Stunting Classification Analysis for Toddlers in Bojongsoang: A Data-Driven Approach

1st Muhammad Ghiyaats Daffa
School of Computing
Telkom University
Bandung, Indonesia
ghiyats@student.telkomunversity.ac.id

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

Abstract—**Stunting is one of the health problem priorities for children in Indonesia. Prevention of stunting in toddlers is needed to avoid the long-term effects for both the toddlers and the public. Stunting prevention can be done by monitoring the growth of toddlers. Therefore, a system that can predict stunting conditions in toddlers is needed. Machine learning offers many methods that can be used to build a system to predict stunting conditions in toddlers. This research analyzes some machine learning models that are potentially suitable to predict stunting classes, which are K-Nearest Neighbor (KNN), Random Forest (RF), and Ensemble Learning called Boosted KNN (BK). The dataset has an imbalance issue in this research, with the stunting data at only 1% of the total dataset. Therefore, oversampling of the dataset is done by generating a random dataset based on the distribution of the data that are classified as the minority class. The results of elaborating on this oversampling are shown to be satisfying. Applying imbalanced data gives an average of 98% accuracy for all methods used; however, the F-1 score macro average is shown not optimal for each of the methods, with 51.95% for KNN, 52.45% for RF, and 53.55% for BK. After the data is balanced by oversampling, the F-1 score macro average for all methods substantially increases. The new results were 93.55% for KNN, 97.70% for RF, and 98.00% for BK, underscoring the critical role of addressing data imbalance in improving predictive accuracy.**

Index Terms—**Stunting, Machine Learning, K-Nearest Neighbor, Random Forest, Ensemble Learning**

I. Introduction

Stunting ranks among the foremost health concerns for children in Indonesia, denoting a condition where children under the age of five fail to thrive due to chronic malnutrition within the initial 1000 days of life, resulting in stunted growth [1], [2], [3]. Addressing this issue is crucial to circumvent enduring consequences for toddlers and the wider public, preventing long-term health decline [4]. Notably, President Joko Widodo has set a target to reduce stunting prevalence from 21% in 2022 to 14% in 2024, as outlined on the https://sehatnegeriku.kemkes.go.id website. Effective stunting prevention involves meticulously monitoring toddler growth, underscoring the need for an implementable system to predict stunting conditions.

To predict stunting conditions in toddlers, a suitable algorithm to perform classification is needed. Based on the research in [5], the K-Nearest Neighbor performs well in predicting stunting conditions. In 2020, the stunting problem was discussed in [5], which used K-Nearest Neighbor (KNN) to predict toddler stunting conditions. The KNN method with K-Fold cross-validation got 95.26% accuracy as the highest result after several iterations. On the other hand, the research in [6] used the Naive Bayes method to predict the same stunting dataset and got 64.36% accuracy as the highest result.

In their effort to address stunting prevention, researchers [7] introduced the Sagita application in their recent research. Sagita is a valuable tool for parents, enabling them to monitor their toddlers' height and weight growth. Beyond mere observation, Sagita also sends timely notifications to parents, alerting them to the necessity of professional medical monitoring to ensure optimal nutrition for their child. The post-test results of the research indicate that Sagita effectively enhances parental understanding of stunting, encourages the adoption of healthier food alternatives, and disseminates other crucial information. Despite its success in information dissemination, Sagita currently possesses limited features [7]. A promising avenue for enhancement involves incorporating a machine learning-based prediction system into Sagita, as suggested in [7].

This study seeks to develop a robust machine-learning model dedicated to detecting stunting conditions in toddlers. Leveraging a dataset sourced from the Bojongsoang Community Health Center, consisting of over 6000 records of toddlers' physical measurements, the research focuses on training a model capable of providing early warnings. The comparison between three machine learning models—K-Nearest Neighbor, Random Forest, and the novel Ensemble Learning, Boosted KNN (a fusion of Random Forest principles with K-Nearest Neighbor)—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.

