MENINGKATKAN DETEKSI STUNTING PADA BALITA DENGAN TEKNIK BOOSTED KNN DAN BOOSTED NAIVE BAYES

Tugas Akhir

diajukan untuk memenuhi salah satu syarat

memperoleh gelar sarjana

pada Program Studi Informatika

Fakultas Informatika Universitas Telkom

1301213398

Gibran Shevaldo

UNIVERSITAS
Telkom

Program Studi Sarjana Informatika

Fakultas Informatika

Universitas Telkom

Bandung

2024

LEMBAR PENGESAHAN

MENINGKATKAN DETEKSI STUNTING PADA BALITA DENGAN TEKNIK BOOSTED KNN DAN BOOSTED NAIVE BAYES

IMPROVING STUNTING DETECTION IN TODDLERS WITH BOOSTED KNN AND BOOSTED NAIVE BAYES TECHNIQUES

NIM :1301213398

Gibran Shevaldo

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagian syarat memperoleh gelar pada Program Studi Sarjana Informatika
Fakultas Informatika
Universitas Telkom

Bandung, 9 Desember 2024
Menyetujui,

Pembimbing I,

Dr. Putu Harry Gunawan, S.Si., M.Si., M.Sc.
NIP: 16860043

Ketua Program Studi

Sarjana Informatika

Dr. Erwin Budi Setiawan, S.Si., M.T.
NIP: 00760045

LEMBAR PERNYATAAN

Dengan ini saya, Gibran Shevaldo, menyatakan sesungguhnya bahwa Tugas Akhir saya dengan judul MENINGKATKAN DETEKSI STUNTING PADA BALITA DENGAN TEKNIK BOOSTED KNN DAN BOOSTED NAIVE BAYES beserta dengan seluruh isinya adalah merupakan hasilkarya sendiri, dan saya tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang belaku dalam masyarakat keilmuan, serta produk dari tugas akhir bukan merupakan produk dari *Generative AI*. Saya siap menanggung resiko/sanksi yang diberikan jika di kemudian hari ditemukan pelanggaran terhadap etika keilmuan dalamLaporan TA atau jika ada klaim dari pihak lain terhadap keaslian karya,

Bandung, 9 Desember 2024

Yang Menyatakan

Gibran Shevaldo

1301213398

Improving Stunting Detection in Toddlers with Boosted KNN and Boosted Naïve Bayes Techniques

1st Gibran Shevaldo
School of Computing
Telkom University
Bandung, Indonesia
arzgbshv@student.telkomuniversity.ac.id

2nd Putu Harry Gunawan
School of Computing
Telkom University
Bandung, Indonesia
phgunawan@telkomuniversity.ac.id

***Abstract*—Stunting is one of the primary health concerns for children in Indonesia. Preventing stunting in toddlers is essential to mitigate long-term effects on both their health and society as a whole. Preventing stunting involves monitoring the growth of toddlers. Therefore, a predictive system for identifying stunting in toddlers is crucial. Machine learning offers many methods that can be used to build a system to predict stunting conditions in toddlers. This research analyzes some potentially suitable machine learning models for predicting stunting classes using Ensemble Learning, which are Boosted K-Nearest Neighbor (BK) and Boosted Naïve Bayes (BN). The boosting is done by assigning an initial weight to each sample and increasing each failed classified sample's weight. This approach enhances the learning done by the machine learning model by focusing on learning more about the failed classified samples. The dataset has an imbalance issue in this research, with the data categorized as short and very short at less than 2% of the total dataset. Therefore, oversampling of the dataset is done by generating a random dataset based on the distribution of the imbalanced dataset. After that, the normal category dataset is reduced to ensure the data is evenly distributed. The result of elaborating on this oversampling has been unsatisfactory, as the data distribution remains imbalanced despite efforts to stabilize the quantity between classes. Therefore, additional boosting is necessary to ensure proper classification. After the data is balanced by oversampling and boosting, the F-1 score macro average reached 97.44% for the BK method and 57.91% for the BN method. Additionally, the accuracy achieved was 98.62% for BK and 80.62% for BN. These results indicate that the BK method outperforms the BN method, despite the BN method achieving better outcomes than the other previous research.**

***Index Terms*—Stunting, Machine Learning, Ensemble Learning, Boosted K-Nearest Neighbor, Boosted Naïve Bayes**

I. INTRODUCTION

Stunting is a chronic nutritional problem caused by prolonged inadequate nutrient intake, leading to growth disorders where a child's height is significantly below the age standard[12]. Although the stunting rate in Indonesia has decreased significantly to 21.6% based on www.kemkes.go.id, it remains a major health concern for children, as the World Health Organization (WHO) recommends a prevalence below 20%[4].

