ISSN 2723-6471

Optimizing Stunting Detection through SMOTE and Machine Learning: a Comparative Study of XGBoost, Random Forest, SVM, and k-NN

Tri Sugihartono[1,*], Benny Wijaya[2], Marini[3] , Ahmad Faqih Alkayes[4], Hendra Agustian Anugrah[5]

[1,2,4,5]*Department of Technology Information, Faculty of Informatics Engineering, Institute Science and Business Atma Luhur, Pangkalpinang, Indonesia*

[3]*Department of Technology Information, Faculty of Information System, Institute Science and Business Atma Luhur, Pangkalpinang, Indonesia*

(Received: September 21, 2024; Revised: October 31, 2024; Accepted: November 21, 2024; Available online: December 31, 2024)

Abstract

Stunting is a vital public health priority that affects millions of children from all over the world, especially in developing countries, where chronic malnutrition impairs their physical growth and cognitive development. Early detection of stunting is necessary for its timely intervention to reduce long-lasting effects. The following study deals with the application of higher-end machine learning techniques in order to detect stunting with more accuracy, using XGBoost, Random Forest, SVM, and k-NN algorithms. Using a dataset sourced from Kaggle, containing 10,000 samples of anthropometric and demographic features, we addressed the significant class imbalance of the data; the number of samples representing stunted children was only 15% of the total. We surmounted this limitation using SMOTE to generate synthetic data in order to balance the representation for this minority class. Further feature selection to improve the performance and interpretability of the model was done using backward elimination, where less impactful features like "Body Length" and "Breastfeeding" were systematically excluded, while putting more emphasis on more predictive variables such as weight, age, and socio-economic indicators. The evaluation of machine learning models showed significant improvements in performance with the integration of SMOTE and optimized feature selection, especially regarding recall and ROC-AUC metrics, which are critical in healthcare settings where the minimization of false negatives is of high importance. XGBoost was the best-performing model among those evaluated, yielding an accuracy of 0.8574, a recall of 0.8914, and an ROC-AUC of 0.9311, hence balancing precision and sensitivity more appropriately than other models. These results emphasize the efficiency of XGBoost in stunting detection while overcoming challenges arising from imbalanced datasets. It then illustrates the potential of merging machine learning techniques with synthetic data augmentation methodologies for the optimization of outcomes related to population health, and forms a basis for healthcare practitioners and policymakers by locating the at-risk children on time. The findings not only point to the importance of advanced data-driven approaches in stunting detection but also lay the ground for future research on machine learning applications in the fight against other malnutrition-related public health challenges, which could be crucial for improving child health and well-being across the world.

Keywords: Stunting Detection, Machine Learning, SMOTE, XGBoost, Random Forest, Support Vector Machine, k-Nearest Neighbors, Healthcare, Class Imbalance, Feature Selection, ROC-AUC, Stunting

1. Introduction

Stunting, a chronic condition characterized by impaired growth and development in children due to malnutrition, is a serious global health issue that disproportionately affects developing countries. According to the World Health Organization (WHO), stunting impacts approximately 149 million children under the age of five worldwide. The consequences of stunting are profound and far-reaching, affecting not only physical development but also cognitive abilities, which can lead to long-term socio-economic disadvantages [1]. Besides impeding a child's physical development, stunting is injurious to a child's cognitive abilities, reducing their potential for learning and changing their IQs irreversibly. Stunting often makes many children fall behind in their classes, thereby limiting their choices of jobs later in life. It is approximated by the World Bank that countries with high rates of stunting could lose as much as 3% of their annual GDP due to low productivity among workers and high medical expenses. In addition, stunting remains a latent threat to long-term sustainability in most developing countries due to its intergenerational effects.

***Corresponding author: Tri Sugihartono (trisugihartono@atmaluhur.ac.id)**
DOI: https://doi.org/10.47738/jads.v6i1.494

Addressing stunting is crucial for improving public health outcomes, particularly in countries with high prevalence rates like Indonesia, where efforts to reduce stunting are a key priority in national health strategies.

Early detection of stunting is critical to enabling timely intervention and treatment, reducing its long-term impact on individuals and society. Traditional methods of detecting stunting rely heavily on anthropometric measurements, such as height-for-age comparisons. While these methods are effective to some extent, they are often limited by the availability of healthcare resources, especially in rural and underdeveloped areas [2]. Moreover, manual measurement processes are prone to errors and inconsistencies, further complicating the early detection of stunting. There is a growing need for more reliable, efficient, and automated solutions that can assist healthcare professionals in identifying children at risk of stunting, especially in resource-limited settings. Machine learning (ML) offers a promising alternative by automating the detection process and leveraging large datasets to improve accuracy and speed.

In recent years, machine learning has been increasingly applied to various health problems, including disease detection, prognosis, and prediction [3]. These models can process large volumes of health data and identify complex patterns that may not be apparent through traditional analysis. For instance, Effendy et al. applied machine learning models to predict thermal sensation in buildings using health data, demonstrating the versatility of ML in non-traditional health settings. Similarly, machine learning could be used to analyze diverse health indicators in the context of stunting detection, improving detection rates and enabling earlier intervention. However, applying ML to health data is challenging, particularly the issue of class imbalance in datasets.

One of the primary challenges in using machine learning for health data analysis is the prevalence of imbalanced datasets. In the context of stunting detection, the number of children with stunting is often much lower than those without, resulting in a dataset where the majority class (non-stunted) dominates the minority class (stunted) [4]. This imbalance can lead to biased models that favor the majority class, which in turn reduces the model's ability to identify stunted children accurately. This is a critical issue in healthcare, where false negatives—failing to detect a child with stunting—can have severe implications for the child's health and development.

Several techniques have been developed to address the issue of imbalanced datasets, with SMOTE (Synthetic Minority Over-sampling Technique) being one of the most widely used. SMOTE generates synthetic instances of the minority class, thereby increasing its representation in the dataset and improving the model's ability to learn from these samples [4]. Studies such as those by Kristiyanti et al. [5] have demonstrated the effectiveness of SMOTE in improving model performance in various contexts, including sentiment analysis and fraud detection. However, while SMOTE has been applied in many domains, its use in stunting detection remains relatively underexplored, particularly in conjunction with feature selection techniques such as backward elimination.

Feature selection is another critical component of machine learning that can significantly impact model performance. Not all features are equally important in large datasets, and including irrelevant or redundant features can reduce the model's efficiency and accuracy [5]. Backward elimination is a popular feature selection technique that iteratively removes the most minor significant features from the dataset, simplifying the model and enhancing its predictive power. Demonstrated the utility of sequential feature selection in improving the performance of k-nearest Neighbors (k-NN) for diabetes prediction [6]. By applying backward elimination, the study reduced model complexity while maintaining high accuracy, underscoring the value of feature selection in health-related machine-learning applications.

