Jurnal Kecerdasan Buatan dan Teknologi Informasi
Vol. 5, No. 2, May 2026, pp. 371-387
e-ISSN: 2964-2922, p-ISSN: 2963-6191
DOI: https://doi.org/10.69916/jkbti.v5i2.423

A MACHINE LEARNING APPROACH FOR PREDICTING STUNTING RISK IN TODDLERS

Ekatri Yulisara[*1], Inggih Permana[2], Febi Nur Salisah[3], M. Afdal[4], Medyantiwi Rahmawita[5]

[1,2,3,4,5]Information Systems Study Program, Faculty of Science and Technology, Sultan Syarif Kasim Riau Islamic University, Pekanbaru, Indonesia
Email: [1]12250325372@students.uin-suska.ac.id, [2]inggihpermana@uin-suska.ac.id, [3]febinursalisah@uin-suska.ac.id, [4]m.afdal@uin-suska.ac.id, [5]medyantiwi.rahmawita@uin-suska.ac.id

(Received: January 16, 2026; Revised: May 3, 2026; Published: May 31, 2026)

Abstract

Stunting is a chronic nutritional problem that remains a major public health challenge, particularly in developing countries such as Indonesia. It results from long-term nutritional deficiencies and can negatively affect physical growth, cognitive development, educational achievement, and future productivity. Early detection of stunting risk is essential to support timely intervention and improve child health outcomes. This study aims to develop and compare the performance of several machine learning algorithms for predicting stunting risk in toddlers using a large-scale nutritional dataset. The dataset was obtained from the Kaggle repository entitled *"Stunting Balita Detection (121K Rows)"* and consists of 120,999 records containing age, gender, height, and nutritional status information. Data preprocessing included categorical data encoding, Min-Max normalization, and dataset partitioning into training and testing sets using an 80:20 ratio. Five classification algorithms were evaluated: K-Nearest Neighbor (KNN), Random Forest, Support Vector Machine (SVM), Naïve Bayes, and Decision Tree C4.5. Model performance was measured using confusion matrix analysis, accuracy, precision, recall, and F1-score. The experimental results showed that KNN achieved the highest performance with an accuracy of 99.94%, precision of 99.90%, recall of 99.93%, and F1-score of 99.92%. Random Forest achieved comparable results with an accuracy of 99.93%, while SVM, Decision Tree C4.5, and Naïve Bayes produced lower performance values. These findings indicate that KNN and Random Forest are highly effective for stunting risk classification and have strong potential to support intelligent decision-support systems for early detection and nutritional monitoring of toddlers.

Keywords: stunting prediction; machine learning; classification algorithm; toddler nutritional status; prediction model.

1. INTRODUCTION

Stunting is one of the chronic nutritional problems that remains a strategic issue in public health, especially in developing countries such as Indonesia[1], [2]. Stunting occurs as a result of long-term malnutrition, especially during the first 1,000 days of life, which causes linear growth disorders in children. The impact of stunting is not limited to physical aspects, but also affects cognitive development, learning abilities, productivity in adulthood, and increases the risk of non-communicable diseases later in life[3]. Therefore, stunting is a multidimensional problem that requires serious attention from various parties, including the government, health workers, and researchers[4].

Various intervention programs have been implemented to reduce the prevalence of stunting, but these efforts still face a number of obstacles, one of which is the limitation in the process of early detection of stunting risk accurately and efficiently[5], [6]. Early detection is crucial because it allows for preventive and remedial measures to be taken before stunting becomes more severe and permanent[7]. In practice, the process of identifying stunting risks still relies heavily on conventional methods, which often take a long time and are prone to errors in data analysis, especially when the amount of data used is large[8].

Advances in information technology and the increasing availability of large-scale health data provide opportunities to utilize data-driven approaches to support decision-making in the health sector[9]. One widely used approach is machine learning, which is capable of learning complex patterns and relationships from historical data to produce accurate predictive models[10]. The application of machine learning in healthcare has shown significant potential, including in disease classification, health risk prediction, and nutritional status analysis in children. By utilizing appropriate classification algorithms, machine learning can help identify the risk of stunting more objectively, quickly, and consistently[11], [12].

Previous studies have examined the use of machine learning algorithms for classifying nutritional status and stunting in toddlers, using various algorithms such as Decision Tree, Naive Bayes, Support Vector Machine, K-Nearest Neighbor, and Random Forest[13], [14]. However, the results show differences in performance between algorithms, which are influenced by data characteristics, sample size, and the evaluation method used[15]. This indicates that there is no single algorithm that consistently provides the best results, so comparative research is needed to determine the most optimal algorithm for a particular dataset[16].

Based on this background, this study aims to develop and compare the performance of several machine learning algorithms in predicting the risk of stunting in toddlers using the Toddler Stunting Detection dataset[17]. The algorithms used in this study include Random Forest, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naive Bayes, and Decision Tree C4.5[18], [19]. The research process was carried out through the stages of data preprocessing, training and testing data division, data normalization, and model training and testing. Model performance was evaluated using accuracy, precision, recall, and F1-score metrics to obtain a comprehensive overview of performance.

The results show that the K-Nearest Neighbor (KNN) algorithm achieved the highest accuracy of 99.94%, followed by Random Forest with an accuracy of 99.93%, while other algorithms such as SVM, Decision Tree, and Naïve Bayes produced relatively lower accuracy values. These findings indicate that KNN and Random Forest provide highly accurate and consistent performance in classifying stunting data, the results of this study are expected to contribute scientifically to the selection of appropriate machine learning algorithms and serve as a basis for the development of data-based decision support systems for early detection of stunting risk[20].

2. RESEARCH METHODOLOGY

The research process consists of the Planning Stage, Data Collection, Data Preprocessing, Machine Learning Modeling, and Model Evaluation and Selection. The methodological flow for this research can be seen in Figure 1.

Figure 1. Research Methodology Flowchart

2.1. Planning Stage

a. Problem Identification
The high rate of stunting and manual early detection result in low prediction accuracy. Previous studies have been limited to a small number of algorithms and small datasets. Therefore, this study compares five classification algorithms using a large dataset of 121K to obtain a more accurate stunting prediction model.

b. Goal Setting
This study aims to produce an accurate stunting risk prediction model by comparing five machine learning algorithms and determining the best model based on accuracy, precision, recall, and F1-score.

c. Defining the Scope of the Problem
The study used Kaggle data with variables of age, gender, and height. The analysis was limited to five algorithms without further optimization, with evaluation using accuracy, precision, recall, and F1-score.

d. Literature Review
The literature review examines the concept of stunting and previous machine learning research, which generally uses limited algorithms and small datasets. This forms the basis for the novelty of this research through the comparison of five algorithms on a large dataset.

2.2. Data Collection

The research dataset uses public data from Kaggle, "Stunting Balita Detection (121K Rows)," which was selected because of its large size and variety. The variables used include age, gender, height, and nutritional status as the target (normal, stunted, severely stunted, and tall). This dataset was processed through preprocessing before being used for machine learning model training.

2.3. Classification Modeling

The classification modeling stage is an important process in this study because it aims to build a machine learning model that can accurately predict the risk of stunting in toddlers. The modeling process is carried out through several systematic steps, starting from dataset processing to evaluation of the model classification results, as shown in Figure 2.