One of the best solutions to stunting is prevention to avoid the problem itself. To reduce stunting rates in Indonesia, preventive measures and improved nutrition must be taken before stunting affects a child. One way to avoid it is to monitor the children's growth regularly[13]. Thus, a system to predict the potential for stunting in children is needed[9]. Machine learning offers methods to predict stunting in toddlers, enabling health workers to provide early nutritional guidance. Based on the research in[2], the Ensemble Learning Boosted K-Nearest Neighbor (BK) performs well in predicting stunting conditions. In 2023, the stunting problem was discussed in[2], which used BK to predict the same stunting dataset. The BK method got 98% accuracy as the highest result after balancing the dataset and several iterations of learning. On the other hand, the research in[7] used the Naïve Bayes method to predict the same stunting dataset and got 64.36% accuracy as the highest result.

In order to cope with imbalanced data classification, research in[6] proposes an ensemble algorithm named BPSO-Adaboost-KNN. The main idea of this algorithm is to integrate feature selection and boosting into an ensemble. On the other hand, research in[10] proposes Naïve Bayes classifier as the solution. But Naïve Bayes' effectiveness still needs to be upgraded, so they presented solutions on using Boosting to improve the Gaussian Naïve Bayes algorithm by combining the Naïve Bayes classifier and Adaboost methods[11].

This study seeks to determine the best machine-learning model dedicated to detecting stunting conditions in toddlers between Ensemble Learning Algorithm. Leveraging a dataset sourced from the Bojongsoang Community Health Center, consisting of over 7,500 records of toddlers' physical measurements, the research focuses on training a model capable of providing early warnings. The comparison between two Ensemble Machine Learning models—Boosted K-Nearest Neighbor (BK) and Boosted Naïve Bayes (BN)—is meticulously outlined to identify the most effective model. This research aims to contribute to stunting prevention in Indonesia by deploying machine learning models to provide timely alerts, potentially mitigating the impact of this critical health issue.

II. METHODS

A. Research Design

This research starts by reviewing related literature to this research to get the background of the problem and acquire the idea of methods used to solve the problem in this research. The research continues by collecting toddler data from the Bojongsoang Community Health Center. The collected data then needs to be understood first. After understanding each feature, some features are selected to be processed later. The features that have been selected are visualized so they can be easier to explain. Before the method chosen is implemented in the data, the data needs to be cleaned and converted to compatible data types to be processed. The data then needs to be checked to see whether it is balanced. The data will be balanced first if it turns out that the data is an imbalanced dataset[2]. After preprocessing, BK and BN will try to classify the data. The performance of each method will be compared to get the conclusion. To get a better explanation, Fig. 1 will show the flowchart used for the research.