While some studies have explored the application of machine learning to health data, particularly in disease detection and prevention, the specific challenges of stunting detection still need to be researched. Previous studies, such as those by Angelica et al. [7], have focused on using machine learning for fraud detection and other non-health domains. Even within the health sector, much of the focus has been on disease prediction, such as cancer and diabetes [7]. For instance, a comparative analysis of explainable AI models for lung cancer prediction, emphasizing the importance of transparency in AI-driven health predictions. However, only some studies have specifically addressed stunting detection, leaving a gap in applying ML models to this pressing global health issue.

Moreover, existing studies on machine learning in health often focus on optimizing accuracy at the expense of other important metrics, such as recall and ROC-AUC, which are crucial for healthcare applications [8]. High recall is essential in stunting detection because it reflects the model's ability to correctly identify stunted children, minimizing

the risk of false negatives. A low recall could result in children being misclassified as non-stunted, delaying critical interventions. Despite this, many studies prioritize accuracy as the primary performance metric, neglecting the importance of recall in high-stakes health applications [8]. This research addresses this gap by evaluating machine learning models using a more holistic set of metrics, including recall, precision, and ROC-AUC.

Several machine learning models have shown promise in various health-related applications, including XGBoost, Random Forest, SVM, and k-NN. These models are known for their strong performance in classification tasks and have been applied in contexts ranging from disease prediction to environmental health monitoring [8]. For instance, SVM has been used effectively for predicting diabetes and thermal sensations in buildings, while XGBoost has been employed for lung cancer and fire hotspot predictions Despite their success in these areas, their comparative performance in stunting detection has yet to be thoroughly investigated. By comparing these models in a stunting detection context, this study seeks to provide insights into which models are best suited for this task [9].

Recent machine learning advancements have shown great potential in the identification and prediction of stunting among under-five children with more accurate and actionable insights. In research [10] employed machine learning algorithms to identify factors associated with undernutrition in under-five-year-old children in Ghana in 2024. Their study showcased how advanced computational models can uncover complex associations in nutrition data sets, which might otherwise be overlooked. In Research [11] applied machine learning techniques to predict stunting in Papua New Guinea. They stressed that these models are applicable across different geographic and socioeconomic contexts and also depend on localized data. Integra approaches into stunting detection frameworks enhance the precision and relevance of the interventions; this underlines the novelty and applicability of machine learning in tackling child malnutrition globally.

In addition to evaluating the performance of different machine learning models, this study also explores the impact of feature selection on model performance [11]. As mentioned earlier, backward elimination removes irrelevant features from the dataset, simplifying the model and potentially improving its predictive power. This is particularly important in health data, where datasets often contain many variables that are not all relevant to the prediction task [12]. By applying backward elimination, this study aims to identify the most relevant features for stunting detection, improving model accuracy and interpretability.

Given the challenges of imbalanced datasets and the importance of feature selection in machine learning, this study aims to fill a gap in the literature by applying SMOTE and backward elimination to stunting detection. Specifically, we evaluate the performance of four machine learning models—XGBoost, Random Forest, SVM, and k-NN—in detecting stunting from an imbalanced dataset [12]. By comparing these models across various performance metrics, including accuracy, recall, precision, and ROC-AUC, we seek to identify the most effective model for early stunting detection. Additionally, we assess the impact of SMOTE and backward elimination on model performance, providing insights into how these techniques can improve stunting detection.

The findings of this study have the potential to contribute to the development of automated stunting detection systems that are both accurate and reliable, particularly in resource-limited settings where early detection is critical for preventing long-term health consequences. By leveraging machine learning, advanced data balancing, and feature selection techniques, this research aims to improve the effectiveness of stunting detection, ultimately contributing to better health outcomes for children in developing countries [13].

2. The Proposed Method/Algorithm

The research workflow for stunting detection, as outlined in figure 1, begins with data preprocessing, a crucial step in preparing the dataset for analysis. This step ensures the removal of incomplete or irrelevant data, improving the overall quality. After preprocessing, a baseline model is implemented without using SMOTE (Synthetic Minority Over-sampling Technique) and backward elimination. This initial model helps identify the most relevant features while eliminating those that don't contribute significantly to the model's performance.

In the next stage, the model is retrained, this time using SMOTE to handle imbalanced data, which is common in stunting detection. SMOTE helps balance the class distribution, improving the model's predictive accuracy. The

 ISSN 2723-6471

workflow continues with backward feature selection, which further refines the feature set by iteratively removing less important features. Finally, the model is evaluated based on multiple performance metrics, ensuring a robust assessment of its ability to detect stunting accurately and reliably.

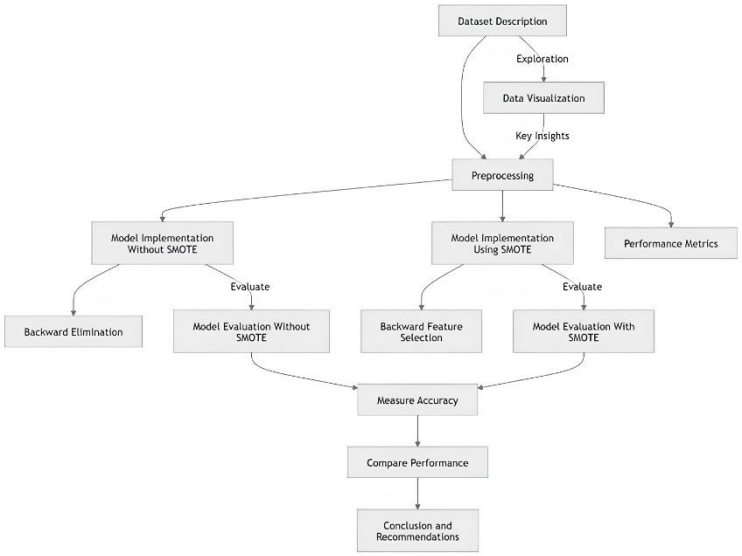

Figure 1. Research Workflow

2.1. Dataset Description

The data is from Kaggle and consists of 10,000 instances used in the detection of stunting in children. Each record contains a set of features in the forms of anthropometric measurements, demographic information, and health indicators. The data consist of both stunted and non-stunted children classified based on WHO's height-for-age standard. From this data, the distribution of male and female children aged 0-5 years has been almost at an equal ratio. While no explicit geographic information is available, there seems to be a mix of rural and urban populations. The socio-economic information also shows that a majority of the children are from low to middle-income families. Similar to many healthcare-related data sets, this suffers from a problem of class imbalance [14]. Non-stunted children are a far heavier majority compared to the stunted cases. This can pose a problem with machine learning models because they tend to favor the majority class, in this case reducing the model's ability to correctly detect stunting.

