Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9903

Comparative Evaluation of Machine Learning Algorithms for Diabetes Prediction with SMOTE and Principal Component Analysis

Badia Inaya Sazrade[1,*], Ken Ditha Tania[1], Ferdiansyah[2]

[1] Computer Science, Information System, Universitas Sriwijaya, Palembang, Indonesia
[2] Computer Science, Computer System, Universitas Indo Global Mandiri, Palembang, Indonesia
Email: [1,*]babadhunain@gmail.com, [2]kenya.tania@gmail.com, [3]ferdi@uigm.ac.id
Correspondence Author Email: babadhunain@gmail.com[*]
Submitted: **11/06/2026**; Accepted: **24/06/2026**; Published: **30/06/2026**

Abstract− Diabetes mellitus is a chronic disease that requires early detection to reduce the risk of severe complications. However, machine learning-based diabetes prediction is often affected by class imbalance and high-dimensional data. This study investigates the effectiveness of integrating Synthetic Minority Over-sampling Technique (SMOTE) and Principal Component Analysis (PCA) for diabetes prediction. A total of 80,437 records from a Kaggle diabetes dataset were processed using the Knowledge Discovery in Databases (KDD) framework. Six machine learning algorithms, namely Random Forest, XGBoost, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naïve Bayes, and Neural Network, were evaluated using train-test split ratios of 70:30, 80:20, and 90:10. Performance was measured using accuracy, precision, recall, and F1-score. Without oversampling, XGBoost consistently achieved the highest accuracy across all split ratios, peaking at 94.04% at the 80:20 ratio; however, recall for the minority (diabetic) class remained substantially lower than for the majority class, indicating that high overall accuracy masked weaker detection of actual diabetes cases. After applying SMOTE, overall accuracy declined across all models (e.g., XGBoost fell to 87.52% at 80:20), but minority-class recall improved markedly, indicating a more balanced classification between classes at the cost of overall accuracy. Notably, at the 80:20 split, the Neural Network achieved a marginally higher accuracy (87.67%) than XGBoost under SMOTE, although XGBoost remained the top performer at the 70:30 and 90:10 ratios, suggesting that its advantage under class-balanced conditions is not uniform across split ratios. PCA was applied to reduce data dimensionality and did not substantially affect predictive performance; however, the present results do not include quantitative evidence, such as the change in feature count or computation time, needed to substantiate claims about its contribution to efficiency. These findings suggest that XGBoost with an 80:20 split is the most effective configuration when class imbalance is not addressed, while the application of SMOTE narrows the performance gap between models and shifts the trade-off toward more balanced, rather than purely accuracy-maximizing, classification.

Keywords: Diabetes Prediction; Machine Learning; SMOTE; PCA; Classification.

1. INTRODUCTION

As one of the most prevalent non-communicable diseases worldwide, diabetes mellitus continues to pose a serious public health challenge, with its growing prevalence placing considerable strain on mortality figures and healthcare expenditure. When diagnosis is delayed, the consequences can be severe, ranging from cardiovascular disorders and kidney failure to stroke, nerve damage, and impaired vision. For this reason, identifying diabetes at an early stage plays a vital role in enabling preventive action and improving patient outcomes. Healthcare practitioners have increasingly turned to machine learning in recent years, owing to its ability to uncover hidden patterns within clinical data and generate reliable predictions. Yet the performance of diabetes prediction models is commonly constrained by two interrelated obstacles: imbalanced data distributions and overly complex, high-dimensional feature sets. When a dataset is skewed toward one class, learning algorithms tend to favor the majority group, which weakens their ability to correctly flag actual diabetes cases. A surplus of features, meanwhile, tends to inflate computational demands, introduce redundancy, and ultimately erode classification quality. To confront these dual obstacles, this study proposes combining the Synthetic Minority Over-sampling Technique (SMOTE) with Principal Component Analysis (PCA) within the diabetes prediction pipeline. Whereas SMOTE works to rebalance class distribution by generating synthetic instances of the underrepresented class, PCA is directed at compressing feature dimensionality and discarding redundant information. Together, these two techniques are anticipated to strengthen the accuracy, efficiency, and robustness of machine learning models applied to diabetes prediction.

A considerable body of prior work has explored how machine learning can be applied to diabetes prediction. Using the Pima Indian Diabetes Dataset, [1] benchmarked several algorithms and reported that Logistic Regression and Support Vector Machine delivered strong predictive results, while Neural Networks reached an accuracy of 88.6%; however, their work stopped short of tackling class imbalance or applying any oversampling strategy. In a related investigation, [2] surveyed a range of machine learning approaches to diabetes prediction and pointed out that many existing models are held back by limited dataset diversity, weak feature selection, and poor generalizability across different populations. [3], by contrast, proposed an ensemble framework built on AdaBoost, Bagging, and Random Forest for early diabetes detection, with Random Forest reaching 97% classification accuracy; this work, however, was largely confined to ensemble strategies and did not venture into dimensionality reduction approaches such as PCA. [4] developed an interpretable clinical decision support system that paired feature selection with SMOTE to forecast gestational diabetes, demonstrating that balancing the data could lift

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9903

predictive performance, yet stopping short of combining SMOTE with PCA to further refine the feature space. [5] drew on biomarker data together with ensemble methods to predict gestational diabetes and achieved encouraging results, though the focus there rested on biomarker selection and ensemble design rather than the joint treatment of dimensionality reduction and class imbalance. Most recently, [6] benchmarked Logistic Regression, AdaBoost, and Naïve Bayes for diabetes classification and reported accuracy levels above 90%, yet this comparison did not extend to examining how oversampling or feature reduction might influence the outcomes.

Collectively, the existing literature reveals a research gap that can be analyzed from two primary perspectives. First, studies on SMOTE, such as [4], demonstrate that rebalancing the class distribution improves predictive performance, particularly for the minority class. However, these investigations have not integrated SMOTE with dimensionality reduction techniques such as principal component analysis (PCA), leaving the combined impact of both methods unexamined. Second, comparative analyses of machine learning algorithms, including [1], [2], [3], and [6], primarily assess model performance across various classifiers. These studies generally do not address class imbalance through oversampling or evaluate the influence of feature reduction on predictive performance.