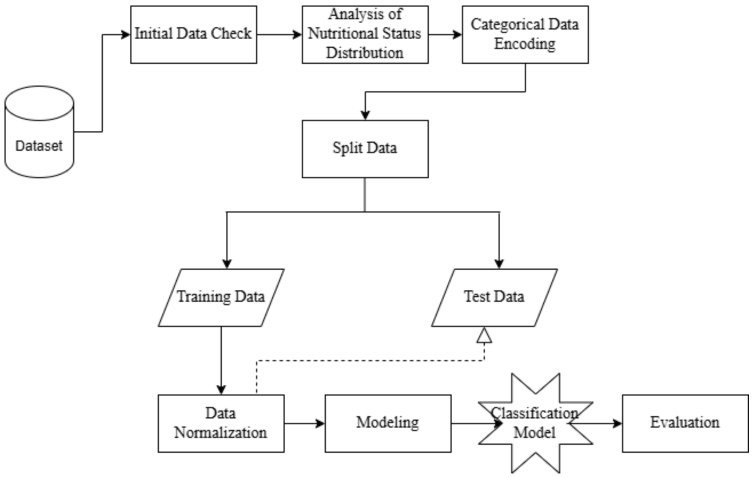

Figure 2. Data Procedure Modeling

a. Dataset
This study uses the public dataset "Stunting Balita Detection (121K Rows)" from Kaggle, which contains information on age, gender, height, and nutritional status labels (Normal, High, Stunted, and Severely Stunted). This dataset was chosen because it has a large amount of data and has been proven relevant in previous studies.

b. Preprocessing
The preprocessing stage involves data normalization using Min-Max Normalization and data separation using the hold-out method into 80% training data and 20% test dataMin-Max Normalization was chosen because it transforms all numerical feature values into a uniform scale within the range of 0 to 1, which is essential for distance-based and gradient-based algorithms such as KNN and SVM. Without normalization, features with larger numerical ranges (such as height) could dominate smaller-scale features (such as age), leading to biased model learningIn addition, Min-Max normalization helps improve computational efficiency, accelerates model convergence, and ensures that each feature contributes proportionally to the model training process. Therefore, this method is considered suitable for improving model performance and stability in this study.

c. Model Development
Five machine learning models were used in this study, namely Decision Tree C4.5, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest, and Naïve Bayes. The main parameters tested included:

1. C4.5: criterion = entropy, min samples leaf = 10–50, random state = 42
2. SVM: kernel = linear, polynomial, RBF, sigmoid
3. KNN: K = 1, 3, 5, 7
4. Random Forest: n_estimators = 10–500, criterion = gini, random state = 42
5. Naïve Bayes: var_smoothing = 1e−9

d. Model Evaluation

Performance evaluation was conducted using a confusion matrix and four key metrics, namely accuracy, precision, recall, and F1-score to assess the effectiveness of the model in predicting the risk of stunting.

2.4. Best Model Contest

The final stage is selecting the best model based on evaluation results using accuracy, precision, recall, and F1-score. Models are selected based on the most accurate and consistent predictive capabilities, while also considering complexity, computation time, and interpretability. This best model is expected to be an effective analytical tool for stunting detection and used in further research and decision making.

3. RESULTS AND DISCUSSION

3.1. Data Collection Process

This study uses a dataset on the nutritional status of toddlers consisting of four variables, namely: age (months), gender, height (cm), and nutritional status. This dataset is original data available on the Kaggle platform. This dataset contains information on 120,999 toddlers, with nutritional status categorized into four classes, namely normal, high, stunted, and severely stunted. This dataset is used to help predict whether a toddler is in the good nutrition category or at risk of stunting based on age, gender, and height. Table 1 below shows the Dataset Attribute List:

Table 1. List of Dataset Attributes

No	Attribute Name	Data Type	Description
1	Age (months)	Numeric (Ratio)	The age of the toddler in months, used to determine child growth standards.
2	Gender	Categorical (Nominal)	Sex of the infant (Male/Female).
3	Height (cm)	Numeric (Ratio)	Height or length of the infant measured in centimeters.
4	Nutritional Status	Categorical (Ordinal)	Label of the child's nutritional status (Normal, Short, Very Short) based on the WHO Z-score..

This research dataset contains four main attributes used to support the analysis of toddler nutritional status. The Age (months) attribute represents the age of the toddler in months and forms the basis for assessing growth according to WHO standards. The Gender attribute records whether the toddler is male or female, which is important because growth standards differ for each gender. The Height (cm) attribute contains the results of measuring the height or length of the toddler in centimeters as the main indicator for determining growth conditions. Meanwhile, the Nutritional Status attribute serves as a target variable that indicates the child's growth category, namely normal, short, or very short based on the WHO Z-score value. These four attributes provide a comprehensive picture of the toddler's condition and form the basis for the analysis and prediction of stunting status. Table 2 below shows the distribution of the Age (months) variable:

Table 2. Distribution of Age Variable (months)

Age (months)	Number
0	2.999
1	2.000
2	2.000
3	2.000
…	…
57	2.000
58	2.000
59	2.000
Total	120.999

The age distribution of toddlers in the dataset shows a relatively stable distribution pattern in the early age range, namely 0 to 3 months. The 0-month age group has the highest number, namely 2,999 toddlers, followed by 1, 2, and 3 months, each with 2,000 toddlers. In the next age range, the data is summarized using ellipsis (...) because the distribution pattern remains consistent. Towards the end of the age range, namely at ages 57 to 59 months, the data is again displayed in full with 2,000 toddlers each. Overall, the total data for the age variable reached 120,999 infants. This relatively even distribution shows that the age variable is well represented across the entire age range, supporting the quality of analysis and modeling in the next stage. Table 3 below shows the distribution of the Gender Variable:

Table 3. Distribution of Gender Variables

Gender	Number
Male	59.997
Female	61.002
Total	120.999

The distribution of gender variables in the dataset shows a relatively balanced proportion between male and female toddlers. Based on the data, the number of male toddlers reached 59,997 children, while female toddlers numbered 61,002 children. Overall, this dataset contains 120,999 toddler records. The balanced distribution between the two gender groups provides a good basis for further analysis, as there is no significant dominance of either category that could potentially cause bias in the modeling process. Table 4 below shows the distribution of the Height Variable (Categories):

Table 4. Distribution of Height Variables (Categories)

Height Category	**Range (cm)**	**Number of Records**
Very Low	<50 cm	2.037
Low	50-55 cm	2.367
Medium	55-65 cm	7.709
Tall	> 65 cm	108.886
Total	-	120.999

The distribution of categorized toddler heights shows that the tallest group is the most numerous, with 108,886 data points. Below that is the medium category with 7,709 data points, followed by the short category with 2,367 data points. Meanwhile, the very short category has the fewest data points, with 2,037. Overall, the total data on the height variable amounts to 120,999. The difference in numbers in each category shows that there is a wide variation in height in the dataset, which can provide a more complete picture in the analysis process. Table 5 below shows the distribution of the Nutritional Status variable:

Table 5. Distribution of Nutritional Status Variables

No	**Class**	**Number of Data**
1.	Normal	67.755
2.	High	19.560
3.	Stunted	13.815
4.	Severely Stunted	19.869
Total		**120.999**

3.2. Data Preprocessing

a. Initial Data Check

The initial data inspection stage is carried out to examine the structure, attributes, and sample contents of the dataset before data cleaning and transformation processes are performed. The dataset used is data collected on toddlers with attributes of age, gender, height, and nutritional status. This dataset is read using the pd.read_csv() command via Google Collab, with the file name data_balita.csv. The results of reading the first few rows of the dataset are shown in Table 6.

