JSCDM

JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

Vol. 7 No. 2 (2026) 388-402
https://publisher.uthm.edu.my/ojs/index.php/jscdm

Analysis of Normalization, Resampling, and Weighted Voting to Improve k-NN Performance in Diabetes Disease Detection

Agustiyar [1]*, R. Rizal Isnanto[1], Budi Warsito[2], Adi Wibowo[3], Ferry Jie[4]

[1] *Doctoral Program of Information Systems Postgraduate School, Universitas Diponegoro, Semarang, INDONESIA*
[2] *Department of Statistics, Faculty of Science and Mathematics, Universitas Diponegoro, Semarang, INDONESIA*
[3] *Department of Informatics, Faculty of Science and Mathematics, Universitas Diponegoro, Semarang, INDONESIA*
[4] *School of Business and Law, Edith Cowan University, 270 Joondalup Drive, Joondalup, WA6065, AUSTRALIA*

*Corresponding Author: agustiyar001@gmail.com
DOI: https://doi.org/10.30880/jscdm.2026.07.02.021

Article Info

Received: 12 January 2026
Accepted: 12 April 2026
Available online: 30 June 2026

Keywords

Diabetes detection, k-nearest neighbors, resampling, normalization, weighted voting

Abstract

Diabetes is a global health issue, and machine learning has become crucial for early detection. The k-Nearest Neighbors (k-NN) machine learning algorithm is widely used because it is simple and easy to implement; however, its performance is sensitive to feature scale differences, class imbalance, and the assumption of uniform decision-making, as commonly found in clinical datasets such as the Pima Indians Diabetes (PID) dataset. Many studies have been conducted to address these issues; however, very few have systematically evaluated the combined effects of preprocessing and distance-based decision-making methods on k-NN performance. This study evaluates the combined effects of preprocessing strategies and distance-based decision-making methods on k-NN performance. Two preprocessing methods, namely normalization and resampling, were evaluated. The normalization methods evaluated are MinMax and Z-Score, while the resampling methods evaluated are SMOTE, ADASYN, and Random Undersampling. Two weighted decision-making schemes, namely Dual Weighted and Gaussian Weighted Voting, were also investigated. Performance was evaluated using Accuracy, Precision, Recall, Specificity, F1-score, AUC, and execution time analysis. The results show that Z-Score normalization combined with SMOTE improves the accuracy of the baseline k-NN from 74.48% to 84.10% and increases recall from 54.90% to 90.80%. Incorporating Dual Weighted Voting achieves the highest overall performance, with 84.20% accuracy, 91.80% recall, 85.32% F1-score, and an AUC of 0.85, while maintaining low computational cost. However, the additional improvement over the best preprocessing configuration is not statistically significant. These findings indicate that performance gains are primarily driven by normalization and resampling strategies, while distance-sensitive

voting provides marginal additional refinement for classical k-NN in imbalanced medical classification tasks.

1. Introduction

Diabetes was the eighth most common cause of death in the world in 2021 [1]. In the same year, 10.5% of adults worldwide aged 20–79 were living with diabetes. The total number is projected to increase to 643 million (11.3%) by 2030 and to 783 million (12.2%) by 2045 [2]. Furthermore, according to the IDF, in 2021 Indonesia was ranked fifth as the country with the largest number of diabetes sufferers. Therefore, this issue remains a major public health concern in Indonesia, considering that diabetes mellitus is known as a source of various diseases.

Diabetes prevention efforts can be done by implementing a healthy lifestyle, including maintaining ideal body weight, implementing a healthy diet, regulating portion sizes, exercising regularly, managing stress, and regularly checking blood sugar levels. In addition to a healthy lifestyle, data mining and machine learning approaches can also be used to detect diabetes, including those carried out by Ahmed et al. [3], who developed a web application for detecting diabetes using the Support Vector Machine, Logistic Regression, Nearest Neighbor, Gradient Boosting, Naive Bayes, Random Forest, and Decision Tree algorithms.

Khanam and Foo [4] evaluated the performance of Logistic Regression and Support Vector Machine for diabetes detection. Rastogi and Bansal [5] applied Random Forest, Support Vector Machine, Logistic Regression, and Naive Bayes for diabetes prediction. Meddage and Rathnayake [6] employed Extreme Gradient Boosting, k-NN, Decision Trees, and Support Vector Classification to identify diabetes cases. Advanced machine learning models combined with resampling techniques have also been applied to predict diabetes [7]. Prasad et al. [8] performed diabetes detection using the Adaptable Fuzzified K-Nearest Neighbor (AF-kNN) model. The results of research conducted by Suyanto et al. [9] using the Multi-Voter Multi-Commission Nearest Neighbor (MVMCNN) outperformed several deep learning approaches. Wee et al. [10] conducted a comprehensive review of more than fifty machine learning and deep learning models and highlighted that each algorithm offers distinct advantages for diabetes detection.

Jafar et al. [11] created a decision support system for diabetic patients undergoing injection therapy using k-NN. Yuk Carrie Lin [12] optimized the k-NN algorithm by selecting relevant features and determining the optimal *k* value to improve the performance of the algorithm on various datasets, one of which is the Diabetes dataset. Li et al. [13] developed V-Detector-kNN, although the processing time was slightly slower but the detection ratio reached 96.38% on the PID dataset. k-NN has been used in the detection of various diseases, including breast cancer detection [14], [15], heart disease detection [16], and even in plant disease detection [17]. k-NN is a classification algorithm that works by measuring the distance from unlabeled test data to labeled training data based on their proximity [3]. The k value describes the number of closest samples that will be used as candidate prediction results. Distance is measured using Euclidean Distance [18], this is the most frequently used metric in k-NN [19].

k-NN has several advantages including simplicity, making it easy to understand [20], and easy to apply [21] by its users. However, despite these advantages, k-NN performance may degrade when applied to small datasets containing outliers [22]. This occurs because k-NN has the principle that each attribute has an equally important role in determining the classification results, while in reality each attribute has a different role [23]. This can be seen when Euclidean Distance does not provide weight when measuring the distance from unlabeled test data to labeled training data.

Although many studies have applied machine learning and k-NN variants to diabetes detection, most focus either on proposing new classifier formulations or on comparing multiple algorithms on the same dataset without systematically analyzing the contribution of preprocessing and voting strategies. A survey conducted by Abu-shareha et al., indicates that improvements in diabetes prediction performance are more heavily influenced by normalization and class-balancing techniques than by increased model complexity [24]. However, there remains a lack of research that combines distance-based weighted voting methods, class-imbalance resampling, and feature normalization within a single experimental framework. Furthermore, there remains insufficient attention to whether preprocessing methods or changes to the voting mechanism are the primary drivers of performance improvements in k-NN.

This study investigates two normalization methods, namely MinMax and Z-Score, three resampling methods, namely SMOTE, ADASYN, and Random Undersampling, along with two voting methods, namely Dual Weighted and Gaussian-Weighted Voting on the performance of k-NN using the Pima Indians Diabetes dataset. The study has three main contributions: first, a systematic evaluation of two normalization methods and three resampling methods for diabetes detection using the k-NN method; second, the integration and comparison of Dual and Gaussian distance-based weighted voting within a unified experimental protocol; and finally, practical insights showing that normalization and resampling have the greatest effect on performance improvement, while weighted voting has the smallest effect on the k-NN method. The results of this research can serve as a guide for configuring the k-NN algorithm when working with imbalanced medical data.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 388-402

This research paper is organized as follows: Section 2 presents the materials and methods. Section 3 presents the experimental results and discussion. Finally, Section 4 presents conclusions of this research.

