LONTAR KOMPUTER VOL. 16, NO. 2 AUGUST 2025 **p-ISSN 2088-1541**
DOI : 10.24843/LKJTI.2025.v16.i2.p03 **e-ISSN 2541-5832**
Accredited Sinta 2 by RISTEKDIKTI Decree No. 158/E/KPT/2021

Breaking Class Imbalance: Machine Learning Solutions for Stunting Detection

Hasna Aqila Raihana[a1], Putu Harry Gunawan[a2], Narita Aquarini[b3]
[a]Data Science, School of Computing, Telkom University, Bandung, Indonesia
[b]Ecole Doctorale Science Economics, Université de Poitiers Intervenant Finance, La Rochelle, France
[1]hasnaaqilar@student.telkomuniversity.ac.id
[2] phgunawan@telkomuniversity.ac.id (Corresponding author), [3]aquarinin@excelia-group.com

Abstract

Stunting is a critical public health issue primarily caused by malnutrition, which hampers the growth of children. This paper evaluates the performance of two machine learning models, K-Nearest Neighbors (KNN) and Decision Tree, in classifying stunting status in toddlers. Three strategies for handling class imbalance, no sampling, Synthetic Minority Over-sampling Technique (SMOTE), and random undersampling, are compared to enhance the detection of the minority class (stunting). The results demonstrate that KNN with SMOTE provides the most effective and balanced performance in stunting detection, as evidenced by achieving an F1-score of 0.99. This value indicates the model's superior ability to balance precision and recall for the minority class. Conversely, although the Decision Tree model without sampling techniques achieved an accuracy 98.64%*, its performance in stunting detection is less reliable due to a lower F1-Score. The application of random under sampling caused a significant decline in performance for both models. These findings underscore the effectiveness of SMOTE in handling class imbalance for stunting detection and provide valuable insights into the application of machine learning techniques in addressing public health issues.*

Keywords: *Class Imbalance, Decision Tree, K-Nearest Neighbors, Machine Learning, Random Under Sampling, SMOTE, Stunting*

1. Introduction

Stunting is a condition that affects the growth of toddlers, primarily caused by malnutrition. As a result of stunting, a child's height becomes shorter than that of other children of the same age [1] Nutritional deficiencies in toddlers begin during pregnancy and continue after birth, with stunting typically becoming noticeable after the child reaches the age of two [2]. Several factors can contribute to stunting in children, such as poor parenting practices, inadequate antenatal care for mothers, and a lack of nutritious food. In addition to these internal factors, external factors such as social, economic, cultural, and political influences also play a role in causing stunting [3].

As of 2020, 149 million toddlers worldwide experienced stunting, with the majority in Asia (54.8%) [4]. Reducing stunting in Indonesia is a key focus of government policy, as highlighted in Presidential Regulation No. 72 of 2021 on the Acceleration of Stunting Reduction. The current stunting prevalence is 21.6%, with a target to reduce it to 14% by 2024 [5]. In line with this, a campaign program providing free lunch and milk for toddlers, pregnant women, and children—benefiting a total of 82.9 million recipients—aims to tackle stunting and support Indonesia's vision of Golden Indonesia 2045 [6].

Stunting is a significant health issue as it can hinder the optimal growth and development of children. The main factors influencing a child's development include health conditions and nutritional intake, including malnutrition during pregnancy, which can affect fetal growth. If these factors persist until the child reaches the age of two, the child is at risk of experiencing delays in growth and development, which is reflected in weight and height that do not meet the standards set by the WHO [7]. Stunting can reduce a child's intelligence and physical capacity, which can lower productivity, slow economic growth, and exacerbate poverty. Furthermore, the impact of

stunting can result in a weakened immune system and increase the risk of chronic diseases, as well as reproductive disorders in women during adulthood [8].

With the advancement of technology, various studies have shown that machine learning (ML) methods have been effectively used to classify stunting in children in regions such as Africa [9]. Machine learning methods have proven to be superior in classification problems compared to traditional statistical methods. ML offers greater flexibility, can handle multidimensional correlations, and can utilize a larger set of predictors [10].

His paper applies two machine learning methods, namely K-Nearest Neighbors (KNN) and Decision Tree. KNN was chosen for its ability to generate accurate and clean data, as well as its effectiveness when implemented on sufficiently large datasets [11]. The Decision Tree method was used due to its ability to identify and analyze the relationships between various factors affecting a particular issue, and to help find the best solution by considering these factors [12]. The KNN method has previously been successfully applied to classify stunting status in toddlers in Bojongemas Village. A paper by Sri et al. (2024) using data from 503 toddlers with attributes such as age, weight, height, and nutritional status showed that the KNN model was able to classify stunting status with an accuracy rate of 92%, thus proving effective in monitoring child growth [13]. Additionally, the Decision Tree method was successfully applied by Amanda et al. (2023). This paper aimed to implement Decision Tree and SVM algorithms to predict the risk of stunting in Dumai City, using 18 attributes and 5021 data points. The Decision Tree method achieved an accuracy of 96.15%, which was higher than the accuracy of the SVM method, which only reached 62.48%, thus making the Decision Tree method more effective in predicting the risk of stunting [14].