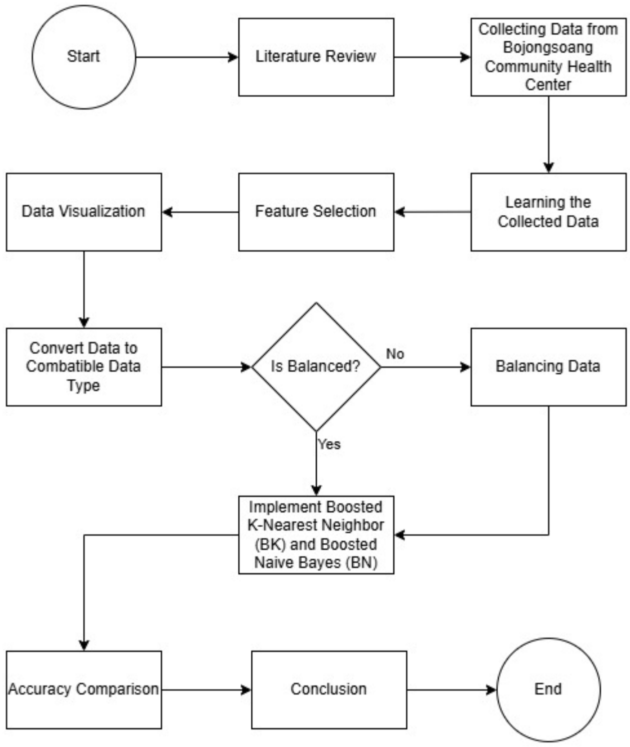

Fig. 1. Research Flowchart

B. K-Nearest Neighbor

K-Nearest Neighbor (KNN) algorithm is one of the most common algorithms used to do classification or regression on data[3]. This algorithm is employed for its simplicity and proven efficacy in classification tasks[14]. The idea of the KNN algorithm is to find similarity in each data, selecting k objects set that are the most similar, and labeling the new data based on the selected k objects set[15]. The similarity between data is determined by calculating the distance between data using Euclidean distance[15]. Euclidean distance from $l = (l_1,\cdots,l_n)$ to $m = (m_1,\cdots,m_n)$ is given as,

$$euc(l,m) = \sqrt{\sum_{i=1}^{n}(l_i - m_i)^2} \tag{1}$$

where n is the number of columns or features in the data and the smaller the distance, the more similar the data are[15]. The KNN algorithm classifies data based on the distance between each unlabeled data and all labeled data in the dataset[2]. The classification is based on KNN (smallest distances), where k is the number of neighbors involved in the voting process[2]. The class label for the test data is determined based on the majority votes[1].

C. Gaussian Naïve Bayes

A particular instance of Naïve Bayes in a continuous case context is the Gaussian Naïve Bayes classifier[11].

With j classes $v_1, v_2, ..., v_j$; with the prior probability $q_i, i = 1, j$; and $X = X_1, X_2, ..., X_m$ the m dimensional data sample[11].

In continuous case, $P(x|v_i)$ is calculated by Eq. (2):

$$P(v_i|x) = \frac{P(v_i)f(x|v_i)}{\sum_{i=1}^{m} P(v_i)f(x|v_i)} = \frac{q_i f_i(x)}{f(x)} \tag{2}$$

With:

- $P(v_i|x)$: the class a prior probability of class v_i,
- $f(x|v_i) = f_i(x)$: the probability density function of class v_i Eq. (3),
- $f(x) = q_i f_i(x) + q_k f_k(x)$, with $f(x)$ detailed in Eq. (3):

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{3}$$

Every probability function is thought to have a Gaussian distribution in practice. Thus, to obtain the density function, as shown in Eq. (3), the mean μ and variance σ^2 are computed[11].

Eq. (4) is then considered:

$$P(x_j|v_i) = f(x_j) \tag{4}$$

The name of this classifier is Gaussian Naïve Bayes. wherein each training samples of classes v_i, the mean μ_i and variance σ^2 must be calculated[11].

D. Adaboost algorithms

The Adaboost algorithm generates a very accurate prediction rule by combining imprecise rules with weak classifiers. By adding a concept of weight, the algorithm concentrates on errors caused by erroneous rules on each iteration[5]. Then, a more accurate rule is created by combining all of the rules[11]. It can be illustrated as in Fig. 2

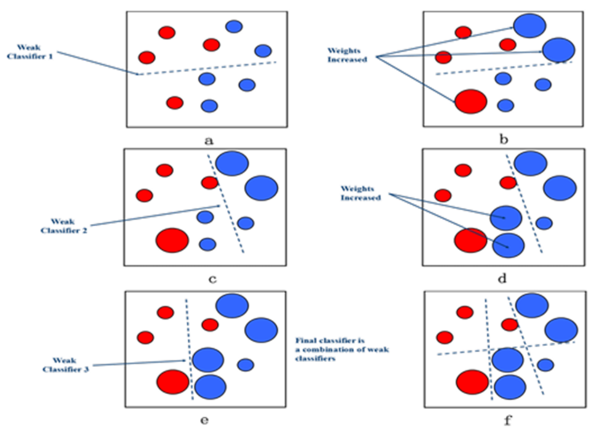

Fig. 2. Adaboost Principle

The misclassified samples are subsequently given weight, Dt, to start a boosting process. The particular misclassified samples will therefore be the subject of the subsequent K-Nearest Neighbor iteration. To guarantee that the final classifier's computation takes into consideration the given weight applied to the incorrectly categorized samples[11].