Table 6. Initial data sample of the dataset

Age (Month)	**Gender**	**Height(cm)**	**Nutritional Status**
0	Male	44.59	Stunted
1	Male	56.70	Height
2	Male	46.86	Normal
3	Male	47.51	Normal
4	Male	42.74	Severely stunted

Table 6 shows some of the initial data from the dataset used in this study. The data shows several key variables that form the basis of the analysis, namely toddler age (in months), gender, height, and nutritional status as output labels. Based on the sample data shown, it can be seen that each entry has different characteristics, both in terms of age and height, which then affect the categorization of nutritional status, such as Normal, High, Stunted, and Severely Stunted. This sample provides an initial overview of the structure and content of the dataset, and shows that the data used has sufficient variation to support the modeling and analysis of toddler nutritional status in the next stage of the research.

b. Analysis of Nutritional Status Distribution

The next step after examining the dataset structure is to analyze the data distribution on the target attribute, namely Nutritional Status. This analysis is performed to determine the distribution of data in each class category contained in the dataset. The results of executing the nutritional status class distribution command can be seen in Table 7.

Table 7. Distribution of Nutritional Status Classes

Nutritional Status	**Number of Data**
Normal	67.755
High	19.560
Saveraly stunted	19.869
Stunted	13.815
Total	120.999

Based on the data presented in Table 7, there is clearly a significant imbalance (class imbalance) in the distribution of the target variable, namely Nutritional Status. Specifically, the Normal class dominates the dataset with 67,755 samples, while the Stunted class has only 13,815 samples, making it the minority class.This imbalance indicates that the dataset is not evenly distributed across all classes, which can significantly affect the performance of machine learning models. Algorithms tend to bias predictions toward the majority class, resulting in high accuracy but poor performance in detecting minority classes, particularly in metrics such as recall and F1-score.In the context of stunting prediction, this issue is critical because misclassification of minority classes (such as Stunted) can lead to serious consequences in real-world applications, where early detection is essential.Therefore, handling class imbalance becomes an important consideration in this study. Although this research focuses on model comparison, the presence of imbalance highlights the need for future improvements, such as applying resampling techniques (oversampling or undersampling) or using class-weighted algorithms to enhance model fairness and sensitivity toward minority classes.

c. Categorical Data Encoding

This step is performed to convert categorical data into numerical form so that it can be processed by machine learning algorithms. Most classification algorithms cannot read data in text or categorical form, so it is necessary to perform encoding on attributes that have categorical data types. In the dataset used, there are two categorical attributes, namely Gender and Nutritional Status. The encoding results can be seen in Table 8.

Table 8. Categorical Data Encoding Results

Age (Month)	**Gender**	**Height (cm)**	**Nutritional Status**
0	0	44.599	1
0	0	56.70	3
0	0	46.86	2
0	0	47.51	2
…	…	…	…
0	0	42.74	0

From these results, it can be explained that:

The Gender attribute is coded into numerical values with the following conditions:

1. male → 0
2. female → 1

The Nutritional Status attribute is coded into numerical values with the following conditions:

1. severely stunted → 0
2. stunted → 1
3. normal → 2
4. tall → 3

This process aims to simplify the classification model in reading and processing data, because machine learning models require numerical representations to perform training and prediction processes. Thus, all data in the dataset is now in numerical format and ready to be used for the model training stage.

d. Training Data and Test Data Distribution

After the categorical data encoding process is complete, the next step is to divide the dataset into training data and testing data. The purpose of this division is so that the machine learning model can be trained using some of the data, then tested on other data that has never been seen before to assess the model's generalization ability.

e. Data Normalization

After the data is divided into training data and test data, the next step is to normalize the feature data. Normalization aims to equalize the scale between numerical variables so that no feature dominates the model training process. This is important because algorithms such as KNN, SVM, and Neural Network are sensitive to differences in scale between features. In this study, normalization was performed using Min-Max Scaler from the scikit-learn library. This method converts the value range of each feature into the interval [0, 1], so that the smallest value in the feature becomes 0 and the largest value becomes 1. The results of training data normalization can be seen in Table 10.

Table 10. Training Data Normalization Results

Age (Month)	Gender	Height
0.18333	0.0	0.272553
0.216667	0.0	0.449868
0.250000	1.0	0.315745
0.533333	1.0	0.514656
0.566667	1.0	0.443048

Table 10 shows the results of training data normalization using the Min-Max Scaler method, so that all values in the Age (Month), Gender, and Height features are in the range of 0 to 1. The values shown in the table are decimal numbers which are the result of scaling the original data. This process is carried out to equalize the scale between features so that no variable dominates during model training, allowing the normalized data to be used optimally in the next modeling stage. The results of the training data normalization can be seen in Table 11.

Table 11. Test Data Normalization Results

Age (Month)	Gender	Height
0.18333	1.0	0.341888
0.050000	0.0	0.288466
0.316667	0.0	0.373713
0.750000	0.0	0.557848
0.983333	0.0	0.613544

Table 11 shows the results of test data normalization using the Min-Max Scaler method. In this table, all values for the Age (Months), Gender, and Height features have been mapped to a range of 0 to 1. The decimal values shown are the result of the transformation of the original data so that each feature has a uniform scale. This normalization process aims to ensure that no variable has a larger scale and dominates the model testing process. Thus, the normalized test data can be used optimally for model performance evaluation in the next stage.

3.3. Classification Model Results

a. K-Nearest Neighbor (KNN)

The performance evaluation of the KNN algorithm in this study was conducted by manually testing several values of the K parameter (number of nearest neighbors). Each K value was tested by building a separate KNN model, training the model using normalized training data, and then evaluating it on the test data. Each model was tested using the Accuracy, Precision, Recall, and F1-score metrics to obtain an overview of the performance produced at each K value. The detailed results of this test can be seen in Table 12.

Table 12. Performance evaluation results of the K-Nearest Neighbor (KNN) algorithm

Algorithm	Parameter Value (K)	Accuracy (CA)	F1-Score	Precision	Recall

K- Nearest Neighbor (KNN)	**1**	**0.999 (99.9%)**	**0.999 (99.9%)**	**0.999 (99.9%)**	**0.999 (99.9%)**
	3	0.998 (99.8%)	0.998. (99.8%)	0.998 (99.8%)	0.998 (99.8%)
	5	0.996 (99.6%)	0.996 (99.6%)	0.996 (99.6%)	0.996 (99.6%)
	7	0.995 (99.5%)	0.995 (99.5%)	0.995 (99.5%)	0.995 (99.5%)

Based on the performance evaluation results shown in Table 12, the KNN algorithm with a parameter value of K = 1 provides the best performance compared to other K values. The model with K = 1 is able to achieve an accuracy of 0.999 or 99.9%, which indicates a very high level of prediction accuracy in classifying data. The F1-Score, Precision, and Recall values are also at 0.999 (99.9%), indicating that the model has an excellent balance between the ability to detect positive classes, avoid prediction errors in negative classes, and maintain overall performance consistency. These results show that the K = 1 configuration works optimally with a very low error rate, so it can be categorized as the model with the best performance compared to other K configurations. The KNN Confusion Matrix with the best values can be seen in Table 13.

Tabel 13. Confusion Matrix K-Nearest Neighbor (KNN)

Confusion Matrix	**True Normal**	**True High**	**True Stunted**	**True Saverely Stunted**
Pred Normal	4129	1	0	0
Pred High	6	2782	2	0
Pred Stunted	0	3	13374	5
Pred Saverely Stunted	0	0	5	3893

Based on the evaluation results through the Confusion Matrix, it can be concluded that the KNN algorithm shows superior classification performance and is close to perfect in all nutritional status categories. The model is able to distinguish each class with a very high level of accuracy, both in minority and majority classes.