Beyond these two strands, none of the reviewed studies have simultaneously examined the role of PCA, the influence of SMOTE, and a comprehensive comparison of classification algorithms within a unified experimental design. The algorithms evaluated in this study represent several learning paradigms, including ensemble-based methods such as Random Forests and XGBoost, kernel-based methods such as SVMs, instance-based methods such as KNN, probabilistic classification methods such as Naïve Bayes, and neural networks. As a result, the existing literature still provides limited evidence on whether improvements in diabetes prediction are primarily due to class balancing, dimensionality reduction, or the intrinsic differences among algorithms. The present study sets out to close this gap by disentangling and evaluating, both separately and jointly, the contributions of SMOTE and PCA across six machine learning algorithms, thereby shedding light on the relative weight each component carries in diabetes prediction performance.

This study pursues five specific aims: (1) establishing the baseline performance of diabetes prediction models in the absence of any optimization technique; (2) examining how SMOTE addresses class imbalance and influences classification performance; (3) determining how effectively PCA reduces feature dimensionality while improving model efficiency; (4) benchmarking Random Forest, XGBoost, SVM, KNN, Naïve Bayes, and Neural Network against one another under optimized conditions; and (5) pinpointing the machine learning model that performs best for diabetes prediction overall. The anticipated outcome is a contribution toward more accurate and efficient diabetes prediction systems, alongside meaningful insight into how SMOTE and PCA can be jointly applied within healthcare-oriented machine learning research.

2. RESEARCH METHODOLOGY

2.1 Research Stages

This research employs the Knowledge Discovery in Databases (KDD) framework to enhance diabetes prediction through the integrated application of machine learning algorithms, Principal Component Analysis (PCA), and the Synthetic Minority Over-sampling Technique (SMOTE). Figure 1 illustrates the revised sequence of research steps, beginning with data collection and concluding with model evaluation. The research commences with the data collection phase, utilizing the Diabetes Prediction Dataset acquired from Kaggle. Following data collection, a preprocessing phase is undertaken to improve data quality prior to subsequent analysis. This phase involves removing duplicate records, restricting the dataset to individuals aged 17 years and above, and correcting inconsistent entries in the gender attribute. These procedures ensure that the dataset used in this study is relevant, internally consistent, and suitable for further processing.

After preprocessing, the dataset was transformed by creating a new feature, age_category, from the existing age attribute. This feature was used to group individuals into several age ranges, such as <30, 30–45, 46–60, and >60. The categorical attributes, namely gender, smoking_history, and age_category, were then encoded into numerical form. One-Hot Encoding was applied to the nominal attributes, namely gender and smoking_history, while Ordinal Encoding was applied to age_category because it represents an ordered category. This encoding process does not rely on the target variable or dataset-wide statistics; therefore, it does not cause information leakage and can be performed before the train-test split.

To prevent data leakage, the dataset was split into training and test subsets after encoding, before applying dimensionality reduction or class balancing. Three split-ratio scenarios were examined: 70:30, 80:20, and 90:10. These were used to evaluate the robustness of the classification models across different training data proportions. For each scenario, stratified splitting was applied using the stratify=y parameter in the train_test_split function to maintain the original proportion of diabetic and non-diabetic cases in both subsets. In addition, a fixed random seed, such as random_state=42, was used to ensure reproducibility.

Once the training and testing subsets are established, Principal Component Analysis (PCA) is applied for dimensionality reduction. PCA is fitted exclusively on the training data; the resulting transformation is then used to project the testing data, without re-fitting PCA on the test set. PCA is configured to retain [specify the number

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9903

of components or cumulative explained-variance threshold, e.g., n_components = X, retaining Y% of variance], reducing the feature space from [original number of features] to [resulting number of components] dimensions while preserving most of the variance in the data. This step is intended to reduce computational complexity and limit the influence of redundant or correlated features on classification performance.

Class imbalance is then addressed using the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic samples for the minority (diabetic) class via the k-nearest-neighbor method. Consistent with the goal of preventing data leakage, SMOTE is applied only to the training subset, after the train-test split and after the PCA transformation; the testing subset is left in its original, unaltered class distribution, so that evaluation reflects real-world class proportions. SMOTE is configured with [specify k_neighbors, e.g., k = 5] and a sampling strategy of [specify, e.g., 1:1 balance between minority and majority classes]. Two experimental conditions are then compared at each split ratio: a "normal data" condition using the original, imbalanced training data, and an "oversampled data" condition using the SMOTE-balanced training data. In both conditions, the testing subset is identical and untouched by SMOTE, ensuring a fair comparison between the two conditions.

The classification stage employs six machine learning algorithms: Random Forest, XGBoost, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naïve Bayes, and Neural Network, configured as follows: Random Forest [specify n_estimators, max_depth, criterion, etc.]; XGBoost [specify n_estimators, learning_rate, max_depth, etc.]; SVM [specify kernel, C, gamma]; KNN [specify number of neighbors and distance metric]; Naïve Bayes [specify variant, e.g., Gaussian Naïve Bayes]; and Neural Network [specify number of hidden layers, neurons per layer, activation function, optimizer, learning rate, epochs, and batch size]. All algorithms are trained on the same training subset, either normal or SMOTE-balanced, and in either the original or PCA-reduced feature space depending on the scenario, and evaluated on the corresponding held-out testing subset.

Model performance was evaluated using the confusion matrix and four standard evaluation metrics: accuracy, precision, recall, and F1-score. These metrics were calculated for both diabetic and non-diabetic classes to identify possible residual imbalance effects after applying SMOTE. The results from each experimental scenario were then compared across data split ratios (70:30, 80:20, and 90:10), balancing conditions (original and SMOTE-oversampled), and feature spaces (original and PCA-reduced). This comparison was conducted to identify the most effective and reproducible configuration and to examine the individual and combined contributions of PCA and SMOTE to diabetes prediction performance.