2.2. Preprocessing

Before applying any machine learning models, the dataset underwent several preprocessing steps to ensure data quality and consistency. The process began with data cleaning, where approximately 2,000 duplicate records were identified and removed, leaving 8,000 unique instances for analysis [15]. The next step involved handling missing values. Depending on the extent of the missing data, incomplete entries were either removed or imputed using statistical methods. For numerical variables, missing data was filled using mean substitution, while categorical variables were imputed using the mode.

2.3. Model Implementation Without SMOTE and Backward Elimination

To establish a baseline, four machine learning models—XGBoost, RF, SVM, and k-NN—were implemented on the preprocessed dataset without applying SMOTE or backward elimination. The purpose of this initial step was to assess how well the models performed on the raw, imbalanced dataset using the full set of features [16]. The dataset was split into training and testing sets using an 80:20 ratio, with 80% of the data used for training and the remaining 20% for testing. This ensured that the models could be evaluated on unseen data. Basic hyperparameter tuning was performed for each model to optimize its performance. For instance, in XGBoost, the learning rate and maximum tree depth were adjusted, Random Forest's number of estimators was fine-tuned, the kernel and regularization parameters were optimized for SVM, and the number of neighbors was varied for k-NN.

2.4. Model Implementation Using SMOTE for Handling Imbalanced Data

Given the significant class imbalance in the dataset, SMOTE was applied to improve the models' ability to detect stunted cases. SMOTE operates by generating synthetic samples of the minority class (stunted children) based on their nearest neighbors, thus increasing the representation of stunted cases in the dataset [17]. This technique enhances the models' capacity to recognize stunting by ensuring that the minority class is adequately represented during the training phase.

In all, the dataset for this study consisted of 10,000 instances, out of which 80% were stunted children totaling 4,873 instances and 20% non-stunted children were 1,185 instances respectively, giving a class imbalance ratio of 1:4. This is relatively comparable with the datasets used in similar studies where usually the class imbalance ranges between 1:3 and 1:6. This imbalance is rather less serious compared to some highly imbalanced healthcare datasets. However, it still can pose some challenges for most machine learning models, in that it might bias the model toward the majority class, impacting its capability of detecting non-stunted cases [17].

2.5. Backward Feature Selection

To further optimize the models and improve their efficiency, backward elimination was employed as a feature selection technique. Backward elimination is a recursive process where the least significant features are systematically removed, enabling the models to focus on the most relevant variables [18]. Initially, all features were included in the models, and their significance was evaluated based on their contribution to model performance. This evaluation was typically done using p-values in statistical models or feature importance scores for tree-based models like XGBoost and Random Forest.

During the elimination process, features that showed the least statistical significance or contributed minimally to model accuracy were removed iteratively. For example, "body length" was identified as a non-essential feature and was subsequently excluded from the models. After each feature removal, the models were retrained to evaluate the impact on performance, ensuring that eliminating less relevant features did not compromise the model’s effectiveness [18].

2.6. Performance Metrics

To evaluate and compare the performance of the models across different configurations—both with and without SMOTE and backward elimination—several key performance metrics were utilized. Accuracy measured the proportion of correctly classified instances (both stunted and non-stunted) out of the total instances, but while commonly used, accuracy alone can be misleading in imbalanced datasets [19]. Precision calculated the proportion of true positive stunting cases out of all cases predicted as stunted, making it particularly important for minimizing false positives. However, the primary focus of this study was recall, which measured the proportion of true positive stunting cases detected out of all actual stunting cases. High recall is crucial in healthcare settings to ensure that stunted children are not overlooked.

The F1-score, which represents the harmonic mean of precision and recall, was used to provide a balanced metric when both precision and recall needed to be optimized simultaneously. This is especially useful in cases where both false positives and false negatives must be carefully managed. Finally, ROC-AUC was employed to illustrate the trade-off between the true positive rate (recall) and the false positive rate. A higher AUC value indicated better overall model performance in distinguishing between stunted and non-stunted cases, offering a comprehensive evaluation of each model's effectiveness in handling the task [19].

3. Methodology

3.1. SMOTE (Synthetic Minority Over-sampling Technique)

SMOTE is a data augmentation technique used to handle class imbalance in datasets, especially in classification problems. In imbalanced datasets, the majority class often overwhelms the minority class, leading machine learning models to bias towards the majority class [20]. SMOTE tackles this issue by generating synthetic samples of the minority class rather than simply duplicating existing samples. This helps the model learn from a broader representation of the minority class and improve its performance in predicting minority cases, such as stunting detection in this study.

 ISSN 2723-6471

$$x_{synthetic} = x + \lambda \times (x_{neighbor} - x) \quad (1)$$

Interpolation in SMOTE works by generating new synthetic data points through a weighted combination of the original data point and one of its nearest neighbors. This is controlled by the parameter λ, a random value between 0 and 1, which determines how far the synthetic point will be placed between the original point and its neighbor. By adjusting λ, the synthetic data point is positioned along the line connecting the two points in the feature space, creating a new sample that reflects the characteristics of both.

3.2. Backward Elimination

Backward Elimination is a widely used feature selection technique in machine learning and statistical modeling that systematically removes the least significant features from a model, one at a time. This technique is especially beneficial when working with datasets that contain a large number of features, many of which may be irrelevant or redundant [21]. By focusing only on the most important features, the model becomes more efficient to train and tends to generalize better to unseen data. In the context of a multiple linear regression model, the relationship between the target variable y and the predictors $x1, x2, \ldots, xn$ can be expressed as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon \quad (2)$$

The goal of backward elimination is to remove features xI that do not contribute significantly to the model's ability to predict y. The algorithm evaluates the statistical significance of each feature by analyzing the p-value of its coefficient βI . If the p-value of a feature is higher than a predetermined significance level (usually 0.05), the feature is considered statistically insignificant and is removed from the model [21]. The process continues until only statistically significant features remain, resulting in a simpler and more efficient model that retains its predictive power while minimizing complexity.

3.3. XGBOOST (Extreme Gradient Boosting)

XGBoost is an optimized version of gradient boosting designed for speed and performance. It has become one of the most popular and powerful machine learning algorithms, widely used for both regression and classification tasks. XGBoost builds an ensemble of decision trees in a sequential manner, where each new tree attempts to correct the errors made by the previously constructed trees [22]. This method uses gradient descent to minimize a loss function by adjusting the model's parameters iteratively.

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (3)$$

In XGBoost, the loss function plays a crucial role by guiding the model to minimize the error between predicted and actual outcomes, ensuring the predictions are as accurate as possible. This function measures how well the model performs during training and helps it adjust parameters to improve accuracy. Additionally, XGBoost incorporates a regularization term to control the complexity of the decision trees, preventing overfitting. By penalizing the model based on the number of leaves and the size of the leaf weights, the regularization term discourages overly complex models that might perform well on training data but fail to generalize to unseen data [22]. This balance between accuracy and simplicity helps XGBoost produce robust models that perform well in real-world scenarios.