1. Normal Class: In this class, the model successfully classified 4,129 data correctly, with only 1 data incorrectly predicted as High class. This very small error indicates that the model has a strong ability to recognize the characteristics of data in the Normal category.
2. High Class: The High Class also showed excellent classification results, where 2,782 data points were correctly predicted, accompanied by 6 data points that were misclassified as Normal and 2 data points that were misclassified as Stunted. The minimal number of errors reflects that the model remains stable even when handling classes with lower frequency of occurrence.
3. Stunted Class: As the class with the largest amount of data, the model's performance remains highly optimal. A total of 13,374 data points were classified correctly, with only 3 data points incorrectly predicted as High and 5 data points incorrectly predicted as Severely Stunted. The consistency of predictions in this class demonstrates the model's excellent generalization ability on majority data.
4. Severely Stunted Class: In this class, the model produced 3,893 correct predictions, with only 5 data points incorrectly classified as Stunted. The very low error rate reinforces the evidence that KNN is capable of accurately recognizing patterns in extreme classes.

Overall, the Confusion Matrix shows that the KNN algorithm has a very high classification ability. The minimal prediction errors indicate that the model is not only accurate, but also stable and consistent in recognizing data patterns across all nutritional status categories. This is reinforced by evaluation metrics such as Accuracy, Precision, Recall, and F1-Score, which are in the range of 99%, proving the model's highly reliable performance.

b. Random Forest

The performance evaluation of the Random Forest classification model was carried out through a process of testing the main parameter, namely the number of decision trees (n_estimators). This test aimed to determine the model's performance in producing accurate and stable predictions for the test data. The Random Forest algorithm combines the prediction results from a number of decision trees to minimize errors and improve model generalization. The evaluation was conducted comprehensively using the Accuracy, Precision, Recall, and F1-score metrics to assess the overall performance of the model. The detailed results of this test can be seen in Table 14.

Table 14. Performance evaluation results of the Random Forest algorithm

Algorithm	Parameter Value (n_estimators)	Accuracy (CA)	F1-Score	Precision	Recall
Random Forest	10	0.999 (99.9%)	0.999 (99.9%)	0.999 (99.9%)	0.999 (99.9%)
	50	0.999 (99.9%)	0.999 (99.9%)	0.999 (99.9%)	0.999 (99.9%)
	100	0.999 (99.9%)	0.999 (99.9%)	0.999 (99.9%)	0.999 (99.9%)
	200	0.999 (99.9%)	0.999 (99.9%)	0.999 (99.9%)	0.999 (99.9%)
	500	0.999 (99.9%)	0.999 (99.9%)	0.999 (99.9%)	0.999 (99.9%)

Based on the evaluation results shown in Table 14, the Random Forest algorithm shows superior and stable performance across all variations of the number of decision trees (n_estimators) parameter. Interestingly, whether using 10, 50, 100, 200, or even 500 trees, the model consistently achieved Accuracy, Precision, Recall, and F1-Score values of 0.999 (99.9%). This consistency confirms that the model is able to recognize data patterns very well, even when the number of trees used is relatively small. In addition, the stable values across all evaluation metrics also indicate that adding more trees does not significantly change the prediction quality, because the model has been working optimally since the initial configuration. Overall, this very high performance indicates that Random Forest has excellent generalization capabilities and is reliable in the classification process. The Confusion Matrix for the Random Forest algorithm can be seen in Table 15.

Tabel 15. Confusion Matrix Random Forest

Confusion Matrix	True Normal	True High	True Stunted	True Saverely Stunted
Pred Normal	4129	1	0	0
Pred High	5	2784	1	0
Pred Stunted	0	6	13376	0
Pred Saverely Stunted	0	0	9	3889

Based on the Confusion Matrix in Table 15, the Random Forest algorithm shows optimal classification performance, as seen from the dominance of diagonal values, which indicates the model's success in accurately predicting each class. In the True Normal class, the model was able to classify 4,129 data correctly with only one prediction error, while in the True High class, 2,784 data were classified accurately with two minor errors scattered across other classes. The model's performance was also very strong in the True Stunted class, with 13,376 data points predicted correctly and very few errors, as well as in the True Severely Stunted class, which showed perfect accuracy with 3,889 data points classified correctly without any errors. The minimal prediction errors across all categories prove that Random Forest is capable of recognizing data patterns very well, resulting in stable, consistent performance that is highly suitable for classification needs in this study.

c. Naïve Bayes Classifier

The performance of the Naïve Bayes classification model was evaluated by testing the main parameters that affect the probability distribution in each class. This test aimed to assess the model's ability to produce accurate and consistent predictions on the test data. The Naïve Bayes algorithm itself operates on a probabilistic basis using Bayes' theorem, assuming that each feature is independent to estimate the probability of a data point belonging to a particular class. To assess the overall performance of the model in the classification process, this evaluation was conducted comprehensively using standard metrics, namely Accuracy, Precision, Recall, and F1-Score. The detailed results of this test are listed in Table 16.

Table 16. Performance evaluation results of the Naïve Bayes Classifier algorithm

Algorithm	Accuracy (CA)	F1-Score	Precision	Recall
Naïve Bayes Classifier	0548 (54.8%)	0.450 (45.0%)	0.390 (39.0%)	0.550 (55.0%)

Based on the evaluation data shown in Table 16, the Naïve Bayes Classifier algorithm shows fairly good classification performance despite still having several limitations. The model obtained an accuracy value of 0.548 (54.8%), which means that about half of the total test data was classified correctly. The precision value of 0.390 (39.0%) indicates that the accuracy of the model's predictions is still relatively low, so that some of the prediction results do not fully match the actual class. However, the recall value of 0.550 (55.0%) indicates that the model has a better ability to recognize positive data in several classes. Meanwhile, the F1-Score value of 0.450 (45.0%) reflects a moderate balance between precision and recall, indicating that the model's performance still needs to be improved to achieve more optimal results. Overall, the performance of the Naïve Bayes algorithm in this study can be considered quite stable as a basic model, but it still requires further development through optimization and data quality improvement. The Confusion Matrix of the Naïve Bayes Classifier algorithm can be seen in Table 17.

Table 17. Confusion Matrix Naïve Bayes Classifier

Confusion Matrix	True Normal	True High	True Stunted	True Saverely Stunted
Pred Normal	1679	0	2451	0
Pred High	599	0	2191	0
Pred Stunted	1801	0	11581	0
Pred Saverely Stunted	724	0	3174	0

Based on the Confusion Matrix in Table 17, the performance of the Naïve Bayes Classifier algorithm shows suboptimal performance and still faces challenges in accurately distinguishing each class. In the True Normal class, the model was only able to classify 1,679 data correctly, while the other 2,451 data were predicted as Stunted, thus showing a significant error rate. This condition also occurs in the True High Class, where all data in that class failed to be predicted correctly and were all classified into other classes. However, the model performed better in the True Stunted Class, with 11,581 data successfully predicted correctly, although there were still a number of classification errors into the Normal and Severely Stunted classes. Meanwhile, in the True Severely Stunted Class, the model was able to classify 3,174 data correctly, but still produced a number of incorrect predictions, especially to the Normal and Stunted classes. Overall, these results show that the Naïve Bayes algorithm has a tendency to concentrate predictions on certain classes, causing an imbalance in performance between classes. This indicates the need for further improvement, either through data imbalance handling or feature optimization, in order to improve the model's ability to classify more accurately and stably across all categories.

d. Decision Tree C4.5

The performance of the C4.5 Decision Tree model was evaluated by testing the main parameters that play a role in forming the decision tree structure. This test aimed to determine the extent to which the model was able to provide accurate and stable prediction results on the test data. The C4.5 algorithm builds a decision tree by considering the information gain ratio as the basis for selecting the best attributes in each data separation process. This approach is designed to minimize the error rate and improve the model's ability to generalize. The model's performance was then analyzed comprehensively using standard evaluation metrics, namely Accuracy, Precision, Recall, and F1-Score. The complete results of the parameter testing can be seen in Table 18.