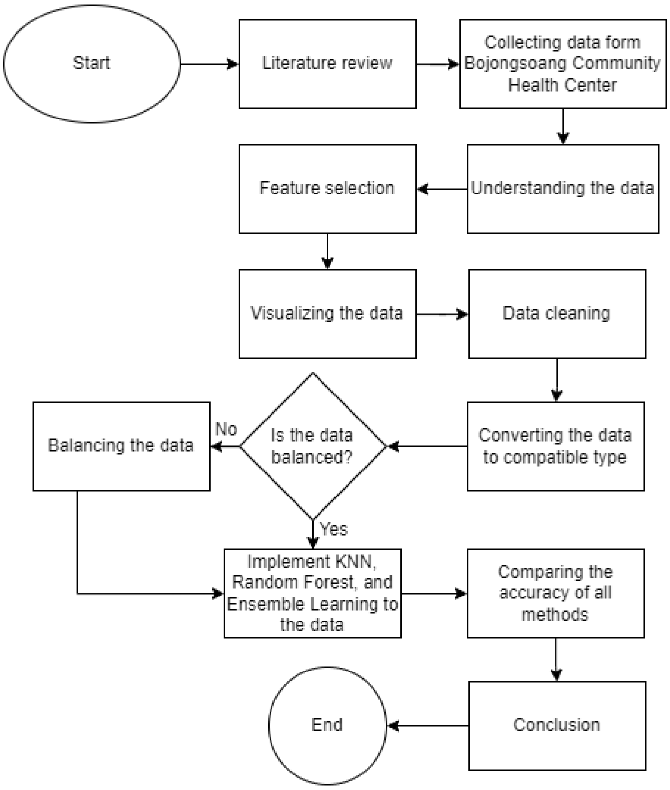
II. Methods

A. Research Design

This research starts by reviewing related literature to this research to get the background of the problem and acquire some method ideas to solve the problem. 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

979-8-3503-1750-3/24/$31.00 ©2024 IEEE

if it turns out that the data is an imbalanced dataset. After preprocessing, K-Nearest Neighbor (KNN), Random Forest (RF), and Boosted KNN (BK) will try to classify the data. The performance of each method will be compared to get a conclusion. In order to get a better explanation, Fig. 1 will show the flowchart used for the research.



Start
Literature review
Collecting data form Bojongsoang Community Health Center
Understanding the data
Feature selection
Visualizing the data
Data cleaning
Converting the data to compatible type
Is the data balanced?
No
Balancing the data
Yes
Implement KNN, Random Forest, and Ensemble Learning to the data
Comparing the accuracy of all methods
Conclusion
End

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[8]. 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[9]. The similarity between data is determined by calculating the distance between data using Euclidean distance[9]. Euclidean distance from $l = (l_1, \cdots, l_n)$ to $m = (m_1, \cdots, m_n)$ is given as,

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

where n is the number of columns or features in the data and the smaller the distance, the more similar the data are[9]. The KNN algorithm classifies data based on the distance between each unlabeled data and all labeled data in the dataset. The classification is based on k-nearest neighbors (smallest distances), where k is the number of neighbors involved in the voting process. The class label for the test data is determined based on the majority votes[10]. The KNN algorithm can be seen in Algorithm 1.

Algorithm 1 K-Nearest Neighbors (KNN)

```
**Require:** $X$: training data
**Require:** $Y$: class labels for $X$
**Require:** $K$: number of nearest neighbors
**Require:** $T$: testing data
**Ensure:** Prediction of class for $T$
  **for all** $l \in T$ **do**
      Calculate the distance from $l$ to $m$ where $\forall m \in X$ using the Euclidean Eq.(1)
  **end for**
  Classify each data point $l \in T$ based on the majority vote of the nearest neighbors
```

C. Random Forest

Random Forest (RF) is a data mining tool commonly used for classification or regression by growing many sets of Decision Trees and determining the result based on voting from those Decision Tree sets[11]. Decision Tree (DT) is a tree that represents each of the attributes of data as a node and the predicted answer as an edge[12]. The RF algorithm selects the most important attributes from the dataset to be used as a node in each DT to perform classification[13]. The DT, which is the base of the RF model, does not perform well on complex datasets. Therefore, the voting in RF boosts the performance of DT to give better results[14]. The RF algorithm can be seen in Algorithm 2[15].