In this paper, the dataset used shows an imbalanced class, where toddlers without stunting significantly outnumber those with stunting, potentially affecting classification results. A comparison is made between methods with and without sampling techniques to tackle this issue. To improve stunting classification accuracy, sampling techniques such as SMOTE and Random Under Sampling are used, as they are effective in handling class imbalance. SMOTE has demonstrated improvements in the accuracy of KNN, Decision Tree, and Random Forest models in employee promotion classification tasks [15]. while RUS has increased both the accuracy and precision of KNN in IoT attack detection [16]. and improved Decision Tree performance in forest fire prediction, achieving high accuracy and ROC values [17]. Optimal hyperparameter selection is crucial to obtaining good model accuracy. Grid Search using Cross Validation provides ease in testing each model parameter without having to perform manual validation one by one, and it can automatically find the best hyperparameter combination [18]. Therefore, the integration of SMOTE, RUS, and GridSearchCV techniques in this paper aims to improve the accuracy and reliability of the model in classifying stunting.

This paper aims to implement the KNN and Decision Tree methods for classifying stunting in toddlers and evaluate their performance to identify the most effective algorithm. It also seeks to determine the factors influencing the performance of both algorithms. Using a machine learning approach, this paper aims to create an accurate and reliable classification model to help healthcare professionals, and the government detect and address stunting more effectively. By identifying the causes and impacts of stunting and supporting nutrition program monitoring, the research contributes to stunting prevention and improving child health quality in Indonesia.

2. Reseach Methods

The research methodology starts with a literature review on stunting and classification algorithms. The dataset, from Bojongsoang Public Health Center, focuses on toddler stunting data. After data exploration and preprocessing (including cleaning and feature selection), both downsampling (Random Under Sampling) and upsampling (SMOTE) are applied to address class imbalance. The data is split into 80% training and 20% test sets. Hyperparameter tuning is performed using GridSearchCV, followed by the implementation of KNN and Decision Tree models. These models are evaluated on accuracy, precision, recall, and F1-score. The process is repeated until all scenarios are tested, with the best model selected. The research flowchart is shown in Figure 1.

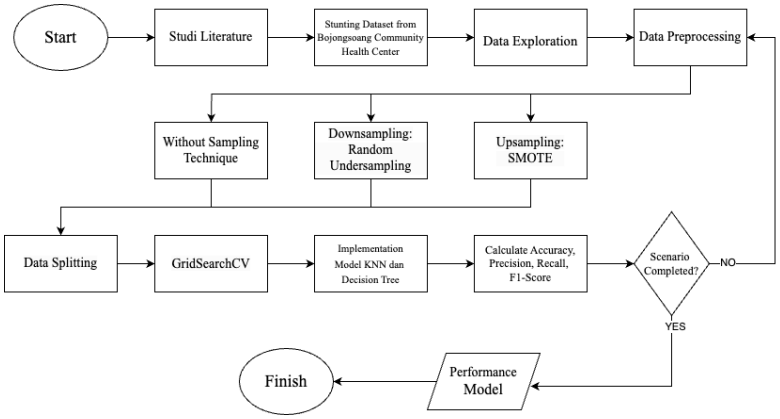


Figure 1. Research Flowchart

2.1. Dataset

The dataset used in this paper, obtained from the Bojongsoang Health Center in CSV format, covers data from August 2024, coinciding with the Toddler Weighing Month and Vitamin A administration activities, ensuring high community participation and completeness. This data forms the basis for developing a stunting detection model. Yuxiang et al. (2024) used gender and age in months to predict stunting risk [18]. Karisma et al. (2022) analyzed birth length and weight in relation to stunting [19]. and Sri et al. (2024) applied weight and height for stunting classification using machine learning [13]. In this paper, seven features are used, including Gender, Birth Weight, Birth Length, Weight, Height, Age in Months, and Result, where the Result contains a value of 1 for stunting and 0 for normal, which is generated from the Z-score of Height-for-Age.

2.2. Stunting

Stunting is a condition commonly caused by malnutrition (including deficiencies in protein, energy, and micronutrients), which can affect a baby's development from the prenatal stage up until birth. Several factors contribute to stunting, such as the mother's body size, the nutrition received during pregnancy, and fetal growth [20]. Factors related to stunting can be observed from the baby's length, which is categorized as either short or very short, determined by calculating the Z-score. Based on the Z-score calculation, there are two categories short, with a range between -3 SD and -2 SD, and very short, with a Z-score below -3 SD [21]. The formula for calculating the Z-score can be found in [21].

2.3. K-Nearest Neighnors (KNN)

The K-Nearest Neighbors (KNN) algorithm is known as a non-parametric data mining algorithm that can be applied to classification and regression tasks. KNN is a supervised learning algorithm that requires training data to classify an object based on its proximity to other data points [22]. The KNN algorithm identifies new data by measuring the shortest distance to the K previously selected neighbors. The class prediction for the new data is made by identifying the most frequent class among the nearest neighbors after all K neighbors are accounted for [23]. The KNN algorithm has been proven to be effective in classifying data [13]. The most used distance measure in the KNN algorithm is Euclidean distance. The formula can be found in [23].