Table 18. Performance evaluation results of the C4.5 Decision Tree algorithm

Algorithm	Parameter Value	Accuracy (CA)	F1-Score	Precision	Recall
Decision Tree C4.5	**10**	**0,992 (99,2%)**	**0,990 (99.0%)**	**0,990 (99.0%)**	**0,990 (99.0%)**
	20	0.980 (98.0%)	0.980 (98.0%)	0.980 (98.0%)	0.980 (98.0%)
	30	0.967 (96.7%)	0.967 (96.7%)	0.970 (97.0%)	0.970 (97.0%)
	40	0959 (95.9%)	0.960 (96.0%)	0.960 (96.0%)	0.960 (96.0%)
	50	0.954 (95.4%)	0.950 (95.0%)	0.950 (95.0%)	0.950 (95.0%)

Based on Table 18, the C4.5 Decision Tree algorithm shows excellent classification performance across all min_samples_leaf parameter tests. Parameter 10 provides the best results with an accuracy of 0.992 (99.2%) and consistent precision, recall, and F1-Score values of 0.990 (99.0%). At parameter 20, the model's performance remained high with an accuracy of 0.980 (98.0%) and other evaluation metrics also at the 98% level. Although there was a slight decrease when the parameter increased to 30, 40, and 50, the model's accuracy remained in the range of 95–97%, indicating that the model was still able to perform very well in classification even though the tree structure became simpler. Overall, these results indicate that the C4.5 Decision Tree has strong generalization capabilities, with optimal performance at smaller parameters and remaining stable at larger parameters. The best Confusion Matrix from the C4.5 Decision Tree algorithm can be seen in Table 19.

Table 19. Confusion Matrix Decision Tree C4.5

Confusion Matrix	True Normal	True High	True Stunted	True Saverely Stunted
Pred Normal	4084	46	0	0
Pred High	39	2723	28	0
Pred Stunted	0	40	13311	31
Pred Saverely Stunted	0	0	17	3881

Based on the Confusion Matrix in Table 19, the Decision Tree algorithm shows excellent classification performance with consistent prediction capabilities for each class. In the Normal class, the model successfully classified 4,084 data correctly, although there were still 46 prediction errors to the High class. For the High class, the model also performed strongly with 2,723 correct predictions, accompanied by minor errors, namely 39 data predicted as Normal and 28 data as Stunted. The Stunted class showed very high accuracy, with 13,311 data points correctly predicted, and relatively low errors of 40 data points classified as High and 31 data points as Severely Stunted. Meanwhile, the Severely Stunted class achieved very accurate performance, with 3,881 data correctly predicted and only 17 data misclassified as Stunted. Overall, these results prove that Decision Tree is able to distinguish each class well, has a low error rate, and shows high stability and accuracy in the classification process.

e. Support Machine Vector (SVM)

The performance of the SVM algorithm was evaluated by testing various types of kernels that play a role in forming the separating boundary (hyperplane) between classes in the data. Each kernel has different capabilities in mapping patterns, so this test aims to identify the kernel that provides the most optimal classification results. Through the principle of maximum margin, SVM seeks to minimize prediction errors and improve the model's ability to generalize to test data. To gain a comprehensive understanding of the model's performance, the evaluation was conducted using the metrics of Accuracy, Precision, Recall, and F1-Score. The detailed results of this test are listed in Table 20.

Table 20. Performance evaluation results of the Support Vector Machine (SVM) algorithm

Algorithm	Parameter Value	Accuracy (CA)	F1-Score	Precisio	Recall
Support Vector Machine (SVM)	Kernel = Linear	0784 (78.4%)	0.770 (77.0%)	0.780 (78.0%)	0.780 (78.0%)
	Kernel = RBF	**0.953 (95.3%)**	**0.950 (95.0%)**	**0.950 (95.0%)**	**0.950 (95.0%)**
	Kernel = Polynomial	0.878 (87.8%)	0.840 (84.0%)	0.860 (86.0%)	0.830 (83.0%)
	Kernel = Sigmoid	0.480 (48.0%)	0.230 (23.0%)	0.440 (44.0%)	0.270 (27.0%)

Based on Table 20, the evaluation of SVM algorithm performance on several kernel types shows significant performance variations in the classification process. The RBF kernel is the parameter with the best results, marked by an Accuracy value of 0.953 (95.3%), which indicates that most of the test data was successfully predicted accurately by the model. In addition, the Precision, Recall, and F1-Score values, which each reached 0.950 (95.0%), indicate that the model is not only capable of producing accurate positive predictions, but also responsive in recognizing all relevant data in each class. Meanwhile, the Linear and Polynomial kernels performed well but were still below the RBF kernel, with accuracy rates of 78.4% and 87.8%, respectively, and lower F1-Scores, Precision, and Recall values. Overall, these results prove that the RBF kernel has the most optimal ability to distinguish data patterns and produce stable and consistent classifications. The best Confusion Matrix from the SVM algorithm can be seen in Table 21.

Table 21. Confusion Matrix Support Machine Vector (SVM)

Confusion Matrix	True Normal	True High	True Stunted	True Saverely Stunted
Pred Normal	3948	182	0	0
Pred High	277	2254	259	0
Pred Stunted	36	146	13106	94
Pred Saverely Stunted	0	0	154	3744

Based on the Confusion Matrix in Table 21, the SVM algorithm with RBF kernel shows excellent classification capabilities across all data categories. In Class 0 (Normal), the model was able to correctly identify 3,948 data points, although there were still 182 data points predicted as Class 1. For Class 1 (High), SVM successfully classified 2,254 data points accurately, with some prediction errors, namely 277 data points that were classified as Class 0 and 259 data points as Class 2. The best performance was seen in Class 2 (Stunted), where 13,106 data points were classified correctly, while the errors that occurred were relatively small, namely 36 data points predicted as Class 0, 146 data points as Class 1, and 94 data points as Class 3. In Class 3 (Severely Stunted), the model also showed high accuracy with 3,744 data correctly identified and only 154 data classified as Class 2. Overall, these results indicate that SVM is able to distinguish each class quite well, with a relatively low error rate, so it can be considered reliable in the process of classifying stunting condition data.

3.4. Model Evaluation

The model evaluation stage was carried out to measure the performance of each classification algorithm applied in this study. The aim was to assess how effective each algorithm was in classifying stunting data, especially after the data had undergone preprocessing and normalization. This study compares the performance of five main algorithms: KNN, SVM, Naïve Bayes, and Decision Tree C4.5. The evaluation process was carried out by calculating the accuracy value of each model using the same test data. The results of the accuracy comparison of these five algorithms can be seen in Figure 3.

