at risk (False Negative). Further explanation of the performance evaluation of the KNN algorithm with a 70:30 data split is shown in Table 3.

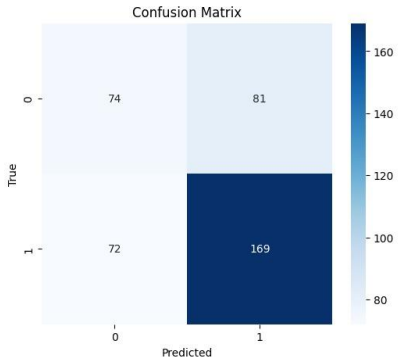

Figure 3. Confusion Matrix KNN with split data 70:30

Table 3. K-NN Performance Evaluation with 70:30 data split

Accuracy	Recall	Precision	F-1 Score
61.36%	47.74%	50.68%	49.17%
	70.12%	67.60%	68.84%

The Table 3 shows the evaluation results of the K-NN algorithm with a data split of 70:30, the model accuracy reaches 61.36%, meaning that KNN can correctly predict about 61% of the overall test data. Meanwhile, Recall values of 47.74% and 70.12% show how well the model detects positive cases (people at risk). Precision values of 50.68% and 67.60% indicate the level of accuracy of positive predictions made by the model. While F1-Score, which combines Recall and Precision, has values of 49.17% and 68.84%, indicating a balance between the model's ability to recognize positive cases and minimize prediction errors.

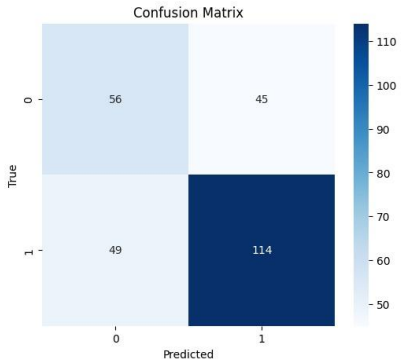

Figure 4. Confusion Matrix KNN with split data 80:20

Based on the confusion matrix in Figure 4, the results of testing the K-NN algorithm with a data split of 80:20 on this heart attack disease dataset successfully predicted 114 positive cases (people at risk of developing the disease) correctly, and 56 negative cases (people who are not at risk) correctly. However, 45 cases were incorrectly predicted to be positive, when in fact the person was not at risk (False Positive), and 49 cases that were incorrectly predicted to be negative, when in fact the person was at risk (False Negative). Further explanation of the performance evaluation of the KNN algorithm with an 80:20 data split is shown in Table 4.

Table 4. K-NN Performance Evaluation with 80:20 data split

Accuracy	Recall	Precision	F-1 Score
64.39%	55.45%	53.33%	54.37%
	69.94%	71.70%	70.81%

The Table 4 shows the results of the K-NN performance evaluation with a data split of 80:20, which has an accuracy of 64.39%, meaning that the model can correctly predict about 64% of the test data. Recall values reached 55.45% and 69.94%, indicating the model's ability to detect positive cases (people at risk). Meanwhile, Precision of 53.33% and 71.70% indicates the level of accuracy of positive predictions made by the model. F1-Score, which describes the balance between Recall and Precision, has values of 54.37% and 70.81%, indicating the model's performance in recognizing and minimizing positive case prediction errors.

3.4. Decision Tree

The test evaluation results of the Decision Tree algorithm with a confusion matrix can be seen in Figure 5 and Figure 6.

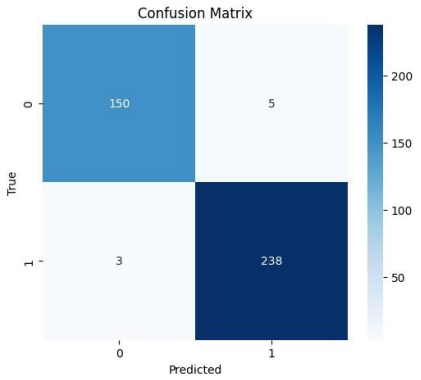

Figure 5. Confusion Matrix Decision Tree with split data 70:30

Based on the confusion matrix in Figure 5, the results of testing the Decision Tree algorithm on a heart attack disease dataset with a data division of 70:30. From these results, the Decision Tree successfully predicted 150 patient data that were not affected by heart attacks (True Negative) and 238 patient data that were affected by heart attacks (True Positive). However, there were 5 prediction errors where the patient was predicted to have a heart attack when they did not (False Positive) and 3 prediction errors where the patient had a heart attack but was predicted not to (False Negative). Further explanation of the performance evaluation of the Decision Tree algorithm with a 70:30 data split is shown in Table 5.