2.4. Decision Tree

The Decision Tree algorithm uses a flowchart-like structure, where each node represents a decision point based on attribute tests of existing variables. It is widely used in classification and prediction analysis, producing a clear and structured model [15]. The decision tree applies a series of rules, represented as branches, to make decisions. At each node, it splits the dataset based on the feature with the highest information gain. This process continues iteratively until reaching the terminal leaf nodes, where the final decision is made based on the dominant class in that node [24]. The goal of the Decision Tree method is to visualize and generate decisions based on specific rules and conditions [25]. The general formula for the Decision Tree can be found in [25].

2.5. Synthetic Minority Oversampling Technique (SMOTE)

SMOTE (Synthetic Minority Over-sampling Technique) addresses class imbalance by increasing data points in the minority class. It creates synthetic samples through random linear combinations of nearby minority class data in the feature space, generating new instances that resemble the minority class without direct duplication [26]. The SMOTE technique begins by randomly selecting samples from the minority class and identifying their KNN. The distance between the sample and each nearest neighbor is then calculated. Synthetic data is generated by interpolating between the selected sample and its neighbors, using random values between 0 and 1 to scale the differences. This process is repeated until the number of samples in the minority class is increased to match that of the majority class [27].

2.6. Random Under-Sampling (RUS)

Random Under Sampling (RUS) is a method that randomly reduces the number of samples from the majority class to balance it with the minority class, making the dataset more proportional before being used in model training. This technique aims to address data imbalance, which can affect the model's performance, by reducing data from the majority class, thereby achieving a more balanced class distribution [16]. This sampling technique is used to randomly decrease the number of instances in the majority class until the class distribution becomes balanced [17]. It is effective in improving the performance of machine learning models, allowing for more optimal classification of both classes [16].

2.7. Grid Search Cross Validation

Grid Search Cross-Validation is a popular method for tuning hyperparameters in machine learning classification algorithms, where all combinations of hyperparameters are tested using cross-validation to find the best model. Although time-consuming, this method is widely used due to its ability to select the combination of parameters that produces optimal predictions [28]. This technique allows for the automatic exploration of various hyperparameter combinations, speeding up the search for the optimal hyperparameters without manual experimentation, and resulting in better-performing models [29][30].

In this paper, hyperparameter optimization was performed on two models with different parameter sets. The KNN model depends on three main hyperparameters. The n_neighbors parameter is crucial for determining the number of nearest neighbors (k value) that will serve as the basis for classification. Neighbor determination is based on the metric that functions to measure distance, this paper employs the Euclidean metric to calculate proximity between data points. Finally, weights regulate the voting scheme of these neighbors, where uniform provides equal influence on all neighbors, while distance gives greater influence on neighbors with closer proximity [31]. Meanwhile, the Decision Tree model is governed by a series of hyperparameters to build an efficient structure. The quality of node splitting is determined by the criterion such as Gini or Entropy that measures data purity, while the splitting process is controlled by the splitter using either best or random approaches. To prevent overfitting and control complexity, three main parameters are utilized max_depth to limit tree depth, min_samples_split and min_samples_leaf to establish the minimum number of data points required for splitting and leaf formation. Additionally, the class_weight parameter is tested with two options None, which treats all classes equally, and balanced, which automatically adjusts weights to provide greater attention to minority classes to handle imbalanced data [32].

The selection of hyperparameter ranges was conducted with the objective of testing various conditions and finding the most suitable combination. For parameters such as n_neighbors in KNN and max_depth in Decision Tree, a range of 1 to 10 was chosen to test various levels of model complexity, from the simplest to the most complex. This approach enables the identification of optimal balance, where the model is neither too rigid (underfit) nor too specific (overfit). Similarly, ranges of 2-10 for min_samples_split and 1-10 for min_samples_leaf were tested to ensure that the rules formed by the decision tree have sufficient data points, making the results more reliable. Meanwhile, for categorical parameters such as weights, criterion, splitter, and class_weight, all standard options were tested with the aim of comparing different strategies, for instance, whether distance ‘weight’ performs better than ‘uniform’, or whether ‘gini’ criterion is more effective than ‘entropy’, to determine which approach is most suitable for the data characteristics in this paper.

2.8. Performance Evaluation Metrics

The confusion matrix is used to measure the performance of a classification model by comparing the predicted results with the test data in four main categories TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative). Model performance is assessed using several key metrics. Precision focuses on the accuracy of positive predictions, measuring the proportion of items flagged as 'positive' that were correct. In contrast, recall (or sensitivity) measures the model's ability to identify all true positive instances within the dataset. To provide a balanced view between these two, the F1-Score is utilized as the harmonic mean of precision and recall, it is an especially valuable metric for imbalanced datasets. Accuracy, meanwhile, offers a more general measure of the ratio of all correct predictions to the total number of instances [27]. Even in the case of imbalanced datasets, this accuracy may provide a less accurate representation of the model's performance. The related formula can be found in [27].

3. Result and Discussion

In this part of the paper, a thorough analysis of the data, including visualization, preprocessing, and splitting to prepare it for modeling. Class balancing techniques, such as SMOTE and RUS, were used alongside a non-sampling approach for comparison. Two modeling algorithms, KNN and Decision Tree, were applied to predict stunting in children, and their performance was compared to assess class imbalance handling and prediction accuracy.