E. Boosted K-Nearest Neighbor

The BK Algorithm combines Adaboost and the KNN classifier. The KNN method is done by using a library in Python called Scikit-learn[2]. The algorithm classifies a dataset using the KNN classifier. The misclassified samples are subsequently given weight, Dt, to start a boosting process. The particular misclassified samples will therefore be the subject of the subsequent K-Nearest Neighbor iteration. To guarantee that the final classifier's computation takes into consideration the given weight applied to the incorrectly categorized samples[11]. The weighted error is determined for each iteration using Eq. (5):

$$\epsilon_t = \sum_{i:f_t(x_i)\neq y_i} D_{(t)}(i) \tag{5}$$

The BK pseudo algorithm can be seen in Algorithm 1.

Algorithm 1 BK Algorithm

Require: S: Training dataset
Require: T: Testing Dataset
Require: K: number of nearest neighbors
Require: N: Maximum number of iterations
Require: m: number of elements in testing data
Require: `arrClassifier`: empty array for base classifier

```
for n = 1 to N do
    Initialize KNeighborsClassifier(K) = knn
    Train Model with knn(S)
    Predict dataset with knn(T)
    for j = 1 to m do
        if Predicted result ≠ values in T then
            Reweight using Eq. (5)
        end if
    end for
    Calculate the accuracy and F1-Score = confusionMatrix
    Assign the classifier to arrClassifier
end for
return last entity of arrClassifier confusionMatrix
```

The last entity from the array of classifiers (arrClassifier) was selected as the chosen classifier, as it is expected to be the most effective among all classifiers.

This is attributed to the fact that it has undergone all iterations of the boosting process, thereby benefiting from the cumulative improvements achieved throughout the training.

F. Boosted Naïve Bayes

The boosting idea behind the BK is also present in the Boosted Naïve Bayes Algorithm. It blends the Gaussian Naïve Bayes classifier with Adaboost. Scikit-learn is a Python library used to implement the Gaussian Naïve Bayes algorithm in this research. The Gaussian Naïve Bayes classifier is used by the method to classify a dataset. The misclassified samples are subsequently given weight, Dt, to start a boosting process. The particular misclassified samples will therefore be the subject of the subsequent Gaussian Naïve Bayes iteration. To guarantee that the final classifier's computation takes into consideration the given weight applied to the incorrectly categorized samples[11].The weighted error is determined for each iteration using Eq. (5). The BN pseudo algorithm can be seen in Algorithm 2.

Algorithm 2 BN Algorithm

Require: S: Training dataset
Require: T: Testing Dataset
Require: N: Maximum number of iteration
Require: m: number of testing data element
Require: `arrClassifier`: empty array for base classifier

```
for n = 1 to N do
    initialize GaussianNB() = nb
    Train Model with nb(S)
    Predict dataset with nb(T)
    for j = 1 to m do
        if Predicted result ≠ values in T then
            Reweight using Eq. (5)
        end if
    end for
    Calculate the accuracy and F1-Score = confusionMatrix
    Assign the classifier to arrClassifier
end for
return last entity of arrClassifier, confusionMatrix
```

III. RESULTS AND DISCUSSIONS

A. Exploratory Data Analysis

The dataset used in this research is a dataset that contains personal information and the health status of toddlers obtained from the Bojongsoang Community Health Center. The dataset contains some columns that explain the child's information, the address, where the child was measured, and the result of the measurements. This data is a measurement in August 2024 consisting of 7,507 records. The selected features are,

- Age (A)
- Weight (W)
- Height (H)
- Z-Score

Besides that, the Height/Age column in the dataset becomes the target class because it contains information on whether the child is **normal**, **short**, or **very short** (has a stunting condition). Here are the distributions of some features to classify the dataset.