3.4. Random Forest

Random Forest, a classification method within ensemble learning, utilizes multiple decision trees to improve prediction accuracy [23]. The process begins by randomly selecting a training subset from the overall training dataset. Each decision tree in the forest is generated and trained using this subset.

$$\hat{y} = \frac{1}{N_{trees}} \sum_{i=1}^{N_{trees}} y_i \quad (4)$$

$\hat{y}$ represents the final prediction or output of the Random Forest model. The variable N_{trees} denotes the total number of decision trees within the forest, and each y_i corresponds to the individual prediction made by the i -th decision tree. The sum of all the predictions is calculated across all trees, and then this total is divided by the number of trees to produce an average. By averaging the predictions, the Random Forest algorithm reduces variability and improves accuracy, leveraging the collective decision-making of all the trees [23]. This ensemble approach allows the model to

generalize better to new, unseen data, mitigating overfitting and yielding a more reliable prediction compared to using a single decision tree [23].

3.5. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a machine learning algorithm designed to classify data by mapping it into a high-dimensional feature space using a nonlinear mapping function [24]. The training data, represented as vectors

$\vec{x}_i$, is classified into two categories, denoted as $\vec{y}_i$, which can take the values -1 or 1, as shown in the formula:

$$G = (\vec{x}_i \in \mathbb{R}^n; y_i = -1 \text{ or } 1; i = 1, 2, \ldots, N) \quad (5)$$

The goal of SVM is to find the optimal hyperplane that best separates the two classes in this feature space. The algorithm begins by identifying the points in each class that are closest to the separating hyperplane [24]—these points are known as the support vectors. Once the support vectors are determined, the distance between the hyperplane and these points is calculated. This distance is called the margin, and the primary objective of SVM is to maximize the margin. A larger margin indicates better generalization and separation between the classes, thus producing a more robust classifier. By maximizing the margin, SVM effectively minimizes the classification error on both the training and unseen data, ensuring high classification performance.

3.6. K-nearest Neighbor

k-NN algorithm is a simple and intuitive machine learning algorithm that classifies data points based on their proximity to other data points [25]. The core idea of k-NN is that a data point is classified by a majority vote of its neighbors, with the data point being assigned to the class most common among its k nearest neighbors. The formula for k-NN classification is as follows:

$$y = \frac{1}{k} \sum_{i=1}^{k} y_i \quad (6)$$

In this formula, y represents the predicted class for a new data point, and yi refers to the class labels of the

k-nearest neighbors. The algorithm identifies the k closest points (neighbors) to the data point in question by calculating the distance between points [25], typically using Euclidean distance, and then it takes the average or majority class of these neighbors to assign the label to the new point.

In practice, k-NN works by comparing a data point to its closest neighbors in feature space. The distance between data points is calculated, and the k-nearest neighbors are selected. The new data point is then assigned the most frequent class label from those neighbors [26]. The choice of k (the number of neighbors) significantly impacts the performance of the algorithm, with smaller values of k leading to more sensitive models that may be prone to noise, while larger values of k create smoother, more generalized models.

3.7. Evaluation Metrics

Evaluation metrics are essential for assessing the performance of classification models, as they determine how accurately a model predicts the correct outcomes. Accuracy is one of the primary metrics used to measure the ratio of correctly classified instances to the total number of instances. This metric provides an overall indication of model performance but may be insufficient when dealing with imbalanced datasets [26]. A confusion matrix is often used to provide more detailed insight into the model's performance by summarizing the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. This matrix is fundamental for calculating additional metrics such as Precision, Recall, and F1-score. As shown in Table 1.

Table 1. Confusion Matrix

	Positive	**Negative**
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Precision measures how many of the instances predicted as positive are actually positive. It is calculated by dividing the number of true positives by the sum of true positives and false positives:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (7)$$

Precision is especially important for applications where minimizing false positives is a priority.

Recall, also known as sensitivity or the true positive rate, calculates how many actual positive instances were correctly predicted by the model [27]. It is computed by dividing the number of true positives by the sum of true positives and false negatives:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (8)$$

Recall focuses on minimizing false negatives, making it crucial in situations where missing positive cases can have serious consequences, such as in medical diagnoses [27].

F1-score is a metric that balances precision and recall, combining them into a single score. It is particularly useful in imbalanced datasets where both precision and recall need to be optimized simultaneously. The formula for F1-score is the harmonic mean of precision and recall:

$$\text{F1} - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Preision+Recall}} \quad (9)$$

Finally, accuracy measures the proportion of correctly classified instances (both positives and negatives) out of the total number of instances, and is calculated using the formula:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (10)$$

4. Results and Discussion

4.1. Dataset Description

The dataset used in this study was obtained from Kaggle and comprised 10,000 samples with anthropometric and demographic features relevant to stunting detection. As shown in table 2.

Table 2. Description of Dataset Features for Stunting Detection

No	Feature	Type	Description
1	Gender	Categorical	Indicates the child's gender, either "Male" or "Female".
2	Age	Numerical	The age of the child, represented in years, a continuous feature.
3	Birth Weight	Numerical	The child's weight at birth, measured in kilograms. Important for assessing early growth patterns.
4	Birth Length	Numerical	The child's length at birth, measured in centimeters. Used to indicate initial growth status.
5	Body Weight	Numerical	The current weight of the child, measured in kilograms. Reflects the child's present health status.
6	Body Length	Numerical	The current length/height of the child, measured in centimeters. Key for assessing growth and development.
7	Breastfeeding	Categorical	Indicates whether the child has been breastfed (Yes/No). Significant for early development and nutrition.
8	Stunting	Binary	Indicates whether the child is stunted (Yes/No). Stunting reflects impaired growth due to malnutrition.

These features included variables such as age, gender, birth weight, current weight, birth length, current height, and breastfeeding status. As shown in table 3.

 ISSN 2723-6471

Table 3. Sample Data from Stunting Detection Dataset

	Gender	Age	Birth Weight	Birth Length	Body Weight	Body Length	Breastfeeding	Stunting
0	Male	17	3.0	49	10.0	72.2	No	No
1	Female	11	2.9	49	2.9	65.0	No	Yes
2	Male	16	2.9	49	8.5	72.2	No	Yes
3	Male	31	2.8	49	6.4	63.0	No	Yes
4	Male	15	3.1	49	10.5	49.0	No	Yes

However, the dataset exhibited a significant class imbalance, with only 15% of the samples labeled as stunted children, while the remaining 85% represented non-stunted children. This imbalance posed a challenge for model training, as the minority class (stunted children) was underrepresented, making it harder for models to accurately detect stunting without applying techniques to address the imbalance.

4.2. Data Preprocessing

Before applying machine learning models, the dataset underwent several important preprocessing steps to ensure data consistency and prepare it for analysis. The first step involved removing approximately 2427 duplicate entries, which reduced the dataset to 7.573 unique instances. This step was crucial to eliminate redundancy and avoid skewing the analysis. Next, missing data in the dataset were addressed. For numerical features such as age, weight, and height, missing values were imputed using the mean or median to maintain consistency, while for categorical features like gender, the mode was used to fill in missing entries. As shown in table 4, Normalized and Encoded Data from Stunting Detection Dataset.

