Received: January 30, 2025. Revised: February 16, 2025.

*International Journal of*
Intelligent Engineering & Systems

http://www.inass.org/

# Investigation of Data Balancing Techniques for Diabetes Prediction

**Ahmad Adel Abu-Shareha[1]*** **Mosleh Abualhaj[2]** **Abdelrahman Hussein[2]**
**Adeeb Al-Saaidah[2]** **Anusha Achuthan[3]**

[1]*Department of Data Science and Artificial Intelligence, Al-Ahliyya Amman University, Amman, 19328, Jordan*
[2]*Department of Networks and Cybersecurity, Al-Ahliyya Amman University, Amman, 19328, Jordan*
[3]*School of Computer Sciences, Universiti Sains Malaysia, Gelugor, Penang 11800, Malaysia*
* Corresponding author's Email: a.abushareha@ammanu.edu.jo

**Abstract:** Diabetes prediction is critical for early intervention and effective disease management. However, the inherent class imbalance in medical datasets, such as the Pima Indians Diabetes dataset, often leads to biased predictions favoring the majority class. This study provides a systematic analysis of various data balancing techniques applied to diabetes prediction, examining their effects on different classifiers and validation techniques. This study uniquely evaluates the impact of prior balancing before data splitting, offering new insights into real-world deployment scenarios. The Pima dataset was used due to its clinical relevance in diabetes prediction and its widespread use as a benchmark, allowing for robust comparison and reproducibility of results. The developed framework consists of preprocessing, data balancing, classification, and evaluation. The performance of different balancing techniques across various classification algorithms and validation techniques was evaluated using accuracy, precision, recall, and F-measure. The results showed that applying data in cross-validation and balancing techniques fails to improve the prediction results, with the accuracy obtained with and without balancing around 90%. The accuracy improved slightly in the train-test percentage split, with the best accuracy of 91%. Finally, when balancing was applied prior to data splitting with cross-validation, the results were improved, as the combined sampling achieved an accuracy of 97.5% and undersampling achieved an accuracy of 94.1%.

**Keywords:** Data balancing, Oversampling, Undersampling, Diabetes prediction.

## 1. Introduction

Diabetes affects the functionality of various body systems, posing serious health risks [1]. This condition is characterized by elevated blood glucose levels, exceeding those in healthy individuals [2]. Glucose, a vital sugar, plays a crucial role in metabolism, providing energy for cells throughout the body [3]. However, when blood glucose levels rise due to insufficient production or ineffective absorption of insulin, serious damage can occur to multiple organs, including the eyes, heart, and kidneys. The global incidence of diabetes is increasing rapidly, as illustrated in Fig. 1, highlighting the urgent need for effective strategies to manage and reduce the risk associated with this life-threatening disease. According to the International Diabetes Federation (IDF), the number of diabetes cases are expected to rise to 783 million by 2045, as illustrated in Fig. 2 [4].

Accurate prediction of diabetes is essential for timely intervention and effective disease management. Early detection plays a crucial role in preventing a wide range of complications associated with diabetes, including heart disease, blindness, vascular problems, stroke, kidney failure, and even limb amputations. The ability to accurately predict diabetes is therefore invaluable, as it can save lives and significantly reduce the disease's impact on patients' health and quality of life. Moreover, early intervention allows for the implementation of tailored treatment plans, which can further slow disease progression and enhance long-term outcomes for individuals at risk [5].

Received: January 30, 2025. Revised: February 16, 2025.

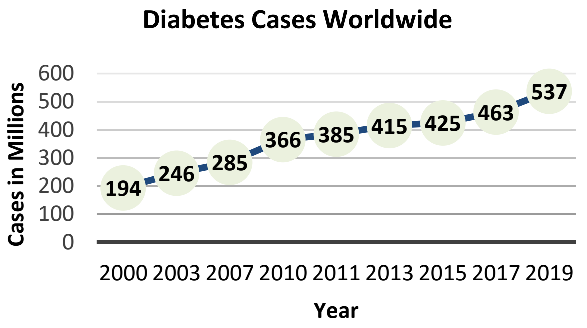

Figure. 1 Number of diabetes cases worldwide 2000-2021 [4]