3.1. Data Visualization

The data visualization reveals important insights into the distribution of children's height-for-age categories and nutritional status, highlighting a significant imbalance between stunted and non-stunted children. Shows varying nutritional conditions, such as undernutrition, risk of overnutrition, and obesity. The clear imbalance emphasizes the need for appropriate sampling techniques to address this issue. Additionally, the distribution of variables like weight, height, age, and the Z-score for height-for-age reveals outliers and variations, indicating areas requiring further attention in managing children's nutrition. These findings form a solid basis for analyzing factors influencing children's physical and nutritional conditions in the context of stunting.

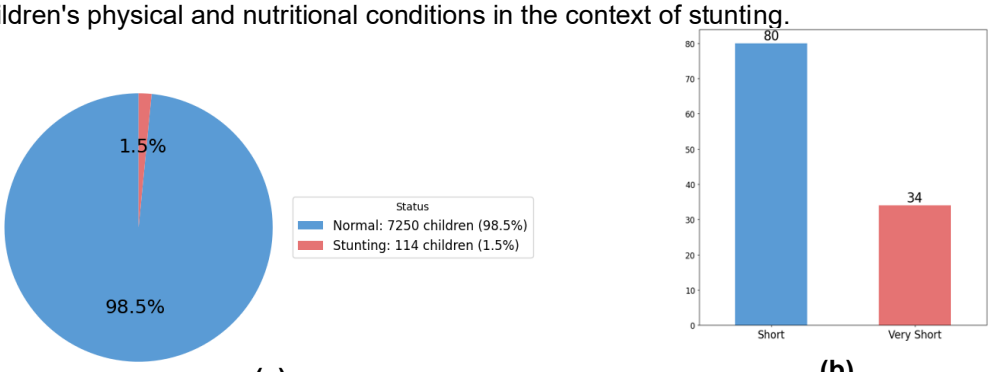
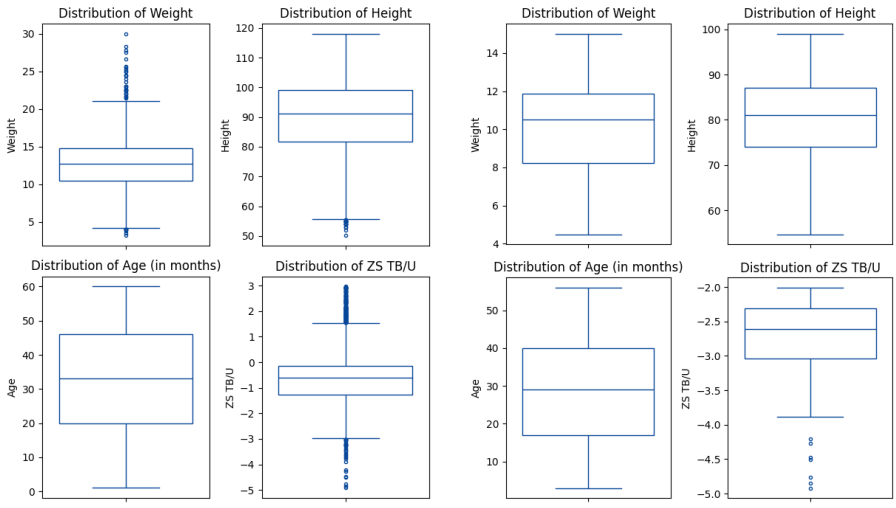


Figure 2. (a) Comparison of normal and stunted children. **(b)** Category Distribution of Stunting.

Figure 2(a) shows the distribution of stunting status in children within the height-for-age category. Most children (98.5%) fall into the normal category, with only 1.5% of children experiencing stunting. This highlights a significant imbalance between the number of children who are stunted and those who are normal. Meanwhile, Figure 2(b) illustrates the distribution of stunting based on the height-for-age category, where 80 children are classified as short and 34 children as very short. This indicates a significant difference in the physical growth conditions of children based on these categories.



(a) (b)

Figure 3. (a) Distribution of selected variables. **(b)** Distribution of variables for stunting class Computer

Figure 3 (a) shows the distribution of weight, height, age, and Z-Score for height-for-age (TB/U). The weight distribution has some high-value outliers, while the height distribution shows low-value outliers. Age data is consistent with no significant outliers. The Z-Score for height-for-age distribution has many extreme outliers, indicating values far from the median and quartiles. Figure 3 (b) presents the stunting-related variables (weight, height, age, and Z-Score). The weight and height distributions have a narrow range with few outliers, while the Z-Score shows a few very low outliers, indicating severe stunting. The age distribution shows no significant outliers, suggesting consistency in age data.

In this paper, most children with stunting were found to have a normal nutritional status (82.5%). However, a small proportion were categorized as underweight (12.3%) or at risk of overweight (3.5%). Additionally, a very small percentage of children were identified as obese (0.9%) or experiencing overnutrition (0.9%). These findings suggest that while most children with stunting appear to receive adequate nutrition, there remains a notable variation in nutritional status. This highlights the need for continued attention to children who are at risk of other nutritional imbalances.