Table 4. Normalized and Encoded Data from Stunting Detection Dataset

	Gender	Age	Birth Weight	Birth Length	Body Weight	Body Length	Breastfeeding	Stunting
0	1	0.246282	0.798750	-0.194058	1.344380	0.327599	0	0
1	0	-0.453041	0.462847	-0.194058	-2.684189	-0.438284	0	1
2	1	0.129728	0.462847	-0.194058	0.493274	0.327599	0	1
3	1	1.878036	0.126945	-0.194058	-0.698275	-0.651030	0	1
4	1	0.013174	1.134652	-0.194058	1.628082	-2.140248	0	1

The correlation matrix illustrates the relationships between key features in the stunting detection dataset, including Gender, Age, Birth Weight, Birth Length, Body Weight, and Stunting. As shown in figure 2. Correlation Matrix of Key Features.

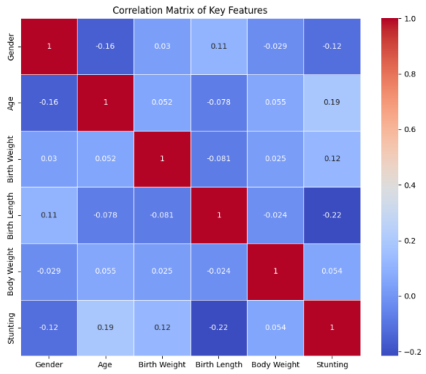

Figure 2. Correlation Matrix of Key Features

 ISSN 2723-6471

The matrix visually represents the strength and direction of the correlations, with values ranging from -1 to 1. Positive correlations, shown in shades of blue, indicate that as one variable increases, the other tends to increase as well. For instance, the positive correlation between Birth Weight and Body Weight suggests that children with higher birth weights are likely to have higher body weights as they grow. This relationship is important for understanding growth patterns in children.

Negative correlations, represented in shades of red, show an inverse relationship, meaning that as one variable increases, the other tends to decrease. However, in this dataset, no strong negative correlations are observed. Variables that have low or near-zero correlation indicate weak or no linear relationship, implying that changes in one variable do not predict changes in another. For example, Gender typically shows weak correlations with anthropometric features like Body Weight or Birth Length, indicating that these characteristics do not strongly vary by gender.

The variable of primary interest, Stunting, shows a moderate negative correlation with Body Weight and Birth Weight, suggesting that lower birth or body weights are associated with higher rates of stunting. These correlations are crucial for identifying which variables are most predictive of stunting, and they provide insights for feature selection during model development. Understanding these relationships helps enhance the accuracy and relevance of machine learning models used for stunting detection.

4.3. Performance Without SMOTE and Backward Elimination

Here is the table representing the performance of the four machine learning models without SMOTE and backward elimination, as shown in Table 5.

Table 5. Performance of Machine Learning Models Without SMOTE and Backward Elimination

Model	Recall	F1score	ROC-AUC	Confusion Matrix
XGBoost	0.9423	0.9066	0.7656	[[98,170], [72,1175]]
Random Forest	0.9591	0.9057	0.7670	[[70,198], [51,1196]]
SVM (Polynomial)	0.9663	0.9043	0.7036	[[55,213], [42,1205]]
KNN	0.9214	0.8910	0.6720	[[85,183], [98,1149]]

The performance comparison of the four machine learning models, XGBoost, Random Forest, Support Vector Machine with a polynomial kernel, and k-Nearest Neighbors, for the stunting detection dataset raises important trade-offs among the different performance metrics. While accuracy, the overall correctness of the models, is highest for XGBoost at 0.8403, in healthcare contexts, accuracy is just not sufficient, since it does not allow for the trade-off between false positives and false negatives. Precision, representing the ratio of true positives in relation to the instances classified as positive, is highest for XGBoost, with 0.8736, and reflects its usefulness for minimizing false positives, a very important aspect of reducing unnecessary interventions in children who are not stunted. Note that the most important class-weighted recall for healthcare applications-that is, for reducing the number of false negatives in healthcare applications-is highest for SVM at 0.9663, hence best capability to find stunted cases and avoid missing children in need of intervention.

The F1-Score, a balanced metric harmonizing precision and recall, is very similar for XGBoost at 0.9066 and Random Forest at 0.9057, reflecting their competence in strong trade-offs between the competing priorities. In terms of the ROC-AUC, which informs about the ability to distinguish between stunted and non-stunted cases, the highest ranking goes to Random Forest at 0.7670, tailed closely by XGBoost at 0.7656, reflecting robust performance in overall class separation. However, k-NN has worse metrics on most parameters: its rate of misclassification is higher, which makes it less suitable for this application.

In healthcare contexts, these are even more critical trade-offs. The excellent precision from XGBoost and, correspondingly, a very well-balanced F1-Score make it ideal for scenarios where keeping the number of false alarms low is just about as important as high recall. On the other hand, high recall with SVM makes it of course compelling when all cases of stunted need to be detected at the cost of increased false positives. In contrast, Random Forest also performed competitively, striking a reasonable balance between precision and recall. This analysis underlines that the

 ISSN 2723-6471

inclusion of accuracy, precision, recall, and F1-score gives broad insight into model performance, satisfying the needs of the specific application of healthcare in stunting detection.

Before the application of SMOTE, the performance of the XGBoost model was very impressive, with an accuracy of 83.43%, a precision of 86.30%, and the highest recall among the models of 94.95%. As shown in figure 3.

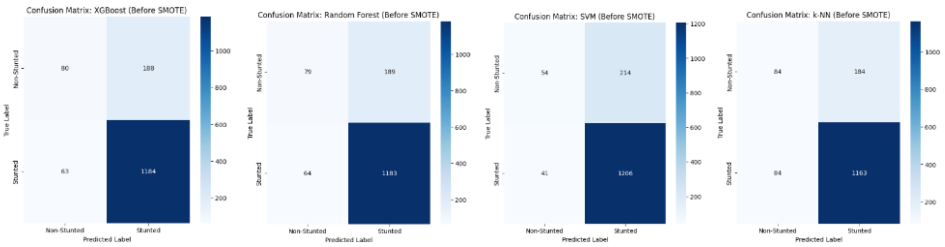

Figure 3. Confusion Matrix Before SMOTE

This is further reflected in its high F1-score of 90.42%, which means that the precision and recall are balanced. However, a ROC-AUC value of 0.7678 suggests there is still room for improvement in the model's power to distinguish between positive and negative classes across various thresholds. The Random Forest model also performed similarly to XGBoost, with an accuracy of 83.30% and precision of 86.22%, though its recall was slightly lower at 94.87%. This yielded a respectable F1-score of 90.34%, somewhat lower than that obtained from XGBoost, but with a much lower value of ROC-AUC at 0.7577.