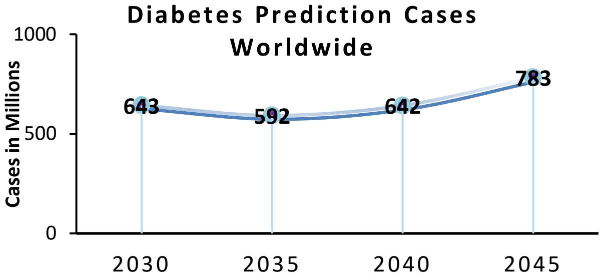

Figure. 2 Predicted number of diabetes cases worldwide 2023-2045 (Age 20-79) according to latest predictions (2013-2021) [4]

Diabetes prediction relies on historical disease data, including tests, examinations, and profile information. The data generated from diabetes-related tests and examinations offers a valuable opportunity to leverage advanced data science and AI techniques. The Pima Indians Diabetes Dataset (PIMA) is one of the well-known diabetes datasets utilized for such predictive modeling. This dataset comprises 768 records and 9 features, including age, body mass index (BMI), insulin levels, and blood pressure. Despite its relatively small size and imbalanced class distribution, the PIMA dataset provides a critical foundation for training machine learning algorithms. This enables the identification of patterns and correlations that improve diabetes prediction accuracy. Additionally, this dataset is often used to develop and test new predictive models that can be generalized to broader populations [6].

Despite its utility, historical data presents several challenges that can hinder the accuracy and effectiveness of predictive models. One major challenge is the presence of missing values in key features such as glucose, blood pressure, skin thickness, insulin, and BMI. These missing values can skew the data analysis and lead to biased or inaccurate model predictions if not handled appropriately. Another significant challenge is class imbalance, where the number of positive cases (diabetic) is significantly lower than negative cases (non-diabetic). This imbalance can bias predictive models toward the majority class, resulting in poor performance in identifying the minority class, which is critical for early detection and intervention. Additionally, the dataset contains features with varying types, scales, and distributions, which can affect the performance of machine learning algorithms. For example, features like age and insulin levels can have wide value ranges, which may lead to model convergence and performance issues if not properly normalized or standardized [7].

Given the challenges posed by missing values, class imbalance, and feature variability in the Pima Indians Diabetes dataset, it is crucial to explore methods that can mitigate these issues to improve model performance. This paper presents a comprehensive comparison of data balancing techniques in diabetes prediction, assessing their impact on multiple machine learning classifiers. Unlike existing studies that primarily focus on classifier performance, this work evaluates the interplay between balancing strategies and different validation methods, providing deeper insights into the effectiveness of each technique.

The primary focus of this paper is to address the class imbalance problem, which can severely affect the accuracy of predictive models by biasing them toward the majority class. Additionally, this work aims to evaluate the effectiveness of various data balancing techniques, oversampling, undersampling, and hybrid methods on classification performance. The objective is to determine which techniques provide the best balance between accuracy and the ability to correctly identify diabetic cases without exacerbating issues such as overfitting or model instability. This study also seeks to assess how different balancing methods influence the performance of multiple machine learning algorithms when applied to the Pima dataset. This study contributes to the systematic analysis of the interaction between balancing techniques (e.g., Synthetic Minority Over-Sampling Technique -SMOTE, Adaptive Synthetic Sampling -ADASYN) and multiple classifiers (e.g., eXtreme Gradient Boosting -XGBoost, Support Vector Machine -SVM) within a diabetes dataset, providing actionable insights for real-world deployment in diabetes prediction systems. The rest of this paper is structured as follows: Section 2 discusses the literature review. Section 3 discusses the proposed framework and the utilized processing components. Section 4 presents the results and evaluation of the proposed framework. Section 5 ends the paper, encapsulating concluding remarks and discussions on future research directions.