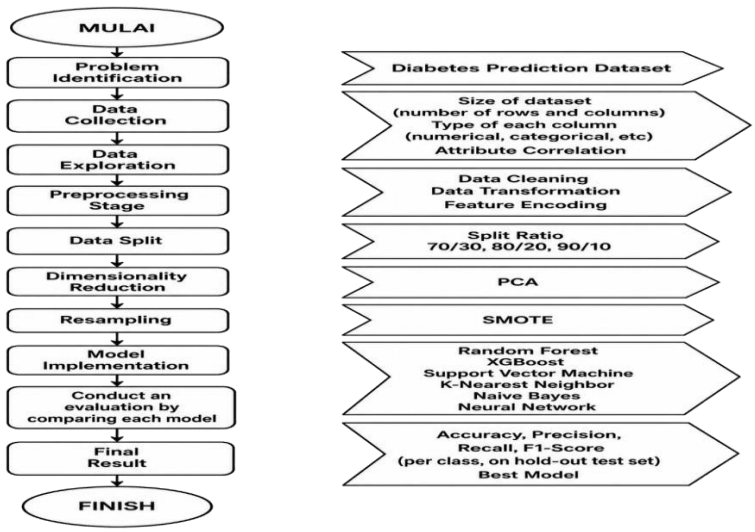


Figure 1. Research Workflow for Diabetes Prediction Using SMOTE and PCA

2.2 Dataset

The dataset used in this study was obtained from the Kaggle platform and contains patient information comprising several attributes, namely gender, age, hypertension, heart disease, smoking history, BMI, HbA1c level, blood glucose level, and diabetes as the class variable. The dataset can be accessed via Kaggle. To align with the study's requirements, data filtering was performed by including only patients aged 17 to 80 years, resulting in a total of 80,437 data points used in the analysis. A description of each attribute is presented in Table 1. The use of relevant and high-quality datasets is crucial as it can influence the performance and generalization ability of the resulting machine learning models [7].

Table 1. Dataset Table

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9903

	gender	age	*hypertension*	*heart_disease*	*smoking_history*	bmi	*HbA1c_level*	*blood_glucose_level*	diabetes
0	Female	80.0	0	1	never	25.19	6.6	140	0
1	Female	54.0	0	0	No Info	27.32	6.6	80	0
…	…	…	…	…	…	…	…	…	…
99998	Female	24.0	0	0	never	35.42	4.0	100	0
99999	Female	29.0	0	0	current	22.43	6.6	90	0

2.3 Data Preprocessing

Data preprocessing is the stage of preparing data prior to the machine learning model development process. During this stage, the data is processed to ensure it is in the appropriate format and ready for use in analysis or classification, so that the model can function optimally [8].

2.3.1 Data Cleaning

Data cleaning is the process of identifying and correcting data that is corrupted, duplicated, incomplete, inaccurate, or contains noise before it is used in the analysis process. This step is performed to ensure that the data used in building machine learning models represents valid information that accurately reflects real-world conditions, thereby preventing errors in the model training process [9].

2.3.2 Data Transformation

Data transformation is the process of changing the form or format of data to meet the requirements of machine learning algorithms. This step can be performed by converting numerical data into categorical data or vice versa to ensure data compatibility with the analysis methods used [10]. In this study, data transformation was performed by creating an age category feature based on the age attribute and converting categorical data into numerical data using encoding techniques so that it could be processed by the classification algorithm used.
The formula for equal-width discretization, which is one of the data transformation techniques, is as follows:

$$X\_normalisasi = \frac{(X - Xmin)}{(Xmax - Xmin)} \quad (1)$$

2.4 Splitting Data

The data splitting stage was performed by dividing the dataset into training data and testing data using three ratio cases: 70:30, 80:20, and 90:10. The use of multiple splitting ratios aims to evaluate the impact of the training data proportion on the model's ability to generalize, given that there is no single data splitting configuration that is always optimal for every dataset characteristic [15]. Based on the test results, the 80:20 ratio on data without oversampling yielded the best performance with an accuracy of 94.04% using the XGBoost algorithm, higher than the 70:30 and 90:10 cases; therefore, this ratio was selected as the most optimal configuration in this study.

2.5 Reduction Dimention

Dimension reduction is a technique used to reduce the number of features in a dataset while retaining the most representative information. This technique plays a role in addressing the curse of dimensionality, a condition where an excessive number of features can increase computational complexity and complicate the data analysis process [11] . In this study, dimensionality reduction was performed using the Principal Component Analysis (PCA).

2.5.1 PCA

In this study, Principal Component Analysis (PCA) was used to reduce the dimensionality of the diabetes dataset prior to the classification process. PCA transforms correlated attributes into a set of principal components that can represent the information from the original data. PCA was used to identify the most representative information in the dataset so that the number of features could be reduced without losing important data characteristics [12]

2.6 Resampling Data

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9903

An imbalanced distribution of data across certain classes, such as the diabetes and smoking history variables, can affect the model's ability to recognize patterns within each class. This condition causes the model to focus more on classes with larger data sets, resulting in suboptimal classification performance for minority classes. To address this issue, this study applies a resampling technique using the Synthetic Minority Over-sampling Technique (SMOTE). This method is used to balance the data distribution by generating synthetic samples in the minority class before the machine learning model is built [13]

2.6.1 SMOTE

The Synthetic Minority Over-sampling Technique (SMOTE) is an oversampling method used to balance class distributions by generating synthetic samples in the minority class. This method works by generating new data based on the characteristics of the existing minority data, thereby making the data distribution more balanced and enabling it to be used more effectively in the classification process [14]

2.7 Model Implementation

The model development phase was conducted after the data underwent preprocessing, resampling using SMOTE, and dimensionality reduction using PCA. In this study, diabetes prediction was formulated as a two-class classification problem: diabetes and non-diabetes. The processed data was then used to train the K-Nearest Neighbor (KNN), Naive Bayes, Support Vector Machine (SVM), XGBoost, and Neural Network algorithms to identify patterns related to diabetes status and determine the model with the best performance [16]

2.7.1 Random Forest

Random Forest is an ensemble learning algorithm that builds a number of decision trees from subsets of data and features selected at random. The final prediction is determined based on the majority vote of all the trees created. In addition to performing classification, Random Forest can also identify the features that most influence the model, thereby helping to reduce data complexity [17]. In this study, Random Forest was used to predict diabetes status based on data that had undergone preprocessing, SMOTE, and PCA.