2. Materials and Methods

2.1 Dataset

This study uses the Pima Indians Diabetes (PID) dataset from Kaggle. The CSV file has 768 rows, each with eight input attributes (Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, and Age) and one output label (Outcome). Before normalization, Fig. 1 shows how these traits are spread out. Pregnancies have a positively skewed distribution, which means that there are a lot of little values and a long tail on the right. Glucose is distributed in a way that is close to normal, but not quite. Blood pressure is also near to normal, however there are a few low outliers. The distribution of skin thickness is favorably skewed, with a lot of small values. There are a lot of little values and a few huge outliers in insulin, which makes it very right-skewed. The BMI is almost normal, however it leans a little to the right. The Diabetes Pedigree Function is heavily right-skewed, with a lot of minor values and a few big outliers. Age also has a positively skewed distribution, meaning that most of the data points are at younger ages.

Most of the features don't fit into a normal distribution, hence data normalization is necessary. Normalization makes outliers less important and stops some features from affecting the distance calculation [25]. Setting feature values on a scale that makes sense is an important step in preparing data. As a result, machine learning algorithms might work better and be more dependable [26]. Fig. 2 shows that the PID dataset's class distribution is not balanced at all. When the data isn't balanced, the model tends to favor the majority class. This makes it more likely to get the minority class wrong [27]. One option to remedy this is to utilize sampling methods to make the distribution of classes more even.

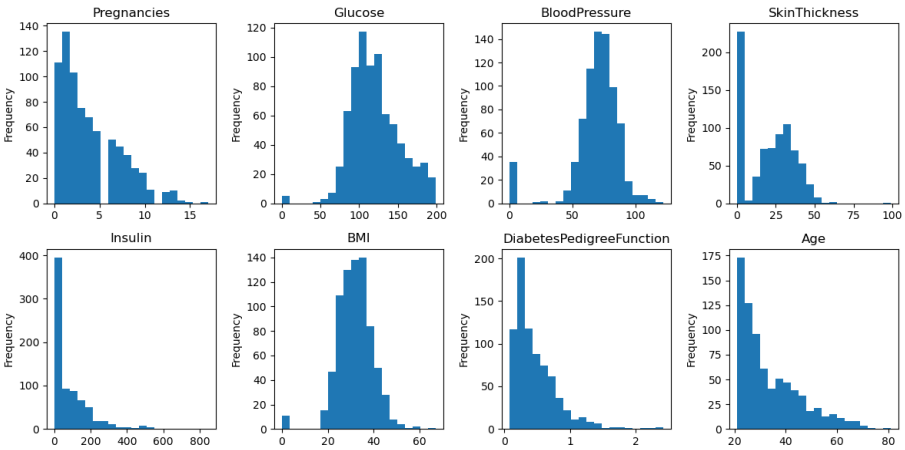

Fig. 1 *PID dataset before normalization*

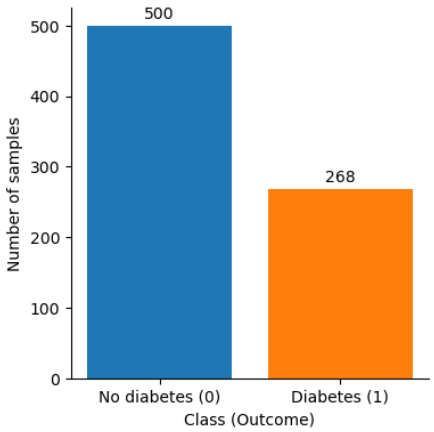

Fig. 2 *PID dataset class distribution*

2.2 Data Cleaning

There are several features in the PID dataset that contain zero values that don't make sense in a medical context. These are Glucose, Blood Pressure, Skin Thickness, Insulin, and BMI. People usually view of these 0 entries as missing data instead of real physiological observations. So, before normalization and resampling, we treated 0 values in these attributes as missing and filled them in with the median. Median imputation was selected because to the skewed distribution of various characteristics and its reduced sensitivity to outliers compared to mean imputation. Before evaluating the model, the imputation was always done on the dataset.

2.3 Data Normalization

Normalizing data is an essential preparatory step for distance-reliant machine learning models like k-NN, as it prevents variables with larger scales from disproportionately influencing the results. In this study, two normalization methods were used: MinMax and Z-score. This was done because without normalization, features with high values, such as glucose would overshadow insulin which has lower values during distance calculations. The original feature values are transformed into the range [0, 1] without altering the original data distribution using MinMax normalization, as shown in Fig. 3.

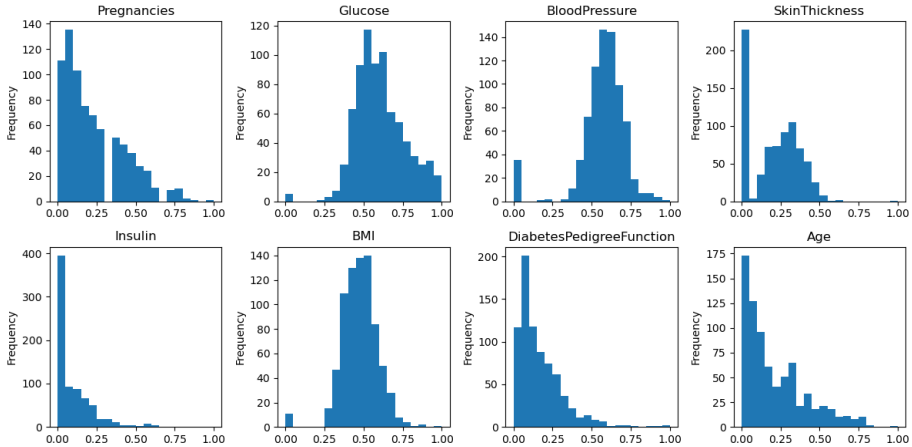

Fig. 3 *MinMax normalization result on PID dataset*

2.3.1 MinMax Normalization

MinMax normalization scales feature values into the range [0, 1]. This method is particularly useful for data with a non-normal distribution, as MinMax normalization eliminates differences in scale while preserving the relative relationships between values [28]. MinMax normalization is defined in Eq. (1):

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where x: initial value of feature, x_{min}: minimum value of feature, x_{max}: maximum value of feature.

Each feature value will be scaled to the range [0, 1]; this helps prevent certain features from dominating the classification when using the k-NN algorithm.

2.3.2 Z-score Normalization

Z-score normalization works by transforming feature values so that the mean is 0 and the standard deviation is 1. Z-score normalization is particularly useful for datasets with varying scales. It can also be used to ensure that each feature contributes equally to distance-based calculations. By using Z-score normalization, the stability and performance of classification models such as k-NN can be improved. Z-score normalization is defined in Eq. (2):

$$z = \frac{X - \mu}{\sigma} \tag{2}$$

where z: normalized value, X: original value, μ: average of features, σ: standard deviation.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 388-402

2.4 Data Sampling

The performance of machine learning models can decline when dealing with datasets that have an imbalanced class distribution [29]. This occurs due to bias toward the majority class. When dealing with such datasets, resampling techniques must be applied to balance the class distribution. The PID dataset is an example of a dataset with an imbalanced class distribution. This is evident in the fact that the "no diabetes" class has nearly twice as many instances as the "diabetes" class. Fig. 4 shows the class distribution after applying the SMOTE method, which balances the dataset by synthetically generating instances of the minority class. Three resampling technique is explored in this research:

- Synthetic Minority Over sampling Technique (SMOTE)
- Adaptive Synthetic Sampling (ADASYN)
- Random Under sampling

These three resampling techniques are capable of balancing datasets with imbalanced class distributions.

