Performance of Random Forest and XGBoost in Stunting Classification for Toddlers in Talaud Island

1st Ninka Natalia Br Singarimbun
School of Computing
Telkom University
Bandung, Indonesia
ninkanatalia@student.telkomuniversity.ac.id

2nd Narita Aquarini
E´cole Doctorale Science Economics
Universit´e de Poitiers Intervenant
Finance La Rochelle, France
aquarinin@excelia-group.com

3nd Putu Harry Gunawan
School of Computing
Telkom University
Bandung, Indonesia
phgunawan@telkomuniversity.ac.id

***Abstract*—Stunting is a problem in toddlers when their height is too low compared to other toddlers their age. Prevention of stunting in toddlers is necessary to avoid long-term effects on toddlers. One way that can be done to reduce stunting rates in Indonesia is to create a system using machine learning that can predict stunting conditions in children to prevent and minimize the occurrence of stunting in children. This research analyzes two machine learning models that are potentially suitable for predicting stunting, namely Random Forest and XGBoost. In this research, the dataset has an imbalance issue. The stunting case is only 3.9% of the total dataset. To overcome this problem, we must perform an oversampling process in the minority class (Stunting). Oversampling is done by generating random data based on the distribution of data classified as minority. The minority class (stunting) data is generated using a quartile-based random sampling method. The research results show that Random Forest and XGBoost demonstrated good predictive capabilities. Random Forest achieved an accuracy of 98.32% and an F1-Score Average of 88.52%, while XGBoost slightly outperformed with an accuracy of 98.42% and an F1-Score Average of 89.06%. The results show that XGBoost has better performance due to its boosting mechanism, but Random Forest remains the right choice when interpretability and simplicity in implementation are prioritized.**

Keywords—Stunting, Random Forest, XGBoost, Performance

I. Introduction

Technological advances in the health sector today have increased the efficiency of health workers in serving the community, especially in diagnosing diseases. In this research, the disease to be diagnosed is stunting. Stunting is a problem in toddler growth characterized by a toddler's height being too short compared to toddlers of the same age [1]. A fetus's inadequate nutritional intake brings on this disorder during the first 1000 days of life or till the age of two. Based on data from the Coordinating Ministry for Human Development and Culture in 2022, Indonesia's stunting prevalence rate ranks twenty-seventh out of 154 countries with stunting data and ranks fifth among Asian countries. In Indonesia, around 6.3 million children under five are affected by this condition out of 149 million children in the world who are affected by stunting [2].

Stunting is a problem because it increases the risk of illness and mortality, delays motor skills, and impairs brain development [3]. The impact of stunting is not only about height but also affects learning abilities and increases vulnerability to chronic diseases. Quoted from the website stunting.go.id, the vice president announced that the Indonesian government has prioritized reducing stunting, achieving a decline from 30.8% in 2018 to 21.6% in 2022. However, further efforts are required to reach the target of 14% by 2024, requiring a reduction of 7.6% within two years. One proposed solution is the utilization of machine learning systems to predict and prevent stunting in children, as machine learning has been successfully applied in the medical field [4] for the early detection of diseases like diabetes [5], heart failure [6], and COVID-19 [7].

In 2022, the stunting problem was discussed in [8]. This research predicts stunting in Zambia using Logistic Regression, Random Forest, Naïve Bayes, Support Vector Machine (SVM), and XGBoost methods. According to the results, the Random Forest model achieves maximum training and testing data accuracy, with corresponding scores of 79.2% and 61.6%. Another study in 2023 was written in [9], which analyzed the application of machine learning methods to classify stunting in Rwanda using Gradient Boosting, Random Forests, SVM, XGBoost, and Logistic Regression. The study finds that XGBoost has the best performance, with an accuracy of 79.13%.

Based on the results of these two studies, where Random Forest (bagging model) and XGBoost (boosting model) showed the best performance in stunting classification in Zambia and Rwanda, this research aims to propose a approach by comprehensively comparing the performance of two algorithms, which are well known as the best models in each approach to the toddler data of Talaud Islands which has different demographic and socioeconomic characteristics compared to other regions. In addition, data imbalance is still a problem when applying classification models [10]. Therefore, this research also utilizes a quartile range-based balancing method (Q1-Q3) to create medically valid synthetic data, ensuring local relevance and effectiveness of the model in detecting stunting. This research is to help related parties find appropriate machine learning for the early prevention and treatment of stunting.