Algorithm 2 Random Forest

```
**Require:** Training dataset, Test dataset, number of Decision Trees
**Ensure:** Classified Test dataset
  Generate a forest:
  **for** $j$ in number of trees **do**
      **for** $i$ in number of nodes **do**
          Randomly select $n$ features from the dataset
          **for** each feature in the random feature set **do**
              Calculate information gain from the feature
          **end for**
          Create a node using the feature with the best information gain
      **end for**
  **end for**
  Determine class prediction:
  **for** each class label **do**
      Calculate probabilities from all trees in the random forest
  **end for**
  Determine the class based on majority vote
```

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 2022 consisting of 6,677 records. The selected features are,

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

- Nutrition Level (W/H)
- Z-Score

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

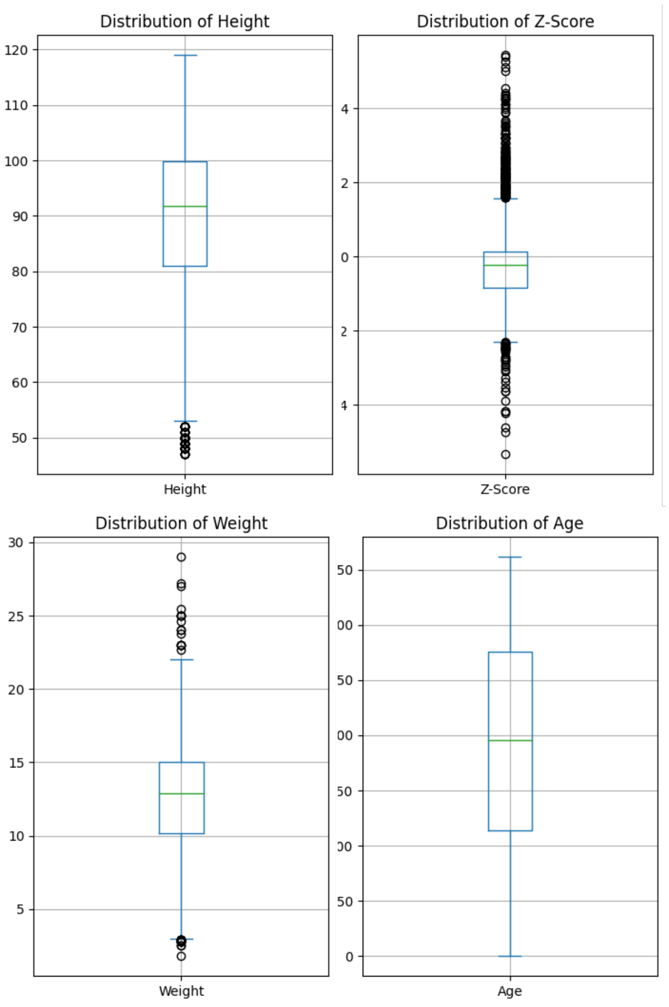


Fig. 2. Distributions of the columns that are used

Fig. 2 shows that the height feature has many low outliers, while the weight feature has many high outliers. On the age feature, there is no outlier, meaning the age feature does not have extreme value. On the other hand, 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. 3 shows the stunting data is only 1% while the normal data is 99% based on the class column. A dataset is imbalanced if the quantity difference between both classes is too high[16]. Thus, this means the dataset used in this research is imbalanced. Despite having a stunting condition, most of the children are still getting good nutrition. The comparison of the nutrition among the stunting children is presented in Fig. 4.

B. Preprocessing

The stunting dataset that will be classified in this research has 6677 records. From those 6677 records, there is a record with a missing value in most of the columns and some records with missing values in the target class column. The records that contain missing values cannot be used in the classification process. Therefore, those records are deleted in order to remove all of the missing values.

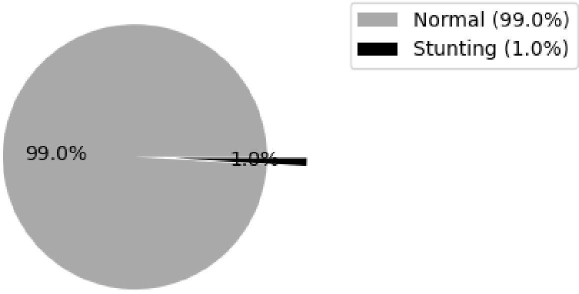


Fig. 3. Comparison of normal and stunting quantity.