## 2. Literature review

In the existing diabetes prediction frameworks, various machine-learning methods have been employed. Karegowda, et al. [8] proposed a framework that uses a Genetic Algorithm (GA) to optimize the weights of a Backpropagation network (BPN). Additionally, feature selection was implemented using Decision Tree (DT) and correlation-based methods. The PIMA dataset was preprocessed to remove the records with missing values, leaving 392 cases for training and testing with a 60-40 split. The results showed that the accuracy of the GA-based framework was 84.7%. Wei, et al. [9] compared the performance of several classifiers, including Deep Neural Network (DNN), Logistic Regression (LR), DT, Naïve Bayesian (NB), and SVM for diabetes prediction. The PIMA dataset was preprocessed by filling in missing values, followed by data transformation and normalization. Feature selection was implemented using Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA). The classification results showed that the DNN achieved the highest accuracy of 77.86% using 10-fold cross-validation.

Kibria, et al. [10] proposed an ensemble method that combines six classifiers using soft voting aggregation. The preprocessing stage includes median-based missing value imputation, SMOTE-based oversampling, feature selection, and normalization. The classifiers used were artificial neural network (ANN), SVM, Random Forest (RF), Adaptive Boosting (AdaBoost), XGBoost, and LR. However, the final ensemble classifier was developed using the two best-performing methods: XGBoost and RF. The results were evaluated using 5-fold cross-validation on the PIMA dataset, with the proposed weighted ensemble model achieving an accuracy of 90% and an F1 score of 89%.

Simaiya, et al. [11] proposed a multistage ensemble classification approach for diabetes prediction. The PIMA dataset was preprocessed by LDA-based dimensionality reduction, while the SMOTE method was employed to address bias during training. The classification methods were organized into three layers: the first layer included NB, K-nearest neighborhood (KNN), and DT; the second layer featured RF and Repeated Incremental Pruning (JRip); and SVM was utilized in the last layer. The results, evaluated using 10-fold cross-validation, showed that the three-layer model achieved a precision of 0.784, a recall of 0.786, and an f-measure of 0.785.

Edeh, et al. [12] proposed a diabetes prediction that uses multiple classifiers. In the preprocessing stage, a k-means clustering algorithm was used to correct data values, serving both to fill in missing values and remove outliers. The classification stage included RF, DT, SVM, and NB. The PIMA dataset was split into an 80%-20% ratio for training and testing. The SVM algorithm achieved the highest accuracy, reaching a value of 83.1%.

Marzouk, et al. [13] developed a model for diabetes prediction by filling in missing values and removing outliers during the preprocessing stage. For classification, the model utilized DT, RF, SVM, Gradient Boosting (GBoost), ANN, KNNm LR, and NB. Using the PIMA dataset and cross-validation evaluation, the ANN achieved the highest prediction accuracy of 81.7%. Chang, et al. [6] compared the performance of three classifiers: NB, RF, and DT. During preprocessing, missing values were filled using the median. PCA, k-means clustering, and importance ranking were used for feature selection. The results using a 70%-30% training-testing split showed that RF achieved the highest accuracy of 86.24% when using the entire feature set.

Yadav and Nilam [14] implemented a normalization for the PIMA dataset and compared the performance of DT, SVM, RF, and KNN classifiers. KNN exhibited the best performance, achieving an accuracy of 80%. Reza, et al. [3] proposed a framework for diabetes prediction that included several preprocessing steps: filling missing data with median value, removing outliers, normalizing the dataset, and addressing class imbalance using SMOTE. For the classification stage, the framework utilized an improved kernel for SVM. The results showed that the enhanced kernel outperformed the traditional kernel, achieving an accuracy of 85.5%, precision of 0.834, recall of 0.87, F1-score of 0.852, and an AUC of 0.855.

Perdana, et al. [7] evaluate the performance of the KNN classifier for diabetes prediction using the PIMA dataset. The preprocessing stage included feature reduction. Various values of *k* were tested in the KNN implementation. The results showed that using a 90%-10% training-testing split with $k$ = 22 achieved an accuracy of 83.12%. Al-Dabbas [15] implemented normalization, filling in missing values, performing outlier imputation, and oversampling. Both SMOTE and SVM-based SMOTE (SVMSMOTE) were implemented for oversampling. For classification, SVM, RF, and XGBoost were utilized. The PIMA dataset was split into a 90%-10% training-testing ratio, with the best results achieved using SVMSMOTE and XGBoost, attaining an accuracy of 91%. A summary of the reviewed literature is given in Table 1.