This paper is organized as follows: Details of the dataset, the Random Forest and XGBoost algorithm principles, and the performance analysis technique are all included in the second section, which also examines the study's theoretical underpinnings and research framework. The data preprocess methods, data exploration procedures, classification model construction, and performance assessment are all covered in the third section. The research is concluded with a summary of the main conclusions in the final section.

II. Methodology

A. Research Framework

These research works are based on stunting classification, and the steps below are applied to perform as shown in Fig. 1.

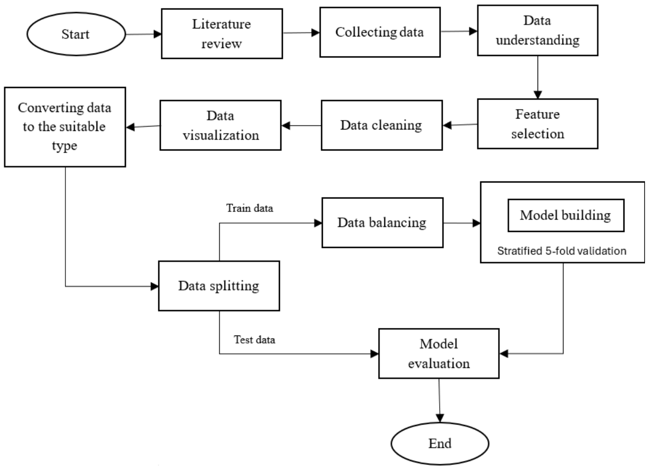

Fig. 1. Research Flowchart

This research starts by reviewing the literature studies related to previous research to understand the topic of stunting, classification methods, and the relevant parameters. Prior research is used as a reference when creating the background and selecting algorithm methods. After reviewing literature studies, data collection was carried out. Data was obtained from the Talaud Islands District Health Service, North Sulawesi. Research continues by understanding the structure, patterns, and distribution of data. At this stage, exploratory data analysis is carried out to understand patterns and relationships between features. After understanding the data, the relevant features for stunting classification are selected. This is done to avoid overfitting and improve model efficiency. Then, data will be cleaned of missing values, outliers, and irrelevant data. After that, the data type is checked. If the data type is not numeric, the data is converted to numeric using encoding techniques and converted to the age in the number of months. The ready data is visualized to understand the pattern and distribution of the data more deeply before being divided into training data (80%) and testing data (20%) using the data splitting method. The research continued by checking whether the data was balanced or not. Imbalanced data means that stunting cases are much less common than typical cases. Therefore, it is important to use balancing techniques, such as oversampling and undersampling, or algorithms such as SMOTE. In this research, data is balanced using the quartile-based random sampling method to oversample the minority class. Quartile-based sampling ensures that the resulting synthetic samples remain within a realistic range of values (interquartile or Q1-Q3). This relates to stunting classification, where features such as weight, height, and age have realistic value boundaries that are important to maintain. The machine learning models, namely Random Forest and XGBoost, are then built using balanced training data, with the application of Stratified 5-Fold Cross Validation to ensure the performance of the models is thoroughly and fairly tested on various subsets of data. The last step is model evaluation, in which the model's performance will be evaluated using accuracy, precision, recall, and F1-score metrics.

B. Dataset

This research obtained data from the Talaud Islands District Health Service, North Sulawesi, based on measurements in June 2024. This dataset consists of 5,050 records covering information about the child, with 197 categorized as stunted and 4,853 as normal. Through a feature selection process, the features selected from a total of 25 features include age (A), gender (G), weight (W), height (H), z-score (ZS), nutritional status (W/H), height/age (H/A), and weight/age (W/A). Table I shows an overview of the dataset.