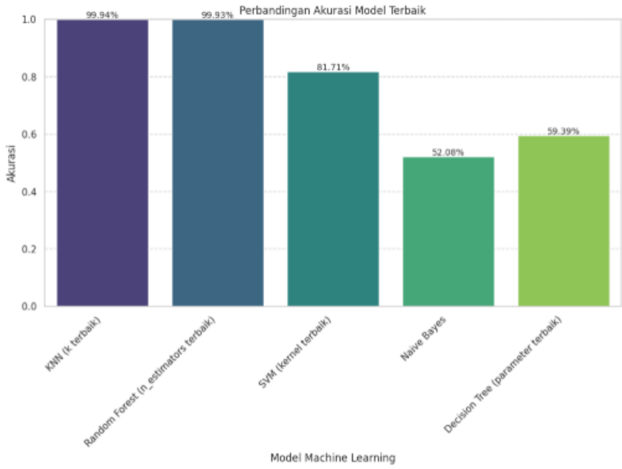

Figure 3. Accuracy Evaluation Results of 5 Machine Learning Algorithms

The comparison of accuracy values in Figure 3 shows a clear difference in accuracy levels among the five algorithms tested. KNN and Random Forest ranked highest with very high accuracy, 99.94% and 99.93% respectively, demonstrating highly reliable and consistent classification capabilities. Next, SVM achieved an accuracy of 81.71%, which, although still quite good, was not as strong as the two previous best models. Meanwhile, the Naïve Bayes and Decision Tree algorithms with optimal parameters obtained lower accuracies of 52.08% and 59.39%, respectively, indicating that both were less effective in optimally recognizing stunting data patterns. Overall, these findings confirm that KNN and Random Forest are the most recommended algorithms in the stunting data classification process because they are able to provide the best prediction performance compared to other models. The results of the F1-Score comparison of these five algorithms can be seen in Figure 4.

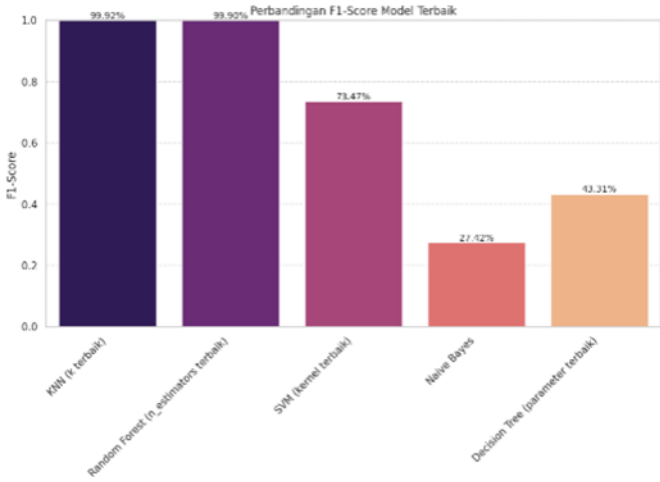

Figure 4. F1-Score Evaluation Results of 5 Machine Learning Algorithms

The comparison of F1-Score values in Figure 4 shows significant variations in performance among the five algorithms tested. KNN and Random Forest ranked highest with very high F1-Scores of 99.92% and 99.90%, respectively, reflecting an almost perfect balance of precision and recall in the stunting data classification process. Next, SVM achieved an F1-Score of 73.47%, indicating that this algorithm is still capable of providing fairly good performance, although not as strong as the top two models. In contrast, Decision Tree with the best parameters only obtained an F1-Score of 43.31%, while Naïve Bayes recorded the lowest value, namely 27.42%, indicating that these two algorithms are less effective in handling data pattern variations. Overall, these results confirm that KNN and Random Forest are the most superior algorithms in this study because they are able to provide the most consistent and accurate prediction performance compared to other models. The results of the Recall comparison of these five algorithms can be seen in Figure 5.

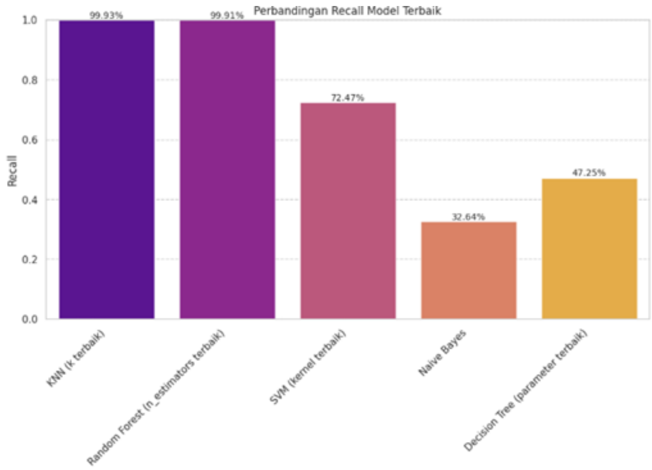

Figure 5. Recall Evaluation Results from 5 Machine Learning Algorithms

The comparison of recall values in Figure 5 shows a striking difference in performance among the five classification algorithms tested. KNN with the best parameters ranks highest with a recall of 99.93%, followed closely by Random Forest (best n_estimators) with a recall of 99.91%, confirming that both models are capable of recognizing almost all positive instances consistently and accurately. On the other hand, the performance of SVM (best kernel) was at an intermediate level with a recall value of 72.47%, indicating that although this algorithm was able to capture most of the positive classes, its sensitivity was still far below the top two models. Decision Tree produced a recall of 47.25%, reflecting that this model still faces difficulties in detecting most positive cases. Naïve Bayes was the model with the lowest performance, with a recall of 32.06%, indicating the algorithm's low ability to identify positive classes, possibly due to independent feature assumptions that did not match the data patterns. Overall, the dominance of recall values in KNN and Random Forest shows that these two models are the most reliable candidates for detecting positive classes in this study. The results of the Precision comparison of these five algorithms can be seen in Figure 6.

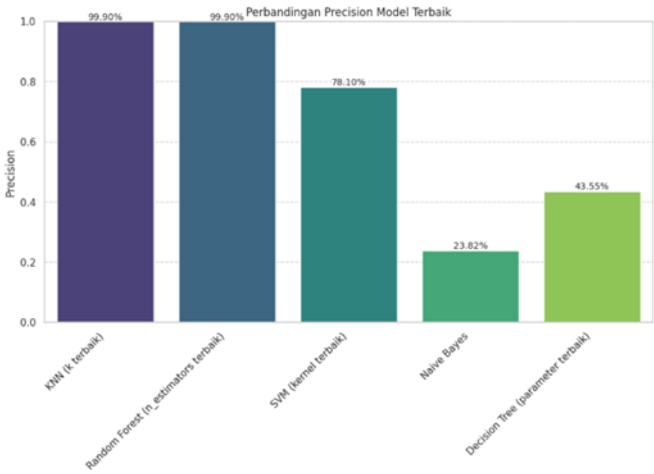

Figure 6. Precision Evaluation Results of 5 Machine Learning Algorithms

The comparison of precision values in Figure 6 shows a clear difference in performance between the five algorithms tested. KNN with the best parameters and Random Forest (best n_estimators) are at the top with the same precision value of 99.90%, reflecting the ability of both models to produce almost entirely accurate positive predictions. SVM (best kernel) follows in the next position with a precision of 78.10%, which still shows good performance even though it is not as accurate as the top two models. Decision Tree recorded a precision of 43.55%, indicating that this model still often produces inaccurate positive predictions. Naïve Bayes was the algorithm with the lowest precision, at 23.82%, indicating the model's low accuracy in predicting positive classes, possibly due to feature independence assumptions that did not fully match the data patterns. Overall, the high precision values for KNN and Random Forest confirm that these two models are the most effective choices for generating accurate positive predictions in this study.