Fig. 3 shows that the age feature has no outlier, meaning the age feature does not have extreme value. On the other hand, the height feature has many low outliers, while the weight feature has many high outliers, and the Z-Score data has many high and low outliers, indicating that this feature has many extreme values that are too far from the median and quartiles.

Fig. 4 shows the **short** and **very short** data combined has less than 2% while the normal data has more than 98% based on the class column. A dataset is imbalanced if the quantity difference between the classes is too high[16]. Thus, this means the dataset used in this research is imbalanced.

B. Preprocessing

The stunting dataset that will be classified in this research has 7,507 records. Feature class and Nutrition level contain string-type data, even though the actual values of those columns are categorical. Therefore, the class and Nutrition

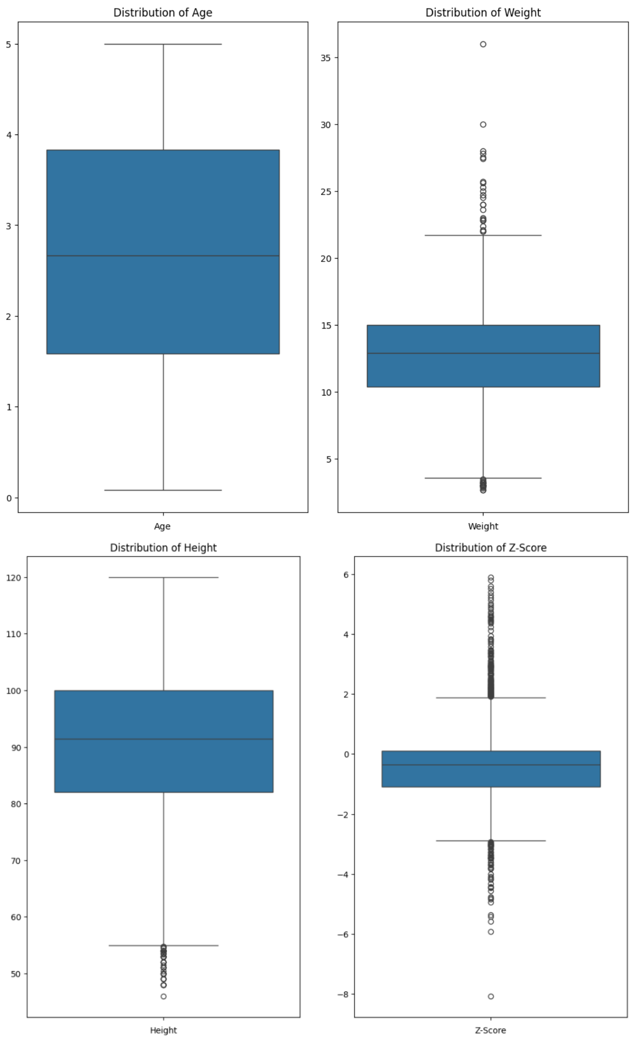

Fig. 3. Distributions of used columns

level columns must be changed into integer-type data for better usage. In the class column, the **very short** value is

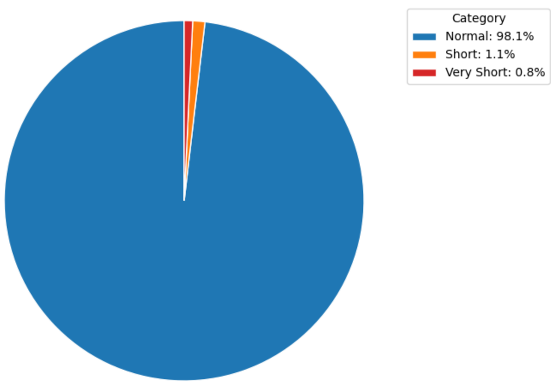

Fig. 4. Comparison before balancing

converted to 2, the **short** value is converted to 1, and the **normal** value is set to 0. Here, the **very short** value means positive stunting, while the **short** value rarely means positive stunting, but indicates that the child is close to stunting, and the **normal** value means negative stunting.