In contrast, the SVM model yielded a high recall of 96.71%, meaning it can capture positive cases with very high efficacy. Still, its precision was a bit lower at 84.93%, hence giving an F1-score of 90.44%, with an ROC-AUC value of 0.7057, lower than the other models, indicating difficulties in the consistent class distinction provided by this model. The k-Nearest Neighbors model gave the lowest accuracy of 82.31%, with a relatively good precision of 86.34%. However, Its recall was lower at 93.26%, giving an F1-score of 89.67%. This model had the lowest ROC-AUC value amongst all models at 0.6914, presenting the weakest discriminative capability.

4.4. Performance After Using SMOTE and Backward Elimination

To address the class imbalance in the dataset, the SMOTE was applied specifically to the training set. SMOTE works by generating synthetic samples of the minority class (stunted children) based on their nearest neighbors, effectively balancing the class distribution. This resampling process ensured that the models had a more balanced dataset to learn from, making them better equipped to detect stunted cases, which were previously underrepresented.

Once SMOTE was applied, the same four machine learning models—XGBoost, Random Forest, SVM, and k-NN—were retrained on the newly balanced dataset. In addition to retraining, the models underwent further hyperparameter tuning to adjust to the updated data distribution. As shown in figure 4.

```
# --- Apply SMOTE to Address Class Imbalance ---
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X, y)

# Split resampled dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=0.2, random_state=42)
```

```
# --- Backward Elimination (Remove "Body Length" and "Breastfeeding") ---
df.drop(columns=['Body Length', 'Breastfeeding'], inplace=True)
```

Figure 4. Code Implementation of the SMOTE and Backward Elimination Techniques

This fine-tuning helped optimize model performance by taking into account the changes in the dataset's structure and ensuring that the models could make more accurate predictions.

 ISSN 2723-6471

The ROC-AUC curves in this figure show the comparative performance of four machine learning models—XGBoost, Random Forest, SVM with a polynomial kernel, and k-NN—on the stunting detection dataset after applying the SMOTE and backward elimination. As shown in figure 5.

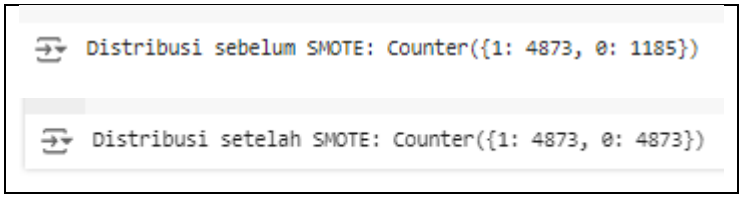

Figure 5. The result of applying SMOTE to the imbalanced stunting detection dataset

The ROC-AUC represents the models' ability to distinguish between stunted and non-stunted children, with higher curves indicating better overall classification performance.

After implementing SMOTE to address class imbalance and backward elimination to remove less relevant features, all models demonstrated improved performance. XGBoost achieved the highest ROC-AUC value of 0.9311, reflecting its superior ability to correctly classify stunted children while minimizing false positives and false negatives. Random Forest followed closely with a ROC-AUC score of 0.8931, indicating strong classification performance as well.

Both k-NN and SVM also showed noticeable improvements, with ROC-AUC scores of 0.8467 and 0.7443, respectively, compared to their pre-SMOTE and backward elimination performance. These results confirm that the application of SMOTE and backward elimination significantly enhances the models' ability to correctly detect stunting, particularly for XGBoost and Random Forest, making them more effective in real-world healthcare scenarios where accurate classification is crucial.

After retraining, the models' performance was evaluated on the original, unbalanced test set to maintain real-world relevance. As shown in Table 6.

Table 6. Performance Metrics of Different Models for Stunting Detection after implementing technique SMOTE and Backward Elimination

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Confusion Matrix
XGBoost	0.8574	0.8413	0.8914	0.8656	0.9311	[[975, 212], [137, 1124]]
Random Forest	0.8207	0.8246	0.8279	0.8263	0.8931	[[965,222], [217,1044]]
SVM (Polynomial Kernel)	0.6977	0.6729	0.8041	0.7327	0.7443	[[694, 493], [247,1014]]
k-NN	0.7708	0.7966	0.7454	0.7702	0.8467	[[947, 240], [321,940]]

The focus of the evaluation was on key metrics such as recall and ROC-AUC, which are particularly critical in healthcare settings. Given that stunting detection is a sensitive issue, minimizing false negatives (i.e., ensuring that all stunted children are correctly identified) was prioritized. ROC-AUC was also used to assess how well the models distinguished between stunted and non-stunted cases, ensuring that the overall detection capability remained high.

After applying SMOTE, the XGBoost model significantly improved ROC-AUC, which increased to 93.11%, indicating better discrimination between classes. As shown in figure 6.

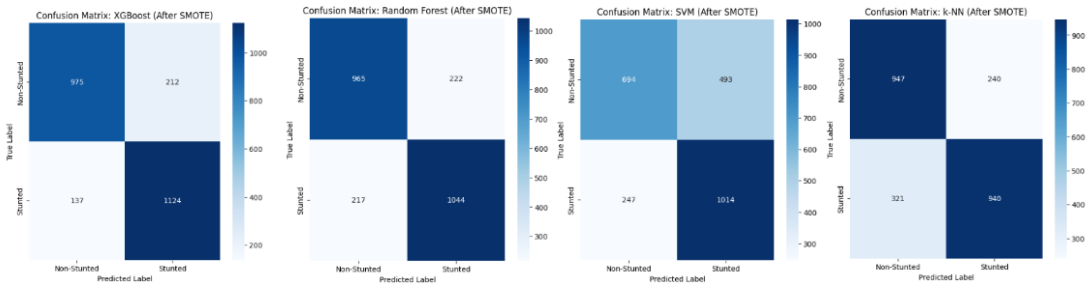

Figure 6. Confusion Matrix After Using Technique SMOTE

 ISSN 2723-6471

Although recall slightly decreased from 94.95% to 89.14%, accuracy improved to 85.74%, reflecting SMOTE's impact in addressing data imbalance while maintaining strong performance. In contrast, the performance of the Random Forest model declined after SMOTE, with accuracy dropping to 82.07%, precision at 82.46%, and recall at 82.79%, resulting in an F1-score of 82.63%. However, its ROC-AUC increased to 89.31%, suggesting that SMOTE enhanced the model's discriminative ability despite the overall performance decline. The SVM model experienced a significant drop in performance following SMOTE, with accuracy falling to 69.77% and an F1-score of 73.27%. ROC-AUC remained low at 74.43%, highlighting the model's difficulty handling the balanced dataset. Additionally, lower precision at 67.29% indicated an increase in false positive classifications. Similarly, the k-NN model experienced a reduction in accuracy, which dropped to 77.08%. Precision and recall were recorded at 79.66% and 74.54%, respectively, resulting in an F1 score of 77.02%. However, its ROC-AUC increased to 84.67%, reflecting improved discriminative ability, even as overall performance metrics declined.