3.5. Model Evaluation Matrix Table

To assess the performance of each algorithm used in this study, measurements were taken using four main evaluation metrics, namely accuracy, precision, recall, and F1-Score. These four metrics were chosen because they provide a more comprehensive picture of the model's ability to perform classification, not only in terms of prediction accuracy, but also in terms of the model's consistency in recognizing positive classes and the data as a whole. This evaluation was applied to all algorithms tested so that the results could be compared objectively. The Machine Learning Model Evaluation Matrix Table can be seen in Table 22.

Table 22 Table Matrix Evaluation Model Machine Learning

No	Algorithm	Accuracy (%)	F1-Score (%)	Recall (%)	Precision (%)
1	**KNN (K1)**	99.94 %	99.92%	99.93	99.90%
2	**Random Forest (10)**	99.93%	99.90%	99.91%	99.90%
3	**SVM (RBF)**	81.71%	73.47%	72.47%	78.10%
4	**Naïve Bayes**	52.08%	27.42%	32.06%	23.82%
5	**Decision Tree (10)**	59.39%	43.31%	47.25%	43.55%

Table 22 shows the performance evaluation results of all algorithms used in the study based on four main metrics, namely accuracy, F1-Score, recall, and precision. Based on this table, KNN and Random Forest show the best performance with consistently high evaluation values across all metrics, making them the most reliable models for classification. Meanwhile, SVM is in the middle category with lower metric values than the top two models, but still shows fairly stable performance. On the other hand, Naïve Bayes and Decision Tree produced relatively low evaluation scores across all metrics, indicating their limitations in optimally recognizing data patterns. Overall, this table shows the differences in the classification capabilities of each algorithm and confirms that KNN and Random Forest are the best-performing models in this study.

3.6. Discussion

The results of this study demonstrate that machine learning algorithms can be effectively applied to predict stunting risk in toddlers using anthropometric and demographic variables, namely age, gender, and height. The comparison of five classification algorithms revealed substantial differences in predictive performance, indicating that the choice of algorithm significantly influences the accuracy and reliability of stunting classification. Among the evaluated models, K-Nearest Neighbor (KNN) achieved the highest performance with an accuracy of 99.94%, followed closely by Random Forest with an accuracy of 99.93%. These findings suggest that both algorithms are highly effective in identifying patterns associated with nutritional status and stunting conditions in large-scale datasets.

The superior performance of KNN can be attributed to the characteristics of the dataset and the preprocessing techniques applied. Since the dataset was normalized using the Min-Max scaling method, the distance calculations used by KNN became more reliable and balanced across all features. KNN classifies data based on the similarity of neighboring instances, making it particularly effective when the dataset contains clear boundaries between classes. The confusion matrix results further confirm this observation, as only a very small number of misclassifications occurred across all nutritional status categories. The optimal performance obtained with $K = 1$ indicates that each toddler record has highly distinctive characteristics that can be accurately represented by its nearest neighbor. This finding suggests that age, height, and gender collectively provide sufficient discriminatory information for stunting classification.

Random Forest also demonstrated outstanding performance, achieving evaluation metrics almost identical to those of KNN. The algorithm benefits from ensemble learning by combining multiple decision trees and aggregating their predictions. This approach reduces variance and improves model robustness, making Random Forest less sensitive to noise and outliers. The consistency of results across different numbers of trees (10 to 500 estimators) indicates that the dataset contains strong and stable patterns that can be effectively captured even with relatively simple ensemble configurations. Furthermore, the minimal classification errors observed in the confusion matrix indicate that Random Forest successfully generalized the relationships between input features and nutritional status categories.

The Support Vector Machine (SVM) algorithm achieved moderate performance compared to KNN and Random Forest. Among the tested kernels, the Radial Basis Function (RBF) kernel produced the best results with an accuracy of 81.71%. This outcome suggests that the relationship between the predictor variables and nutritional status is nonlinear, allowing the RBF kernel to construct more flexible decision boundaries. However, despite its relatively good performance, SVM was unable to match the accuracy achieved by KNN and Random Forest. One possible explanation is that the dataset contains overlapping characteristics among some nutritional status classes, making it difficult for SVM to establish optimal hyperplanes that separate all categories perfectly. In addition, the large dataset size may increase computational complexity and limit the effectiveness of parameter settings.

In contrast, Naïve Bayes and Decision Tree C4.5 produced considerably lower performance. The Naïve Bayes classifier achieved an accuracy of only 52.08%, indicating substantial limitations in handling the characteristics of the dataset. This result is likely caused by the independence assumption underlying the Naïve Bayes algorithm, which assumes that all features are conditionally independent of one another. In the context of stunting prediction, age, height, and nutritional status are naturally interrelated variables, causing this assumption to be violated. Consequently, the probabilistic estimations generated by Naïve Bayes become less accurate, leading to a high number of classification errors.

Similarly, the Decision Tree C4.5 algorithm obtained lower performance compared to KNN and Random Forest. Although the model achieved good results during parameter testing, its overall evaluation metrics remained below expectations. Decision trees are known to be sensitive to variations in data distribution and can easily overfit or underfit depending on the tree structure. The relatively lower precision and recall values indicate that the generated decision boundaries were insufficient to capture the complex patterns present in the dataset. Nevertheless, the Decision Tree model still offers the advantage of interpretability, allowing healthcare practitioners to understand the reasoning behind classification decisions more easily than with black-box models.

Another important finding of this study is the presence of class imbalance in the dataset. The Normal class contained significantly more records than the Stunted class, which may influence model behavior and evaluation results. Despite this imbalance, KNN and Random Forest maintained exceptionally high recall and precision values, indicating their ability to recognize minority classes effectively. However, future studies should investigate the impact of balancing techniques such as Synthetic Minority Oversampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), or class-weighting strategies to further improve fairness and robustness across all nutritional status categories.

From a practical perspective, the results indicate that KNN and Random Forest have strong potential for implementation in stunting early detection systems. Their high predictive performance suggests that these algorithms can assist healthcare workers, nutritionists, and government agencies in identifying children at risk of stunting more quickly and accurately. Early identification enables timely nutritional interventions and monitoring programs, which are essential for reducing the prevalence and long-term impacts of stunting. Furthermore, integrating these models into web-based or mobile health applications could support real-time decision-making in community health centers and child growth monitoring programs.

Overall, the findings confirm that machine learning approaches can significantly enhance the effectiveness of stunting risk prediction. Among the evaluated models, KNN emerged as the best-performing algorithm, followed closely by Random Forest. These results provide valuable evidence that data-driven approaches can support public health initiatives aimed at improving child nutrition and growth outcomes. Future research should consider incorporating additional anthropometric, socioeconomic, maternal, and environmental variables to develop more comprehensive predictive models and improve the generalizability of stunting detection systems across different populations and regions.