2.7.2 XGBoost

XGBoost (eXtreme Gradient Boosting) is a boosting algorithm that builds models incrementally by correcting prediction errors from previous iterations. The algorithm's ability to integrate multiple predictors and learn nonlinear relationships among features allows the information contained in the data to be utilized more effectively than models that rely on linear relationships between variables [18]

2.7.3 Support Vector Method

Support Vector Machines (SVMs) are classification algorithms that work by constructing optimal hyperplanes to separate data into different classes. In this study, SVMs were used to examine the relationship between various diabetes risk factors, such as age, BMI, HbA1c levels, blood glucose levels, and other health histories. The ability of SVMs to construct optimal decision boundaries allows patterns related to diabetes status to be identified more effectively. This approach aligns with the use of machine learning in the field of diabetes, which supports decision-making based on patterns derived from health data [19]

2.7.4 K-Nearest Neighbor

K-Nearest Neighbor (KNN) is a classification algorithm that determines the class of a data point based on its proximity to a set of its nearest neighbors. In this study, KNN is used to identify patterns of diabetes status based on health attributes that have undergone preprocessing, SMOTE, and PCA. KNN performance is influenced by feature representation and the measurement of distances between data points; therefore, dimensionality reduction using PCA is expected to produce a more representative feature space in the classification process [20]

2.7.5 Naïve Bayes

In this study, the Naïve Bayes classifier was used to analyze the relationship between various diabetes risk factors, such as age, hypertension, history of heart disease, BMI, HbA1c levels, and blood glucose levels. The classification process was performed by calculating the probability of each attribute relative to diabetes status, so that the contribution of each factor could be taken into account in the prediction process. This approach allows diabetes risk patterns to be identified based on the combination of health characteristics possessed by each individual [21]

2.7.6 Neural Network

A neural network processes data through a number of interconnected neurons in the input layer, hidden layer, and output layer. During the training process, the weights of each connection are updated iteratively to learn the relationships between attributes that contribute to diabetes status. In this study, a neural network was used to process various health risk factors simultaneously so that complex patterns among features could be better

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9903

identified. This capability allows the information contained in attributes such as age, BMI, HbA1c levels, blood glucose levels, hypertension, and history of heart disease to be utilized more optimally in the diabetes classification process [22

2.8 Model Evaluation

The next step is to evaluate the performance of all models in detecting diabetes using the prepared dataset. This evaluation aims to provide a clearer picture of each model's ability to identify cases of diabetes. Some of the evaluation methods used in this study include

a. *Confusion Matrix*
A confusion matrix is a 2 × 2 table used in binary classification to summarize the performance of a model, categorizing predictions into true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) [23]. The *Confusion Matrix* is shown in Table 2.

Table 2. *Confussion Matrix*

	Prediksi Positif	**Prediksi Negatif**
Aktual Positif	*True Positive* (TP)	*True Negative* (TN)
Aktual Negatif	*False Positive* (FP)	*False Negative* (FN)

b. *Accuracy*
Accuracy measures the proportion of correctly classified predictions out of the total evaluated data. This metric provides an overall picture of the model's accuracy in performing classification[24]. Equation (1) is used to calculate accuracy.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

c. *Precision*
Precision is a classification metric that measures the ratio of true positive predictions to the total predicted positives. It is crucial for evaluating model performance, especially in cases where false positives carry significant consequences, ensuring reliability in machine learning applications [25]. Equation (2) is used to calculate Precision.

$$Precision = \frac{TP}{TP+FP} \quad (2$$

d. *Recall*
Recall in model evaluation measures the ability to identify true positive cases among all actual positives. In the context of traffic accident severity detection, prioritizing recall helps minimize the risk of missing high-risk crashes, which is crucial for safety. Equation 3 is used to calculate Recall.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

e. *F1-Score*
The F1 score is a performance metric for binary classification models, balancing precision and recall. However, it varies with the positive class ratio in training data, necessitating calibration for fair comparisons, as discussed in the study. Formula (4) is used to calculate F-1 Score

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision \times Recall} \quad (4)$$

3. RESULT AND DISCUSSION

Several sequential stages preceded the actual processing of the dataset in this research. Each step of data handling and analysis was captured visually, allowing the changes occurring at every stage to be tracked and illustrated.

3.1 Data Preprocessing

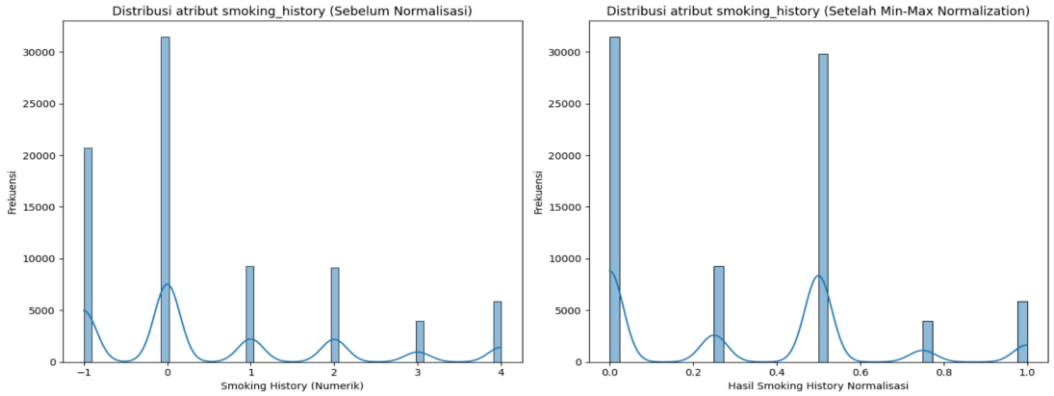
Two procedures take place during this phase: cleaning the data and transforming it. Cleaning is carried out to confirm the absence of null entries and duplicate records within the dataset. No missing values were found in this dataset; however, a check for duplicate entries revealed 3,854 such records, which once removed left 96,146 unique entries. The dataset was then narrowed further by retaining only records corresponding to individuals between 17 and 80 years of age, yielding a final count of 80,437 records. The data transformation phase follows, during which certain attribute values are converted into numerical form. Specifically, Minimum–Maximum scaling was applied to the smoking_history attribute so that its values would fall within a consistent range relative to the other features. Figure 2 and Table 3 present the outcome of this normalization step.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9903