TABLE I. DATASET STUNTING

A	W	H	ZS	W/H	W/A	H/A
3 years	13.5	99	-0.74	Normal	Normal	Normal
4 years	23.3	104.2	-0.18	Obese	Risk of over-weight	Normal
4 years	15.4	97	-2.59	Normal	Normal	Stunted
4 years	12.3	97	-1.99	Seve-rely wasted	Under-weight	Normal
4 years	12.6	96.6	-2.37	Normal	Under-weight	Stunted

[a] A: Age, W: Weight, H: Height, ZS: Z-Score, W/H: Weight/Height, W/A: Weight/Age, H/A: Height/Age

The target class in this research is Height/Age because it directly describes whether children are stunted based on their height and age ratio. So, the selected features for this research are 8 features.

C. Random Forest

Random Forest is a machine learning algorithm commonly used to classify data that works by combining the result of multiple decision trees to get one more accurate result [11], [12]. This algorithm is a group of untrained classification or regression trees, which are trained using bootstrap of training data and use random feature selection during the tree formation. After creating many trees, each tree votes to determine the most common class. The collective voting process is the concept of random forest. In this classification, two important parameters are required: the number of trees and features used to multiply the number of trees [13]. How random forest works can be seen in Fig. 2.

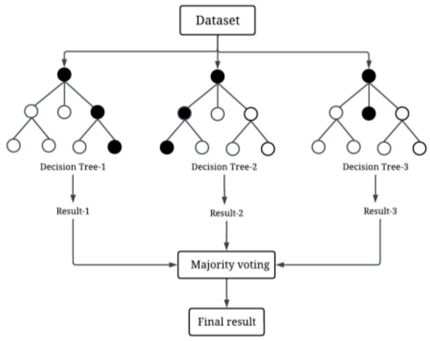

Fig. 2. Random Forest Flow

D. XGBoost

XGBoost (Extreme Gradient Boosting) is a development of gradient boosting that can boost decision trees in a parallel and distributed way. With the specified probability initialization, the first tree in XGBoost will be weak in classification. Then, each tree built will receive weight updates, resulting in a collection of strong classification trees [9]. XGBoost uses a gradient boosting framework, which optimizes a predefined objective function by minimizing the residual error at each iteration. Equipped with regularization techniques, the algorithm can effectively overcome overfitting. XGBoost is known for its speed, scalability, and outstanding performance in various data science applications [14]. How XGBoost works can be seen in Fig. 3.

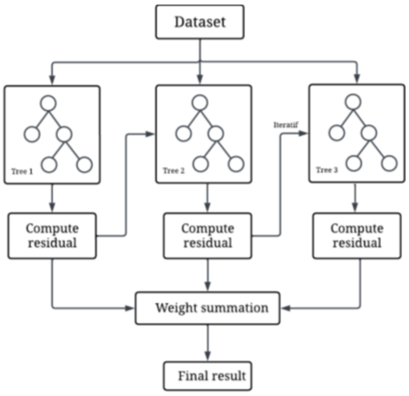

Fig. 3. XGBoost Flow

E. *Stratified K-Fold Cross-Validation*

Stratified K-Fold cross-validation is a model evaluation technique used in data analysis to improve prediction accuracy. This approach helps provide a more reliable estimate of a model's generalization ability, particularly in imbalanced datasets [15]. It works by splitting the dataset into k-equal parts and ensuring each part has the same distribution as the original dataset. Thus, Stratified K-Fold Cross Validation can help improve classification accuracy and ensure the model can be used effectively on various dataset sizes. In this research, stratified K-Fold cross-validation is only used to train data. It is used to estimate the performance of the model more accurately [16]

F. *Evaluation Metrics*

This research uses the confusion matrix method for performance analysis. In machine learning, a confusion matrix can summarize the models, allowing for a more in-depth analysis of the algorithm’s performance [17]. The confusion matrix consists of accuracy, precision, recall, and F1-score. The confusion matrix table can be seen on Fig. 4.