Comparison among the performances of the four models in this study using metrics such as recall, F1-score, ROC-AUC, and accuracy reveals that the models significantly differ in their performance on both the imbalanced and the balanced dataset resulting after SMOTE. As shown in figure 7.

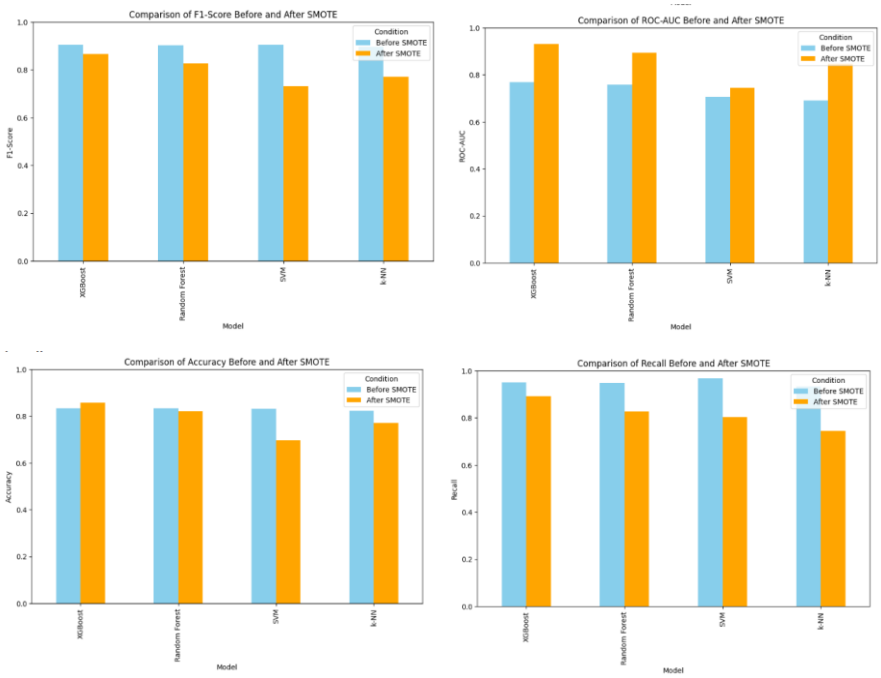

Figure 7. Comparison Performance Recall, F1-Score, ROC-AUC, and Accuracy avoid Four Modell Classification

Recall, which measures how well the model detects true positives, was highest from SVM before SMOTE at 96.71%, dropping significantly to 80.41% after SMOTE. In contrast, XGBoost maintained relatively high recall values, from 94.95% (before SMOTE) to 89.14% (after SMOTE), showing more stability than the other models. This suggests XGBoost's robustness in capturing positive cases even with a balanced dataset.

For the F1-score, which balances precision and recall, XGBoost excelled with 90.42 before SMOTE and 86.56 after SMOTE. This highlights its consistent performance, unlike SVM and k-NN, which saw notable declines after SMOTE. Speaking directly to the model's ability to differentiate between classes, as indicated by ROC-AUC, XGBoost saw quite a remarkable improvement after SMOTE, from 0.7678 to 0.9311-highest among the models. Random Forest improved in ROC-AUC from 0.7577 to 0.8931, but this same improvement was not shared explicitly in SVM and k-NN.

The accuracy that measures the overall correctness was consistently the highest for XGBoost both before (83.43%) and after SMOTE (85.74%), further emphasizing its adaptability for handling an imbalanced dataset to a balanced one. Random Forest trailed closely in its ranking by accuracy before SMOTE with 83.30%, although its performance

declined a bit after balancing the dataset. Meanwhile, both SVM and k-NN suffered dramatic decreases in accuracy after using SMOTE to balance the dataset.

Among the metrics, ROC-AUC is the most critical in finding out what model performed the best since it reflects the ability to distinguish between the positive and negative classes over different thresholds and provides a complete picture of model performance. Recall and F1-score are essential, but they only offer some photos of the model's discrimination capability. ROC-AUC shows that the best performance came from XGBoost, with the most consistent and significant gain post-SMOTE. The immediate implication is that XGBoost can maintain balanced performance along all critical metrics, making it the most reliable model for this task.

5. Conclusion

This study demonstrates the effectiveness of applying machine learning techniques combined with data balancing strategies for stunting detection in children. By addressing the class imbalance in the dataset through the SMOTE and optimizing feature selection using backward elimination, we were able to significantly enhance the performance of machine learning models, particularly in terms of recall and ROC-AUC, which are critical in healthcare applications.

Among the models evaluated, XGBoost emerged as the best-performing model, achieving an accuracy of 85.74%, a recall of 89.14%, and an ROC-AUC of 93.11% after applying SMOTE and backward elimination. Random Forest also performed well, demonstrating high accuracy and class separation capability. SVM and k-NN, while slightly less effective, still showed improvements after applying the balancing and feature selection techniques.

The results indicate that handling class imbalance and refining feature sets are crucial steps for improving machine learning models' ability to accurately detect stunting. The findings of this study can guide healthcare practitioners and researchers in adopting machine learning for early stunting detection, enabling timely interventions to mitigate long-term impacts on children's growth and development.

6. Declarations

6.1. Author Contributions

Conceptualization: T.S., B.W., M., A.F.A., and H.A.A.; Methodology: M.; Software: T.S.; Validation: T.S., M., and H.A.A.; Formal Analysis: T.S., M., and H.A.A.; Investigation: T.S.; Resources: M.; Data Curation: M.; Writing Original Draft Preparation: T.S., M., and H.A.A.; Writing Review and Editing: M., T.S., and H.A.A.; Visualization: T.S. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Ipa, Y. Yuliasih, E. P. Astuti, A. D. Laksono, and W. Ridwan, "STAKEHOLDERS' ROLE IN THE IMPLEMENTATION OF STUNTING MANAGEMENT POLICIES IN GARUT REGENCY," *Jurnal Administrasi Kesehatan Indonesia*, vol. 11, no. 1, pp. 26–35, Jun. 2023, doi: 10.20473/jaki.v11i1.2023.26-35.

[2] S. Munawaroh, M. N. Fajri, and S. R. Ajija, "THE EFFECTS OF SOCIAL ASSISTANCE PROGRAMS ON STUNTING PREVALENCE RATES IN INDONESIA," *Indonesian Journal of Health Administration*, vol. 12, no. 1, pp. 74–85, Jun. 2024, doi: 10.20473/jaki.v12i1.2024.74-85.