Table 3. Minimum–Maximum Range for the Smoking_History Attribute

	Smoking_history	*Smoking_numeric*	*Smoking_normalized*
1	*Never*	0	0.0
2	*No info*	2	0.5
…	…	…	…
99998	*Never*	0	0.0
99999	*Current*	2	0.5

Min-Max Normalization was applied to rescale the smoking_history attribute into a range between 0 and 1. As shown in Figure 2, the normalization process changes the numerical scale of the attribute while maintaining its distribution pattern. This preprocessing step is important to improve data consistency before the model training stage.



(a) Before (b) After

Figure 2. Normalization: (a) Before and (b) After

3.2 Data Resampling, Data Reduction and Data Splitting.

To address the unequal distribution of classes within the dataset, this research incorporates the Synthetic Minority Over-sampling Technique (SMOTE), while Principal Component Analysis (PCA) is used to condense the dataset's information into a reduced set of representative components. The extent of this dimensionality reduction achieved through PCA is summarized numerically in Table 4.

Table 4. Dimensionality Reduction Summary

Metric	Value
Number of original features (input to PCA)	9 (gender, age, hypertension, heart_disease, bmi, HbA1c_level, blood_glucose_level, age category, smoking_numeric)
Number of principal components retained	2
Reduction in dimensionality	77.8% (9 → 2 features)
Cumulative explained variance retained	63.4%

To determine the influence of data balancing on model performance, experiments were conducted under two conditions: with SMOTE-based oversampling and without SMOTE. In addition, the training and testing data proportions were varied using three configurations: 70:30, 80:20, and 90:10. Six classification algorithms were evaluated in this study, namely Random Forest, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), XGBoost, Naïve Bayes, and Neural Network.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9903

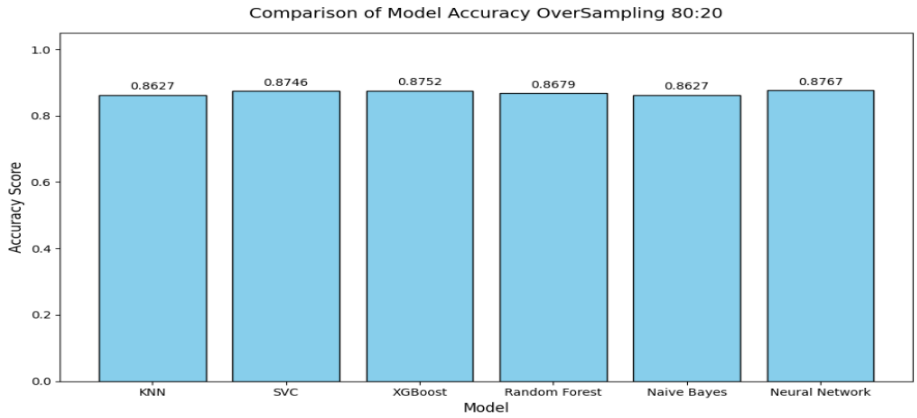


Figure 3. Comparison of Model Accuracy OverSampling 80:20

Figure 4 presents model performance metrics under oversampling using an 80:20 split for comprehensive model evaluation.

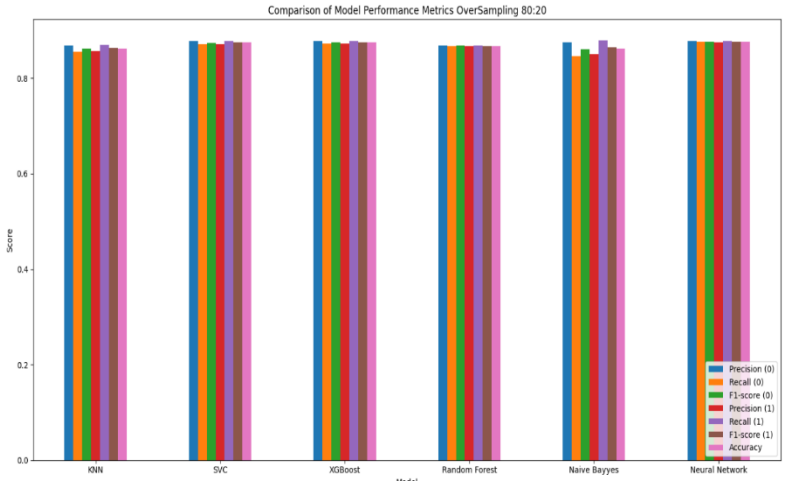


Figure 4. Comparison of Model Performance Metrics Over Sampling, 80:20 Ratio

The model accuracy under the no-oversampling condition with an 80:20 split is presented in Figure 5.