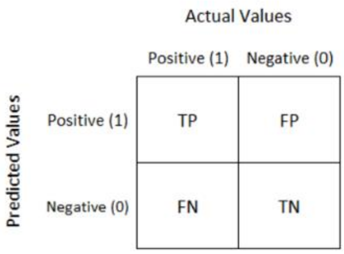

Fig. 4. Confusion Matrix

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\% \quad (4)$$

Based on Fig. 5, “TP’ or True Positive describes home many positive cases the model has classified correctly. “TN” or True Negatives means how many negative cases were successfully classified by the model. “FP” of False Positive indicates some negative cases incorrectly classified as positive. “FN” of False Negative describes misclassified positive cases as negative[18]. Accuracy represents the percentage of all accurate predictions. Precision (positive prediction accuracy) measures the extent to which the positive values have been identified correctly compared to the total number of positive values. Recall (TP rate) reflects the extent to which the model identified positive cases. F1-Score is the harmonic means of sensitivity and precision, where both factors are considered significant [19].

III. Result and Discussion

A. *Exploratory Data Analysis*

Exploratory data analysis is a step in research that aims to understand the dataset, see errors, find outliers, and understand relations and patterns between data.

The dataset used in this research is a collection of toddler data with nutritional status obtained from the Health Office of the Talaud Island Regency, North Sulawesi, based on measurements in June 2024. The selected features for this research are age, gender, weight, height, z-score, nutritional status, and weight/age. And for the class target, this research uses Height/Age (H/A). The statistical distribution of each column can be seen in Fig. 5.

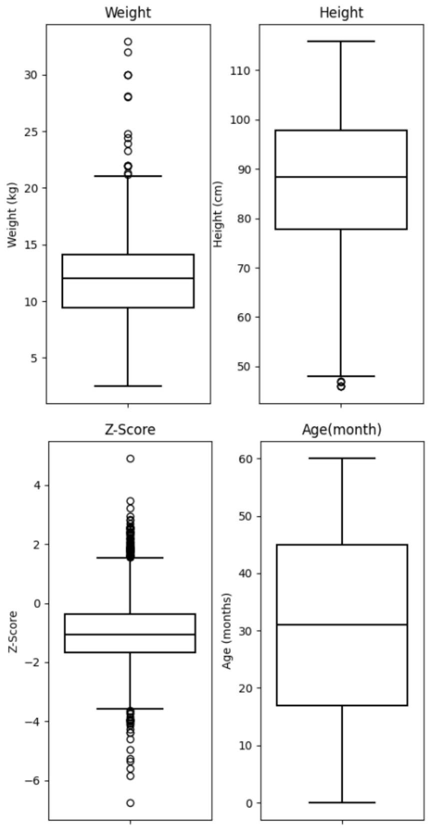

Fig. 5. Distributions of Data

Fig. 5 shows an overview of the characteristics of the features that will be used for classification. In the weight feature, the median is around 12 to 15kg, which indicates that most individuals weigh in that range. Outliers above 20kg indicate individuals with a weight much higher than the average population. The median is around 90cm for height, and outliers below 50cm indicate individuals with much lower height. In the Z-Score, the median is close to 0. It means most individuals are growing according to usual standards, but significant outliers above and below indicate stunting or overweight issues. Meanwhile, the age distribution has a median of 30 to 35 months with no outliers, meaning this feature has no extreme values.

B. Data Preprocessing

TABLE II. DATA PREPROCESSING OVERVIEW

Step	Description	Output Example
Feature Selection	Select the relevant features	Final Features: 8
Data Cleaning	Handle missing values by dropping incomplete rows. No duplicated data was identified	Total Data: 5,048
Feature Transformation	Age in string format 'X Years–Y Months–Z Days' converted to number of months	Age: 3 Years-11 Months-1 Day → 47 months
Feature Encoding	Features encoded into numerical categories	Nutritional status: 0: Normal, 1: Overnutrition, 2: Undernutrition
Outlier Handling	Remove outliers based on interquartile range (IQR)	Values outside Q1-Q3 replaced with median
Data Splitting	Split data into 80% training and 20% testing sets	Training: 4,038, Testing: 1,010
Oversampling	Quartile ranges for the oversampling method	Quartile: Q1 (low) to Q3 (high)

