SAINSTECH
INNOVATION JOURNAL

SIJ, Volume 9, Nomor 1, May 2026, Hal. 612-620

Performance Analysis of K-Nearest Neighbors and Naive Bayes Algorithms in Stunting Risk Classification in Toddlers Using Public Dataset

[Nurhikmayani][1*], [Syahrani Lonang][2], [Ahmad Fatoni Dwi Putra][3]

[1] [Program Studi Ilmu Komputer, Fakultas Sains dan Teknologi, Universitas Qamarul Huda Badaruddin Bagu1)]
[2] [Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Qamarul Huda Badaruddin Bagu)]

Corresponding Author

Received on July 01, 2026, first revision on July 30, 2026, approved on July 31, 2026, published online on July 30, 2026

Abstract: *Stunting is a chronic nutritional problem in toddlers that can affect physical growth, cognitive development, and children's health quality in the future. This study aims to analyze and compare the performance of the K-Nearest Neighbors (KNN) and Naïve Bayes algorithms in classifying stunting risk in toddlers using a public Kaggle dataset. The research was conducted through several stages, including data preprocessing, data cleaning, normalization using the Min-Max method, data balancing using SMOTE-ENN, splitting training and testing data, and parameter optimization using Grid Search. Model evaluation was carried out using a Confusion Matrix with accuracy, precision, Recall, F1-score, ROC Curve, and AUC Score metrics. The results showed that the KNN algorithm performed better than the Naïve Bayes algorithm in classifying stunting risk. The KNN algorithm produced higher accuracy, precision, Recall, and F1-score values, as well as more optimal ROC-AUC values for each classification class. Based on these evaluation results, the KNN algorithm was considered more effective and stable in detecting stunting risk in toddlers compared to the Naïve Bayes algorithm. Therefore, the KNN algorithm can be used as an effective method to support early stunting risk detection based on Machine Learning.*

Keywords: *Stunting, Machine Learning, K-Nearest Neighbors, Naïve Bayes, Classification*

1. INTRODUCTION

Stunting is a condition of stunted growth in toddlers caused by long-term chronic malnutrition. This condition can impact physical growth, cognitive development, and the child's future health[1]. Indonesia still faces a high stunting prevalence rate, which is 24.22% in 2022, so a more effective approach is needed in the process of early detection and classification of stunting risks. [2].

The development of machine learning technology provides solutions in the health sector, especially for disease classification and prediction [3]. Machine learning algorithms such as K-Nearest Neighbors (KNN) and Naive Bayes are widely used due to their distinct characteristics in data classification. However, research specifically comparing these two algorithms in stunting risk classification is still limited[4].

Beberapa penelitian terdahulu menunjukkan bahwa KNN memiliki performa yang baik dalam klasifikasi status gizi dan stunting, sedangkan Naïve Bayes unggul pada proses klasifikasi probabilistik. Penelitian ini menggunakan public dataset dari Kaggle dengan jumlah data yang besar serta menerapkan metode SMOTE-ENN dan Grid Search untuk meningkatkan kualitas klasifikasi.

Penelitian ini bertujuan menganalisis performa algoritma KNN dan Naïve Bayes dalam klasifikasi risiko stunting menggunakan evaluasi accuracy, precision, recall, F1-score, ROC Curve, dan AUC Score.

2. RESEARCH METHOD

This research was used to analyze the accuracy of the K-Nearest Neighbors (KNN) and Naïve Bayes (NB) models in predicting the risk of stunting in toddlers. The description in this chapter includes the type and approach of the research, implementation procedures, tools and materials used, data collection techniques, and data analysis methods applied in the research process.

2.1 Dataset

Tabel 2.1 Dataset

No	age of the month	gender	height (cm)	nutritional status
1	0	man	44.5919732943438	*Stunted*
2	0	man	56.7052033668847	height
3	0	man	46.8633575967919	ordinary
4	0	man	47.5080256315438	normal
5	0	man	42.7434938911793	*Severely Stunted*
	.....	.....	.....	.....
120996	60	Pemale	100.6	ordinary
120997	60	Pemale	98.3	*Stunted*
120998	60	Pemale	121.3	ordinary
120999	60	Pemale	112.2	ordinary
121000	60	pemale	109.8	ordinary

2.2 Preprocessing Data

The data will go through a pre-processing stage which includes handling empty values,

normalization, and other processing steps, so that the data is ready to be used optimally in the classification process[5].