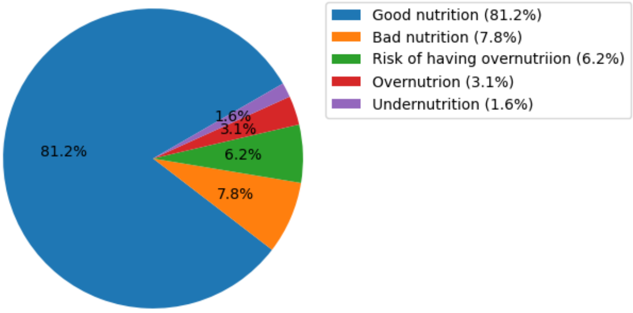


Fig. 4. Comparison of stunting children's nutrition.

Feature class and Nutrition level contain string-type data, even though the actual values of those columns are categorical. Therefore, the class and Nutrition level columns must be changed into integer-type data for better usage. In the class column, the ”Stunting” value is converted to 1 and the ”Normal” value is set to 0. Here, the ”Normal” value means negative stunting and the ”Stunting” value means positive stunting. Every value in the Nutrition level column is transformed into an integer in the range of 0 to 5 representing each category which is ”Obesity”, ”Overnutrition”, ”Bad Nutrition”, ”Risk of having overnutrition”, ”Undernutrition”, and ”Good nutrition”.

Based on Fig. 3, the dataset still has an imbalance issue, with the stunting and normal ratio being 1:99. If we perform classification on an imbalanced dataset, the majority class will perform effectively and more accurately than the minority class. In contrast, the minority class will be falsely classified as the majority class most of the time[17]. In order to handle this issue, we need to increase the quantity of the minority class by oversampling the dataset so the difference between both classes is reduced. Oversampling the dataset can be done by generating a random dataset based on the distribution of the data that are classified as the minority class. In order to generate a random sample, the upper and lower limits for the random data need to be determined.

According to the boxplots shown in Fig. 5, while the first quartile of each feature will determine the lower limit for the random sample, the third quartile of each feature is the upper limit for the random sample so that the random sample generated is undoubtedly classified as stunting. The stunting labeled data must be increased until it reaches at least 6000 records (similar to ”normal” labeled data) to balance the dataset. Therefore, 6000 iterations will generate random data, and each iteration will produce data that is not higher than the upper limit or lower than the lower limit. The number of records in the stunting class is now 6064, whereas the ”normal” class has 6608 records due to oversampling the minority class.

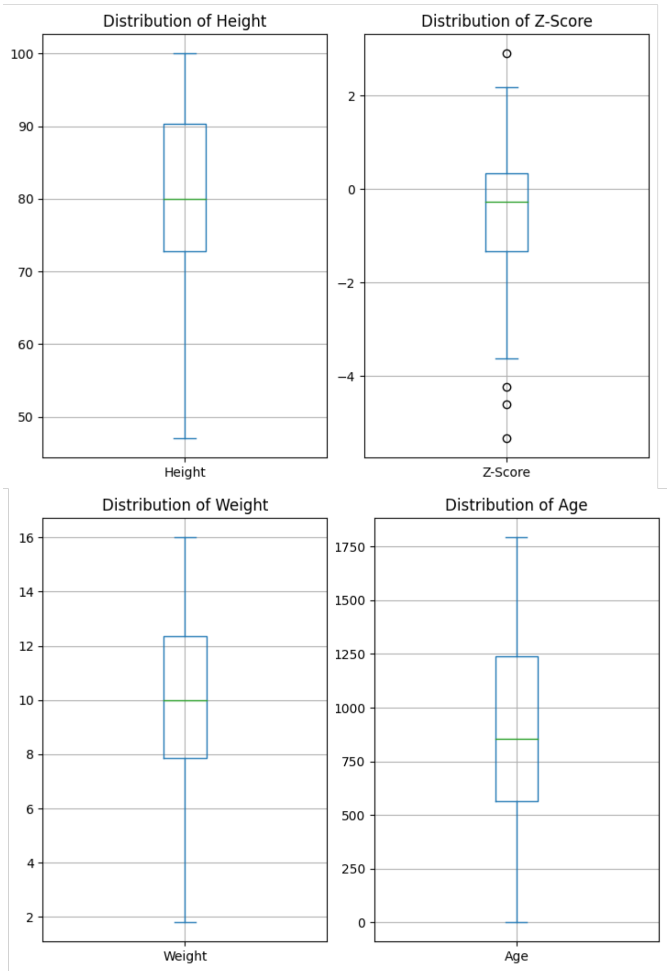


Fig. 5. Distributions of the positive stunting data

C. K-Nearest Neighbor Implementation

KNN is used to classify stunting data in this research. The implementation of the KNN method is done by using a library in Python called Scikit-learn. After several tests, the best input for the n parameter in this KNN implementation is $n = 3$. The dataset is split into train and test data with the ratio $7 : 3$ and then scaled into the interval of $[0, 1]$ before it is classified. After the train and test data are scaled, the KNN model is fitted into the dataset, and the results are presented in Table I.