Table 1. Summary of literature review on diabetes prediction using Pima dataset

| Ref. | Preprocessing | Balance | Acc. | Split |
|---|---|---|---|---|
| [8] | GA for weight adjustments | None | 84.7% | Percentage split using a 60-40 ratio |
| [9] | Filling in missing values, normalization, transformation, and feature selection | None | 77.86 % | 10-fold cross-validation |
| [10] | Filling in missing values, normalization, and feature selection | SMOTE | 90% | 5-fold cross-validation |
| [11] | Feature selection | SMOTE | 0.784 Prec. | 10-fold cross-validation |
| [12] | Filling in missing values and outlier removal | SMOTE | 83.1% | Percentage using an 80-20 ratio |
| [13] | Filling in missing values and normalization | None | 81.7% | 10-fold cross-validation |
| [14] | Normalization | **None** | **80%** | Percentage using a 90-10 ratio |
| [6] | Filling in missing values and feature selection | None | 86.24 %. | Percentage using a 70-30 ratio |
| [3] | Filling in missing values, normalization, outlier removal, and transformation | SMOTE | 85.5% | 10-fold cross-validation |
| [7] | Feature selection | **None** | **83.12 %** | Percentage using a 90-10 ratio |
| [15] | Filling in missing values, normalization, outlier removal, and transformation | **SMOTE, SMOTE SVM** | **91%** | Percentage using a 90-10 ratio |

The challenge of diabetes prediction highlights the necessity of effective data preprocessing and balancing techniques to enhance model accuracy. As summarized in Table 1, previous studies using the Pima dataset demonstrate varying results, with accuracies ranging from 77.86% to 91%. While many researchers have focused on traditional preprocessing methods like filling missing values, the impact of advanced balancing strategies, particularly oversampling and undersampling techniques, remains underexplored. This paper stands out by comprehensively comparing these balancing methods, which are crucial for addressing class imbalance—a significant issue in medical datasets. The variations in accuracies among the studies emphasize that achieving high predictive performance depends not only on the choice of classifiers but also on the balancing techniques employed. By demonstrating how different balancing techniques interact with machine learning algorithms, this research contributes valuable insights to the field of diabetes prediction. This work underscores that selecting the appropriate balancing method is as vital as the choice of algorithm, ultimately paving the way for more accurate and reliable predictions in clinical practice.

## 3. The proposed framework

A framework is developed for diabetes detection, integrating various machine-learning algorithms and balancing techniques to evaluate these techniques and enhance the predictive performance. The framework consists of four main stages, as illustrated in Fig. 3: data preprocessing, data balancing, classification, and evaluation. A binary classification problem (diabetic or non-diabetic) is used for prediction.

### 3.1 Data preprocessing

In data preprocessing, missing values are handled, and data scaling is applied to ensure consistency across features. Missing values for key features such as glucose, blood pressure, skin thickness, insulin, and BMI, where unrealistically low values (such as zeros) are present, are treated as missing data. These missing values are filled with the median of the respective feature to maintain the statistical balance without introducing bias. This imputation helps prevent the skewing of model predictions due to incomplete or inaccurate data points. Data scaling is crucial to ensure that all features are on the same scale, preventing any particular feature from dominating the learning process due to its magnitude.