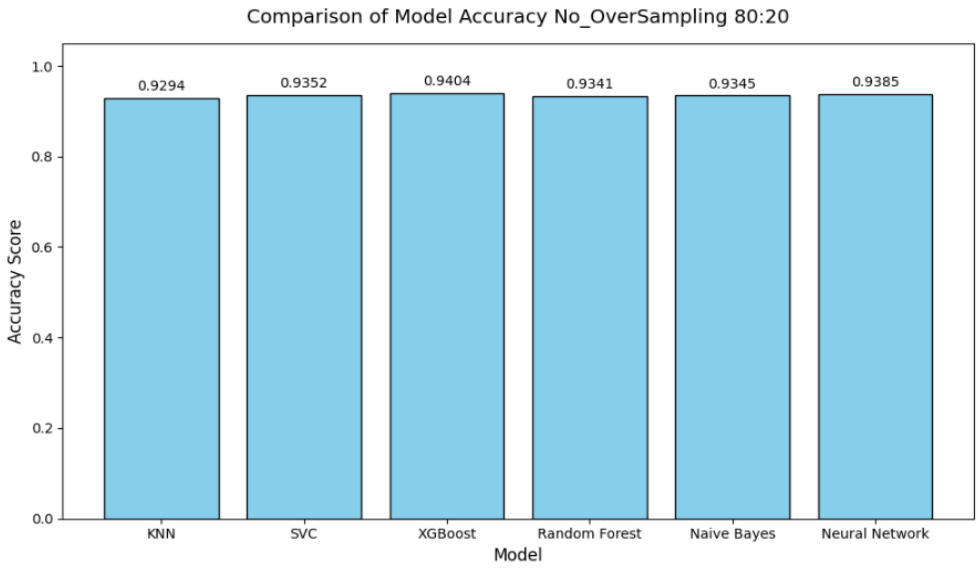


Fig 5. Comparison of Model Accuracy with No Over Sampling 80:20 Ratio

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9903

The performance metrics of each model under the no-oversampling condition with an 80:20 split are shown in Figure 6.

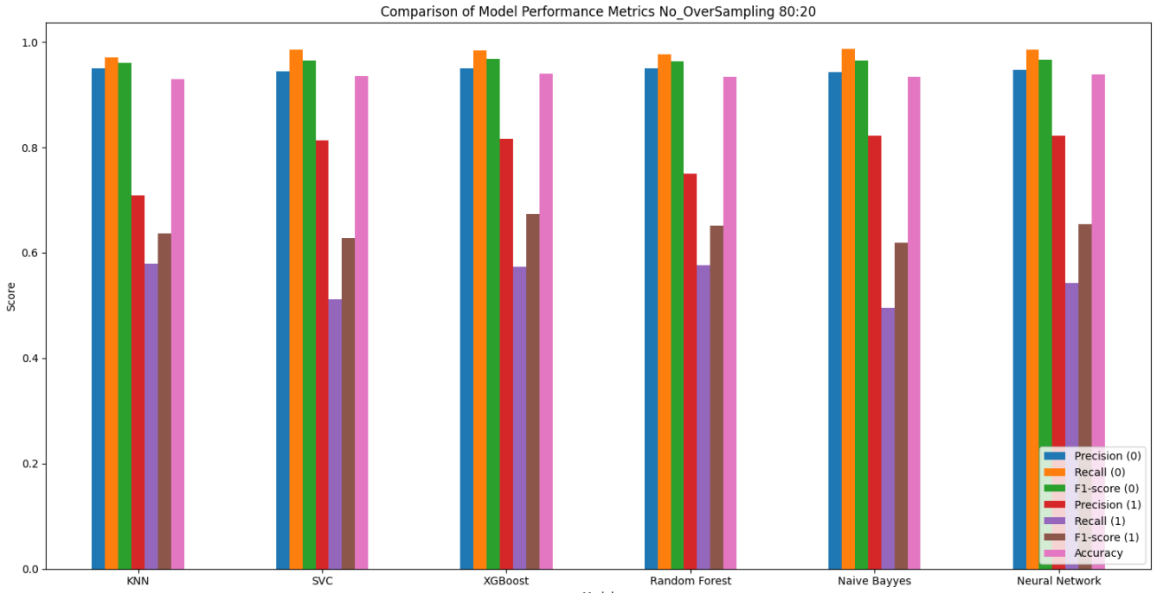


Fig 6. Comparison of Model Performance Metrics with No Over Sampling 80:20 Ratio

At the 80:20 split, results obtained under the oversampling condition show accuracy figures spanning from 86.27% (recorded by both KNN and Naïve Bayes) up to 87.67% (recorded by the Neural Network). The Neural Network emerged as the top performer at this ratio, with XGBoost (87.52%) and SVM (87.46%) trailing closely behind. Given that only 0.15 percentage points separate the Neural Network from XGBoost, it appears that once the classes have been balanced, several algorithms reach a comparable level of performance rather than any single model standing out decisively. To move beyond accuracy alone, Table 7 lays out precision, recall, and F1-score figures for each class individually, lending concrete support to the assertion that SMOTE contributes to a more even balance between classes.

By contrast, when oversampling was not applied, accuracy at the same 80:20 ratio climbed to between 92.94% and 94.04%. Here, XGBoost again led with 94.04%, trailed by the Neural Network at 93.85% and SVM at 93.52%. Yet this headline accuracy figure obscures a sizable disparity at the class level: of the 1,721 diabetic cases present in the test set, the XGBoost model, without oversampling, correctly flagged only 986 (a recall of 0.57), leaving 735 genuine diabetic cases undetected despite its otherwise impressive overall accuracy. Table 7 lays this disparity out in full.

Across this investigation, the 80:20 split consistently yields the strongest outcomes under both experimental conditions, making it the optimal configuration overall. Where accuracy is the sole criterion, the combination of XGBoost and no oversampling at this ratio stands out, delivering 94.04%. Once class-wise balance enters the equation, though, the picture shifts: applying SMOTE at the same 80:20 ratio produces results that better reflect the practical demands of diabetes prediction, a context in which overlooking an actual diabetic case is far costlier than raising a false alarm.

3.3 Cross-Models Testing Results

Table 5 presents the accuracy comparison of six classification models under the oversampling condition across three data split ratios: 70:30, 80:20, and 90:10. This comparison is used to evaluate the consistency of each model after applying SMOTE.

Table. 5 Oversampling Accuracy

Ratio	KNN	SVM	XGBoost	Random Forest	NaiveBayyes	Neural Network
70:30	0.86	0.87	0.87	0.86	0.86	0.87
80:20	0.86	0.87	0.87	0.86	0.86	0.87
90:10	0.86	0.87	0.87	0.87	0.86	0.87

Table 6 shows the accuracy comparison of the classification models without applying oversampling. The results are presented across three data split ratios to examine model performance under the original class distribution.