Table 5. Decision Tree Performance Evaluation with 70:30 data split

Accuracy	Recall	Precision	F-1 Score
97.98%	96.77%	98.04%	97.40%
	98.76%	97.94%	98.35%

The Table 5 shows the results of evaluating the performance of the Decision Tree with a 70:30 data split, resulting in an accuracy of about 97.98%, which means that almost all predictions are correct. With a recall of 96-98%, the model was able to identify most of the patients who were truly at risk of heart attack. A precision of 97-98% ensures that the model's positive predictions are generally accurate. The balanced combination of recall and precision resulted in an F1-score of about 97-98%, indicating the model is effective and reliable for predicting heart attack risk.

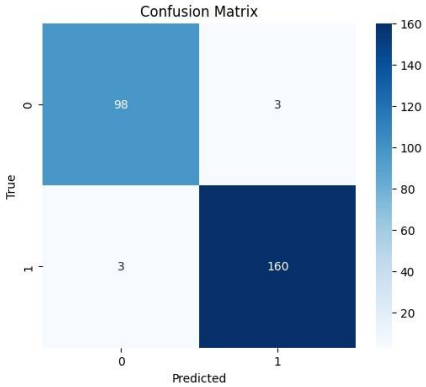

Figure 6. Confusion Matrix Decision Tree with split data 80:20

Based on the confusion matrix in Figure 6, the results of testing the Decision Tree algorithm on a heart attack disease dataset with a data division of 80:20. The model successfully predicted 98 patient data that were not affected by heart attacks (True Negative) and 160 patient data that were affected by heart attacks (True Positive). However, there were 3 prediction errors where patients were predicted to have a heart attack when they did not (False Positive) and another 3 errors where patients had a heart attack but were predicted not to (False Negative). Further explanation of the performance evaluation of the Decision Tree algorithm with an 80:20 data split is shown in Table 6.

Table 6. Decision Tree Performance Evaluation with 80:20 data split

Accuracy	Recall	Precision	F-1 Score
97.73%	97.03%	97.03%	97.03%
	98.16%	98.16%	98.16%

The Table 6 shows the performance evaluation results of the Decision Tree with a data split of 80:20, achieving an accuracy of about 97.73%, which means that almost all predictions it makes are correct. With recall and precision ranging from 97.03% to 98.16% respectively, this means the model was able to identify most of the truly high-risk patients while providing accurate positive predictions. The F1-score, which is also in the range of 97.03% to 98.16%, shows that the model has an optimal balance between recall and precision.

3.5. Naïve Bayes Classifier (NBC)

The test evaluation results of the Naïve Bayes algorithm with a confusion matrix can be seen in Figures 7 and 8.

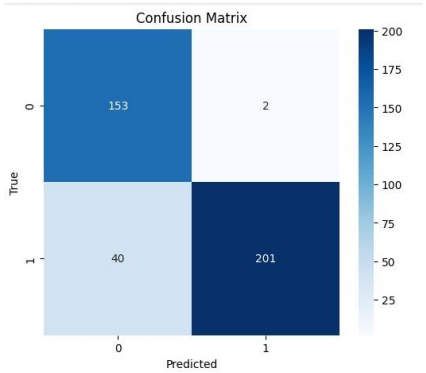

Figure 7. Confusion Matrix NBC with split data 70:30

Based on the confusion matrix in Figure 7, the results of testing the NBC algorithm on a heart attack disease dataset with a data division of 70:30. From these results, the model successfully predicts 153 patient data that are not affected by heart attacks (True Negative) and 201 patient data that are affected by heart attacks (True Positive). However, there were 2 mispredictions where patients were predicted to have a heart attack when they did not (False Positive) and 40 other errors where patients had a heart attack but were predicted not to (False Negative). Further explanation of the performance evaluation of the NBC algorithm with a 70:30 data split is shown in Table 7.

