Advance Sustainable Science, Engineering and Technology (ASSET)
Vol. 6, No.1, January 2024, pp. 02401012-01 ~ 02401012-08
ISSN: 2715-4211 DOI: https://doi.org/10.26877/asset.v6i1.17685

Stroke Classification Comparison with KNN through Standardization and Normalization Techniques

Muhammad Raihan Firmansyah*, Yani Parti Astuti

Faculty of computer Science, Universitas Dian Nuswantoro, Imam Bonjol No.207 Semarang, Central Java , 50131, Indonesia

*111202013184@mhs.dinus.ac.id

Abstract. This study explores the impact of z-score standardization and min-max normalization on K-Nearest Neighbors (KNN) classification for strokes. Focused on managing diverse scales in health attributes within the stroke dataset, the research aims to improve classification model accuracy and reliability. Preprocessing involves z-score standardization, min-max normalization, and no data scaling. The KNN model is trained and evaluated using various methods. Results reveal comparable performance between z-score standardization and min-max normalization, with slight variations across data split ratios. Demonstrating the importance of data scaling, both z-score and min-max achieve 95.07% accuracy. Notably, normalization averages a higher accuracy (94.25%) than standardization (94.21%), highlighting the critical role of data scaling for robust machine learning performance and informed health decisions.

Keywords: KNN, Z-Score Standardization, Min Max Normalization, Stroke Classification, Data Scaling

(Received 2023-12-10, Accepted 2023-12-22, Available Online by 2024-01-02)

1. Introduction

The implementation of AI and ML in the medical field holds revolutionary potential, enhancing the accuracy of diagnosis, treatment planning, and patient monitoring [1]. Its ability to process vast medical data enables the development of innovative diagnostic tools and treatment plans that can improve patient outcomes, identify individual risks, and personalize treatment plans. [2].In the context of classifying stroke datasets, the process of data standardization and normalization demonstrates significant urgency. Standardization and normalization are crucial steps in data preprocessing that play a major role in improving the performance of classification models. Through standardization, attributes in the dataset are transformed into a uniform scale, ensuring that each variable has an equal impact in the classification process [3]. Meanwhile, normalization adjusts attribute values into a more controlled range, minimizing the impact of outliers and improving the distribution of data [4].

In this study, the focus on data standardization and normalization aims to enhance the accuracy and reliability of classification models, particularly machine learning algorithms like KNN, in identifying stroke risk patterns. Both processes are crucial because health attributes in the stroke dataset exhibit diverse scales. By avoiding the dominance of large-scale attributes, standardization and normalization ensure a balanced contribution of each attribute, enabling the model to provide more accurate and consistent predictions. Through the analysis of its positive impact, this research highlights the vital role of standardization and normalization in improving prediction accuracy, providing a reliable foundation for result interpretation, and supporting more precise health decision-making.

Based on previous research related to the topic of this study, which is the Evaluation of Stroke Classification with KNN through Standardization and Normalization Techniques, there are several findings. The results obtained from the study titled 'Analysis of the Influence of Data Scaling on the Performance of Machine Learning Algorithms for Plant Identification' [5] indicate that the difference in accuracy and recall between standardization and normalization is not significantly different for the KNN machine learning algorithm. For standardization, the accuracy obtained is 76%, while the normalization result is slightly higher, with an accuracy of 77.33%.

Similar research on the comparison of data scaling, specifically standardization and normalization, is also found in a journal regarding the comparison of data normalization for wine classification using the K-NN algorithm [6]. The accuracy results from Min-Max normalization are 57.41%, while for Z-score standardization, it is 56.40%.

A similar journal discussing the difference in the Evaluation of Stroke Classification with KNN through Standardization and Normalization techniques is also found in the following journal. This journal compares Min-Max normalization with Z-Score standardization to test the accuracy of Breast Cancer types using the KNN algorithm [7]. The accuracy results obtained are 97% for standardization and 98% for Min-Max normalization.