Based on Fig. 4, the dataset still has an imbalance issue, between the category of **normal** and **very short** with the

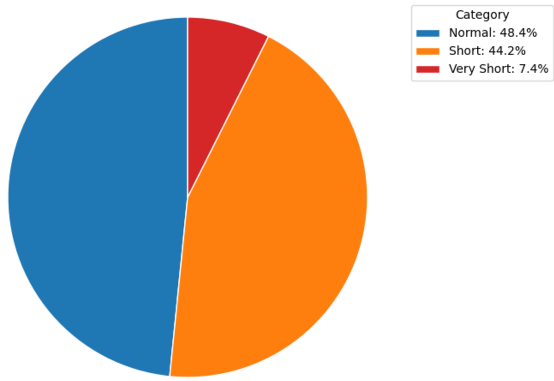

Fig. 5. Comparison after balancing

short combined ratio being 98.1 : 1.9. When classification is performed on an imbalanced dataset, it is observed that the majority class is classified more effectively and accurately than the minority class. In contrast, the minority class is often misclassified as the majority class[8]. To address the class imbalance issue, minimizing the gap between categories through oversampling and trimming the oversized dataset is essential. Oversampling can be achieved by generating

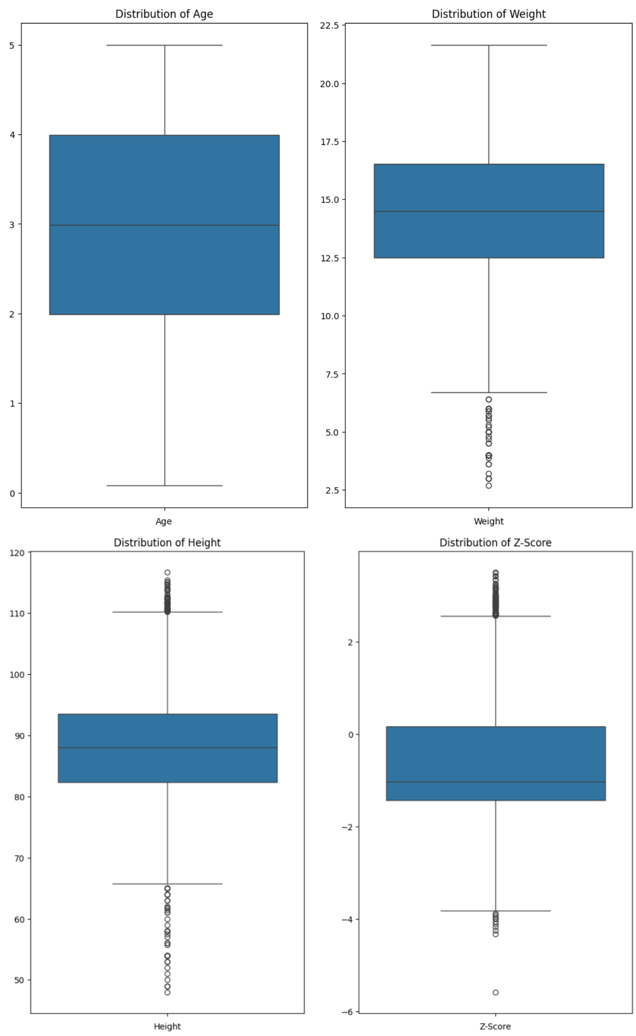

Fig. 6. Distributions after balancing

synthetic data that reflects the distribution of the existing dataset, thereby ensuring a sufficient number of samples. This approach helps to prevent undersampling in both the training and testing phases. During the data generation process, it is ensured that the generated data accurately represents the original distribution by employing appropriate techniques, such as the use of WHO standardized standard deviation. After a substantial dataset was generated, the data was categorized based on the calculated Z-scores. Specifically, every data with Z-Scores less than -2 was classified as **very short**, those with Z-Scores less than -1 were labeled as **short**, and all the other data were considered **normal**. Following this categorization, the **normal** data was trimmed to minimize the gap between the combined **very short** and **short** categories and the **normal** category.