The dataset contains 25 features, but for analysis, it uses 8 features. Data cleaning is done in this step, including missing values and duplicating data. The results show no duplicate data; the missing values are handled by dropping the data. In addition, encoding techniques have processes for converting data into compatible types. The data types of the features were converted from string to numeric. The age data type, a string, is converted to the number of months to be numeric. The nutritional status features five values, namely Normal, At Risk of Overnutrition, Undernutrition, Overnutrition, Obesity, and Malnutrition, which will be narrowed down to three types. At risk of overnutrition and obesity will be combined into overnutrition. Malnutrition will be combined with undernutrition. So, nutritional status has three types: normal, overnutrition, and undernutrition. This feature is then set to an integer from 0 to 2. The target class also changes the data type from string to numeric. The normal value is set to 0, and the stunted value is set to 1. The Interquartile Range (IQR) method replaces outliers with median values, and Winsorization ensures that the data is cleaner and can be processed properly.

The data preprocessing process is continued by analyzing the amount of stunting and normal data shown in Table III.

TABLE III. DATA BEFORE BALANCING

	Normal (0)	Stunting (1)
Amount	4,853	195

Table III shows that the data from the 5,048 data is imbalanced. So, it can be indicated that the data needs to be balanced and has an issue with overfitting. This becomes a problem because, with less data in positive classes, the model will take longer to learn negative data than positive data. It causes the model to be biased towards new data. The model tends to overfit, and the performance in performing classification on positive data is significantly reduced [10].

To overcome this problem, we must perform an oversampling process in the minority class (stunting). Before balancing the data, the data is first split into 80% training data and 20% testing data. The training data amounted to 4,038, and the testing data amounted to 1,010. Data balancing only works on training data to avoid data leakage; when testing data, information leaks into training data. In this research, oversampling is done by generating random data based on the distribution of data classified as minority. The minority class (stunting) data is generated using a quartile-based random sampling method, where new values for features such as weight, height, age at measurement, and nutritional status are generated in the interquartile range (Q1 to Q3) to keep the oversampled data relevant without any outliers. The random sample generated in this study is around 4,000. The random sample data will be combined with the initial data so that the stunting data becomes as many as 4,156. The amount of stunting and normal data after oversampling can be seen in Table IV.

TABLE IV. DATA AFTER BALANCING

	Normal (0)	Stunting (1)
Amount	3,882	4,156

C. Model Building

After the data was balanced, the Stratified K-Fold Cross Validation method built the model. This method balances the proportion of Normal and Stunted classes in each fold during the training. This approach reduces the risk of overfitting, ensures stable evaluation, and maintains a consistent class representation. The final evaluation was performed on pre-separated testing data (20% of the dataset) without modifications, such as data balancing, to ensure the model's performance on the original data distribution as done in [16]

1) Random Forest

The first model uses the Random Forest algorithm, implemented with the Scikit-learn library. In this research, to generate 100 random decision trees, the model in this study was trained using training data for 100 iterations. By assigning the minority class more weight, the class weight='balanced' parameter was applied to handle data imbalance automatically. A preliminary evaluation uses the training data after training to ensure the model can detect patterns without overfitting. Hyperparameter optimization using Grid Search was performed during model building to optimize accuracy. Nevertheless, the outcomes demonstrated that the model's performance reached ideal accuracy without optimization and did not significantly improve model performance.

A preliminary evaluation uses the training data after training to ensure the model can detect patterns without overfitting. The result of the Random Forest model when using Stratified 5-fold Cross-Validation for data training is represented using a confusion matrix, as shown in Table V.

TABLE V. RANDOM FOREST PERFORMANCE FOR DATA TRAINING

Random Forest							
Fold	Precision		Recall		F1-Score		Accuracy
	0	1	0	1	0	1	
Fold 1	97.65%	98.19%	98.46%	97.26%	98.05%	97.72%	97.90%
Fold 2	97.13%	97.13%	97.53%	96.66%	97.33%	96.89%	97.13%
Fold 3	96.95%	97.71%	98.04%	96.42%	97.49%	97.06%	97.29%
Fold 4	97.20%	95.98%	96.49%	96.78%	96.84%	96.38%	96.63%
Fold 5	97.33%	97.48%	97.84%	96.90%	97.58%	97.19%	97.40%
Average	97.27%		97.24%		97.25%		97.26%

Based on Table V, the performance is consistent between folds with an average accuracy, precision, recall, and F1-score of 97.26%. No fold shows a significant decrease in performance, which indicates that the Random Forest model has excellent generalization. The model that has been trained using training data is then tested on testing data to evaluate its performance on data that has never been seen before. The confusion matrix for the testing data is shown in Fig. 7.