2. Methods

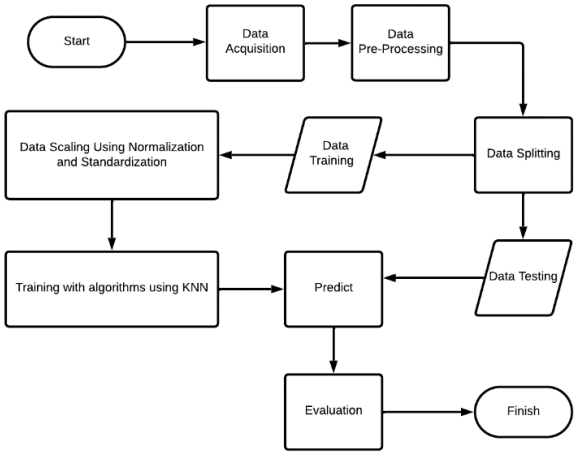

Figure 1. Research Methodology

This study focuses on comparing the performance of the K-Nearest Neighbors (KNN) model in stroke classification. The process begins with data acquisition, followed by data preprocessing. Subsequently, the dataset is divided into training and testing data. The training data undergoes three different preprocessing conditions: Z-score standardization, Min-Max normalization, and no preprocessing (raw data). The KNN model is then trained on the preprocessed training data to identify patterns in stroke detection. Its performance is evaluated on the testing data using standard metrics such as accuracy, precision, and recall. The research aims to provide a comprehensive understanding of the impact of data preprocessing on model performance in the context of stroke classification using KNN.

2.1. Data Acquisition

The dataset used in this study, sourced from Kaggle.com ('stroke prediction'), contains 5110 rows and eleven columns. The dataset for stroke classification encompasses several attributes. The 'Id' serves as a unique identifier for each record. Gender information is provided in the 'Gender' column, categorized as 'Male,' 'Female,' or 'Other.' The 'Age' column denotes the patients' ages, while 'Hypertension' and 'Heart_Disease' are binary attributes indicating specific health conditions. Marital status is detailed in the 'Ever_married' column, and employment type is specified in 'Work_type.' The 'Residence_type'

column distinguishes between rural and urban residences. Health metrics include 'Avg_glucose' for blood sugar levels and 'Bmi' for Body Mass Index. Smoking habits are described in the 'Smoking_status' column, and the 'Stroke' column indicates the stroke status (label).

2.2. Data Pre-processing

In the data preprocessing process for the stroke dataset study, crucial steps are undertaken to ensure the quality of the dataset used in stroke classification. Firstly, irrelevant attributes, such as identification numbers, are removed to simplify the dataset and focus on attributes that have a significant impact on stroke risk, such as age, blood pressure, and smoking history. Next, handling missing values becomes a primary focus, where missing values are deleted from the dataset with the consideration that their presence could affect the quality and integrity of stroke classification analysis. Finally, label encoding is performed as a technique to transform categorical data into numeric form. This allows machine learning algorithms to more effectively understand and analyze categorical variables in the dataset, ensuring accurate representation in the model. By using label encoding, data analysis and modeling become more efficient, preparing the dataset optimally for the training and prediction processes of the model.

2.3. Outlier Handling

Identifying and addressing outliers in data is crucial in the context of machine learning and predictive modeling. This action helps reduce noise, detect erroneous records, and prevent overfitting, providing insights into patterns and trends in the data. In model development, handling outliers is necessary to enhance reliability and accuracy. This study utilizes the Interquartile Range (IQR) method to identify outliers in relevant attributes. The initial steps involve calculating the first quartile (q1) and third quartile (q3), which are used to compute the IQR as the difference between q3 and q1 [8].

$$IQR = q3 - q1 \tag{1}$$

Outliers are identified by calculating the lower bound as q1 minus 1.5 times the IQR and the upper bound as q3 plus 1.5 times the IQR [9].

Outlier handling is performed on the numerical attributes 'bmi' and 'avg_glucose_level' using the interquartile range (IQR) method. Visualization with boxplots is used for outlier identification, and the outliers are removed from the dataset to ensure data integrity and quality. This process enhances the reliability of analysis and modeling by eliminating potentially disruptive data. After outlier handling, the dataset is reduced from 4909 to 4260 rows.