Table 7. NBC Performance Evaluation with 70:30 data split

Accuracy	Recall	Precision	F-1 Score
89.39%	98.71%	79.27%	87.93%
	83.40%	99.01%	90.54%

Table 7 shows the performance evaluation results of Naïve Bayes with a 70:30 data split, with an accuracy of about 89.39%, meaning that almost 90% of the model's predictions are correct. The model achieves a high recall of 98.71% in one class, indicating an excellent ability to identify high-risk patients. However, the recall in the other class is lower at 83.40%, suggesting potential issues such as data imbalance. Precision reached 99.01% for one class, meaning the positive predictions were highly accurate. F1-score values were in the range of 87.93% to 90.54%, reflecting a fairly good balance between recall and precision, although there is room for improvement in handling both classes more equally.

Based on the confusion matrix in Figure 8, the results of testing the NBC algorithm with an 80:20 split data on the heart attack disease dataset show that this algorithm is able to predict quite well. A total of 108 positive data (having a heart attack) were correctly predicted, while 100 negative data (not having a heart attack) were also correctly predicted. However, there were 55 False Negative prediction errors, where data that was positive was predicted as negative, as well as 1 False Positive prediction error. Further explanation of the performance evaluation of the NBC algorithm with an 80:20 data split is shown in Table 8.

The Table 8 shows the performance evaluation results of Naïve Bayes with 80:20 data split, the accuracy is about 78.79%, which means almost 79% of the model's predictions are correct. The model has a very high recall in one of the classes (99.01%), indicating its excellent ability to identify at-risk patients. Precision reached 99.08% for one class, indicating that the positive predictions were very accurate, but the

lower precision another class (64.52%) indicates some misclassification. F1-score values in the range of 78.12% to 79.41% reflect a fairly good balance between recall and precision.

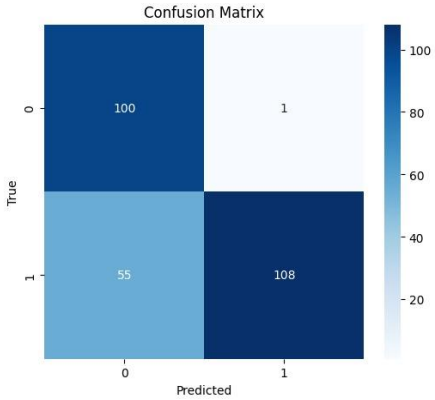

Figure 8. Confusion Matrix NBC with split data 80:20

Table 8. NBC Performance Evaluation with 80:20 data split

Accuracy	Recall	Precision	F-1 Score
78.79%	99.01%	64.52%	78.12%
Pred. Yes	66.26%	99.08%	79.41%

3.6. Random Forest

The test evaluation results of the Random Forest algorithm with a confusion matrix can be seen in Figure 9 and Figure 10.

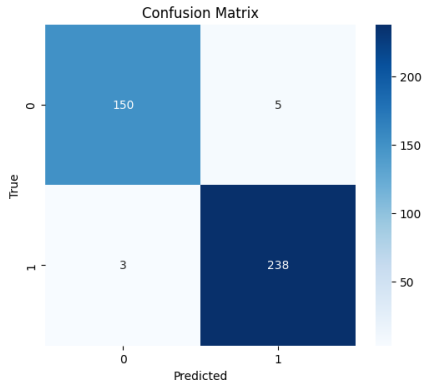

Figure 9. Confusion Matrix Random Forest with split data 70:30

Based on the confusion matrix in Figure 9, the results of testing the Random Forest algorithm on a heart attack disease dataset with a data division of 80:20. From the test results, the algorithm successfully predicted correctly as many as 150 cases for patients who were not affected by the disease (True Negative) and 238 cases for patients affected by the disease (True Positive). Meanwhile, there were 5 mispredictions where patients were affected but predicted not to be (False Positive), and 3 mispredictions where patients were not affected but predicted to be affected (False Negative). Further explanation of the performance evaluation of the Random Forest algorithm with a 70:30 data split is shown in Table 9.

Table 9. Random Forest Performance Evaluation with 70:30 data split

Accuracy	Recall	Precision	F1 Score
97.98%	96.77%	98.04%	97.40%
	98.76%	97.94%	98.35%

Table 9 shows the results of the Random Forest performance evaluation with a 70:30 data split, and the accuracy reached 97.98%. Recall of 96.77% and 98.76% reflects the model's ability to identify positive data well. Precision which ranges from 97.94% to 98.04% shows that the positive predictions made by the

model have a high level of accuracy. F1-Score with values of 97.40% and 98.35% shows a balance between recall and precision.