Table 6. No Over Sampling Accuracy

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9903

Ratio	KNN	SVM	XGBoost	Random Forest	Naive Bayes	Neural Network
70:30	0.92	0.93	0.93	0.93	0.93	0.93
80:20	0.92	0.93	0.94	0.93	0.93	0.93
90:10	0.92	0.93	0.93	0.93	0.93	0.93

Table 7 presents the per-class performance evaluation of each model at the 80:20 split ratio without oversampling. The evaluation includes precision, recall, and F1-score for both class 0 and class 1 to assess the classification balance of each model.

Table 7. Per-Class Precision, Recall, and F1-score at the 80:20 Split Ratio (No Oversampling)

Model	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)
KNN	0.95	0.97	0.96	0.71	0.58	0.64
SVM	0.94	0.99	0.96	0.81	0.51	0.63
XGBoost	0.95	0.98	0.97	0.82	0.57	0.67
Random Forest	0.95	0.98	0.96	0.75	0.58	0.65
Naïve Bayes	0.94	0.99	0.96	0.82	0.50	0.62
Neural Network	0.95	0.99	0.97	0.82	0.54	0.65

Examining the results overall, XGBoost emerges as the strongest performer under the non-oversampled condition regardless of split ratio, posting 0.9397, 0.9404, and 0.9393 at the 70:30, 80:20, and 90:10 ratios respectively. Its dominance is less consistent, however, once oversampling is introduced: while it leads at the 70:30 (0.8739) and 90:10 (0.8786) ratios, the Neural Network edges it out at 80:20, posting 0.8767 against XGBoost's 0.8752. Such a slim margin — under half a percentage point —makes clear that accuracy alone cannot reliably crown a single winner once the classes have been balanced.

The decrease in accuracy of approximately six to seven percentage points after applying SMOTE, such as from 94.04% to 87.52% for XGBoost at the 80:20 ratio, reflects an inherent trade-off rather than a limitation of the technique itself. By rebalancing the training data, SMOTE pushes the model's decision boundary to classify a greater number of borderline instances as diabetic. The consequence is twofold: true positives among the minority class rise, lifting recall for class 1, while false positives among the majority class also climb, dragging down overall accuracy. Because non-diabetic cases make up 89.5% of the dataset, even a modest uptick in false positives within that majority group is enough to produce a sizable dip in the accuracy figure. From this perspective, the main issue is not why accuracy decreases after SMOTE is applied, but whether the improvement in minority-class recall outweighs the loss in majority-class precision in this clinical application. This comparison can be further supported by Table 7 once the SMOTE-based results are completed.

Overall, these results suggest that XGBoost without oversampling at the 80:20 split achieved the highest raw accuracy in this study. However, SMOTE produced a more balanced classification performance and may offer greater clinical relevance, despite a measurable decrease in overall accuracy.

4. CONCLUSION

This study evaluated the impact of integrating Synthetic Minority Over-sampling Technique (SMOTE) and Principal Component Analysis (PCA) on diabetes prediction using six machine learning algorithms, namely KNN, SVM, XGBoost, Random Forest, Naïve Bayes, and Neural Network. Experiments were conducted using three train-test split ratios (70:30, 80:20, and 90:10) under both oversampling and non-oversampling conditions. The findings consistently showed that XGBoost achieved the best overall performance among all evaluated models. The highest accuracy was obtained by XGBoost without oversampling at the 80:20 split ratio, reaching 94.04%, while the best result under the oversampling scenario was 87.52%. The results indicate that the 80:20 data split ratio provided the most effective balance between training and testing data, yielding the highest predictive performance across most experimental settings. Although the application of SMOTE improved the balance of classification performance between majority and minority classes, as reflected by more consistent precision, recall, and F1-score values, it did not improve overall accuracy. In fact, all models achieved higher accuracy when trained on the original non-oversampled data. Therefore, the contribution of SMOTE in this study should be interpreted primarily as improving class balance rather than enhancing predictive accuracy. Similarly, PCA successfully reduced data dimensionality and contributed to a more efficient learning process without causing a substantial decline in model performance. However, the present study did not include a direct comparison between models with and without PCA; therefore, the extent of PCA's contribution to predictive improvement cannot be conclusively determined. Several limitations should be considered when interpreting these findings. First, the study

relied on a single publicly available Kaggle dataset, which may not fully represent broader clinical populations. Second, model evaluation was limited to internal train-test split validation, and no external validation dataset was employed to assess generalizability. Third, despite the use of SMOTE to address class imbalance, the original dataset remained inherently imbalanced, which may introduce bias in model learning and performance evaluation. Future studies should incorporate multi-source clinical datasets, perform external validation, and explore additional imbalance-handling techniques and feature engineering approaches to improve the robustness and generalizability of diabetes prediction models. Overall, XGBoost with an 80:20 data split ratio and without oversampling emerged as the most effective configuration for diabetes prediction in this study.