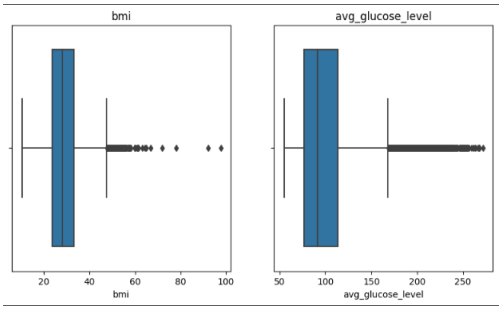

Figure 2. Before Outliers Handling

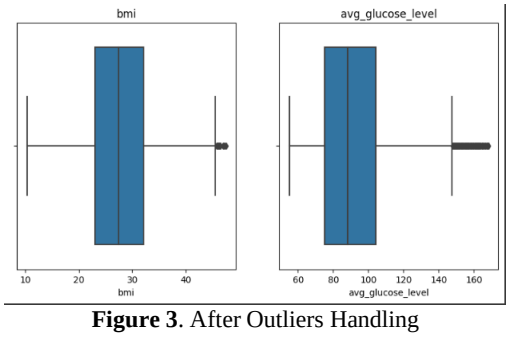

Figure 3. After Outliers Handling

2.4. Data Spliting

In the preprocessing stage, the dataset is divided into training and testing subsets using five variations of ratios, namely 90%:10%, 80%:20%, and 70%:30%. This process aims to objectively test the stroke classification model on data not used during training, preventing overfitting, and ensuring the model's generalizability. The division is performed randomly for objectivity and data representation in both training and model validation.

Table 1. Rasio Split Train Data and Test Data

Rasio Split Train Data and Test Data	Jumlah Data Training	Jumlah Data Testing
90:10	3834	426
80:20	3408	852
70:30	2982	1278

2.5. Imbalance data Handling

The research employs the SMOTE technique to address imbalanced data, a method effective in classification tasks. SMOTE creates synthetic samples from the minority class by selecting reference points and generating synthetic samples through a formula incorporating a parameter δ, allowing flexibility in adjusting synthetic results to the dataset's characteristics [10].

$$\text{Xsyn} = \text{Xi} + (\text{Xknn} - \text{Xi}) * \delta \quad (2)$$

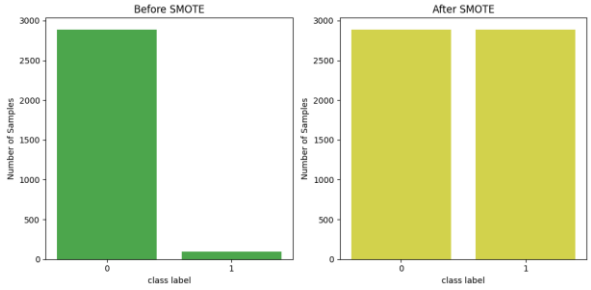

Figure 4. comparison of the amount of data before and after SMOTE Oversampling

The implementation of SMOTE increases the number of stroke samples from 3408 to 6610 in the training data.

2.6. Min Max Normalization

MinMax normalization is a process to transform the range of data values into between 0 and 1[6]. The primary objective of this normalization is to ensure that all data attributes have a uniform scale, avoiding the dominance of attributes with large scales over others. By rescaling the data value range, the interpretation of analysis results becomes more consistent, and the performance of classification models can be enhanced [11], especially in the task of stroke risk classification. The MinMax normalization process is applied to each data value (*xi*) using the formula:

$$X' = \frac{xi - \min(x)}{max(x) - \min(x)} \quad (3)$$

Where *x'* is the normalized result value of the original observation value or initial value of a data point (*xi)*, while min(*x*) and max(*x*) represent the minimum and maximum values across all the data.

2.7. Z-Score Standardization

Z-Score standardization is a standardization technique that uses the mean and standard deviation of each feature attribute to transform the scale of data values. This standardization procedure is applied to reduce the impact of outliers and ensure that each attribute has a consistent scale. The main goal of Z-Score standardization is to enhance the stability of analysis results and improve the consistency of data interpretation [12].

The Z-Score standardization process is applied to each data value (*xi*)using the formula:

$$Z = \frac{X - \mu}{\sigma} \quad (4)$$

The z-score formula standardizes data by transforming its distribution to a mean of 0 and a standard deviation of 1, facilitating comparisons. It involves subtracting the observation value from the population mean and dividing by the population standard deviation. The resulting Z score indicates the value's distance from the mean in standard deviation units, crucial for statistical analysis and machine learning.