3.2. Data Preprocessing

The data cleaning and transformation process involved several steps to ensure quality and consistency. First, irrelevant columns, such as personal and administrative information, were removed. Numerical data in columns like 'BB Lahir' (birth weight) and 'Berat' (weight) were converted to numeric types after removing commas. The 'Usia Saat Ukur' (age at measurement) column, initially text-based, was transformed into months for easier analysis. Rows with an age of 0 months were deleted, and missing values and irrelevant hyphens were removed. Duplicate rows were also eliminated to prevent bias, and important columns were selected, with categorical variables like 'JK' (gender) converted to numeric values. Finally, commas in the 'Tinggi' (height) and 'Berat' (weight) columns were replaced with periods for proper conversion. The data is now ready for further analysis.

3.3. Data Splitting

The data is split into 80% training data and 20% testing data. This division allows the model to learn patterns from the training data and be evaluated on its ability to make predictions on unseen testing data. Table 1 shows the distribution of sample counts for each class in both the training and testing datasets.

Table 1. Data Splitting Results without Sampling Technique

Result	Normal	Stunting	Total
Training Data	5804	87	5891
Testing Data	1446	27	1473

Table 2. Data Splitting Results with SMOTE

Result	Normal	Stunting	Total
Training Data	5826	5774	11600
Testing Data	1424	1476	2900

Table 3. Data Splitting Results with RUS

Result	Normal	Stunting	Total
Training Data	94	88	182
Testing Data	20	26	46

Table 1 shows the data split without sampling technique, highlighting the imbalance between the Normal and Stunting classes in both training and testing data. Table 2 displays the use of SMOTE, which balances the sample sizes between the classes. Table 3 presents the results of RUS, which reduces the majority class (Normal) to achieve balance, though it reduces the total data amount.

3.4. Classification Model Performance Analysis

In this section, the results of KNN and Decision Tree model implementation in three imbalanced data handling scenarios are presented. Before evaluating final performance, a tuning process using GridSearchCV was conducted to find the best hyperparameter combination for each scenario. The results of this tuning process, which served as the basis for model configuration in testing, are summarized in Table 4. recognizable.

Table 4. Best Hyperparameter Tuning Results with GridSearchCV

Technique Sampling	Model	Best Hyperparameter
Without Sampling Technique	KNN	metric: 'euclidean', n_neighbors: 3, weights: 'distance'
	Decision Tree	class_weight: None, criterion: 'entropy', max_depth: 9, min_samples_leaf: 1, min_samples_split: 6, splitter: 'best'
SMOTE	KNN	metric: 'euclidean', n_neighbors: 2, weights: 'uniform'
	Decision Tree	class_weight: None, criterion: 'gini', max_depth: 10, min_samples_leaf: 1, min_samples_split: 3, splitter: 'best'
RUS	KNN	metric: 'euclidean', n_neighbors: 5, weights: 'uniform'
	Decision Tree	class_weight: 'balanced', criterion: 'entropy', max_depth: 10, min_samples_leaf: 2, min_samples_split: 7, splitter: 'best'

Using the optimal hyperparameter configuration from Table 4, each model was then evaluated for its performance. A comprehensive performance comparison of all scenarios, including precision, recall, F1-Score, and accuracy, is presented in Table 5.

Table 5. Model Performance Classification Report

Technique Sampling	Model	Class	Precision	Recall	F1-Score	Accuracy (%)
Without Sampling Technique	KNN	1	0.81	0.33	0.47	98.64%
		0	0.98	0.99	0.99	
		Avg	0.90	0.66	0.73	
	Decision Tree	1	0.73	0.40	0.52	98.64%
		0	0.98	0.99	0.99	

		Avg	0.86	0.70	0.75	
SMOTE	KNN	1	0.98	0.99	0.99	99.17%
		0	0.99	0.98	0.99	
		Avg	0.99	0.99	0.99	
	Decision Tree	1	0.96	0.98	0.97	97.37%
		0	0.98	0.96	0.97	
		Avg	0.97	0.97	0.97	
RUS	KNN	1	0.91	0.80	0.85	84.78%
		0	0.78	0.90	0.83	
		Avg	0.84	0.85	0.84	
	Decision Tree	1	0.87	0.80	0.84	82.60%
		0	0.77	0.85	0.80	
		Avg	0.82	0.82	0.82	

In this paper, the performance of KNN and Decision Tree models was compared using three approaches to handle class imbalance, without sampling technique, with SMOTE (Synthetic Minority Over-sampling Technique), and with RUS. Model performance evaluation in this paper prioritized the F1-Score metric, particularly for the stunting class (class 1). Given the highly imbalanced data condition (1.5% stunting vs 98.5% normal), the accuracy metric can provide misleading results. F1-Score, which balances precision and recall, serves as a far more reliable benchmark for assessing the model's ability to identify stunting cases, which is the primary detection target.

As presented in Table 5, there are drastic performance differences across scenarios. In the No Sampling scenario, an anomaly occurred where despite both models achieving very high accuracy of 98.64%, the F1-Score for the stunting class was extremely low, at 0.47 for KNN and 0.52 for Decision Tree. Recall values of only 0.33 for KNN and 0.40 for Decision Tree provide clear evidence of model failure, where more than half of stunting cases were not successfully detected. This demonstrates that without special handling, even optimized models will be highly biased and tend to ignore minority classes.