[3] R. K. Sari, C. P. Mayangsari, I. D. Mashoedi, Y. S. N. Intan, S. Trisnadi, and D. F. Aprilyanti, "Strengthening emotional intelligence intervention on behavior changes of mothers in stunting prevention," *International Journal of Public Health Science (IJPHS)*, vol. 13, no. 2, p. 536, Jun. 2024, doi: 10.11591/ijphs.v13i2.23652.

[4] F. Hilali Moh'd, K. Anwar Notodiputro, and Y. Angraini, "Enhancing interpretability in random forest: Leveraging inTrees for association rule extraction insights," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 4, p. 4054, Dec. 2024, doi: 10.11591/ijai.v13.i4.pp4054-4061.

[5] E. H. Nugrahani, S. Nurdiati, F. Bukhari, M. K. Najib, D. M. Sebastian, and P. A. N. Fallahi, "Sensitivity and feature importance of climate factors for predicting fire hotspots using machine learning methods," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 2, pp. 2210–2223, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp2212-2225.

[6] N. Effendy, M. Z. A. Fadhilah, D. W. Kraton, and H. A. Abrar, "The prediction of thermal sensation in building using support vector machine and extreme gradient boosting," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 3, pp. 2963–2970, Sep. 2024, doi: 10.11591/ijai.v13.i3.pp2963-2970.

[7] A. Febriani, "Improved Hybrid Machine and Deep Learning Model for Optimization of Smart Egg Incubator," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1052–1068, Sep. 2024, doi: 10.47738/jads.v5i3.304.

[8] H. A. Abdelhafez and A. A. Amer, "Machine Learning Techniques for Diabetes Prediction: A Comparative Analysis," *Journal of Applied Data Sciences*, vol. 5, no. 2, pp. 792–807, May 2024, doi: 10.47738/jads.v5i2.219.

[9] D. A. Kristiyanti, S. A. Sanjaya, V. C. Tjokro, and J. Suhali, "Dealing imbalance dataset problem in sentiment analysis of recession in Indonesia," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 2, pp. 2058–2070, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp2060-2072.

[10] W. Chimphlee and S. Chimphlee, "Hyperparameters optimization XGBoost for network intrusion detection using CSE-CIC-IDS 2018 dataset," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, pp. 817–826, Mar. 2024, doi: 10.11591/ijai.v13.i1.pp817-826.

[11] C. Angelica, C. Charleen, and A. Wibowo, "Elevating fraud detection: machine learning models with computational intelligence optimization," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 4, p. 4273, Dec. 2024, doi: 10.11591/ijai.v13.i4.pp4273-4280.

[12] S. Makubhai, G. R. Pathak, and P. R. Chandre, "Comparative analysis of explainable artificial intelligence models for predicting lung cancer using diverse datasets," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 2, pp. 1978–1989, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp1980-1991.

[13] R. Govindarajan, V. Balaji, J. Arumugam, T. A. Assegie, and R. Mothukuri, "Evaluation of sequential feature selection in improving the K-nearest neighbor classifier for diabetes prediction," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 2, pp. 1567–1573, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp1567-1573.

[14] S. H, "Multi-Label Feature Aware XGBoost Model For Student Performance Assessment Using Behavior Data in Online Learning Environment," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 4, p. 4537, Dec. 2024, doi: 10.11591/ijai.v13.i4.pp4537-4543.

[15] I. Slamet, "Retinopathy Classification using Convolutional Neural Network Method with Adaptive Momentum Optimization and Applied Batch Normalization," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1123–1133, Sep. 2024, doi: 10.47738/jads.v5i3.309.

[16] S. Armoogum, "Breast Cancer Prediction Using Metrics-Based Classification," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1508–1519, Sep. 2024, doi: 10.47738/jads.v5i3.351.

[17] I. E. Purba, Y. G. Tarigan, A. Zendrato, A. Purba, and T. Sinaga, "Maternal factors associated with stunting among children under two years in South Nias, Indonesia: a cross-sectional study," *International Journal of Public Health Science (IJPHS)*, vol. 13, no. 3, p. 1349, Sep. 2024, doi: 10.11591/ijphs.v13i3.24316.

[18] R. Tanjung, D. Lestrina, and J. Sinaga, "Spatial analysis of environmental sanitation and stunting incidents," *International Journal of Public Health Science (IJPHS)*, vol. 13, no. 4, p. 1968, Dec. 2024, doi: 10.11591/ijphs.v13i4.23442.

[19] M. K. Anam, S. Defit, Haviluddin, L. Efrizoni, and M. B. Firdaus, "Early Stopping on CNN-LSTM Development to Improve Classification Performance," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1175–1188, Sep. 2024, doi: 10.47738/jads.v5i3.312.

[20] A. R. Hananto, "Identifying Student Learning Styles Using Support Vector Machine in Felder-Silverman Model," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1495–1507, Sep. 2024, doi: 10.47738/jads.v5i3.337.

[21] E. K. Anku and H. O. Duah, "Predicting and identifying factors associated with undernutrition among children under five years in Ghana using machine learning algorithms," *PLoS One*, vol. 19, no. 2 February, Feb. 2024, doi: 10.1371/journal.pone.0296625.

[22] H. Shen, H. Zhao, and Y. Jiang, "Machine Learning Algorithms for Predicting Stunting among Under-Five Children in Papua New Guinea," *Children*, vol. 10, no. 10, Oct. 2023, doi: 10.3390/children10101638.

[23] A. Azis, "Application of the Vector Machine Support Method in Twitter Social Media Sentiment Analysis Regarding the Covid-19 Vaccine Issue in Indonesia," *Journal of Applied Data Sciences*, vol. 2, no. 3, pp. 102–108, 2021.

[24] P. S. Siregar, R. G. Hatika, and B. H. Hayadi, "Multiple Choice Question Difficulty Level Classification with Multi Class Confusion Matrix in the Online Question Bank of Education Gallery," *Journal of Applied Data Sciences*, vol. 4, no. 4, pp. 392–406, Dec. 2023, doi: 10.47738/jads.v4i4.132.

[25] L. Jen and Y.-H. Lin, "A Brief Overview of the Accuracy of Classification Algorithms for Data Prediction in Machine Learning Applications," *Journal of Applied Data Sciences*, vol. 2, no. 3, pp. 84–92, 2021.

[26] A. Suryaputra Paramita, I. Maryati, and L. M. Tjahjono, "Implementation of the K-Nearest Neighbor Algorithm for the Classification of Student Thesis Subjects," *Journal of Applied Data Sciences*, vol. 3, no. 3, pp. 128–136, 2022.

[27] N. Trianasari and T. A. Permadi, "Analysis of Product Recommendation Models at Each Fixed Broadband Sales Location Using K-Means, DBSCAN, Hierarchical Clustering, SVM, RF, and ANN," *Journal of Applied Data Sciences*, vol. 5, no. 2, pp. 636–652, May 2024, doi: 10.47738/jads.v5i2.210.