2.8. KNN

The K-Nearest Neighbors (KNN) algorithm falls into the category of supervised learning, classifying data based on their proximity or distance to other data points. In the implementation of KNN, the Euclidean distance formula is commonly used to measure the proximity between training and test data points [13].

$$di = \sqrt{\sum_{i=1}^{n}(xi - yi)^2} \quad (5)$$

The K-Nearest Neighbors (KNN) algorithm employs the Euclidean formula, where 'di' represents the distance between training and test data, 'xi' is the training data, 'yi' is the test data, 'n' is the data dimension, and 'i' is the data variable. KNN's operation involves initializing 'K', calculating distances, sorting distances, selecting the nearest 'K' neighbors, applying majority rule, and predicting the category. This method focuses on the relationships among data points in feature space, with the value of 'K' determining the number of nearest neighbors considered, a critical factor in stroke classification[14]. For model development, cross-validation is used to find the optimal 'K'. The model is trained and evaluated on subsets, with accuracy recorded as an evaluation metric. The optimal 'K' is selected based on the average evaluation metric across all cross-validation iterations, ensuring the best choice for stroke classification.

2.9. Evaluate metrics

In this study, the evaluation of the classification model heavily relies on the Confusion Matrix, a powerful tool that summarizes model performance. The matrix's four main components—True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN)—facilitate the measurement of accuracy, precision, recall, and F1-score. These metrics offer comprehensive insights into the model's ability to distinguish between different classes. The confusion matrix table forms the foundation for calculating accuracy, precision, recall, and F1-score, essential in evaluating the model's effectiveness and accuracy in class prediction. [15].

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (6)$$

$$\text{Presicion} = \frac{TP}{(TP+FP)} \quad (7)$$

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad (8)$$

$$\text{F1-Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

3. Results and Discussion

3.1. Preprocessing

Several steps are taken in preprocessing the stroke dataset as follows. First, remove irrelevant attributes such as identification numbers. Second, address missing values by removing them from the dataset. Finally, apply label encoding to convert categorical data into numerical form. This process enhances the efficiency of data analysis and modeling, preparing the dataset optimally for model training and prediction.

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

Figure 5. Dataset Before Pre-processing

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
2	1	80.0	0	1	1	0	1	105.92	32.5	1	1
6	1	74.0	1	1	1	0	1	70.09	27.4	1	1
7	0	69.0	0	0	0	0	0	94.39	22.8	1	1
9	0	78.0	0	0	1	0	0	58.57	24.2	3	1
10	0	81.0	1	0	1	0	1	80.43	29.7	1	1

Figure 6. dataset after Pre-processing

3.2. Data Scaling

Here are the results of the data that have undergone Z-Score standardization and Min-Max normalization.

```
array([[1.        , 0.74340176, 0.        , ..., 0.23068061, 0.88978495,
        0.66666667],
       [1.        , 0.74340176, 0.        , ..., 0.75983694, 0.63709677,
        0.66666667],
       [1.        , 0.18132942, 0.        , ..., 0.0871145 , 0.3844086 ,
        1.        ],
       ...,
       [1.        , 0.78040887, 0.        , ..., 0.16761628, 0.49438401,
        0.98566909],
       [1.        , 0.93943462, 0.        , ..., 0.72027028, 0.43853531,
        0.        ],
       [0.        , 0.70469002, 0.        , ..., 0.46023389, 0.75668409,
        0.01401407]])
```

Figure 7. Data after normalization

```
array([[ 1.29173623,  0.32715671, -0.45883538, ..., -0.4815554 ,
         2.57731674,  0.54535679],
       [ 1.29173623,  0.32715671, -0.45883538, ...,  2.17202351,
         0.987685  ,  0.54535679],
       [ 1.29173623, -1.6985405 , -0.45883538, ..., -1.20150144,
        -0.60194673,  1.55768981],
       ...,
       [ 1.29173623,  0.46052958, -0.45883538, ..., -0.79780632,
         0.08989569,  1.51416686],
       [ 1.29173623,  1.03365513, -0.45883538, ...,  1.97360718,
        -0.26144194, -1.47930924],
       [-0.92286263,  0.18764039, -0.45883538, ...,  0.66959354,
         1.7399948 , -1.43674851]])
```

Figure 8. Data after standardization

3.3. Modelling

Searching for the best K value for each ratio with accuracy testing experiments for K values ranging from k=1 to k=8. The following is the experimental table.