The most significant and effective improvement was achieved through the SMOTE technique. A deeper analysis of the tuning results in Table 4 can explain why this approach succeeded. The KNN model with SMOTE achieved a nearly perfect F1-Score of 0.99 with hyperparameter configuration k=2. This very small k value indicates a strong synergy between SMOTE and the fundamental nature of KNN. SMOTE successfully created dense and informative stunting data clusters, allowing KNN to make highly accurate decisions by examining only its two nearest neighbors. The Decision Tree model with SMOTE also showed remarkable improvement with an F1-Score of 0.97. Its best hyperparameter, max_depth=10, indicates that this model has the capacity to learn complex patterns. The failure in the no-sampling scenario was not due to model incapability, but rather due to insufficient data to build representative rules. SMOTE provided the raw material needed for the decision tree to form effective and unbiased data partitions.

The RUS technique proved less optimal. Although F1-Score improved compared to no sampling, its performance was far below SMOTE. The sharp decrease in accuracy to 84.78% for KNN and 82.60% for Decision Tree indicates crucial information loss. This is reflected in the KNN tuning results, where the optimal k value jumped to 5. This indicates that after many majority data points were removed, the data space became sparser, forcing the model to search for neighbors in broader and less reliable areas. An interesting finding in the Decision Tree with RUS scenario was the selection of class_weight balanced as the best hyperparameter, despite the training data already being made nearly balanced. The explanation for this case lies in the GridSearchCV mechanism that uses cross-validation. During the tuning process, training data is split again into several folds, and this random division does not guarantee that each fold has perfect class balance. Therefore, class_weight 'balanced' was selected as the best strategy, this setting

functions as a safeguard that keeps the model alert and provides more attention to minority classes to address imbalances that may randomly emerge during the training process.

Overall, the analysis shows that model success depends not only on tuning, but also on the compatibility between the model's fundamental nature and the data structure modified by sampling techniques. SMOTE's success lies in its ability to build clearer data patterns for minority classes, while RUS's failure is caused by the risk of losing important information. This difference reinforces the conclusion that KNN with SMOTE is the most reliable solution in this paper.

4. Conclusion

This paper compares KNN and Decision Tree models for detecting stunting in toddlers using three class imbalance handling strategies: no sampling, SMOTE, and RUS. This paper concludes that the combination of KNN model with SMOTE oversampling technique represents the most superior and reliable solution, achieving a nearly perfect F1-Score of 0.99. This success is also attributed to the highly compatible combination between KNN's working mechanism and SMOTE's results. KNN heavily relies on nearest neighbor data and SMOTE excels at making minority data groups denser and more recognizable.

The success of the KNN and SMOTE combination demonstrates that a sampling technique's effectiveness is highly dependent on the model's working mechanism. In this case, successful stunting detection occurred because KNN's working method is highly compatible with how SMOTE creates synthetic data that makes minority class patterns denser and clearer. SMOTE makes KNN's task significantly easier, resulting in highly accurate outcomes. Furthermore, this paper confirms that F1-Score is a far superior and more reliable evaluation metric compared to accuracy for problems with critical class imbalance, where identifying every minority case is the primary priority.

Nevertheless, this research has limitations, including the use of data from one geographical region and testing limited to two algorithms. Future research can be expanded by testing this interaction on other more complex algorithms (such as XGBoost or deep learning) on more diverse datasets to ensure model generalization. Overall, this paper offers both a reliable, practical approach for detecting stunting early and significant methodological guidance for using machine learning to address pressing public health issues involving imbalanced data.

References

[1] S. Lonang and D. Normawati, “Jurnal Media Informatika Budidarma Klasifikasi Status Stunting Pada Balita Menggunakan K-Nearest Neighbor Dengan Feature Selection Backward Elimination,” *Jurnal Media Informatika Budidarma*, vol. 6, no. 1, pp. 49–56, 2021, doi: 10.30865/mib.v5i1.2293.

[2] J. Aurima, S. Susaldi, N. Agustina, A. Masturoh, R. Rahmawati, and M. Tresiana Monika Madhe, “Faktor-Faktor yang Berhubungan dengan Kejadian Stunting pada Balita di Indonesia,” *Open Access Jakarta Journal of Health Sciences*, vol. 1, no. 2, pp. 43–48, Nov. 2021, doi: 10.53801/oajjhs.v1i3.23.

[3] A. S. Vinci and A. Bachtiar, “Efektivitas Edukasi Mengenai Pencegahan Stunting Kepada Kader: Systematic Literature Review,” *Jurnal Endurance: Kajian Ilmiah Problema Kesehatan*, vol. 7, no. 1, pp. 66–73, 2022, doi: 10.22216/endurance.v7i1.822.

[4] Z. Wardani, D. Sukandar, Y. F. Baliwati, and H. Riyadi, “Sebuah Alternatif: Indeks Stunting Sebagai Evaluasi Kebijakan Intervensi Balita Stunting di Indonesia,” *Gizi Indonesia*, vol. 44, no. 1, pp. 21–30, Mar. 2021, doi: 10.36457/gizindo.v44i1.535.