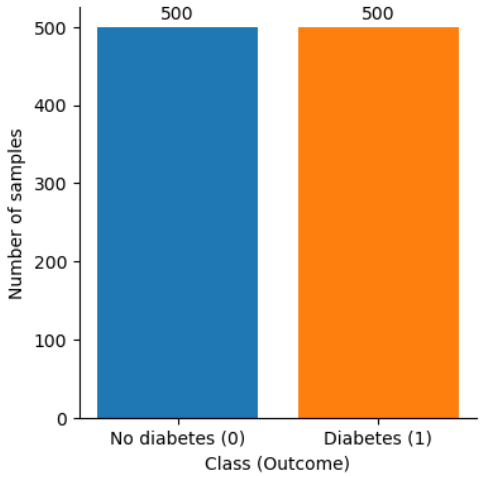

Fig. 4 *SMOTE make PID dataset balance*

2.4.1 SMOTE

The Synthetic Minority Oversampling (SMOTE) technique generates synthetic samples for the minority class, thereby balancing the class composition [30]. This technique uses interpolation between randomly selected minority samples. SMOTE is defined in Eq. (3):

$$x_{new} = x_{original} + \lambda\,(x_{neighbor} - x_{original}) \tag{3}$$

where λ: random number between 0 and 1.

2.4.2 ADASYN

Adaptive Synthetic Sampling (ADASYN) is an improved version of SMOTE that can be used to balance datasets with imbalanced classes [31]. SMOTE generates synthetic samples uniformly [32], whereas ADASYN generates synthetic samples based on the local density of minority instances and their proximity to majority-class neighbors [33]. The number of synthetic samples N_i generated for each minority instance x_i is determined by the local density of majority-class neighbors, as defined by Eq. (4):

$$N_i = round(\frac{D_i}{\sum D_i}.N_{total}) \tag{4}$$

where D_i represents the number of majority class neighbors for instance x_i, $\sum D_i$ is the total count across all minority instances, and N_{total} is the total number of synthetic samples to generate.

2.4.3 Random Undersampling

Random undersampling randomly selects a subset of majority class samples to match the number of minority class instances, thereby balancing the dataset. The new size of the majority class is defined as Eq. (5):

$$N'_{majority} = N_{minority} \tag{5}$$

The resulting balanced dataset is obtained by combining the reduced majority class with the original minority class Eq. (6):

$$D_{balanced} = D'_{majority} \cup D_{minority} \tag{6}$$

where $N_{minority}$: number of samples in the minority class, $N'_{majority}$: new number of samples in the majority class. $D_{balanced}$: new balance dataset, $D'_{majority}$: new majority data, $D_{minority}$: minority data.

While undersampling is computationally efficient and simple to implement, it may lead to the loss of important information by discarding potentially useful samples from the majority class [34]. Therefore, it is best applied when the dataset is large and the majority class is significantly overrepresented.

2.5 *k*-Nearest Neighbor

The *k*-Nearest Neighbor (*k*-NN) is a non-parametric classification technique that labels an unknown sample based on the majority class among its *k* nearest neighbors. It operates on the principle that similar instances exist in close proximity in feature space. The most common distance metric used in k-NN is Euclidean distance, calculated as Eq. (7):

$$d(x', x) = \sqrt{\sum_{a=1}^{n} (x'_a - x_a)^2} \tag{7}$$

where x' is the test sample, x is a training sample, and n is the number of features. Once the distances are computed, the class label is determined through majority voting Eq. (8):

$$y' = arg\,max_y \sum_{i=1}^{k} y = y_i \tag{8}$$

y': prediction class, y: class, y_i: i-th label of k nearest neighbors, k: number of neighbor.

Fig. 5 illustrates the traditional k-NN algorithm in the voting process. A test data point is classified based on the labels of the k nearest data points, with each vote having equal weight regardless of distance. Although effective in some cases, this assumption of equal weight can lead to misclassification, especially in datasets containing outliers.

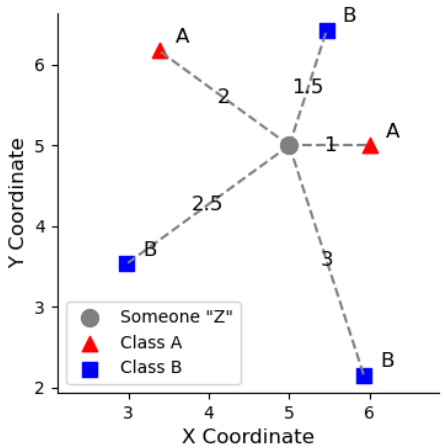

Fig. 5 *Voting process in traditional k-NN*

2.6 Interpersonal Trust and Weighted Voting

In traditional k-NN prediction, the class is determined based on the majority class among the k nearest neighbors, without considering the weight of the distance between the neighbors and the query point. However, this strategy does not reflect real-world decision-making processes, where closer sources tend to have a greater influence. The k-NN voting mechanism is based on interpersonal trust theory [36] and introduces a distance-based weighting into the study. Fig. 6 illustrates the effect of trust-based weighting on classification. For instance, if "Z" has 5

nearest neighbors: 2 from A (1.0 and 2.0), and 3 from B (3.0, 4.0 and 5.0) standard majority voting would give "Z" class B as the predicted label. However, if we apply weights based on proximity, class A may receive a higher cumulative weight and become the predicted class.

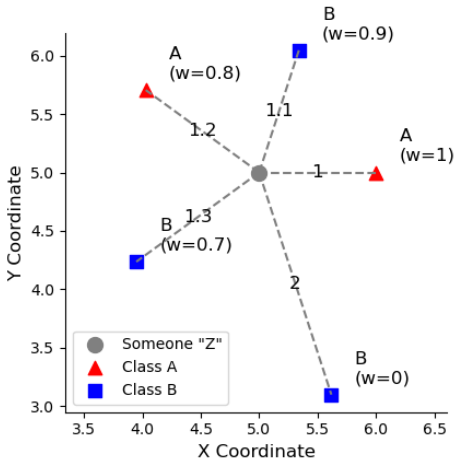

Fig. 6 *Interpersonal trust theory applied on k-NN*

Through this interpersonal trust theory approach, the core limitation of conventional k-NN namely the lack of distance sensitivity in decision making, can be addressed.

2.7 Dual Weighted Voting k-NN

Dual Weighted Voting [35] is a distance-based weighting strategy that assigns greater influence to closer neighbors in the k-NN algorithm. With this method, the weights are scaled linearly based on the distance between the nearest and farthest neighbors in the k-nearest-neighbor set. Dual Weighted Voting is defined as Eq. (9):

$$w_i = \frac{d_k - d_i}{d_k - d_1} \times \frac{d_k - d_i}{d_k - d_1} \tag{9}$$

where w_i is the weight assigned to the i-th neighbor, d_i is the distance from the test point to the i-th neighbor, d_1 and d_k are the distances to the nearest and farthest neighbors, respectively.