Table 2. Best k value in each ratio.

Rasio Split	**Akurasi**							
	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8
90% and 10%	92.71%	92.72%	90.43%	90.99%	89.56%	90.33%	88.97%	89.24%
80% and 20%	93.50%	93.64%	91.54%	91.96%	90.51%	90.98%	89.83%	90.25%
70% and 30%	93.61%	93.73%	91.67%	91.92%	90.96%	91.19%	90.19%	90.72%

In the results of searching for the optimal K value using cross-validation and performance curves, based on the table above utilizing three ratios (90% and 10%, 80% and 20%, 70% and 30%), it is found that the best K value is at k=2.

3.3.1. KNN using Standardization k = 2

Here is the KNN model evaluation table for Z-Score standardization in each data splitting ratio.

Table 3. Classification report for KNN using Standardization

Rasio Split Train Data and Test Data	Accuracy	Presisi	Recall
90% and 10%	95.07%	98.3%	96.65%
80% and 20%	93.9%	97.44%	96.26%
70% and 30%	93.66%	96.76%	96.68%

3.3.2. KNN using Normalization k =2

Here is the KNN model evaluation table for Min Max Normalization in each data splitting ratio.

Table 4. Classification report for KNN using Normalization

Rasio Split Train Data and Test Data	Accuracy	Presisi	Recall
90% and 10%	95.07%	98.3%	96.65%
80% and 20%	93.54%	97.2%	96.14%
70% and 30%	94.13%	97.33%	96.62%

3.3.3. KNN without Standardization and Normalization

Here is the KNN model evaluation table for data without Z-Score standardization and Min-Max normalization in each data splitting ratio.

Table 5. Classification report for KNN without Data Scaling

Rasio Split Train Data and Test Data	Accuracy	Presisi	Recall
90% and 10%	88.73%	90.75%	97.39%
80% and 20%	88.85%	91.33%	96.9%
70% and 30%	87.8%	89.87%	97.28%

4. Conclusion

In the study during the data preprocessing stage, two different conditions were applied. The first condition involved Z-Score standardization of the stroke classification dataset, and the second condition involved Min-Max normalization. Subsequently, training was conducted using the KNN machine learning algorithm. Based on the results and discussions, several conclusions can be drawn:

a. The highest accuracy for both Z-Score standardization and Min-Max normalization is 95.07% for the 90:10 data splitting ratio.
b. The average accuracy of Z-Score standardization is lower at 94.21% compared to the average accuracy of normalization at 94.25%.
c. The optimal K value after cross-validation is found to be k=2 after comparing for each ratio.
d. The comparison results between Z-Score standardization and Min-Max normalization with data without Z-Score standardization and Min-Max normalization reveal differences. This underscores the importance of performing data scaling on the dataset to achieve high machine learning performance.

References

[1] F. D. Telaumbanua, P. Hulu, T. Z. Nadeak, R. R. Lumbantong, and A. Dharma, "Penggunaan Machine Learning Di Bidang Kesehatan," *JURNAL TEKNOLOGI DAN ILMU KOMPUTER PRIMA (JUTIKOMP)*, vol. 2, no. 2, Art. no. 2, 2019, doi: 10.34012/jutikomp.v2i2.657.

[2] A. S. Fahmy *et al.*, "An Explainable Machine Learning Approach Reveals Prognostic Significance of Right Ventricular Dysfunction in Nonischemic Cardiomyopathy," *JACC: Cardiovascular Imaging*, vol. 15, no. 5, pp. 766–779, May 2022, doi: 10.1016/j.jcmg.2021.11.029.