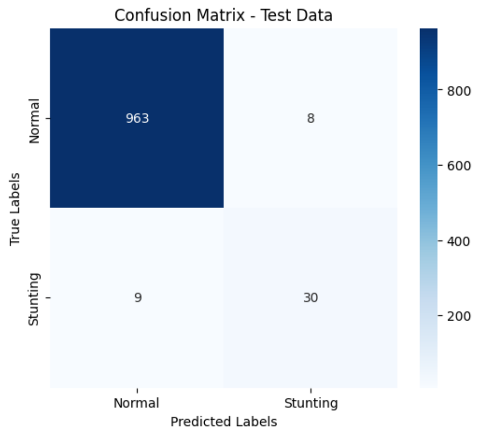

Fig. 7. Confusion Matrix Random Forest for Data Testing

2) *XGBoost*

The second model uses an XGBoost algorithm, which is implemented using the Python xgboost module. The model was trained using default parameters provided by the library. To create a model, 100 boosting rounds, or iterations, are built during the training phase. A preliminary assessment uses training data to ensure the model can recognize patterns without overfitting. Furthermore, Grid Search is used for hyperparameter optimization to determine the optimal combination of parameters. The outcomes demonstrated that following hyperparameter modification, the model's performance improved. The best combination obtained is 'colsample_bytree': 0.8, 'learning_rate': 0.2, 'max_depth': 10, 'n_estimators': 300, 'subsample': 1.0. The use of these parameters resulted in higher accuracy compared to the default model. This appears that hyperparameter optimization improves the model's ability to perform the stunting classification task.

Then, the evaluation is repeated to provide the model's performance using the testing data objectively. The result of the XGBoost model when using Stratified 5-fold Cross-Validation for data training is represented using a confusion matrix, as shown in Table VI.

TABLE VI. XGBOOST PERFORMANCE FOR DATA TRAINING

XGBoost							
Fold	Precision		Recall		F1-Score		Accuracy
	0	1	0	1	0	1	
Fold 1	98.07%	99.39%	99.49%	97.74%	98.77%	98.56%	98.67%
Fold 2	97.64%	97.84%	98.15%	97.26%	97.89%	97.55%	97.73%
Fold 3	96.75%	97.94%	98.25%	96.19%	97.50%	97.05%	97.29%
Fold 4	96.65%	97.70%	98.04%	96.07%	97.34%	96.88%	97.13%
Fold 5	97.24%	97.95%	98.25%	96.78%	97.74%	97.36%	97.57%
Average	97.71%		97.61%		97.66%		97.67%

Based on Table VI, the average accuracy of the model consistently reached 97.67%. This demonstrates the model's ability to detect stunting cases with high accuracy without many false positives or negatives. The consistent performance across folds reflects the strong generalization ability of the XGBoost model. The model trained on the training data was then tested on the testing data to evaluate its performance on data that had never been seen before. The confusion matrix for the testing data is shown in Fig. 8.

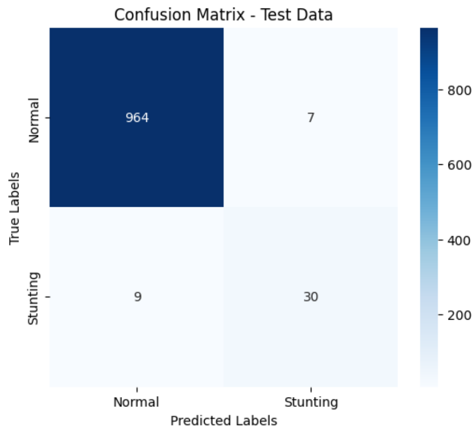

Fig. 8. Confusion Matrix XGBoost for Data Testing

D. *Model Evaluation*

TABLE VII. MODELS PERFORMANCE FOR DATA TESTING

Model	Precision	Recall	F1-Score	Accuracy
Random Forest	89.01%	88.05%	88.52%	98.32%
XGBoost	90.08%	88.10%	89.06%	98.42%