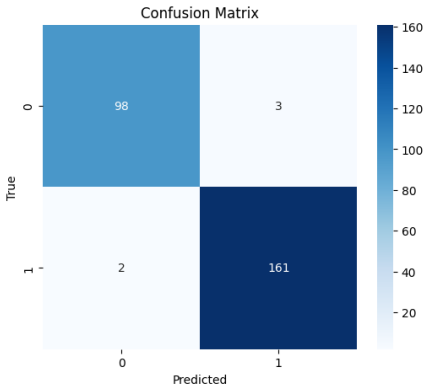

Figure 10. Confusion Matrix Random Forest with split data 80:20

Based on the confusion matrix in Figure 10, the results of testing the Random Forest algorithm on the heart attack disease dataset with a data division of 80:20. The algorithm managed to correctly predict 98 cases for patients who were not affected by the disease (True Negative) and 161 cases for patients affected by the disease (True Positive). Meanwhile, there were 3 mispredictions where patients were affected but predicted not to be (False Positive) and 2 mispredictions where patients were not affected but predicted to be affected (False Negative). Further explanation of the performance evaluation of the Random Forest algorithm with an 80:20 data split is shown in Table 10.

Table 10. Random Forest Performance Evaluation with 80:20 data split

Accuracy	Recall	Class Precision	F1 Score
97.98%	96.77%	98.04%	97.40%
	98.76%	97.94%	98.35%

The Table 10 shows the results of the Random Forest performance evaluation with 80:20 data split, the accuracy was 97.98% in the first test and 98.76% in the second test, which indicates a high level of prediction accuracy. Recall was obtained at 96.77% and 98.76%, indicating the ability of the model to detect positive cases correctly. Class Precision was 98.04% and 97.94% respectively, indicating an excellent level of precision of positive predictions. In addition, F1 Score was recorded at 97.40% and 98.35%, reflecting the balance between recall and precision.

3.7. Support Vector Machine (SVM)

The test evaluation results of the SVM algorithm with a confusion matrix can be seen in Figures 11 and 12.

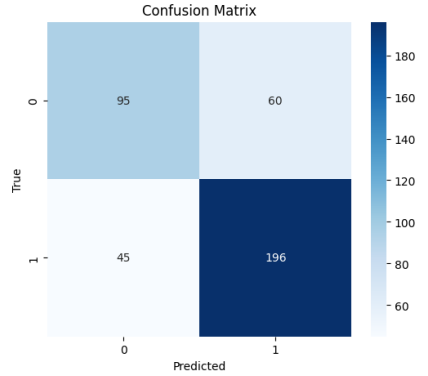

Figure 11. Confusion Matrix SVM with split data 70:30

Based on the confusion matrix in Figure 11, the results of testing the SVM algorithm on a heart attack disease dataset with a data division of 70:30. From the results of this test, the algorithm managed to correctly predict 95 cases for patients who were not affected by the disease (True Negative) and 196 cases for patients affected by the disease (True Positive). However, there were 60 mispredictions where patients were not

affected but were predicted to be (False Positive) and 45 mispredictions where patients were affected but were predicted not to be (False Negative). Further explanation of the performance evaluation of the SVM algorithm with a 70:30 data split is shown in Table 11.

Table 11. SVM Performance Evaluation with 70:30 data split

Accuracy	Recall	Class Precision	F1 Score
77.79%	61.29%	67.86%	64.41%
	81.33%	76.56%	78.87%

The Table 11 shows the results of the SVM performance evaluation with a 70:30 data split, accuracy with a value of 77.79% in the first test and 81.33% in the second test, which reflects a fairly good level of prediction accuracy. The recall was recorded at 61.29% and 81.33%, indicating the model's ability to detect positive cases with significant variation between the two tests. Class Precision was obtained at 67.86% and 76.56%, indicating the accuracy of the model's positive predictions. F1 Score was 64.41% and 78.87%, illustrating the balance between precision and recall.

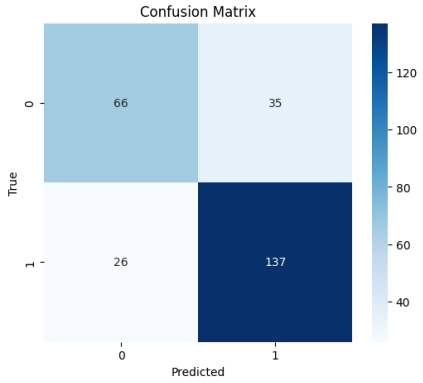

Figure 12. Confusion Matrix SVM with split data 80:20

Based on the confusion matrix Figure 12, the results of testing the SVM algorithm on the heart attack disease dataset with a data split of 80:20 for training and testing. From these results, the model successfully predicted 137 cases correctly as positive (experiencing the risk of disease) and 66 cases correctly as negative (not experiencing the risk of disease). However, there were 35 cases where the model incorrectly predicted a person as having a disease risk when they were not (False Positive) and 26 cases where the model incorrectly predicted a person as not having a disease risk when they were (False Negative).