Dual Weighted Voting allows classification methods to make predictions that are more context-sensitive. Neighbors closer to the query point have a greater influence than those farther away. With this approach, the influence of outliers or less relevant neighbors can be reduced.

2.8 Gaussian-Weighted Voting k-NN

Gaussian-Weighted Voting [36] weights each of the k nearest neighbors by applying a Gaussian distribution. Compared to a linear weighting scheme, this method is more flexible and refined. The weight assigned to the i-th neighbor is defined as Eq. (10):

$$w_i = \exp(-\frac{d_i^2}{d\sigma^2}) \tag{10}$$

where w_i is the weight for the i-th neighbor, d_i is the distance between the query point and the i-th neighbor, σ is a smoothing parameter that controls the rate at which weight decays with distance. In this study, σ was set to the standard deviation of distances within each neighborhood, following common practice in distance-weighted k-NN.

Because the function is exponential, closer neighbors are given substantially more weight than farther neighbors. It is especially convenient on high-dimensional or noisy data, where the surrounding points can be regarded as a more significant determinant of classification.

2.9 Weighted Voting *k*-NN Mechanism

The traditional k-NN algorithm employs majority voting, where each neighbor contributes equally to the final decision. In contrast, weighted k-NN voting adjusts each neighbor's contribution based on its proximity to the query point, as determined by a weighting function. In this study, both Dual-Weighted Voting and Gaussian-

Weighted Voting calculate a weighted score for each class label based on the neighbors in that class. The class score is calculated as Eq. (11):

$$\text{Score}(y_j) = \sum_{i \in N_j} w_i \tag{11}$$

The predicted class $\hat{y}$ is the one with the highest total score:

$$\hat{y} = argmax_k Score(y_j) \tag{12}$$

where y_j: j class which the score is being calculated, N_j: The set of neighbors with label y_j, w_i: weight of the i-th neighbor, typically based on distance, $\hat{y}$: predicted class.

The classification method is more robust against noise and outliers and better captures local structures in the data thanks to distance-sensitive weighting connections. Compared to uniform voting, this mechanism improves the decision-making process.

2.10 Experiment Method

This study systematically evaluated normalization, resampling, and weighted voting strategies on the performance of k-NN for diabetes detection. The process was divided into six main stages, as shown in Fig. 7.

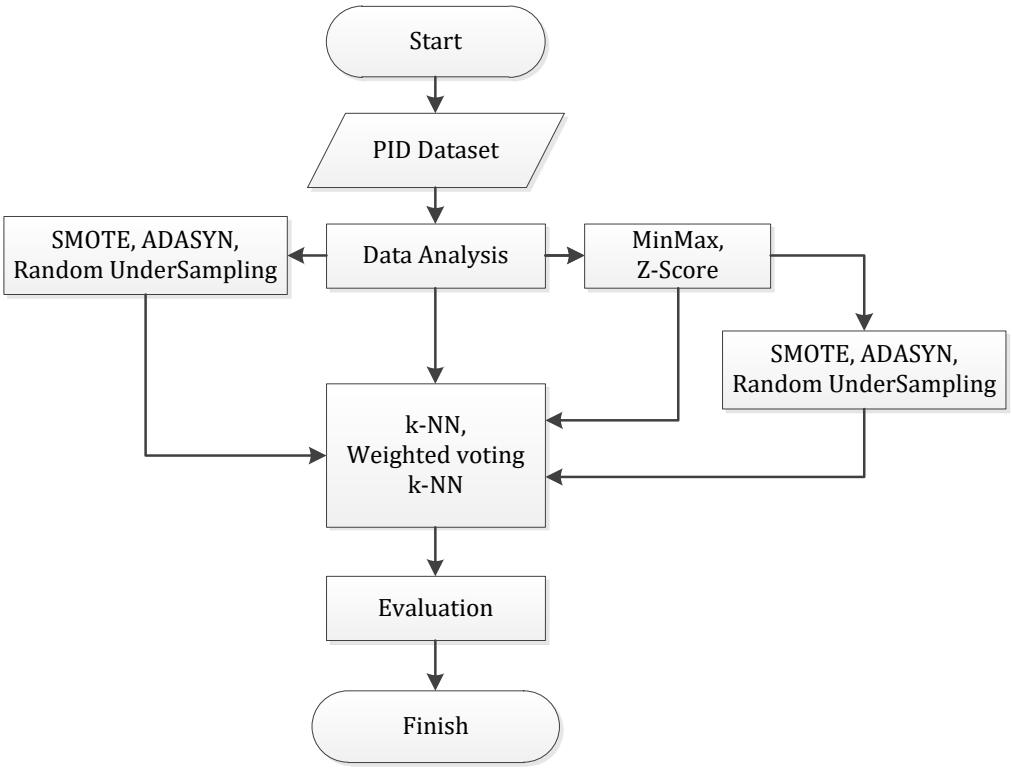

Fig. 7 *Experiment method flow diagram*

- **Data Preparation**: We explored the PID dataset to assess the feature and class distributions. This determined whether data normalization or resampling was needed by evaluating feature skewness and class imbalance.
- **Normalization and Initial Resampling**: Two normalization techniques, MinMax and Z-score, are applied individually, followed by SMOTE, ADASYN, and Random Undersampling to assess their effects independently.
- **Combined Preprocessing**: The best performing normalization method is combined with each resampling technique to evaluate their synergistic impact on classification performance.
- **Model Training**: Both classical k-NN and weighted voting k-NN (using Dual and Gaussian weighting schemes) are trained on the preprocessed datasets.

- **Evaluation**: All models are validated using stratified 10-fold cross-validation. Odd values of k from 1 to 15 are used to avoid tie conditions in the voting process.
- **Model Selection**: The optimal configuration was identified based on the consistency of overall multi-metric performance across cross-validation folds.

Performance metrics such as Accuracy, Precision, Recall, Specificity, F1-score, AUC, and time are used to gain a more comprehensive understanding of the model's performance. Accuracy is used to observe general performance trends across various preprocessing steps and voting strategies. Meanwhile, Recall and F1-score can indicate how well the minority class is handled, an aspect that becomes particularly important when the data distribution is imbalanced. AUC can assess how effectively the model distinguishes between classes. By combining these performance metrics, the comparison of the proposed configurations becomes more balanced and methodologically sound.

2.11 Parameter Sensitivity Analysis

The sensitivity of the k-NN classifier was examined using two complementary strategies. First, the neighborhood size *k* was varied from 1 to 15 using odd values (1, 3, 5, 7, 9, 11, 13, and 15) in order to analyze the balance between overfitting and generalization, while simultaneously preventing tie outcomes during the classification process. Second, we performed stratified 10-fold cross-validation to retain the original proportion of classes in each fold, allowing a more accurate and unbiased evaluation of model performance.

2.12 Statistical Significance Testing

To empirically assess statistical significance of performance improvements over baseline k-NN model versus configurations proposed (the extra features that were deemed informative) we conducted stratified 10-fold cross-validation which provided through each validation fold an potential optimal classifier means towards each validation fold. Results from each validation folds respective accuracies were treated as paired observations and the differences of those pairwise model comparisons put to a paired t-test with significance level $\alpha = 0.05$. By fixing the data partitions for all models under consideration, this approach provides a stable basis to assess whether any differences observed in performance reflect real methodological improvements or simply noise.