1. Celeaning Data

Data cleansing is the process of removing common errors in data, such as missing values, duplicate data, or inconsistent formats. This step aims to ensure the data is clean and ready for use in the analysis process[6].

2. Normalization

Data normalization is the process of adjusting the values of various variables in a dataset to have a uniform range and be comparable to each other. The main purpose of normalization is to convert the values of numeric attributes to a simpler scale, usually between 0 and 1, so that each feature has a balanced influence in the analysis or modeling[7].

3. Synthetic Minority Oversampling Technique – Edited Nearest Neighbors (SMOTEENN)

Synthetic Minority Oversampling – Edited Nearest Neighbors, or SMOTEENN, is a data resampling technique that combines the SMOTE and ENN methods. SMOTE adds synthetic data to the minority class to improve its distribution. ENN classifies data based on nearest neighbors or missing data in noise. By eliminating irrelevant data, SMOTEENN improves dataset quality and increases the number of minority data points. This technique is typically applied to classification problems with imbalanced data[8].

4. Spliting Data

Data splitting is the process of dividing a dataset into two main parts. 80% of the training data is used. 20% of the testing data is used. The model learns data patterns and structures through the training data. Testing is performed on previously unknown data. This process aims to measure the model's ability to classify accurately. The 80% and 20% splits provide sufficient time for training and evaluation. The K-Nearest Neighbors and Naive Bayes algorithms are applied to the training data and tested with the testing data to assess the performance of stunting risk classification in toddlers[9].

2.3 Algoritma K-Nearest Neighbors

The KNN algorithm is a simple yet effective classification method for data processing. This algorithm works by classifying data based on its patterns and similarity to other data. In practice, KNN is constructed using several commonly used k values, then selecting the k value that provides the best performance[10]. This study used three distance metrics: Euclidean Distance, Manhattan Distance, and Minkowski Distance. A comparison of these three metrics was conducted to determine which distance metric performed best in classifying stunting risk in toddlers[11]. The K-NN algorithm formula is as follows:

1. Euclidean Distance Calculate the distance between the testing data and all training data. This calculation applies the Euclidean Distance formula in Formula (1). Where pi is the training data, qi is the testing data, i indicates the variable data, and n is the number of data[12].

$$euc = \sqrt{\sum_{i=1}^{n}(pi - qi)^2} \qquad (1)$$

Description:

Pi = sample data/training data

qi = test data/testing data

i = data variable/feature

n = data dimension (number of features being compared)

2. Manhattan Distance Distance where the Manhattan distance is used to calculate the distance between two points in n-dimensional space. The Manhattan Distance formula is in formula (3). where the pi value indicates the value of the i-th feature of the

training data. The qi value indicates the value of the i-th feature of the test data. The I value is the feature index. The n value is the number of features being compared, while the d value indicates the distance between two data points.

$$d = \sum_{i=1}^{n} | \, pi - qi \, | \qquad (2)$$

Description:
Pi = sample data/training data
qi = test data/testing data
i = data variable/feature
n = data dimension (number of features being compared)
d = Manhattan distance between two data sets

3. Minkowski distance is a general metric used to measure the distance between two points in a vector space with a certain number of dimensions. This metric is a generalization of the Euclidean and Manhattan distances, which are controlled by the parameter p. Minkowski's formula in formula (4) Where the D value (A, B) indicates the distance between two data points. The x_i value indicates the value of the i-th feature b from point A. The y_i value indicates the value of the i-th feature from point B. The i value is the feature index. The n value is the number of features being compared. The p value is an order parameter that determines the type of distance with p $\geqslant$ 1. The calculation is done by subtracting the feature value of point B from the feature value of point A.

$$D(A,B) = \left(\sum_{i=1}^{n} |x_i - y_i \, | \right)^{p} \Bigg)^{\frac{1}{p}} \qquad (3)$$

Description:
D (A, B) = Minkowski distance between two points A and B
n = number of dimensions or features
x_i, y_i = value between the i-th feature of data A and B
p = order parameter (p $\geq$ 1) that determines the type of distance