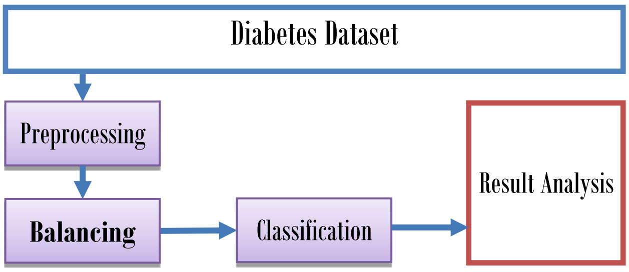

Figure. 3 The proposed framework

Table 2. Summary of the oversampling techniques

| Tech. | Description | Pros | Cons |
|---|---|---|---|
| Random [16] | Duplicates random samples of the minority class. | Simple and effective at balancing classes. | High risk of overfitting. |
| SMOTE [16] | Generates synthetic samples by interpolating between existing minority class samples. | Reduces overfitting and improves generalization. | May introduce noise. |
| Border-line SMOTE [17] | Generates synthetic samples near the decision boundary. | Enhances decision boundary robustness. | Can still lead to some overfitting. |
| ADASYN [18] | Generates synthetic samples based on density distribution. | Improves accuracy. | Overemphasize noisy instances. |
| SVM-SMOTE [19] | Generates synthetic samples based on SVM support vectors. | Improves accuracy. | Computationally intensive. |

Besides, the min-max scaling technique is employed, which transforms the data to a common range of [0, 1]. This method not only simplifies the model training process but also improves the convergence of optimization algorithms used in machine learning models. By scaling all features uniformly, the models can perform more effectively, especially those that rely on distance metrics or gradient-based optimization.

### 3.2 Data balancing

The PIMA dataset consists of 678 samples, with 500 non-diabetic cases and 268 diabetic cases, presenting a significant class imbalance. This imbalance can lead to a biased model that performs well in predicting the majority class (non-diabetic) but struggles to accurately identify the minority class (diabetic), which is critical for early detection and intervention. To address this issue, several data balancing techniques are employed, including oversampling methods to increase the number of minority class instances, undersampling to reduce the majority class, and combined methods that apply both oversampling and undersampling. These approaches aim to create more balanced datasets, enabling models to learn effectively from both classes and improving their predictive performance.

Oversampling is a technique that balances the dataset by increasing the number of samples in the minority class. This can be achieved by duplicating existing samples or generating new synthetic ones. In random oversampling, random samples of the minority class are duplicated until the class distribution is balanced. While this effectively balances the dataset, random oversampling can lead to overfitting, as the model may become too reliant on repeated instances. SMOTE [16] is a more advanced oversampling technique that generates new synthetic samples by interpolating between existing minority class samples. By creating new, plausible samples, SMOTE reduces overfitting and improves the model's generalization ability. Several extensions of SMOTE have been proposed, including borderline smote [17], ADASYN [18], and SVMSMOTE [19]. Borderline SMOTE focuses on generating synthetic samples near the decision boundary between classes, where misclassification is more likely. ADASYN adjusts the number of synthetic samples generated based on the learning difficulty of minority class instances, giving more weight to harder-to-learn cases. SVMSMOTE integrates SMOTE with SVM by using the support vectors to generate synthetic samples, focusing on key decision boundaries. Table 2 provides a comparison of these techniques.

Undersampling addresses data imbalance by reducing the number of samples in the majority class. This technique involves selecting a representative subset of the majority class to match the size of the minority class. In random undersampling, random samples from the majority class are removed until the dataset is balanced. However, this method can result in the loss of valuable information and may lead to underfitting. Advanced methods for undersampling have been proposed, including Cluster Centroids [20], Edited Nearest Neighbors (ENN) [21], All KNN [22], Condensed Nearest Neighbor (CNN) [23], and One-Sided Selection (OSS) [24]. The Cluster Centroids technique clusters the majority class into several groups using a clustering algorithm (like k-means) and then replaces these clusters with their centroids. The result is a reduced number of representative samples for the majority class. ENN removes samples from the majority class that are misclassified by their k-nearest neighbors. Similar to ENN, All KNN applies the ENN method iteratively with different values of k for the nearest neighbors, removing misclassified instances across all iterations. CNN reduces the majority class by selecting a subset of samples that maintain the decision boundary. It starts with a small subset and iteratively adds instances only if they contribute to the correct classification of the remaining samples. OSS combines CNN with ENN. It first applies CNN to reduce the majority class and then uses ENN to remove noisy instances from the reduced dataset. Table 3 provides a comparison of these techniques.