[3] Mohammed Z. Al-Faiz, Ali A. Ibrahim, and Sarmad M. Hadi, "The effect of Z-Score standardization (normalization) on binary input due the speed of learning in back-propagation

neural network," *Iraqi Journal of Information and Communications Technology(IJICT)*, vol. 1, no. 3, pp. 42–48, Feb. 2019, doi: 10.31987/IJICT.1.3.41.

[4] Lei Huang, ie Qin, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao, "Normalization Techniques in Training DNNs: Methodology, Analysis and Application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 8, pp. 10173–10196, Aug. 2023, doi: 10.1109/tpami.2023.3250241.

[5] gus Ambarwari, Qadhli Jafar Adrian, and Yeni Herdiyeni, "Analysis of the Effect of Data Scaling on the Performance of the Machine Learning Algorithm for Plant Identification | Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)," *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 1, pp. 117–122, 2020, doi: https://doi.org/10.29207/resti.v4i1.1517.

[6] D. A. Nasution, H. H. Khotimah, and N. Chamidah, "Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN," *CESS (Journal of Computer Engineering, System and Science)*, vol. 4, no. 1, Art. no. 1, Jan. 2019, doi: 10.24114/cess.v4i1.11458.

[7] Henderi, Tri Wahyuningsih, and Efana Rahwanto, "Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer," *International Journal of Informatics and Information System*, vol. 4, no. 1, pp. 13–20, Mar. 2021, doi: 10.47738/IJIIS.V4I1.73.

[8] MADISON WENZLICK, OSMAN MAMUN, RAM DEVANATHAN, KELLY ROSE, and JEFFREY HAWK, "Assessment of Outliers in Alloy Datasets Using Unsupervised Techniques," *JOM*, vol. 74, no. 7, pp. 2846–2859, May 2022, doi: 10.1007/s11837-022-05204-4.

[9] Amerah Alabrah, "An Improved CCF Detector to Handle the Problem of Class Imbalance with Outlier Normalization Using IQR Method," in *Sensors*, Apr. 2023, pp. 4406–4406. doi: 10.3390/s23094406.

[10] Shobha Aswal, Neelu Jyothi Ahuja, and Ritika Mehra, "Feature Selection Method Based on Honeybee-SMOTE for Medical Data Classification," *Informatica*, vol. 46, no. 9, pp. 111–118, Feb. 2023, doi: 10.31449/inf.v46i9.4098.

[11] Gde Agung Brahmana Suryanegara, Adiwijaya, and Mahendra Dwifebri Purbolaksono, "Peningkatan Hasil Klasifikasi pada Algoritma Random Forest untuk Deteksi Pasien Penderita Diabetes Menggunakan Metode Normalisasi | Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)," *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 1, pp. 114–122, 2021, doi: https://doi.org/10.29207/resti.v5i1.2880.

[12] I. Permana and F. N. S. Salisah, "Pengaruh Normalisasi Data Terhadap Performa Hasil Klasifikasi Algoritma Backpropagation: The Effect of Data Normalization on the Performance of the Classification Results of the Backpropagation Algorithm," *Indonesian Journal of Informatic Research and Software Engineering (IJIRSE)*, vol. 2, no. 1, Art. no. 1, Mar. 2022, doi: 10.57152/ijirse.v2i1.311.

[13] S. K. P. Loka and A. Marsal, "Perbandingan Algoritma K-Nearest Neighbor dan Naïve Bayes Classifier untuk Klasifikasi Status Gizi Pada Balita: Comparison Algorithm of K-Nearest Neighbor and Naïve Bayes Classifier for Classifying Nutritional Status in Toddlers," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 1, Art. no. 1, May 2023, doi: 10.57152/malcom.v3i1.474.

[14] R. D. Y. Prakoso, B. S. Wiriaatmadja, and F. W. Wibowo, "Sistem Klasifikasi Pada Penyakit Parkinson Dengan Menggunakan Metode K-Nearest Neighbor," *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS)*, vol. 1, no. 1, Art. no. 1, Feb. 2020.

[15] Dubravka Bożi´ c, Biserka Runje, Dragutin Lisjak, and Davor Kolar, "Metrics Related to Confusion Matrix as Tools for Conformity Assessment Decisions," *Applied Sciences*, vol. 13, pp. 8187–8205, Jul. 2023, doi: 10.3390/app13148187.