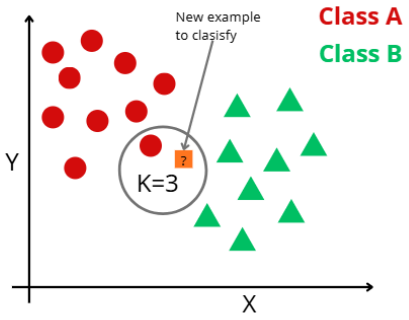

Figure 2.1 Working Principle of the K-Nearest Neighbors (KNN) Algorithm

2.4 Algoritma Naïve Bayes

The Naïve Bayes method is one of the simplest yet most effective algorithms for probability-based classification. This algorithm relies on a statistical approach based on Bayes' Theorem. The main goal of this method is to group data into specific categories based on existing features[13]. Formula (4) is Bayes' theorem, which is used to calculate the probability of a class based on observed data. The formula is:

$$P(C|X) = \frac{P(\mathrm{C}).\ P(\mathrm{X}|\mathrm{C})}{P(X)} \qquad (5)$$

Description:
P(C|X) = probability of class C given feature X (posterior probability)
P(C) = initial probability of class C (prior probability)
P(X|P) = probability of feature X given class C (likelihood)
P((X) = total probability of feature X (evidence)

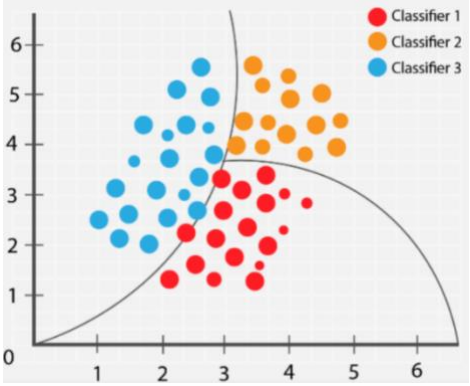

Figure 2.2 Working Principle of the Naïve Bayes Algorithm

2.5 Grid Search

Grid search is a parameter optimization technique used to determine the best combination of hyperparameters for a given machine learning algorithm. The search process is performed exhaustively (an exhaustive search) for each predetermined parameter combination based on the number of steps, lower bounds, and upper bounds. Each combination is systematically evaluated using study data to determine the optimal parameters that yield the best model performance[14].

2.6 Evaluasi Model

Model evaluation is the process of assessing the performance of an algorithm in classifying or predicting based on available data. In this study, the evaluation aimed to measure the algorithm's effectiveness in predicting the risk of stunting in toddlers. The evaluation process was conducted using various measurement metrics, such as accuracy, precision, recall, F1-score, and computation time[15]. In addition to using the Confusion Matrix, the performance of the classification model was also evaluated using the ROC (Receiver Operating Characteristic) curve and AUC (Area Under the Curve) value. The ROC Curve is a graph that shows the relationship between the True Positive Rate and False Positive Rate at various threshold values[16]. The AUC score is the area under the ROC curve. AUC values range from zero to one. The higher the AUC, the better the model's classification ability. An AUC value close to one indicates high performance, while a value close to 0.5 indicates low performance[17].

Evaluation in classification research aims to measure how well a model accurately predicts the target class. Evaluation is typically conducted using metrics such as accuracy, precision, recall, and F1-score, which are calculated based on prediction results against test data. The ROC curve is a graph showing the relationship between the True Positive Rate and False Positive Rate at various threshold values. This graph is used to evaluate the model's ability to distinguish between positive and negative classes. The True Positive Rate measures the proportion of positive data successfully recognized by the model. The False Positive Rate measures the proportion of negative data incorrectly classified as positive. The AUC score indicates the area under the ROC curve. AUC values range from zero to one. The higher the AUC value, the better the model's classification ability.

RESULTS AND DISCUSSION

The dataset used in this study is public stunting data taken from the Kaggle site https://www.kaggle.com/*datasets*/rendiputra/stunting-balita-detection-121k-rows *on June 23 - December 2025.*

3.1 ***Data Preprocessing Results***

1. Deleting Duplicate Data

The data size was reduced to 4,066, with 4 rows. This cleaned dataset was then ready for use in the exploration and modeling phase of stunting risk classification.