Based on Table VII, the XGBoost model shows better performance than Random Forest in classifying stunting. XGBoost achieved an accuracy of 98.42%, higher than Random Forest, which reached 98.32%. In addition, XGBoost's F1-Score Macro Average reached 89.06%, reflecting a better balance between precision (90.08%) and recall (88.10%) for both classes. XGBoost model training results in Table VI are also better than Random Forest. Although XGboost performance on testing is lower than the average training value, this decrease is reasonable considering the difference in testing data. But testing performance is still better than Random Forest. This superiority is due to the boosting mechanism employed by XGBoost, where the model learns iteratively by correcting the prediction error at each step. This approach allows XGBoost to handle minority classes, such as “Stunting,” supported by built-in regularization (L1 and L2) to prevent overfitting and weight adjustment capabilities to handle class imbalance. This observation aligns with the findings in [14]. On the other hand, Random Forest uses a bagging approach that is less responsive to complex patterns than boosting because it does not learn from previous prediction errors. However, XGBoost needs to set parameters while Random Forest does not. As a result, this research analyzes that XGBoost is superior in terms of stunting classification compared to Random Forest, especially in handling imbalanced dataset patterns.

IV. Conclusion

According to the research results for classifying stunting among toddlers, Random Forest and XGBoost show significant predictive ability for classifying toddler stunting. Random Forest with an accuracy of 98.32% and an F1-Score Average of 88.52%, whereas XGBoost obtained an accuracy of 98.42% and an F1-Score Average of 89.06%. The results show that XGBoost performs better due to its boosting mechanism, which allows the model to handle more significant and complex datasets. XGBoost is better at generalization because it has a more minor difference between training and testing performance. This confirms that XGBoost is more effective for handling unseen data with higher accuracy and better evaluation metrics. Its high performance across all metrics suggests that the XGBoost model is suitable for implementation in real-world applications in child health or nutrition. However, Random Forest remains the right option if interpretability and ease of use in implementation are prioritized. In addition, a quartile range-based balancing method is proven to work on training data based on the performance of both models. This research highlights the potential of both models as reliable tools to assist in the classification of stunting and supporting early detection and intervention strategies in public health initiatives.

Reference

[1] A. Fadilah, M. N. Pangestu, S. Lumbanbatu, and S. Defiyanti, “Pengelompokan Kabupaten/Kota Di Indonesia Berdasarkan Faktor Penyebab Stunting Pada Balita Menggunakan Algoritma K-Means,” *JIKO (Jurnal Informatika dan Komputer)*, vol. 6, no. 2, p. 223, 2022, doi: 10.26798/jiko.v6i2.581.

[2] A. Bilqis, I. Putri, and S. A. Okdiyanti, “Analisis Kebijakan Pemerintah Kabupaten Jember Dalam Menangani Stunting Dengan Perspektif Problem Tree Analysis,” *JMA)*, vol. 2, no. 5, pp. 3031–5220, 2024, doi: 10.62281.

[3] D. Agung Susanti, S. Patonah, F. Azizah, and S. Tinggi Ilmu Kesehatan Rajekwesi Bojonegoro, “Edukasi Orang Tua Tentang Pencegahan Stunting Dengan Perbaikan Pola Makan, Pola Asuh & Sanitasi Di Ra Nafisa Kendal Sidodadi Sukosewu,” 2024.

[4] M. A. Moreno-Ibarra, Y. Villuendas-Rey, M. D. Lytras, C. Yáñez-Márquez, and J. C. Salgado-Ramírez, “Classification of diseases using machine learning algorithms: A comparative study,” *Mathematics*, vol. 9, no. 15, Aug. 2021, doi: 10.3390/math9151817.

[5] S. Usabili and U. Indahyanti, “Pemodelan Deteksi Dini Diabetes Mellitus menggunakan Pendekatan Ensemble Learning,” *Manajemen Pelayanan Kesehatan*, vol. 1, no. 1, p. 12, Jan. 2024, doi: 10.47134/mpk.v1i1.2924.