Table 12. SVM Performance Evaluation with 80:20 data split

Accuracy	Recall	Class Precision	F1 Score
76.4%	65.35%	71.74%	68.39%
	84.05%	79.65%	81.79%

The Table 12 shows the results of the SVM performance evaluation with a data split of 80:20, the accuracy reached 76.4%, which indicates the level of accuracy of the model's prediction of the test data as a whole. The Recall value of 65.53% indicates the model's ability to recognize positive cases or patients who actually have heart attack disease. Meanwhile, the Class Precision of 71.74% illustrates the level of accuracy of the model's predictions when stating positivity. F1-Score of 68.39% shows the balance between Precision and Recall.

3.8. Comparative Analisys

In this section, we compare the performance of five algorithms K-NN, Decision Tree, Naïve Bayes, Random Forest, and SVM using 70:30 and 80:20 data splits. The evaluation includes accuracy, precision, recall, and F1-score, as presented in Table 13.

From the results, Decision Tree and Random Forest consistently achieved the highest accuracy, recall, and precision values, both above 97% in both data split scenarios. This superior performance reflects their capability to capture complex decision boundaries and handle non-linear relationships, which fits well with the heterogeneous patterns in the heart attack dataset. Naïve Bayes demonstrated very high recall above 98%, indicating strong sensitivity in detecting positive cases, but its lower precision in the 80:20 split 66.26% suggests a higher number of false positives due to the independence assumption among features. SVM

showed moderate but stable performance around 76–78% accuracy, which indicates good generalization but limited effectiveness when class boundaries overlap. In contrast, KNN consistently produced the lowest accuracy 61–64%, likely because of its sensitivity to feature scaling and the curse of dimensionality, making it less suitable for this dataset. The comparison graph of the accuracy values of the five algorithms can be seen in Figure 13.

Table 13. Comparison of Algorithm Performance

No.	Algorithm	Split Data	Recall		Precision	
			True No	True Yes	Pred. No	Pred. Yes
1	K-NN	70:30	47.74%	70.12%	50.68%	67.60%
		80:20	55.45%	69.94%	53.33%	71.70%
2	Decision Tree	70:30	96.77%	98.76%	98.04%	97.94%
		80:20	97.03%	98.16%	97.03%	98.16%
3	NBC	70:30	98.71%	83.40%	79.27%	99.01%
		80:20	99.01%	66.26%	64.52%	99.08%
4	Random Forest	70:30	97.03%	98.16%	97.03%	98.16%
		80:20	96.77%	98.76%	98.04%	97.94%
5	SVM	70:30	61.29%	81.33%	67.86%	76.56%
		80:20	65.35%	84.05%	71.74%	97.93%