REFERENCES

[1] J. J. Khanam and S. Y. Foo, "A comparison of machine learning algorithms for diabetes prediction," *ICT Express*, vol. 7, no. 4, pp. 432–439, 2021, doi: 10.1016/j.icte.2021.02.004.
[2] V. Jaiswal, A. Negi, and T. Pal, "A review on current advances in machine learning based diabetes prediction," *Prim. Care Diabetes*, vol. 15, no. 3, pp. 435–443, 2021, doi: 10.1016/j.pcd.2021.02.005.
[3] U. e. Laila, K. Mahboob, A. W. Khan, F. Khan, and W. Taekeun, "An Ensemble Approach to Predict Early-Stage Diabetes Risk Using Machine Learning: An Empirical Study," *Sensors*, vol. 22, no. 14, pp. 1–15, 2022, doi: 10.3390/s22145247.
[4] Y. Du, A. R. Rafferty, F. M. McAuliffe, L. Wei, and C. Mooney, "An explainable machine learning-based clinical decision support system for prediction of gestational diabetes mellitus," *Sci. Rep.*, vol. 12, no. 1, pp. 1–14, 2022, doi: 10.1038/s41598-022-05112-2.
[5] Y. N. Chan *et al.*, "A machine learning approach for early prediction of gestational diabetes mellitus using elemental contents in fingernails," *Sci. Rep.*, vol. 13, no. 1, pp. 1–11, 2023, doi: 10.1038/s41598-023-31270-y.
[6] M. M. Mijwil and M. Aljanabi, "A Comparative Analysis of Machine Learning Algorithms for Classification of Diabetes Utilizing Confusion Matrix Analysis," *Baghdad Sci. J.*, vol. 21, no. 5, pp. 1712–1728, 2024, doi: 10.21123/BSJ.2023.9010.
[7] Y. Gong, G. Liu, Y. Xue, R. Li, and L. Meng, "A survey on dataset quality in machine learning," *Inf. Softw. Technol.*, vol. 162, no. September 2022, p. 107268, 2023, doi: 10.1016/j.infsof.2023.107268.
[8] C. Y. Chou, D. Y. Hsu, and C. H. Chou, "Predicting the Onset of Diabetes with Machine Learning Methods," *J. Pers. Med.*, vol. 13, no. 3, 2023, doi: 10.3390/jpm13030406.
[9] P. O. Côté, A. Nikanjam, N. Ahmed, D. Humeniuk, and F. Khomh, "Data cleaning and machine learning: a systematic literature review," *Autom. Softw. Eng.*, vol. 31, no. 2, 2024, doi: 10.1007/s10515-024-00453-w.
[10] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data," *Front. Energy Res.*, vol. 9, no. March, pp. 1–17, 2021, doi: 10.3389/fenrg.2021.652801.
[11] W. Jia, M. Sun, J. Lian, and S. Hou, "Feature dimensionality reduction: a review," *Complex Intell. Syst.*, vol. 8, no. 3, pp. 2663–2693, 2022, doi: 10.1007/s40747-021-00637-x.
[12] Z. Jayidan, A. M. Siregar, S. Faisal, and H. Hikmayanti, "Peningkatan Akurasi Prediksi Penyakit Jantung Menggunakan Principal Component Analysis (PCA) Pada Algoritma Machine Learning," J. Tek. Inform., vol. 5, no. 3, pp. 821-830, 2024, doi: 10.52436/1.jutif.2024.5.3.2047.
[13] X. Yi, Y. Xu, Q. Hu, S. Krishnamoorthy, W. Li, and Z. Tang, "ASN-SMOTE: a synthetic minority oversampling method with adaptive qualified synthesizer selection," *Complex Intell. Syst.*, vol. 8, no. 3, pp. 2247–2272, 2022, doi: 10.1007/s40747-021-00638-w.
[14] A. Davinka, S. Depari, K. D. Tania, and P. E. Sevtiyuni, "Penerapan Metode Machine Learning dan Teknik SMOTE untuk Prediksi Diabetes," J. Sist. Komput. dan Inform. (JSON), vol. 7, no. 2, pp. 436-447, 2025, doi: 10.30865/json.v7i2.9032.
[15] V. R. Joseph, "Optimal ratio for data splitting," *Stat. Anal. Data Min.*, vol. 15, no. 4, pp. 531–538, 2022, doi: 10.1002/sam.11583.
[16] E. Dritsas and M. Trigka, "Data-Driven Machine-Learning Methods for Diabetes Risk Prediction," *Sensors*, vol. 22, no. 14, 2022, doi: 10.3390/s22145304.
[17] S. Gündoğdu, "Efficient prediction of early-stage diabetes using XGBoost classifier with random forest feature selection technique," *Multimed. Tools Appl.*, vol. 82, no. 22, pp. 34163–34181, 2023, doi: 10.1007/s11042-023-15165-8.
[18] X. Hu, X. Hu, Y. Yu, and J. Wang, "Prediction model for gestational diabetes mellitus using the XG Boost machine learning algorithm," *Front. Endocrinol. (Lausanne).*, vol. 14, no. March, pp. 1–10, 2023, doi: 10.3389/fendo.2023.1105062.
[19] M. A. Makroum, M. Adda, A. Bouzouane, and H. Ibrahim, "Machine Learning and Smart Devices for Diabetes Management: Systematic Review," *Sensors*, vol. 22, no. 5, pp. 1–24, 2022, doi: 10.3390/s22051843.
[20] K. Alnowaiser, "Improving Healthcare Prediction of Diabetic Patients Using KNN Imputed Features and Tri-Ensemble Model," *IEEE Access*, vol. 12, no. February, pp. 16783–16793, 2024, doi:

10.1109/ACCESS.2024.3359760.

[21] A. Hennebelle, H. Materwala, and L. Ismail, “HealthEdge: A Machine Learning-Based Smart Healthcare Framework for Prediction of Type 2 Diabetes in an Integrated IoT, Edge, and Cloud Computing System,” *Procedia Comput. Sci.*, vol. 220, pp. 331–338, 2023, doi: 10.1016/j.procs.2023.03.043.

[22] M. Kowsher, M. Y. Turaba, T. Sajed, and M. M. M. Rahman, "Prognosis and treatment prediction of type-2 diabetes using deep neural network and machine learning classifiers," in Proc. 22nd Int. Conf. Comput. Inf. Technol. (ICCIT), 2019, doi: 10.1109/ICCIT48885.2019.9038574.

[23] D. Chicco and G. Jurman, “An Invitation to Greater Use of Matthews Correlation Coefficient in Robotics and Artificial Intelligence,” *Front. Robot. AI*, vol. 9, no. March, pp. 1–4, 2022, doi: 10.3389/frobt.2022.876814.

[24] D. Chicco and G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,” *BMC Genomics*, vol. 21, no. 1, pp. 1–14, 2020, doi: 10.1186/s12864-019-6413-7.

[25] A. Muniasamy, AI Model Design and Data Management for Disease Prediction. Hershey, PA, USA: IGI Global, 2025, doi: 10.4018/979-8-3373-5137-7.

[26] M. Taz, "Diabetes Prediction Dataset," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset. [Accessed: Jun. 21, 2026]