[5] S. Aulia *et al.*, “Jurnal Rekayasa Teknologi Nusa Putra: Implementasi Metode Simple Additive Weighting untuk Menurunkan Jumlah Balita Stunting di Posyandu Dahlia 10,” *Jurnal Rekayasa Teknologi Nusa Putra*, vol. 10, no. 2, pp. 65–74, 2024, [Online]. Available: https://rekayasa.nusaputra.ac.id/index

[6] D. Febriali Sahl and A. Mauluddin, “Elemen-elemen Politik sebagai Strategi Mengkapitalisasi Perilaku Pemilih dalam Kontestasi Pemilu Presiden tahun 2024 di Indonesia,” *JCIC : Jurnal CIC Lembaga Riset dan Konsultan Sosial*, vol. 6, no. 1, pp. 13–28, Mar. 2024, doi: 10.51486/jbo.v6i1.116.

[7] M. Damanik, E. Sitorus, and I. M. Mertajaya, "Sosialisasi Pencegahan Stunting pada Anak Balita di Kelurahan Cawang Jakarta Timur," *Jurnal Comunita Servizio*, vol. 3, no. 1, pp. 552–560, 2021, doi: 10.33541/cs.v3i1.2909.

[8] L. Masan, "Penyuluhan Pencegahan Stunting Pada Balita," *Jurnal Altifani Penelitian dan Pengabdian kepada Masyarakat*, vol. 1, no. 1, pp. 58–62, Jan. 2021, doi: 10.25008/altifani.v1i1.121.

[9] O. N. Chilyabanyama *et al.*, "Performance of Machine Learning Classifiers in Classifying Stunting among Under-Five Children in Zambia," *Children*, vol. 9, no. 7, Jul. 2022, doi: 10.3390/children9071082.

[10] H. M. Fenta, T. Zewotir, and E. K. Muluneh, "A machine learning classifier approach for identifying the determinants of under-five child undernutrition in Ethiopian administrative zones," *BMC Med Inform Decis Mak*, vol. 21, no. 1, Dec. 2021, doi: 10.1186/s12911-021-01652-1.

[11] F. Putra, H. F. Tahiyat, R. M. Ihsan, R. Rahmaddeni, and L. Efrizoni, "Penerapan Algoritma K-Nearest Neighbor Menggunakan Wrapper Sebagai Preprocessing untuk Penentuan Keterangan Berat Badan Manusia," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 1, pp. 273–281, Jan. 2024, doi: 10.57152/malcom.v4i1.1085.

[12] L. Noorca Rachmadi, A. Prasetya Wibawa, U. Pujianto, and I. Artikel Abstrak, "belantika Pendidikan Rekomendasi Jurusan dengan Menggunakan Decision Tree pada Sistem Penerimaan Peserta Didik Baru SMK Widya Dharma Turen," *Belantika Pendidikan*, vol. 4, no. 1, pp. 29–36, 2021, doi: https://doi.org/10.47213/bp.v4i1.95.

[13] S. W. Pebrianti, R. Astuti, and F. M. Basysyar, "Penerapan Algoritma K-Nearest Neighbor dalam Klasifikasi Status Stunting Balita di Desa Bojongemas," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 2, pp. 2479–2488, 2024, doi: https://doi.org/10.36040/jati.v8i2.8448.

[14] A. I. Putri, Y. Syarif, P. Jayadi, F. Arrazak, and F. N. Salisah, "Implementasi Algoritma Decision Tree dan Support Vector Machine (SVM) untuk Prediksi Risiko Stunting pada Keluarga," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 2, pp. 349–357, Feb. 2024, doi: 10.57152/malcom.v3i2.1228.

[15] L. Madaerdo Sotarjua, D. Budhi Santoso, U. H. Singaperbangsa Karawang Jl Ronggo Waluyo, K. Telukjambe Timur, K. Karawang, and J. Barat, "Perbandingan Algoritma KNN, Decision Tree, dan Random Forest pada Data Imbalanced Class untuk Klasifikasi Promosi Karyawan," *INSTEK: Informatika Sains dan Teknologi*, vol. 7, no. 2, pp. 192–200, 2022, doi: https://doi.org/10.24252/instek.v7i2.31385.

[16] N. Dwi Primadya, A. Nugraha, S. Yudha Fahrezi, A. Luthfiarta, and U. Dian Nuswantoro, "Optimizing Imbalanced Data Classification: Under Sampling Algorithm Strategy with Classification Combination," *Techné: Jurnal Ilmiah Elektroteknika*, vol. 23, no. 2, pp. 277–288, 2024, doi: https://doi.org/10.31358/techne.v23i2.435.

[17] I. Kurniawan, D. Cahya Putri Buani, W. Apriliah, and E. Fitriani, "Penerapan Teknik Random Undersampling untuk Mengatasi Imbalance Class dalam Prediksi Kebakaran Hutan Menggunakan Algoritma Decision Tree," *AJCSR: Academic Journal of Computer Science Research*, vol. 5, no. 1, p. 1, 2023, doi: 10.38101/ajcsr.v5i1.617.

[18] Y. Xiong, X. Hu, J. Cao, L. Shang, and B. Niu, "A predictive model for stunting among children under the age of three," *Front Pediatr*, vol. 12, Sep. 2024, doi: 10.3389/fped.2024.1441714.