3. Results and Discussion

In this study, we use the Pima Indians Diabetes (PID) dataset. A stratified 10-fold cross-validation process was used in both training and evaluation steps, where the dataset was divided into ten equally sized parts, and learning procedure was applied over the ten folds iteratively. The values of *k* that were utilized in this experiment are 1, 3, 5, 7, 9, 11, 13 and finally the last value is equal to15. The accuracy results from the 10-fold cross-validation of several variants of k-NN algorithms on the PID dataset are outlined in Table 1, indicating which combinations outperformed others (the best model configuration is highlighted in bold).

Table 1 *Accuracy of the k-NN variant on the PID dataset*

Methods / *k* value	1	3	5	7	9	11	13	15	Average of Accuracy
k-NN	67.97%	70.31%	72.14%	73.96%	73.83%	73.69%	74.22%	74.48%	72.58%
k-NN+MinMax	70.83%	73.95%	75.00%	75.39%	74.22%	73.96%	75.65%	76.05%	74.38%
k-NN+Z-Score	70.57%	74.34%	73.82%	74.22%	74.61%	74.48%	74.74%	74.74%	73.94%
k-NN+SMOTE	81.50%	76.30%	75.40%	75.10%	74.70%	74.60%	73.80%	72.20%	75.45%
k-NN+ADASYN	79.69%	74.24%	72.39%	70.75%	70.03%	70.95%	71.26%	70.24%	72.44%
k-NN+Random Undersampling	65.48%	66.98%	69.61%	69.80%	70.34%	72.03%	71.48%	70.90%	69.58%
k-NN + MinMax + SMOTE	83.20%	79.10%	78.90%	76.30%	75.20%	75.50%	75.60%	75.60%	77.43%
k-NN+Z-Score+SMOTE	84.10%	80.10%	80.90%	78.70%	78.40%	77.70%	77.20%	77.10%	**79.28%**

During the initial phase of the analysis, the k-NN model was evaluated under baseline conditions without any preprocessing, resulting in an average accuracy of 72.58%. After normalizing, performance improved significantly. Min–Max normalization provided the largest increase in accuracy, from 70.66% to 74.38%, compared to Z-Score normalization, which provided a more modest improvement of accuracy (from 70.66% to 73.94%). Widespread discrepancies were observed when resampling methods to rectify class imbalance were utilized. Of the considered approaches, SMOTE showed the most significant gain with an accuracy of 75.45%. On the other hand, ADASYN reached almost similar accuracy of 72.44% to the baseline performance, and random undersampling has led to a reduced performance which was equal to only 69.58%. Fortunately, these findings indicate that the performance impairment can be attributed to a huge elimination of majority-class samples, which hinders the model's ability for capturing underlying data distribution.

Accordingly, based on these observations, SMOTE was selected as the appropriate resampling strategy for experimental studies going forward. So, the next experiment was carried out by applying a combination of normalization method and SMOTE to train k-NN model. The comparison is shown in Fig. 8 and it indicates that pairing these techniques consistently produced better results. When MinMax scaling and SMOTE were used together, the model achieved an average accuracy of 77.43%. The configuration using Z-Score normalization with SMOTE performed even better, reaching 79.28%, making it the top-performing setup in this series of tests on the PID dataset.

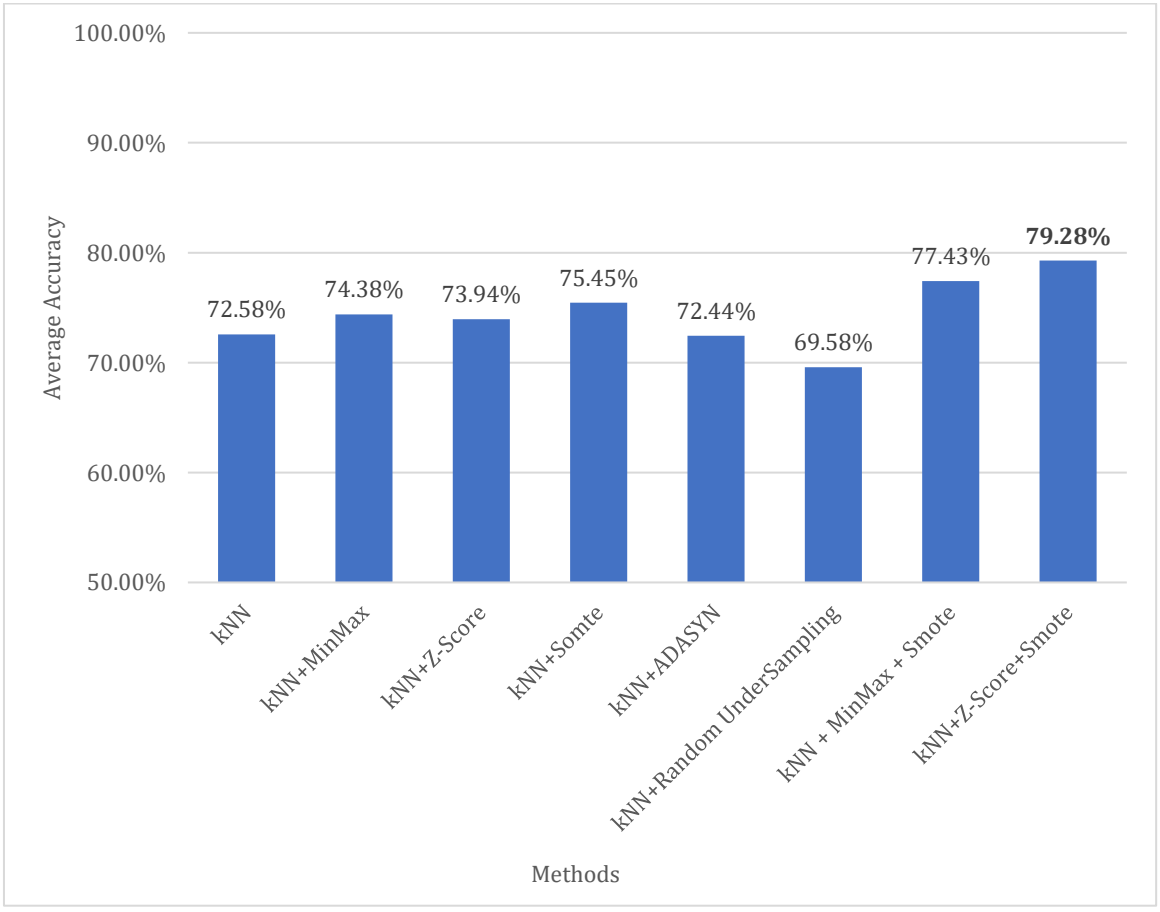

Fig. 8 *k-NN accuracy without and with normalization and resampling methods for the PID dataset*

As the next step, the standard majority voting mechanism in k-NN is replaced with weighted voting. Two weighting schemes are evaluated, namely Dual Weighted Voting and Gaussian Weighted Voting, applied on top of the best preprocessing configurations identified in the previous experiment, i.e., MinMax + SMOTE and Z-Score + SMOTE. Table 2 shows the stratified 10-fold accuracy of the weighted voting k-NN variants on the PID dataset. Best-performing configuration highlighted in bold.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 388-402

Table 2 *Stratified 10-fold accuracy of weighted voting k-NN variants on the PID dataset*