4. CONCLUSION

This study successfully implemented and compared five machine learning algorithms, namely K-Nearest Neighbor (KNN), Random Forest, Support Vector Machine (SVM), Naïve Bayes, and Decision Tree C4.5, for predicting stunting risk in toddlers using the "Stunting Balita Detection (121K Rows)" dataset. The dataset consisted of 120,999 records with variables including age, gender, height, and nutritional status. Prior to model development, data preprocessing was conducted through categorical encoding, dataset partitioning, and Min-Max normalization to improve data quality and model performance. The experimental results demonstrate that the choice of machine learning algorithm significantly affects classification performance. Among all evaluated models, KNN achieved the best performance with an accuracy of 99.94%, precision of 99.90%, recall of 99.93%, and F1-score of 99.92%. Random Forest produced comparable results with an accuracy of 99.93%, confirming its effectiveness and robustness in handling large-scale stunting datasets. In contrast, SVM achieved moderate performance, while Naïve Bayes and Decision Tree C4.5 showed relatively lower classification accuracy and consistency. The confusion matrix analysis further confirmed that KNN and Random Forest were able to classify all nutritional status categories with very low error rates, including minority classes such as Stunted and Severely Stunted. These findings indicate that both algorithms are highly reliable for identifying stunting risk and can effectively support data-driven decision-making in child health monitoring programs. Based on the overall evaluation, KNN is recommended as the most suitable algorithm for stunting classification in this dataset, while Random Forest serves as a strong alternative due to its stable and consistent performance. The results of this study contribute to the development of machine learning applications in public health and demonstrate the potential of artificial intelligence to support early stunting detection. Future research is recommended to incorporate additional health, socioeconomic, maternal, and environmental factors, as well as apply class-balancing and hyperparameter optimization techniques, to further improve prediction performance and model generalization.

REFERENCES

[1] P. M. Putri, A. S. Shafira, dan G. S. Mahardhika, "Stunting Reduction Strategy in Indonesia: Maternal Knowledge Aspects," *Indones. J. Public Heal.*, vol. 19, no. 2, hal. 329–343, 2024, doi: 10.20473/ijph.v19i2.2024.329-343.

[2] N. N. Laloan, S. Masithah, A. Rahmaniar MB, F. Wahyuni, dan S. Selvia, "Faktor Risiko Status Gizi Kurang Pada Anak Usia 0-59 Bulan Di Wilayah Kerja Puskesmas Kimi Nabire Papua Tahun 2023," *J. Kesehat. Tambusai*, vol. 5, no. 4, hal. 11432–11440, 2024, doi: 10.31004/jkt.v5i4.35925.

[3] M. Paramita, S. Helmyati, R. W. Basrowi, dan C. Dilantika, "Risk Factors for Stunting among Infants on Java Island: A Cross-Sectional Study using the 2021 Indonesian Nutritional Status Survey Data," *Amerta Nutr.*, vol. 7, no. 3, hal. 20–29, 2023, doi: 10.20473/amnt.v7i3SP.2023.20-29.

[4] H. Rahman, M. Rahmah, dan N. Saribulan, "Upaya Penanganan Stunting Di Indonesia," *J. Ilmu Pemerintah. Suara Khatulistiwa*, vol. VIII, no. 01, hal. 44–59, 2023.

[5] N. F. Sahamony, T. Terttiaavini, dan H. Rianto, "Analisis Perbandingan Kinerja Model Machine Learning untuk Memprediksi Risiko Stunting pada Pertumbuhan Anak," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 2, hal. 413–422, 2024, doi: 10.57152/malcom.v4i2.1210.

[6] A. I. Putri *et al.*, "Implementation of K-Nearest Neighbors, Naïve Bayes Classifier, Support Vector Machine and Decision Tree Algorithms for Obesity Risk Prediction," *Public Res. J. Eng. Data Technol. Comput. Sci.*, vol. 2, no. 1, hal. 26–33, 2024, doi: 10.57152/predatecs.v2i1.1110.

[7] M. Fazlur Rahman Assauqi dan Zaehol Fatah, "Prediksi Risiko Demam Berdarah Menggunakan Decision Tree Berdasarkan Gejala Klinis Dan Data Laboratorium," *J. Ilm. Sains Teknol. Dan Inf.*, vol. 2, no. 4, hal. 42–53, 2024, doi: 10.59024/jiti.v2i4.972.

[8] R. Sumanti, "Collaborative Governance: Strategies To Prevent And Reduce Of Stunting Prevalence," *J. Dev. Policy*, vol. 19, no. 1, hal. 13–26, 2024.

[9] I. P. Putri, T. Terttiaavini, dan N. Arminarahmah, "Analisis Perbandingan Algoritma Machine Learning untuk Prediksi Stunting pada Anak," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 1, hal. 257–265, 2024, doi: 10.57152/malcom.v4i1.1078.

[10] S. Anwar, E. Winarti, dan S. Sunardi, "Systematic Review Faktor Risiko, Penyebab Dan Dampak Stunting Pada Anak," *J. Ilmu Kesehat.*, vol. 11, no. 1, hal. 88, 2022, doi: 10.32831/jik.v11i1.445.

[11] B. M. K. Neighbors, A. Putra, T. Djoru, dan S. Yulianto, "Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi) Pendekatan Machine Learning untuk Deteksi Stunting pada," vol. 9, no. June, hal. 664–672,

2025.
[12] V. No, N. R. Febriyanti, dan A. D. Hartanto, “Edumatic : Jurnal Pendidikan Informatika Analisis Perbandingan Algoritma SVM , Random Forest dan Logistic Regression untuk Prediksi Stunting Balita,” vol. 9, no. 1, hal. 149–158, 2025, doi: 10.29408/edumatic.v9i1.29407.
[13] T. Septiyanto, F. A. Mufarroha, D. R. Anamisa, dan A. Jauhari, “Prediksi Jumlah Penderita Stunting di Madura Dengan Pendekatan Machine Learning,” *JoMMiT J. Multi Media dan IT*, vol. 7, no. 2, hal. 070–076, 2023, doi: 10.46961/jommit.v7i2.901.
[14] D. T. H. Hutabarat, M. A. Bima, N. Syahfitri, dan S. D. Manurung, “Kajian Literatur Tentang Upaya Pencegahan Stunting Anak Melalui Imunisasi Dan Asupan Gizi,” *J. Ilm. Wahana Pendidik.*, vol. 10, no. April 2024, hal. 298–310, 2024.
[15] Fadellia Azzahra, N. Suarna, dan Y. Arie Wijaya, “Penerapan Algoritma Random Forest Dan Cross Validation Untuk Prediksi Data Stunting,” *Kopertip J. Ilm. Manaj. Inform. dan Komput.*, vol. 8, no. 1, hal. 1–6, 2024, doi: 10.32485/kopertip.v8i1.238.
[16] M. Jafar dan D. Aisyah, “Romlah1, Sutan Faisal2, Rahmat3, Jamaludin Indra4,” vol. 2338, hal. 13–34, 2022.
[17] U. R. Gurning, S. F. Octavia, D. R. Andriyani, N. Nurainun, dan I. Permana, “Prediksi Risiko Stunting pada Keluarga Menggunakan Naïve Bayes Classifier dan Chi-Square,” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 1, hal. 172–180, 2024, doi: 10.57152/malcom.v4i1.1074.
[18] W. C. Wahyudin, F. M. Hana, dan A. Prihandono, “Prediksi Stunting Pada Balita Di Rumah Sakit Kota Semarang Menggunakan Naive Bayes,” *J. Ilmu Komput. dan Matemtika*, vol. 2019, hal. 32–36, 2023.
[19] E. S. Yulia Wardita, Eka Meiri Kurniyati, Cory Nelia Damayanti, Mujib Hannan, “Model Prediksi Kejadian Stunting Pada Balita Berdasarkan Faktor Personal Ibu Dan Pola Asuh,” *J. Keperawatan*, vol. 14, hal. 1047–1056, 2022.
[20] A. J. Wahidin dan T. H. Andika, “Deteksi Dini Stunting Pada Anak Berdasarkan Indikator Antropometri dengan Menggunakan Algoritma Machine Learning,” hal. 378–387, 2024, doi: 10.33364/algoritma/v.21-2.2122.