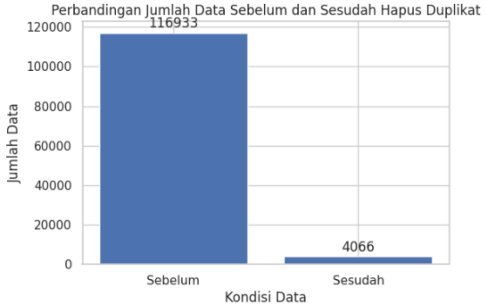

Figure 3.1 Comparison of the Number of Data and After Deleting Duplicates

2. Data Balancing

Checking the data distribution for the target variable, Nutritional Status. Results before handling imbalanced data. The distribution indicates an imbalanced data set, with class 0 having significantly more data than the other classes. This condition can lead to a model bias toward the majority class.

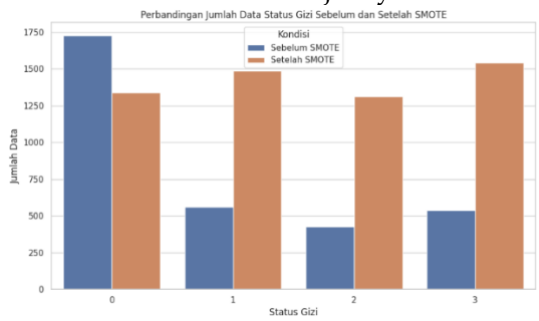

Figure 3.1 Comparison of the Number of Data Before and After SEMOTE

3.2 Classification Results

Table 1 Comparison Table of All Models

Model	Precision	Recall	F1-score	Accuracy
KNN before Tuning	0.99	0.83	0.90	87%
Naïve Bayes before Tuning	0.74	0.62	0.62	52%
KNN after Tuning	0.94	0.84	0.84	85%
Naïve Bayes after Tuning	0.74	0.62	0.68	52%

Based on Table 1, the comparison of all models, the KNN model before Tuning has an accuracy of 87%, Precision 99%, Recall 83%, F1-score 90%. Best performance. Optimal default parameters. KNN after Tuning has an accuracy of 85%, Precision 94%, Recall 84%, F1-score 89%. There is a decrease. Tuning does not improve performance. Naïve Bayes before Tuning has an accuracy of 52%, Precision 74%, Recall 62%, F1-score 68%. Naïve Bayes after Tuning has the same value. No change. Limited parameters. The assumption of feature independence is not met. KNN before Tuning is the best model.

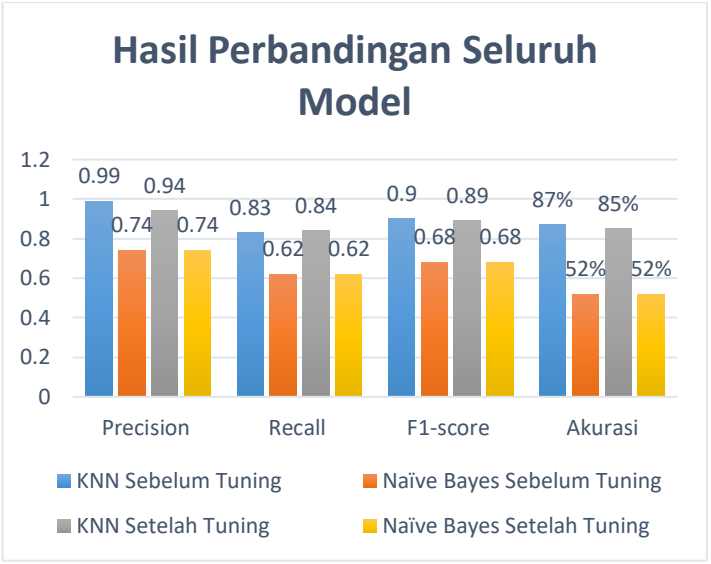

Figure 3.2 Comparison of All Models

Based on Figure 3.2, it shows a comparison of the performance of the Naïve Bayes and K-Nearest Neighbors algorithms before and after the Tuning process based on Precision, Recall, F1-score, and accuracy values. The KNN model obtained the best results in almost all metrics with Precision values approaching 0.99, F1-score around 0.90, Recall around 0.83, and accuracy around 0.87. KNN after Tuning still showed high performance although some values experienced a slight decrease. The Naïve Bayes algorithm produced lower performance than KNN, but the Tuning process was able to improve the evaluation results on each metric. Overall, KNN was the model with the best classification level in the test.