Methods / *k* value	1	3	5	7	9	11	13	15	Average of Accuracy
k-NN + MinMax + SMOTE +Dual Weighted	83.20%	83.20%	83.90%	83.30%	82.50%	81.60%	81.70%	80.80%	82.53%
k-NN + MinMax + SMOTE +Gaussian-Weighted	83.20%	79.10%	78.90%	76.30%	75.20%	75.50%	75.60%	75.60%	77.43%
k-NN + Z-Score + SMOTE +Dual Weighted	84.10%	84.10%	84.20%	83.30%	82.60%	82.10%	81.90%	81.70%	**83.00%**
k-NN + Z-Score + SMOTE +Gaussian-Weighted	84.10%	80.60%	81.20%	80.30%	80.30%	79.50%	79.10%	78.90%	80.50%

Fig. 9 demonstrates how the four different types of weighted voting for k-NN did on average. When utilized with MinMax normalization and SMOTE, the Dual Weighted Voting setup was roughly 82.53% accurate. The same preprocessing steps were followed, but the weighting method was changed to the Gaussian version, which caused the value to decrease a lot, to roughly 77.43%. When Z-Score normalization takes the role of MinMax, the same thing happens. The Dual Weighted Voting method wins again, this time with Z-Score and SMOTE, getting about 83.00%. The Gaussian-weighted model, on the other hand, levels off at 80.50%. The study did not conduct a formal statistical comparison between the Dual and Gaussian weighting schemes; however, the overall tendency is evident from the descriptive results. The Dual Weighted method was always more accurate than the Gaussian method in every setup we tried. This means that it works better in this scenario and with this dataset.

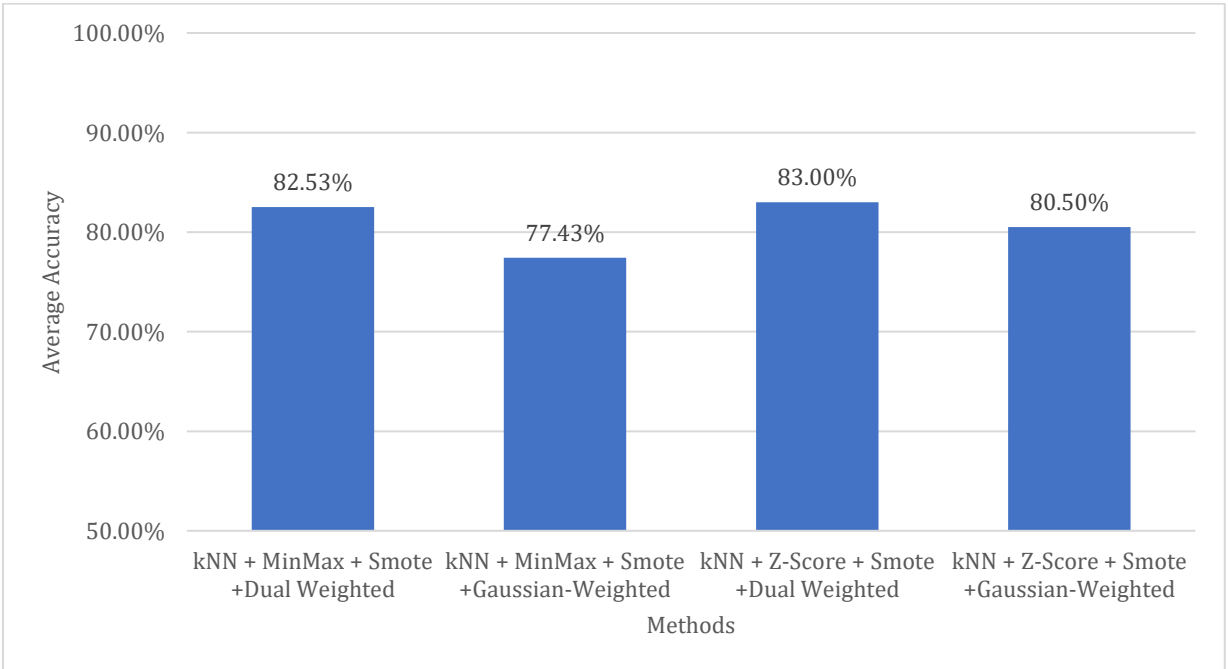

Fig. 9 *Weighted voting k-NN accuracy without and with normalization and resampling methods for the PID dataset*

In the statistical evaluation, the analysis relied on the accuracy scores collected from each fold, using the best k value identified earlier for every model during cross-validation. For each pair of models being compared, the accuracy values obtained from the same folds were treated as paired observations and analyzed using a paired t-test. The paired t-test between the baseline k-NN and the configuration using Z-Score normalization with SMOTE yielded $t = 6.37$, $p = 0.00013$, indicating a statistically significant improvement. However, when the model using Z-Score normalization with SMOTE was compared with its Dual Weighted extension, the paired t-test yielded $t = 0.10$, $p = 0.92117$, indicating that the difference is not statistically significant. This suggests that most of the

performance improvement is primarily driven by preprocessing, while weighted voting provides only marginal refinement under the current experimental setting. It should be noted that the statistical test was applied to evaluate improvements from preprocessing configurations relative to the baseline model, while a formal statistical comparison between Dual Weighted and Gaussian weighting schemes was not conducted and is therefore interpreted descriptively.

A comparison between the k-NN baselines, its improved counterparts, and multiple standard machine learning algorithms is shown in Table 3. As it is, the baseline model yields an accuracy of 74.48% but its recall is relatively weak (54.90%), i.e., it does not identify a significant number of diabetes-positive cases but has high specificity (85.00%). Grabbing the Z-Score normalization and SMOTE leads to a very good improvement with 84.10% in accuracy and 90.80% in recall. These results highlight how much preprocessing and class-balancing affect the model's performance for detecting positive cases. Improving (>2%) performance with Dual Weighted Voting yields an accuracy of 84.20%, a recall of 91.80% and F1-score of 85.32% as well the best AUC of 0.85. Traditional models such as Random Forest and Logistic Regression perform around mid–70% accuracy (76.56% and 76.69%, respectively) with significantly lower recall (58.58% and 53.73%). The XGBoost also shows a between the accuracy (74.61%) and recall (60.07%). These results suggest that the performance improvements on the PID dataset are mostly driven by preprocessing and resampling, with additional but small gains in producing more sensitive models regarding positive cases due to the Dual Weighted Voting mechanism.

Now, the examination of execution times illustrates computational behaviors aligned with the nature of each assessed algorithm. The baseline k-NN model shows the minimum execution time of 0.0146 seconds because it does not need complicated training processes. The execution time increases by 0.2025 seconds when applying the Z-Score normalization and SMOTE resampling technique. Introducing the Dual Weighted Voting mechanism pushed execution time to 0.4847 seconds, with the added burden primarily stemming from calculating additional weights in the prediction phase. However, this increase remains modest. While ensemble-based methods like Random Forest or XGBoost take much longer to compute 5.7787 and 2.2031 seconds, respectively as this method builds trees in an ensemble manner that work either together or one after the other to improve prediction performance. The computational efficiency remains high with an execution time of 0.0478 seconds for Logistic Regression. In summary, the resulting k-NN configuration of this approach provides better predictive performance than existing individual methods with computational costs more competitive than those of more sophisticated ensemble methods.