D. Random Forest Implementation

TABLE I
CONFUSION MATRIX FOR THE KNN METHOD

		Predicted Values	
		Stunting	**Normal**
Actual Values	**Stunting**	1902	117
	Normal	136	1647

Apart from KNN, RF is also used to try to classify the stunting data. Using Scikit-learn, the same library as the KNN method, the RF generates 100 random decision trees to classify the data. The RF model is fitted to the dataset using the same ratio for the train and test dataset that has been scaled. The result of the RF method is represented using a confusion matrix, as shown in Table II.

TABLE II
CONFUSION MATRIX FOR THE RF METHOD

		Predicted Values	
		Stunting	**Normal**
Actual Values	**Stunting**	1921	66
	Normal	40	1787

E. Boosted KNN Implementation

After using each KNN and RF, the dataset will try to be classified by a model that combines KNN and RF concept called Boosted KNN (BK). While RF usually contains many DT [11], the DT is swapped with many KNN with different random inputs of k neighbors. By ensembling the concept of RF to KNN, the accuracy of the classification result is expected to be higher than the original KNN model. Using the same dataset that has been split and scaled, 100 BK are generated to classify those data. In generating those KNN models for BK, an odd number between 1 and 200 is chosen as the k neighbors for each KNN model. Since selecting the k neighbors is random, and there is no restriction on duplicate k neighbors, the k neighbors in each KNN model could be the same or different. Table III shows the results of the BK model for the stunting dataset represented in the confusion matrix.

TABLE III
CONFUSION MATRIX FOR THE BK MODEL

		Predicted Values	
		Stunting	**Normal**
Actual Values	**Stunting**	1935	57
	Normal	27	1783

F. Accuracy Comparison

The methods that are used to classify the stunting dataset perform differently. Besides using different methods, the classification performs differently when imbalanced and balanced datasets are classified. Here are the results of imbalanced data classification after 20 iterations.

TABLE IV
ACCURACY AND F-1 SCORE FOR IMBALANCED DATASET

Method	**Accuracy**	**F-1 Score Macro Avg**
KNN	98.92%	51.95%
Random Forest	98.83%	52.45%
Boosted KNN	98.80%	53.55%

As seen in Table IV, even though the accuracy of each method is up to 98%, the F-1 score macro averages are only approximately 50%. This means the classifications do not perform well for one of the classes, in this case, the stunting class. Here are the results of balanced data classification after 20 iterations.

TABLE V
ACCURACY AND F-1 SCORE FOR BALANCED DATASET

Method	Accuracy	F-1 Score Macro Avg
KNN	93.67%	93.55%
Random Forest	97.76%	97.70%
Boosted KNN	97.94%	98.00%

Table V shows that even though the accuracy is not as high as the imbalanced dataset, the classification on the balanced dataset performs better based on the F-1 score macro average reaching more than 90%. Besides that, the performance of Boosted KNN is higher than that of KNN and Random Forest. Therefore, the ensemble learning method has successfully boosted the KNN method and surpassed Random Forest's performance.