3.3 *Comparison of ROC-AUC Results of KNN and Naïve Bayes*

Table 2 ROC-AUC Comparison Results

Model	Class 0	Class 1	Class 2	Class 3	average value
KNN	0.924	0.935	0.896	0.973	0.9317
Naïve Bayes	0.699	0.762	0.721	0.822	0.7508

Based on table 2, KNN has an ROC-AUC value of 0.924 in class 0, 0.935 in class 1, 0.896 in class 2, 0.973 in class 3, with an average of 0.9317. The classification ability is very good. Naïve Bayes has a value of 0.699 in class 0, 0.762 in class 1, 0.721 in class 2, 0.822 in class 3, with an average of 0.7508. The classification ability is sufficient. KNN shows superior performance compared to Naïve Bayes.

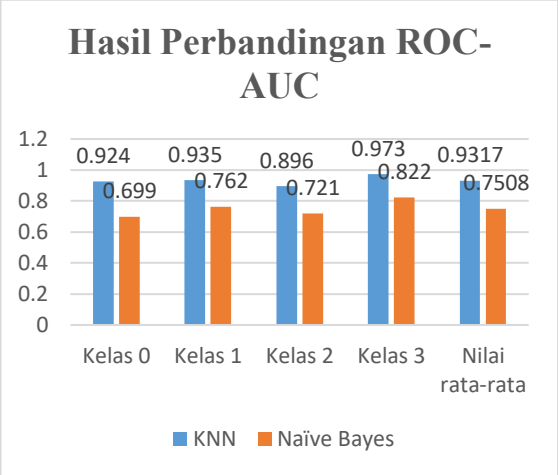

Figure 3.3 ROC-AUC Comparison Results

Based on Figure 3.3 ROC-AUC comparison, the KNN model shows better classification performance than Naïve Bayes in all classes. The KNN AUC value in class 0 is 0.924, class 1 is 0.935, class 2 is 0.896, class 3 is 0.973. The Naïve Bayes AUC value in class 0 is 0.699, class 1 is 0.762, class 2 is 0.721, class 3 is 0.822. The average ROC-AUC value of KNN is 0.932, the average value of Naïve Bayes is 0.751. These results indicate that KNN classification ability is better in distinguishing each class of data..

3.4 Discussion

KNN has superior performance because it can group data based on the proximity between features. The stunting dataset has data patterns that are more suitable for distance-based approaches than the probabilistic Naïve Bayes approach. Furthermore, the feature independence assumption in Naïve Bayes does not align well with the characteristics of stunting data, thus affecting classification performance.

4 CONCLUSION

Based on the research results, the K-Nearest Neighbors (KNN) algorithm performed better than Naïve Bayes in classifying stunting risk in toddlers. The KNN model achieved 87% accuracy, a 90% F1-score, and an average ROC-AUC of 0.9317. Meanwhile,

Naïve Bayes only achieved 52% accuracy with an ROC-AUC of 0.7508. The results showed that KNN is more effective for machine learning-based stunting risk classification.

THANK-YOU NOTE

The author would like to thank Qamarul Huda Badaruddin Bagu University, the supervising lecturer, family, and all parties who have provided support in this research process.

BIBLIOGRAPHY

[1] U. Malikussaleh, "SENASTIKA Universitas Malikussaleh," pp. 1–6, 2024.

[2] V. No, N. R. Febriyanti, and A. D. Hartanto, "Edumatic : Jurnal Pendidikan Informatika Analisis Perbandingan Algoritma SVM , Random Forest dan Logistic Regression untuk Prediksi Stunting Balita," vol. 9, no. 1, pp. 149–158, 2025, doi: 10.29408/edumatic.v9i1.29407.

[3] R. H. Prayetno, R. D. Purba, K. Wirawan, and K. Sweet, "Purchasing Prediction Using Machine Learning Algorithms for Optimizing Inventory Management," vol. 11, no. 1, pp. 150–168, 2025, doi: https://doi.org/10.37012/jtik.v11i1.2522.