[6] Y. Pratama, A. Prayitno, D. Azrian, N. Aini, Y. Rizki, and E. Rasywir, “Klasifikasi Penyakit Gagal Jantung Menggunakan Algoritma K-Nearest Neighbor,” *Bulletin of Computer Science Research*, vol. 3, no. 1, pp. 52–56, Dec. 2022, doi: 10.47065/bulletincsr.v3i1.203.

[7] P. Podder, S. Bharati, M. R. H. Mondal, and U. Kose, “Application of machine learning for the diagnosis of COVID-19,” in *Data Science for COVID-19 Volume 1: Computational Perspectives*, Elsevier, 2021, pp. 175–194. doi: 10.1016/B978-0-12-824536-1.00008-3.

[8] 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.

[9] S. Ndagijimana, I. H. Kabano, E. Masabo, and J. M. Ntaganda, “Prediction of Stunting among Under-5 Children in Rwanda Using Machine Learning Techniques,” *Journal of Preventive Medicine and Public Health*, vol. 56, no. 1, pp. 41–49, Jan. 2023, doi: 10.3961/jpmph.22.388.

[10] M. I. Anugrah, J. Zeniarja, and D. S. Setiawan, “Peningkatan Performa Model Hard Voting Classifier dengan Teknik Oversampling ADASYN pada Penyakit Diabetes,” *Edumatic: Jurnal Pendidikan Informatika*, vol. 8, no. 1, pp. 290–299, Jun. 2024, doi: 10.29408/edumatic.v8i1.25838.

[11] Z. Jin, J. Shang, Q. Zhu, C. Ling, W. Xie, and B. Qiang, “RFRSF: Employee Turnover Prediction Based on Random Forests and Survival Analysis,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2020, pp. 503–515. doi: 10.1007/978-3-030-62008-0_35.

[12] G. Daffa and P. Gunawan, *Stunting Classification Analysis for Toddlers in Bojongsoang: A Data-Driven Approach*. IEEE, 2024.

[13] R. Annisa, “Analisis Komparasi Algoritma Klasifikasi Data Mining Untuk Prediksi Penderita Penyakit Jantung,” *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 3, no. 1, 2019.

[14] N. T. Hastuti and F. Budiman, “Edumatic: Jurnal Pendidikan Informatika Optimasi Klasifikasi Stunting Balita dengan Teknik Boosting pada Decision Tree,” vol. 8, no. 2, pp. 655–664, 2024, doi: 10.29408/edumatic.v8i2.27913.

[15] M. T, K. Vinoth, K. Dhilip, O. Geman, M. Margala, and M. Guduri, “The stratified K-folds cross-validation and class-balancing methods with high-performance ensemble classifiers for breast cancer classification,” *Healthcare Analytics*, vol. 4, Dec. 2023, doi: 10.1016/j.health.2023.100247.

[16] R. Rabiei, S. M. Ayyoubzadeh, S. Sohrabei, M. Esmaeili, and A. Atashi, “Prediction of Breast Cancer using Machine Learning Approaches,” *J Biomed Phys Eng*, vol. 12, no. 3, pp. 297–308, Jun. 2022, doi: 10.31661/jbpe.v0i0.2109-1403.

[17] B. Sahu, S. N. Mohanty, and S. K. Rout, “A Hybrid Approach for Breast Cancer Classification and Diagnosis,” *EAI Endorsed Transactions on Scalable Information Systems*, vol. 6, no. 20, 2019, doi: 10.4108/eai.19-12-2018.156086.

[18] A. Vanacore, M. S. Pellegrino, and A. Ciardiello, “Fair evaluation of classifier predictive performance based on binary confusion matrix,” *Comput Stat*, vol. 39, no. 1, pp. 363–383, Feb. 2024, doi: 10.1007/s00180-022-01301-9.

[19] S. Lonang, A. Yudhana, and M. K. Biddinika, “Analisis Komparatif Kinerja Algoritma Machine Learning untuk Deteksi Stunting,” *Jurnal Media Informatika Budidarma*, vol. 7, no. 4, p. 2109, Oct. 2023, doi: 10.30865/mib.v7i4.6553.