Table 3 *Performance comparison of baseline and best-performing model*

Methods	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1-score (%)	AUC	Time (s)
k-NN	74.48	66.84	54.90	85.00	59.99	0.80	0.0146
k-NN + Z-Score + SMOTE	84.10	80.08	90.80	77.40	84.99	0.84	0.2025
k-NN + Z-Score + SMOTE +Dual Weighted	84.20	79.69	91.80	76.60	85.32	0.85	0.4847
Random Forest	76.56	69.47	58.58	86.20	63.56	0.83	5.7787
Logistic Regression	76.69	72.36	53.73	89.00	61.67	0.83	0.0478
XGBoost	74.61	64.66	60.07	82.40	62.28	0.81	2.2031

The Dual-Weighted Voting strategy outperforms the Gaussian-based weighting scheme across the board and this result has contributed heavily towards hell when using a PID dataset. A lot of the features in this dataset are skewed and contain outliers; this is very apparent from features such as Insulin, SkinThickness and FamilyHistory. Under such circumstances, policies based on linear distance-based weighting schemes have more stability because they are less sensitive to the unexpected changes of distances values. These weights still favor nearby neighbors, but do not excessively amplify distance differences. Consequently, when tested on small and moderately noisy clinical datasets such as PID, the Dual-Weighted Voting method proved to be more robust.

Conversely, the Gaussian weighting strategy applies an exponential decay function which can be too strong for differently distributed inter-sample distances. In such cases, neighboring instances that are relatively close might have little influence on the decision process due to asymmetrical distance distributions. While this behaviour is acceptable in datasets with very large sample size and low variance, it can cause excessive variation in aggregated class scores for limited-variance, small-sample datasets that would otherwise attenuate the model's generalization capability. It can also be observed that the Dual Weighted Voting mechanism consistently outperforms the Gaussian approach across different values of k, suggesting that, at least in this case, a simpler linear weighting formulation exhibits more stable performance when combined with normalization and resampling to tackle class imbalance commonly found in medical datasets.

From a clinical standpoint, such use of a predictive model is crucial for diabetes screening, as false negatives would cause delayed treatment and an associated increase in serious complications from the disease. The k-NN model established in the present work produced a recall of 91.80%, representing a meaningful enhancement on its respective baseline model and suggesting an improved capacity with respect to identifying individuals that might be at risk for diabetes. Although this increase in sensitivity results in a small compromise of specificity, the trade-off is usually acceptable at the initial screening stage where we seek to detect as many true positive cases as possible and control over false positives is less critical. As such, this approach is most appropriate in a decision-support capacity for initial risk assessment rather than as a final diagnostic tool.

4. Conclusions

The performance of k-NN for diabetes prediction relies substantially on the characteristics of the dataset and the data preprocessing steps that were taken before using a specific model. Weights adjustable by simple linear distances appear to work better on the PID dataset that contained multiple skewed features and significant outliers. Overall, our experimental results were shown the Dual Weighted Voting method exceed to Gaussian-based method in every evaluation metrics we use for both datasets which means that the linear approach is more stable than its counterpart and indeed it has a better working fitting for a small number of data with several noise since they are relatively correlated with the normalization and resample methods.

These results suggest that much of the performance gained comes from data cleaning rather than altering the voting mechanism. The top results from the evaluation showed that a Z-Score normalized data sample classed with SMOTE (Synthetic Minority Oversampling Technique) outperformed other configurations used alongside a simple majority voting system. Using weighted voting based on prediction confidence and implementing it to this setting only showed negligible improvements in performance (none statistically significant) as compared to just using preprocessing. Though, Dual Weighted Voting with Z-Score normalization and SMOTE performed best overall accuracy of 84.20%. These results indicate that the performance is in a greater extent affected by preprocessing strategies carefully designed and proper voting mechanisms rather than changes made to the k-NN algorithm at its core.

In conclusion, the combination of normalization and resampling strategy greatly enhances k-NN performance with weighted voting being minimal fine-tuning. While the PID dataset is a benchmark in diabetes prediction research, reliance on a single dataset may limit the generalizability of findings. Future research will confirm the proposed configuration on larger and multi-center clinical data sets to assess robustness over varying demographic distributions. Additionally, future studies could investigate the integration of feature selection and dimensionality reduction techniques to further optimize model performance.

Acknowledgement

The authors used ChatGPT for coding assistance and Grammarly for grammar checking and language editing. All content generated was reviewed and verified by the authors, who take full responsibility for the final submission.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

***Agustiyar**: Conceptualization, writing – original draft; **R. Rizal Isnanto**: Funding acquisition; **Catur Edi Widodo**: Improve the writing quality; **Budi Warsito**: Methodology, review and editing; **Adi Wibowo**: Data curation, validation, software; **Ferry Jie**: Supervision.*

References

[1] WHO (2024) The top 10 causes of death - Factsheet, *WHO reports*, no. 7 August 2024, [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death

[2] International Diabetes Federation (2021, November 2) Diabetes now affects one in 10 adults worldwide, pp. 1–7, [Online]. Available: https://idf.org/news/diabetes-now-affects-one-in-10-adults-worldwide/

[3] N. Ahmed *et al.* (2021) Machine learning based diabetes prediction and development of smart web application, *Int. J. Cogn. Comput. Eng.*, vol. 2, no. December, pp. 229–241, https://doi.org/10.1016/j.ijcce.2021.12.001

[4] J. J. Khanam and S. Y. Foo (2021) A comparison of machine learning algorithms for diabetes prediction, *ICT Express*, vol. 7, no. 4, pp. 432–439, https://doi.org/10.1016/j.icte.2021.02.004

[5] R. Rastogi and M. Bansal (2023) Diabetes prediction model using data mining techniques, *Meas. Sensors*, vol. 25, no. October 2022, p. 100605, https://doi.org/10.1016/j.measen.2022.100605

[6] G. Dharmarathne, T. N. Jayasinghe, M. Bogahawaththa, D. P. P. Meddage, and U. Rathnayake (2024) A novel machine learning approach for diagnosing diabetes with a self-explainable interface, *Healthc. Anal.*, vol. 5, no. December 2023, p. 100301, https://doi.org/10.1016/j.health.2024.100301

[7] A. A. Nafea *et al.* (2025) A Machine Learning Technique for Early Detection of Gestational Diabetes Mellitus Using SMOTE and Optimized Light Gradient Boosting Machine, vol. 1, no. 1, https://doi.org/https://doi.org/10.65511/jaima.v1i1.739

[8] B. V. V. S. Prasad, S. Gupta, N. Borah, R. Dineshkumar, H. K. Lautre, and B. Mouleswararao (2023) Predicting diabetes with multivariate analysis an innovative KNN-based classifier approach, *Prev. Med. (Baltim).*, vol. 174, no. May, p. 107619, https://doi.org/10.1016/j.ypmed.2023.107619

[9] S. Suyanto, S. Meliana, T. Wahyuningrum, and S. Khomsah (2022) A new nearest neighbor-based framework for diabetes detection, *Expert Syst. Appl.*, vol. 199, no. April 2021, https://doi.org/10.1016/j.eswa.2022.116857

[10] B. F. Wee, S. Sivakumar, K. H. Lim, W. K. Wong, and F. H. Juwono (2024) Diabetes detection based on machine learning and deep learning approaches, *Multimed. Tools Appl.*, vol. 83, no. 8, pp. 24153–24185, https://doi.org/10.1007/s11042-023-16407-5