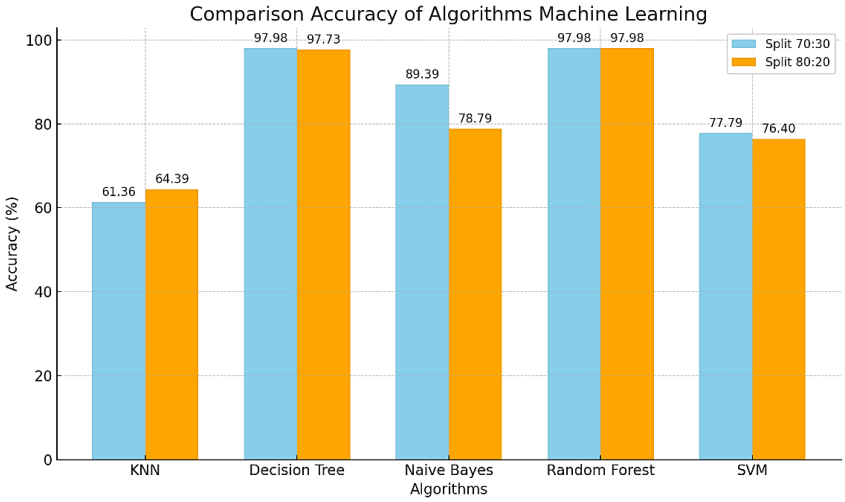

Figure 13. Comparison of Algorithm Accuracy Values

Figure 13 visually illustrates the accuracy distribution of all algorithms across two different data splits, highlighting key performance patterns. It can be observed that Decision Tree and Random Forest not only lead in accuracy but also maintain exceptional stability between the 70:30 and 80:20 splits, indicating strong robustness against variations in training data size. In contrast, Naïve Bayes shows a significant drop in accuracy when the training set is reduced, suggesting that its performance is sensitive to changes in data distribution due to its independence assumption.

The gap between K-NN and the other algorithms is also evident, emphasizing its poor adaptability in handling high-dimensional medical features. Meanwhile, SVM remains relatively stable, but its lower overall accuracy compared to tree-based models indicates that it may struggle with non-linear relationships present in the dataset. Overall, the visualization confirms the consistency of tree-based methods while exposing the vulnerability of instance-based and probabilistic models under different data splits. This reinforces the importance of selecting algorithms that can maintain reliability even with variations in dataset composition.

4. CONCLUSION

This study demonstrates the effectiveness of machine learning algorithms in predicting heart attack risk. Of the five algorithms tested K-NN, Decision Tree, Naïve Bayes Classifier, Random Forest, and SVM. Decision Tree and Random Forest algorithms showed the best performance with the highest accuracy, recall, precision, and F1-score at the 70:30 and 80:20 data split. Both algorithms proved to be very good at recognizing patterns in the dataset and predicting heart attack risk. Naïve Bayes and SVM gave fairly good results, where Naïve Bayes excelled in recall but showed variations in accuracy depending on the data split. Meanwhile, KNN had the lowest performance, making it less suitable for this dataset. This study shows that Decision Tree and Random Forest algorithms are the most effective tools for medical diagnosis, especially in

heart attack prediction. Further research can be conducted using larger datasets and additional features to improve prediction accuracy.

REFERENCES

[1] M. S. Iqbal, M. Adnan, S. E. G. Mohamed, and M. Tariq, "A hybrid deep learning framework for short-term load forecasting with improved data cleansing and preprocessing techniques," *Results Eng.*, vol. 24, no. November, p. 103560, 2024, doi: 10.1016/j.rineng.2024.103560.

[2] A. Tawakuli and T. Engel, "Make your data fair: A survey of data preprocessing techniques that address biases in data towards fair AI," *J. Eng. Res.*, no. July, 2024, doi: 10.1016/j.jer.2024.06.016.

[3] S. P. Patro, G. S. Nayak, and N. Padhy, "Heart disease prediction by using novel optimization algorithm: A supervised learning prospective," *Informatics Med. Unlocked*, vol. 26, 2021, doi: 10.1016/j.imu.2021.100696.

[4] S. Aziz, N. Afreen, F. Akram, and M. Ahmed, "A Framework for Cardiac Arrest Prediction via Application of Ensemble Learning Using Boosting Algorithms," *Procedia Comput. Sci.*, vol. 235, no. 2023, pp. 3293–3304, 2024, doi: 10.1016/j.procs.2024.04.311.

[5] M. Wang, X. Yao, and Y. Chen, "An Imbalanced-Data Processing Algorithm for the Prediction of Heart Attack in Stroke Patients," *IEEE Access*, vol. 9, pp. 25394–25404, 2021, doi: 10.1109/ACCESS.2021.3057693.

[6] G. Sugendran and S. Sujatha, "Earlier identification of heart disease using enhanced genetic algorithm and fuzzy weight based support vector machine algorithm," *Meas. Sensors*, vol. 28, no. May, p. 100814, 2023, doi: 10.1016/j.measen.2023.100814.

[7] M. W. Rasheed, A. Mahboob, and I. Hanif, "An estimation of physicochemical properties of heart attack treatment medicines by using molecular descriptor's," *South African J. Chem. Eng.*, vol. 45, no. April, pp. 20–29, 2023, doi: 10.1016/j.sajce.2023.04.003.

[8] J. Gamboa-Cruzado, R. Crisostomo-Castro, J. Vilabuleje, J. López-Goycochea, and J. N. Valenzuela, "Heart Attack Prediction Using Machine Learning: a Comprehensive Systematic Review and Bibliometric Analysis," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 5, pp. 1930–1944, 2024.