[4] D. Fitria, R. R. Suryono, C. Science, and U. T. Indonesia, "CLASSIFICATION OF PUBLIC SENTIMENT TOWARDS STUNTING PREVENTION PROGRAM USING NAÏVE BAYES AND SUPPORT VECTOR MACHINE ON X APPLICATION KLASIFIKASI SENTIMEN PUBLIK TERHADAP PROGRAM PENCEGAHAN STUNTING MENGGUNAKAN NAÏVE BAYES DAN SUPPORT VECTOR MACHINE," vol. 5, no. 6, pp. 1839–1847, 2024.

[5] D. Nasien *et al.*, "Perbandingan Implementasi Machine Learning Menggunakan Metode KNN , Naive Bayes , Dan Logistik Regression Untuk Mengklasifikasi Penyakit Diabetes," vol. 4, no. 1, 2024.

[6] U. R. Gurning, S. F. Octavia, D. R. Andriyani, N. Nurainun, and I. Permana, "Prediksi Risiko Stunting pada Keluarga Menggunakan Naïve Bayes Classifier dan Chi-Square," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 1, pp. 172–180, 2024, doi: 10.57152/malcom.v4i1.1074.

[7] A. D. Putri, F. Sholekhah, and E. Dadynata, "The Application of C4 . 5 Decision Tree Algorithm for Predicting the Survival Rate of Thyroid Cancer Patients Penerapan Algoritma Decesion Tree C4 . 5 untuk Memprediksi Tingkat Kelangsungan Hidup Pasien Kanker Tiroid," vol. 4, no. October, pp. 1485–1495, 2024.

[8] B. Alareeni and I. Elgedawy, *Indonesia's Path to Sustainability: Exploring the Intersections of Ecological Footprint, Technology, Global Trade, Financial Development and Renewable Energy Asif*, vol. 1. 2024.

[9] L. Shen, Y. Sun, Z. Yu, L. Ding, X. Tian, and D. Tao, "On Efficient Training of Large-Scale Deep Learning Models," *ACM Comput. Surv.*, vol. 57, no. 3, pp. 1–60, 2024, doi: 10.1145/3700439.

[10] E. Sahelvi, P. Cikita, and R. M. Sapitri, "Comparison of K-Nearest Neighbors and Random Forest Algorithms for Recommendations for a Healthy Lifestyle in Prevent Heart Disease Perbandingan Algoritma K-Nearest Neighbors dan Random Forest untuk Rekomendasi Gaya Hidup Sehat dalam Mencegah Penyakit Jan," vol. 5, no. July, pp. 830–840, 2025.

[11] S. Rahmi, A. Rachman, and R. Puspitasari, "K-Nearest Neighbor dengan Jarak Euclidean , Manhattan , dan Minkowski pada klasifikasi sampah," vol. 2, no. 3, pp. 294–301, 2025.

[12] P. P. Allorerung, A. Erna, M. Bagussahrir, and S. Alam, "Analisis Performa Normalisasi Data untuk Klasifikasi K-Nearest Neighbor pada Dataset Penyakit," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 9, no. 3, pp. 178–191, 2024, doi: 10.14421/jiska.2024.9.3.178-191.

[13] M. F. A. Ad-duali, D. P. Adinata, T. Informatika, F. Teknik, U. Nusantara, and P. Kediri, "Komparasi Algoritma Naive Bayes dan Random Forest untuk Identifikasi Kata Berpotensi Spam," vol. 5, pp. 439–448, 2026.

[14] W. Nugraha and A. Sasongko, "Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search Hyperparameter Tuning on Classification Algorithm with Grid Search," *Sist. J. Sist. Inf.*, vol. 11, no. 2, pp. 2540–9719, 2022, [Online]. Available: https://doi.org/10.32520/stmsi.v11i2.1750

[15] R. Merdiansah and A. A. Ridha, "Analisis Sentimen Pengguna X Indonesia Terkait Kendaraan Listrik Menggunakan IndoBERT," vol. 7, pp. 221–228, 2024.

[16] K. Kristiawan and A. Widjaja, "Perbandingan Algoritma Machine Learning dalam Menilai Sebuah Lokasi Toko Ritel," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 35–46, 2021, doi: 10.28932/jutisi.v7i1.3182.

[17] D. Chicco and G. Jurman, "The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification," *BioData Min.*, vol. 16, no. 1, pp. 1–23, 2023, doi: 10.1186/s13040-023-00322-4.