The number of records in the **normal** class now has 148,865 records, the **short** class has 135,825 records, and the **very short** class has 22,816 records due to oversampling and trimming the **normal** data. The comparison between classes after balancing the data can be seen in Fig. 5.

Despite balancing the data and stabilizing the comparison between classes, the distribution within the dataset remains significantly unbalanced. Fig. 6 illustrates the distribution post-balancing, revealing numerous outliers that persist due to the extensive variability in the dataset, similar to the original distribution. Consequently, a standard machine learning model is insufficient to address this issue. Therefore, as discussed in this research, an additional approach is required to support the model, specifically boosting.

C. Boosted K-Nearest Neighbor Implementation

In the initial implementation, each data is assigned a weight of $\frac{1}{m}$ where m represents the total number of samples. The dataset is then divided into training and testing subsets using a $9:1.$ ratio. Subsequently, the classes within both the training and testing data are separated into two sets: x, which includes the classes that the model will learn to classify, and y, which represents the target classes. The model is trained and tested using the prepared data, with a reweighting mechanism applied to each misclassified sample during each iteration. This approach boosted the base classifier by ensuring the model focuses more on learning from the misclassified samples in each iteration. After several tests, the optimal value for the neighbors' parameter in the K-Nearest Neighbors (KNN) algorithm, utilized as the base classifier, is determined to be $n = 3$. The results of these experiments are presented in Table I.

TABLE I
CONFUSION MATRIX FOR THE BOOSTED KNN METHOD

		Predicted Values		
		Normal	*Short*	*Very Short*
Actual Values	***Normal***	14899	0	0
	Short	179	13406	0
	Very Short	243	0	2042

D. Boosted Naïve Bayes Implementation

This method has the same initial implementation as the BK method, with the primary distinction being the use of Gaussian Naïve Bayes as the base classifier. Each data is assigned a weight of $\frac{1}{m}$ where m represents the total number of samples. The dataset is then divided into training and testing subsets using a $9:1.$ ratio. Subsequently, the classes within both the training and testing data are separated into two sets: x, which includes the classes that the model will learn to classify, and y, which represents the target classes. The model is trained and tested using the same prepared data as the BK method, with a reweighting mechanism applied to each misclassified sample during each iteration. This approach boosted the base classifier by ensuring the model focuses more on learning from the misclassified samples in each iteration. The results of these experiments are presented in Table II.

TABLE II
CONFUSION MATRIX FOR THE BN METHOD

		Predicted Values		
		Normal	*Short*	*Very Short*
Actual Values	***Normal***	14899	0	0
	Short	3753	9832	0
	Very Short	2209	0	76

E. Accuracy Comparison

The methods that are used to classify the stunting dataset perform differently. Here are the classification results between BK and BN.

TABLE III
ACCURACY AND F-1 SCORE FOR COMPARISON

Method	Accuracy	F-1 Score Macro Avg
BK	98.62%	97.44%
BN	80.62%	57.91%

Table III demonstrates that although the accuracy of the BN method is not as high as that of the BK method, the classification performance of the BN method surpasses that reported in previous research[7], achieving an accuracy exceeding 80%. Furthermore, the F-1 score macro average of the BK method is nearly identical to that of the BK method in previous research, despite the preprocessing results in this study being significantly more imbalanced compared to those in earlier work[2]. These findings indicate that the boosting technique has effectively enhanced both the KNN and Naïve Bayes methods, despite the unsatisfactory results of the preprocessing step.

IV. CONCLUSION

This research demonstrates that the BK and BN methods yield different results. As shown in Table III, the accuracy of the BN method is lower than that of the BK method. However, the BN method exhibits a significant improvement in their accuracy compared to previous studies, indicating enhanced performance overall. Table III further illustrates the comparison between the BK and BN models. The BK model, which employs AdaBoost to enhance KNN as the base classifier, achieves an accuracy of 98.62% and an F-1 score of 97.44%. Meanwhile, the BN model, which uses AdaBoost to boost Naïve Bayes as the base classifier, achieves an accuracy of 80.62% and an F-1 score of 57.91%. Therefore, the BK and BN have succeeded in boosting the performance of the original KNN and Naïve Bayes method. Future work should focus on further balancing the training and testing data prior to implementing the BN method, as this is expected to enhance both the F-1 score and the accuracy of the model. Additionally, this research concludes that the BK method is superior for classifying stunting data among toddlers.