[9] S. S. Shijer, A. H. Jassim, L. A. Al-Haddad, and T. T. Abbas, "Evaluating electrical power yield of photovoltaic solar cells with k-Nearest neighbors: A machine learning statistical analysis approach," *e-Prime - Adv. Electr. Eng. Electron. Energy*, vol. 9, no. July, p. 100674, 2024, doi: 10.1016/j.prime.2024.100674.

[10] M. Ozcan and S. Peker, "A classification and regression tree algorithm for heart disease modeling and prediction," *Healthc. Anal.*, vol. 3, no. December 2022, p. 100130, 2023, doi: 10.1016/j.health.2022.100130.

[11] N. Gul, W. K. Mashwani, M. Aamir, S. Aldahmani, and Z. Khan, "Optimal model selection for k-nearest neighbours ensemble via sub-bagging and sub-sampling with feature weighting," *Alexandria Eng. J.*, vol. 72, pp. 157–168, 2023, doi: 10.1016/j.aej.2023.03.075.

[12] W. J. Sari *et al.*, "Performance Comparison of Random Forest, Support Vector Machine and Neural Network in Health Classification of Stroke Patients," *Public Res. J. Eng. Data Technol. Comput. Sci.*, vol. 2, no. 1, pp. 34–43, 2024, doi: 10.57152/predatecs.v2i1.1119.

[13] M. R. Anugrah, N. A. Al-Qadr, N. Nazira, and N. Ihza, "Implementation of C4.5 and Support Vector Machine (SVM) Algorithm for Classification of Coronary Heart Disease," *Public Res. J. Eng. Data Technol. Comput. Sci.*, vol. 1, no. 1, pp. 20–25, 2023, doi: 10.57152/predatecs.v1i1.805.

[14] H. Hidayat, A. Sunyoto, and H. Al Fatta, "Klasifikasi Penyakit Jantung Menggunakan Random Forest Clasifier," *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, vol. 7, no. 1, pp. 31–40, 2023, doi: 10.47970/siskom-kb.v7i1.464.

[15] Y. H. Shakir, E. Aziz, A. Al, and A. Alkhazraji, "Leveraging Machine Learning for Early Risk Prediction in Cirrhosis Outcome Patients," vol. 3, no. July, pp. 22–30, 2025.

[16] S. Lee, C. Lee, K. G. Mun, and D. Kim, "Decision Tree Algorithm Considering Distances between Classes," *IEEE Access*, vol. 10, no. April, pp. 69750–69756, 2022, doi: 10.1109/ACCESS.2022.3187172.

[17] A. F. Lubis *et al.*, "Classification of Diabetes Mellitus Sufferers Eating Patterns Using K-Nearest Neighbors, Naïve Bayes and Decission Tree," *Public Res. J. Eng. Data Technol. Comput. Sci.*, vol. 2, no. 1, pp. 44–51, 2024, doi: 10.57152/predatecs.v2i1.1103.

[18] B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *J. Appl. Sci. Technol. Trends*, vol. 2, no. 01, pp. 20–28, 2021, doi: 10.38094/jastt20165.

[19] T. Kim and J. S. Lee, "Exponential Loss Minimization for Learning Weighted Naive Bayes Classifiers," *IEEE Access*, vol. 10, pp. 22724–22736, 2022, doi: 10.1109/ACCESS.2022.3155231.

[20] M. Libnao, M. Misula, C. Andres, J. Mariñas, and A. Fabregas, "Traffic incident prediction and classification system using naïve bayes algorithm," *Procedia Comput. Sci.*, vol. 227, pp. 316–325, 2023, doi: 10.1016/j.procs.2023.10.530.

[21] M. Artur, "Review the performance of the Bernoulli Naïve Bayes Classifier in Intrusion Detection Systems using Recursive Feature Elimination with Cross-validated selection of the best number of features," *Procedia Comput. Sci.*, vol. 190, no. 2019, pp. 564–570, 2021, doi: 10.1016/j.procs.2021.06.066.

[22] O. Peretz, M. Koren, and O. Koren, "Naive Bayes classifier – An ensemble procedure for recall and precision enrichment," *Eng. Appl. Artif. Intell.*, vol. 136, no. PB, p. 108972, 2024, doi: 10.1016/j.engappai.2024.108972.