[19] G. D. Karisma, S. Fauziyah, and S. Herlina, "pengaruh antropometri bayi baru lahir dan prematuritas dengan kejadian stunting pada balita di desa baturetno," *Journal of Community Medicine*, vol. 10, no. 2, pp. 1–10, 2022, Accessed: Jun. 03, 2025. [Online]. Available: https://jim.unisma.ac.id/index.php/jkkfk/article/view/18051

[20] A. Dermawan, M. Mahanim, and N. Siregar, "Upaya Percepatan Penurunan Stunting Di Kabupaten Asahan," *Jurnal Bangun Abdimas*, vol. 1, no. 2, pp. 98–104, Nov. 2022, doi: 10.56854/ba.v1i2.124.

[21] A. Penelitian, K. Putri, and A. F. Lestari, "Attribution-ShareAlike 4.0 International Some rights reserved Transformasi Data Menjadi Tindakan : Menguak Stunting Di Desa Bayuning," *Jurnal Kesehatan dan Pelayanan Masyarakat*, vol. 1, no. 2, 2024, doi: 10.69688/jkpm.v1i2.168.

[22] J. Homepage, Q. A'yuniyah, and M. Reza, "Engineering Application Of The K-Nearest Neighbor Algorithm For Student Department Classification At 15 Pekanbaru State High School Penerapan Algoritma K-Nearest Neighbor Untuk Klasifikasi Jurusan Siswa Di Sma Negeri 15 Pekanbaru," *IJIRSE: Indonesian Journal of Informatic Research and Software*, vol. 3, no. 1, pp. 39–45, Aug. 2023, doi: https://doi.org/10.57152/ijirse.v3i1.484.

[23] W. Z. Aprilita, R. Akbar, R. Cahyadi Prayogi, T. Informatika, and S. Amik Riau, "Comparison of K-Nearest Neighbor (KNN) and Naive Bayes Algorithms in the Classification of Parkinson's Disease Komparasi Algoritma K-Nearest Neighbor (KNN) dan Naive Bayes dalam Klasifikasi Penyakit Parkinson," *SENTIMAS: Seminar Nasional Penelitian dan Pengabdian Masyarakat*, vol. 1, no. 1, pp. 188–193, Aug. 2023, Accessed: Jun. 03, 2025. [Online]. Available: https://journal.irpi.or.id/index.php/sentimas/article/view/612

[24] A. F. Najwa and P. H. Gunawan, "Enhancing Stunting Prediction for Indonesian Children Using Machine Learning with SMOTE Data Balancing," *Journal of Computing Science and Engineering*, vol. 18, no. 4, pp. 203–213, 2024, doi: 10.5626/JCSE.2024.18.4.203.

[25] D. A. Mukhsinin, M. Rafliansyah, S. A. Ibrahim, R. Rahmaddeni, and D. Wulandari, "Implementasi Algoritma Decision Tree untuk Rekomendasi Film dan Klasifikasi Rating pada Platform Netflix," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 2, pp. 570–579, Mar. 2024, doi: 10.57152/malcom.v4i2.1255.

[26] R. S. Abdulsadig and E. Rodriguez-Villegas, "A comparative study in class imbalance mitigation when working with physiological signals," *Front Digit Health*, vol. 6, 2024, doi: 10.3389/fdgth.2024.1377165.

[27] H. Adzkia Delfina, U. Novia Wisesty, and I. Kurniawan, "Jurnal Teknik Informatika (JUTIF) Gastroesophageal Reflux Disease Early Detection using XGBoost Method Classifier," vol. 6, no. 6, pp. 855–870, 2025, doi: 10.52436/1.jutif.6.6.3540.

[28] W. Nugraha and A. Sasongko, "Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search Hyperparameter Tuning on Classification Algorithm with Grid Search," *SISTEMASI: Jurnal Sistem Informasi*, vol. 11, no. 2, pp. 2540–9719, Accessed: Jun. 03, 2025. [Online]. Available: https://sistemasi.ftik.unisi.ac.id/index.php/stmsi/article/view/1750

[29] M. Adnan, A. A. S. Alarood, M. I. Uddin, and I. ur Rehman, "Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models," *PeerJ Comput Sci*, vol. 8, 2022, doi: 10.7717/PEERJ-CS.803.

[30] N. H. Dharma, D. I. Gede, and S. Astawa, "Hyperparameter Tuning Algoritma KNN Untuk Klasifikasi Kanker Payudara Dengan Grid Search CV," *Jurnal Nasional Teknologi Informasi dan Aplikasinya*, vol. 1, no. 1, pp. 397–402, 2022, Accessed: Jun. 03, 2025. [Online]. Available: https://ojs.unud.ac.id/index.php/jnatia/article/view/92647/47020

[31] J. Gou, L. Du, Y. Zhang, and T. Xiong, "A New Distance-weighted k-nearest Neighbor Classifier," *Journal of Information & Computational Science*, vol. 9, no. 6, pp. 1429–1436, 2012, [Online]. Available: http://www.joics.com

[32] A. K. Pathak, M. Chaubey, and M. Gupta, "Randomized-Grid Search for Hyperparameter Tuning in Decision Tree Model to Improve Performance of Cardiovascular Disease Classification," Nov. 2024, Accessed: Jun. 25, 2025. [Online]. Available: https://arxiv.org/abs/2411.18234