IV. CONCLUSION

This research has shown that using imbalanced and balanced datasets gives different results. Based on Table IV, when the methods classify the imbalanced dataset, all methods have good accuracy. However, the F-1 score macro average of every method indicates that the performance of the models needs to be improved. The imbalanced data training resulted in the models classifying most cases as the majority class, so the models do not perform well on the rare class or the minority class. Table V shows that the accuracy of the balanced dataset is lower than that of the imbalanced dataset. However, each method's F-1 score macro average is significantly improved rather than the imbalanced dataset. Therefore, the performance of every method for both classes has improved. Apart from good results of F-1 scores, Table V shows a comparison between KNN, RF, and BK. The KNN method has 93.67% of accuracy and 93.55% of F-1 score. Meanwhile, the Random Forest has 97.76% of accuracy and 97.70% of F-1 score. Thus, this result indicates that Random Forest performs better than the KNN method. On the other hand, the BK model (combining the concept of Random Forest and Ensemble Learning) achieves 97.94% of accuracy and 98.00% of F-1 score. Therefore, the BK has succeeded in boosting the performance of the original KNN method. Future work in developing a machine learning model for classification can be related to implementing the RF concept to another machine learning method, such as Naive Bayes or Logistic Regression. Besides that, the BK model can be implemented into text or image classification.

REFERENCES

[1] D. D. Astuti, R. B. Adriani, and T. W. Handayani, "Pemberdayaan masyarakat dalam rangka stop generasi stunting," *JMM (Jurnal Masyarakat Mandiri)*, vol. 4, no. 2, pp. 156–162, 2020.

[2] C. R. Titaley, I. Ariawan, D. Hapsari, A. Muasyaroh, and M. J. Dibley, "Determinants of the stunting of children under two years old in indonesia: A multilevel analysis of the 2013 indonesia basic health survey," *Nutrients*, vol. 11, no. 5, p. 1106, 2019.

[3] Ramli, K. E. Agho, K. J. Inder, S. J. Bowe, J. Jacobs, and M. J. Dibley, "Prevalence and risk factors for stunting and severe stunting among under-fives in north maluku province of indonesia," *BMC pediatrics*, vol. 9, pp. 1–10, 2009.

[4] S. Sagita and K. N. Siregar, "Faktor-faktor risiko stunting pada balita di indonesia: Suatu scoping review," *Media Publikasi Promosi Kesehatan Indonesia (MPPKI)*, vol. 5, no. 6, pp. 654–661, 2022.

[5] H. H. Sutarno, R. Latuconsina, and A. Dinimaharawati, "Prediksi stunting pada balita dengan menggunakan algoritma klasifikasi k-nearest neighbors," *eProceedings of Engineering*, vol. 8, no. 5, 2021.

[6] V. Herliansyah, R. Latuconsina, and A. Dinimaharawati, "Prediksi stunting pada balita dengan menggunakan algoritma klasifikasi naïve-bayes," *eProceedings of Engineering*, vol. 8, no. 5, 2021.

[7] M. H. Barri, F. Alia, L. Novamizanti, R. Purnamasari, F. Akhyar, T. Fahrudin, P. H. Gunawan, and S. Mandala, "Aksi cegah stunting melalui aplikasi sagita: Status gizi balita," *JMM (Jurnal Masyarakat Mandiri)*, vol. 7, no. 2, 2023.

[8] 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)*. IEEE, 2019, pp. 679–684.

[9] 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, Rocquencourt, France, March 28-30, 2007. Proceedings 3*. Springer, 2007, pp. 555–566.

[10] 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*, vol. 34, no. 6, pp. 3815–3827, 2022.

[11] N. Dogru and A. Subasi, "Traffic accident detection using random forest classifier," *2018 15th Learning and Technology Conference, L and T 2018*, pp. 40–45, 2018.

[12] V. Matzavela and E. Alepis, "Decision tree learning through a predictive model for student academic performance in intelligent m-learning environments," *Computers and Education: Artificial Intelligence*, vol. 2, p. 100035, 2021.

[13] V. Jackins, S. Vimal, M. Kaliappan, and M. Y. Lee, "Ai-based smart prediction of clinical disease using random forest classifier and naive bayes," *The Journal of Supercomputing*, vol. 77, pp. 5198–5219, 2021.

[14] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, "A comparison of random forest variable selection methods for classification prediction modeling," *Expert systems with applications*, vol. 134, pp. 93–101, 2019.

[15] R. G. Devi and P. Sumanjani, "Improved classification techniques by combining knn and random forest with naive bayesian classifier," in *2015 IEEE international conference on engineering and technology (ICETECH)*. IEEE, 2015, pp. 1–4.

[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*, vol. 7, pp. 71 013–71 037, 2019.

[17] T. R. Hoens and N. V. Chawla, *Imbalanced Datasets: From Sampling to Classifiers*. John Wiley & Sons, Ltd, 2013, ch. 3, pp. 43–59.