[23] W. Putri, D. Hastari, K. U. Faizah, S. Rohimah, and D. Safira, "Implementation of Naïve Bayes Classifier for Classifying Alzheimer's Disease Using the K-Means Clustering Data Sharing Technique," *Public Res. J. Eng. Data Technol. Comput. Sci.*, vol. 1, no. 1, pp. 47–54, 2023, doi: 10.57152/predatecs.v1i1.803.

[24] A. Tariq *et al.*, "Modelling, mapping and monitoring of forest cover changes, using support vector machine, kernel logistic regression and naive bayes tree models with optical remote sensing data," *Heliyon*, vol. 9, no. 2, p. e13212, 2023, doi: 10.1016/j.heliyon.2023.e13212.

[25] K. Maxwell, M. Rajabi, J. Esterle, M. Tivane, and D. Travassos, "Spatial modelling and classification of altered coal using random forest-based methods at Moatize Basin, Mozambique," *J. African Earth Sci.*, vol. 215, no. March, p. 105279, 2024, doi: 10.1016/j.jafrearsci.2024.105279.

[26] P. F. Pratama, D. Rahmadani, R. S. Nahampun, D. Harmutika, A. Rahmadeyan, and M. F. Evizal, "Random Forest Optimization Using Particle Swarm Optimization for Diabetes Classification," *Public Res. J. Eng. Data Technol. Comput. Sci.*, vol. 1, no. 1, pp. 41–46, 2023, doi: 10.57152/predatecs.v1i1.809.

[27] E. Asamoah, G. B. M. Heuvelink, I. Chairi, P. S. Bindraban, and V. Logah, "Random forest machine learning for maize yield and agronomic efficiency prediction in Ghana," *Heliyon*, vol. 10, no. 17, p. e37065, 2024, doi: 10.1016/j.heliyon.2024.e37065.

[28] P. Josso, A. Hall, C. Williams, T. Le Bas, P. Lusty, and B. Murton, "Application of random-forest machine learning algorithm for mineral predictive mapping of Fe-Mn crusts in the World Ocean," *Ore Geol. Rev.*, vol. 162, no. September, p. 105671, 2023, doi: 10.1016/j.oregeorev.2023.105671.

[29] M. Muta'alimah, C. K. Zarry, A. Kurniawan, H. Hasysya, M. F. Firas, and N. Nadhirah, "Classifications of Offline Shopping Trends and Patterns with Machine Learning Algorithms," *Public Res. J. Eng. Data Technol. Comput. Sci.*, vol. 2, no. 1, pp. 18–25, 2024, doi: 10.57152/predatecs.v2i1.1099.

[30] M. Wahba, R. Essam, M. El-Rawy, N. Al-Arifi, F. Abdalla, and W. M. Elsadek, "Forecasting of flash flood susceptibility mapping using random forest regression model and geographic information systems," *Heliyon*, vol. 10, no. 13, p. e33982, 2024, doi: 10.1016/j.heliyon.2024.e33982.

[31] Z. Liu *et al.*, "Enhancing XRF sensor-based sorting of porphyritic copper ore using particle swarm optimization-support vector machine (PSO-SVM) algorithm," *Int. J. Min. Sci. Technol.*, vol. 34, no. 4, pp. 545–556, 2024, doi: 10.1016/j.ijmst.2024.04.002.

[32] R. Krishna and S. Hemamalini, "Improved TLBO algorithm for optimal energy management in a hybrid microgrid with support vector machine-based forecasting of uncertain parameters," *Results Eng.*, vol. 24, no. July, p. 102992, 2024, doi: 10.1016/j.rineng.2024.102992.

[33] V. Asadpour, E. J. Puttock, D. Getahun, M. J. Fassett, and F. Xie, "Automated placental abruption identification using semantic segmentation, quantitative features, SVM, ensemble and multi-path CNN," *Heliyon*, vol. 9, no. 2, p. e13577, 2023, doi: 10.1016/j.heliyon.2023.e13577.

[34] M. L. De Klerk and A. K. Saha, "Performance analysis of DTC-SVM in a complete traction motor control mechanism for a battery electric vehicle," *Heliyon*, vol. 8, no. 4, p. e09265, 2022, doi: 10.1016/j.heliyon.2022.e09265.

[35] S. Zhao, X. Liang, L. Wang, H. Zhang, G. Li, and J. Chen, "A fault diagnosis method for analog circuits based on EEMD-PSO-SVM," *Heliyon*, vol. 10, no. 18, p. e38064, 2024, doi: 10.1016/j.heliyon.2024.e38064.