References

[1] A. Almomany, W. R. Ayyad, and A. Jarrah. Optimized implementation of an improved knn classification algorithm using intel fpga platform: Covid-19 case study. *Journal of King Saud University-Computer and Information Sciences*, 34(6):3815–3827, 2022.

[2] M. G. Daffa and P. H. Gunawan. Stunting classification analysis for toddlers in bojongsoang: A data-driven approach. In *2024 2nd International Conference on Software Engineering and Information Technology (ICoSEIT)*, pages 42–46, Bandung, Indonesia, 2024.

[3] R. Devika, S. V. Avilala, and V. Subramaniyaswamy. Comparative study of classifier for chronic kidney disease prediction using naive bayes, knn and random forest. In *2019 3rd International conference on computing methodologies and communication (ICCMC)*, pages 679–684. IEEE, 2019.

[4] C. Fannany, P. H. Gunawan, and N. Aquarini. Machine learning classification analysis for proactive prevention of child stunting in bojongsoang: A comparative study. In *Proceedings of the Conference*, volume 4.1, pages 1–5, 2024.

[5] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[6] G. Haixiang, L. Yijing, L. Yanan, L. Xiao, and L. Jinling. Bpso- adaboost-knn ensemble learning algorithm for multi-class imbalanced data classification. *Engineering Applications of Artificial Intelligence*, 49:176–193, 2016.

[7] V. Herliansyah, R. Latuconsina, and A. Dinimaharawati. Prediksi stunting pada balita dengan menggunakan algoritma klasifikasi naïve bayes. *eProceedings of Engineering*, 8(5), 2021.

[8] T. R. Hoens and N. V. Chawla. *Imbalanced Datasets: From Sampling to Classifiers*. John Wiley & Sons, Ltd, 2013.

[9] H. Janawisuta, P. H. Gunawan, and N. Indwiarti. Early detection of stunting in indonesian toddlers: A machine learning approach. In *Proceedings of the Conference*, volume 2, pages 12–16, 2024.

[10] G. A. F. Khansa and P. H. Gunawan. Predicting stunting in toddlers using knn and naïve bayes methods. In *Proceedings of the Conference*, volume 3, pages 17–21, 2024.

[11] A. Pizzo, P. Teyssere, and L. Vu-Hoang. Boosted gaussian bayes classifier and its application in bank credit scoring. *Journal of Advanced Engineering and Computation*, 2(2):131–138, 2018.

[12] I. W. Pranata, R. A. Yuniawati, N. A. Robbika, G. H. Permadi, M. N. Anwar, R. A. D. E. Putri, H. U. Wusqo, S. S. Arsyie, N. Novel, and S. M. Fransisca. Prevention of stunting through improving maternal parenting and early detection of pregnancy risk factors. *Jurnal Pengabdian Masyarakat Bestari*, 1(9):1025–1034, 2022.

[13] K. Rahmadhita. Permasalahan stunting dan pencegahannya. *Jurnal Ilmiah Kesehatan Sandi Husada*, 11(1):225–229, 2020.

[14] A. T. A. Sibuea, P. H. Gunawan, and N. Indwiarti. Classifying stunting status in toddlers using k-nearest neighbor and logistic regression analysis. In *Proceedings of the Conference*, volume 216, pages 6–11, 2024.

[15] A. S. M. Sohail and P. Bhattacharya. Classification of facial expressions using k-nearest neighbor classifier. In *Computer Vision/Computer Graphics Collaboration Techniques: Third International Conference, MIRAGE 2007*, pages 555–566. Springer, 2007.

[16] M. A. U. H. Tahir, S. Asghar, A. Manzoor, and M. A. Noor. A classification model for class imbalance dataset using genetic programming. *IEEE Access*, 7:71013–71037, 2019.