[11] A. Jafar, M. R. Pasqua, B. Olson, and A. Haidar (2024) Advanced decision support system for individuals with diabetes on multiple daily injections therapy using reinforcement learning and nearest-neighbors: In-silico and clinical results, *Artif. Intell. Med.*, vol. 148, no. December 2023, p. 102749, https://doi.org/10.1016/j.artmed.2023.102749

[12] K. Yuk Carrie Lin (2024) Optimizing variable selection and neighbourhood size in the K-nearest neighbour algorithm, *Comput. Ind. Eng.*, vol. 191, no. April, https://doi.org/10.1016/j.cie.2024.110142

[13] Z. Li, T. Li, J. He, Y. Zhu, and Y. Wang (2021) A hybrid real-valued negative selection algorithm with variable-sized detectors and the k-nearest neighbors algorithm, *Knowledge-Based Syst.*, vol. 232, p. 107477, https://doi.org/10.1016/j.knosys.2021.107477

[14] S. R. Sannasi Chakravarthy, N. Bharanidharan, and H. Rajaguru (2023) Deep Learning-Based Metaheuristic Weighted K-Nearest Neighbor Algorithm for the Severity Classification of Breast Cancer, *Irbm*, vol. 44, no. 3, p. 100749, https://doi.org/10.1016/j.irbm.2022.100749

[15] E. K. Nyarko, I. Vidović, K. Radočaj, and R. Cupec (2018) A nearest neighbor approach for fruit recognition in RGB-D images based on detection of convex surfaces, *Expert Syst. Appl.*, vol. 114, pp. 454–466, https://doi.org/10.1016/j.eswa.2018.07.048

[16] M. M. Ali, B. K. Paul, K. Ahmed, F. M. Bui, J. M. W. Quinn, and M. A. Moni (2021) Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison, *Comput. Biol. Med.*, vol. 136, no. July, p. 104672, https://doi.org/10.1016/j.compbiomed.2021.104672

[17] Y. Guo, C. Du, Y. Zhao, T. F. Ting, and T. A. Rothfus (2021) Two-level K-nearest neighbors approach for invasive plants detection and classification, *Appl. Soft Comput.*, vol. 108, p. 107523, https://doi.org/10.1016/j.asoc.2021.107523

[18] A. İ. Çetin and A. H. Büyüklü (2025) A new approach to K-nearest neighbors distance metrics on sovereign country credit rating, *Kuwait J. Sci.*, vol. 52, no. 1, https://doi.org/10.1016/j.kjs.2024.100324

[19] K. Schenatto, E. G. de Souza, C. L. Bazzi, A. Gavioli, N. M. Betzek, and H. M. Beneduzzi (2017) Normalization of data for delineating management zones, *Comput. Electron. Agric.*, vol. 143, no. October, pp. 238–248, https://doi.org/10.1016/j.compag.2017.10.017

[20] K. Taunk (2019) A Brief Review of Nearest Neighbor Algorithm for Learning and Classification, *2019 Int. Conf. Intell. Comput. Control Syst. ICCS 2019*, no. Iciccs, pp. 1255–1260.

[21] R. K. Halder, M. N. Uddin, M. A. Uddin, S. Aryal, and A. Khraisat (2024) Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications, *J. Big Data*, vol. 11, no. 1, https://doi.org/10.1186/s40537-024-00973-y

[22] J. Gou, H. Ma, W. Ou, S. Zeng, Y. Rao, and H. Yang (2019) A generalized mean distance-based k-nearest neighbor classifier, *Expert Syst. Appl.*, vol. 115, pp. 356–372, https://doi.org/10.1016/j.eswa.2018.08.021

[23] Y. Ma and X. Zhao (2021) POD: A Parallel Outlier Detection Algorithm Using Weighted kNN, *IEEE Access*, vol. 9, pp. 81765–81777, https://doi.org/10.1109/ACCESS.2021.3085605

[24] A. A. Abu-shareha, M. M. Abualhaj, A. Mohammad, A. Al-saaidah, and A. Achuthan (2025) Diabetes Prediction Through Classification Using Pima Dataset : Survey and Evaluation, vol. 6, no. 1, pp. 1–20.

[25] D. Singh and B. Singh (2022) Feature wise normalization: An effective way of normalizing data, *Pattern Recognit.*, vol. 122, p. 108307, https://doi.org/10.1016/j.patcog.2021.108307

[26] R. C R and D. S. C P (2024) Evaluating Deep Learning with different feature scaling techniques for EEG-based Music Entrainment Brain Computer Interface, *e-Prime - Adv. Electr. Eng. Electron. Energy*, vol. 7, no. December 2023, p. 100448, https://doi.org/10.1016/j.prime.2024.100448

[27] K. Kim (2021) Normalized class coherence change-based kNN for classification of imbalanced data, *Pattern Recognit.*, vol. 120, p. 108126, https://doi.org/10.1016/j.patcog.2021.108126

[28] J. Han, J., Kamber, M., Pei (2012) *Data Mining: Concepts and Techniques*. Elsevier, 2012. [Online]. Available: https://www.sciencedirect.com/book/monograph/9780123814791/data-mining-concepts-and-techniques

[29] D. Elreedy and A. F. Atiya (2019) A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance, *Inf. Sci. (Ny).*, vol. 505, pp. 32–64, https://doi.org/10.1016/j.ins.2019.07.070

[30] Q. Chen, Z. L. Zhang, W. P. Huang, J. Wu, and X. G. Luo (2022) PF-SMOTE: A novel parameter-free SMOTE for imbalanced datasets, *Neurocomputing*, vol. 498, pp. 75–88, https://doi.org/10.1016/j.neucom.2022.05.017

[31] R. Mitra, A. Bajpai, and K. Biswas (2023) ADASYN-assisted machine learning for phase prediction of high entropy carbides, *Comput. Mater. Sci.*, vol. 223, no. November 2022, p. 112142, https://doi.org/10.1016/j.commatsci.2023.112142

[32] A. Imakura, M. Kihira, Y. Okada, and T. Sakurai (2023) Another use of SMOTE for interpretable data collaboration analysis, *Expert Syst. Appl.*, vol. 228, no. May, p. 120385, https://doi.org/10.1016/j.eswa.2023.120385

[33] M. Zakariah, S. A. AlQahtani, and M. S. Al-Rakhami (2023) Machine Learning-Based Adaptive Synthetic Sampling Technique for Intrusion Detection, *Appl. Sci.*, vol. 13, no. 11, https://doi.org/10.3390/app13116504

[34] S. M. Liu, J. H. Chen, and Z. Liu (2023) An empirical study of dynamic selection and random under-sampling for the class imbalance problem, *Expert Syst. Appl.*, vol. 221, no. January, p. 119703, https://doi.org/10.1016/j.eswa.2023.119703

[35] J. Gou, M. Luo, and T. Xiong (2011) Improving k-nearest neighbor rule with dual weighted voting for pattern classification, *Commun. Comput. Inf. Sci.*, vol. 159 CCIS, no. PART 2, pp. 118–123, https://doi.org/10.1007/978-3-642-22691-5_21

[36] T. H. Sarma, P. Viswanath, D. S. K. Reddy, and S. S. Raghava (2013) An improvement to k-nearest neighbor classifier, *Comput. Vis. Pattern Recognit.*, https://doi.org/https://doi.org/10.1109/RAICS.2011.6069307