Table 3. Summary of the undersampling techniques

| Tech. | Description | Pros | Cons |
|---|---|---|---|
| Random [20] | Randomly removes samples from the majority class. | Simple to implement. | Leads to information loss and underfitting. |
| Cluster Centroids [20] | Clusters the majority class and replaces each cluster's samples with their centroids. | Preserves diversity and structure of the majority class. | Risk of information loss. |
| Edited Nearest Neighbors (ENN) [21] | Removes majority class samples misclassified by the k-nearest neighbors. | Effectively reduces noise and ambiguous instances. | Computationally expensive and Information loss. |
| All KNN [22] | Iteratively applies ENN with different values of k. | Improves performance by removing more noise. | Computationally expensive. |
| Condensed Nearest Neighbor (CNN) [23] | Selects a subset of majority class samples that maintain decision boundaries. | Helps preserve decision boundaries. | Sensitive to noise and outliers. |
| One-Sided Selection (OSS) [24] | Combines CNN and ENN to reduce the majority class and remove noisy samples. | Effectively reduces noise while balancing the data. | Computationally expensive and Information loss. |

Combined sampling methods implement both oversampling and undersampling techniques to achieve a balanced dataset. SMOTEENN [25] and SMOTE with Tomek Links (SMOTE-TOMEK) [26] are used for this purpose. SMOTEENN combines the Synthetic Minority Over-sampling Technique (SMOTE) with Edited Nearest Neighbors (ENN). First, SMOTE is applied to generate synthetic instances for the minority class. Then, ENN cleans the dataset by removing noisy or ambiguous instances from both the majority and minority classes. SMOTE-TOMEK is a combination of SMOTE and Tomek Links. After applying SMOTE to oversample the minority class, Tomek Links are used to identify and remove borderline instances close to the decision boundary between classes. Table 4 compares these techniques.

Table 4. Summary of the combined sampling techniques

| Tech. | Description | Pros | Cons |
|---|---|---|---|
| SMOTEENN [25] | Combines SMOTE with ENN | Reduces noise and improves decision boundary clarity | Computationally expensive and information loss |
| SMOTE Tomek [26] | Combines SMOTE with Tomek | Refines decision boundary by removing overlaps | Computationally expensive and information loss |

The choice of data balancing technique significantly affects model performance. Oversampling, particularly with techniques like SMOTE, helps to improve the model's sensitivity to the minority class, reducing the risk of bias towards the majority class. Undersampling, on the other hand, can simplify the model by reducing the size of the dataset but may also lead to the loss of important information. Combined sampling techniques offer a balanced approach, optimizing both class representation and model generalization.

### 3.3 Machine learning algorithms

Various machine-learning classifiers were used, referring to the classification implemented for diabetes prediction in the literature (See Table 1) and the general state of the art of data classification. KNN is a non-parametric, instance-based algorithm that classifies data points based on the majority label of their k-nearest neighbors. KNN is helpful in cases where the decision boundary is complex. Gaussian NB is a probabilistic classifier based on Bayes' theorem, assuming that the features follow a normal (Gaussian) distribution. NB is effective when the data distribution closely matches the Gaussian assumption. SVM is a supervised learning model that finds the optimal hyperplane to separate different classes in the feature space. SVM is well-known for its high performance, especially for high-dimensional spaces. DT is split into subsets based on the value of the input features; thus, it is known for its interpretability and ability to model complex relationships, and it is less sensitive to outliers. Ripper is a rule-based classifier that generates rules to classify data, create interpretable models, and handle balanced and imbalanced datasets. RF is an ensemble learning method that builds multiple decision trees and merges them to get a more accurate and stable prediction. RF is robust, can handle large datasets, and effectively reduces variance through ensemble learning. NN can model complex, non-linear relationships and can learn deep representations. AdaBoost is an ensemble technique that combines multiple weak classifiers to create a strong classifier. AdaBoost can improve model performance by focusing on difficult-to-classify samples. XGBoost is an advanced gradient-boosting technique that builds models in a stage-wise fashion. It is known for its speed, accuracy, and ability to handle large datasets with complex patterns. Table 5 compares these classifiers.

### 3.4 Data splitting and cross-validation

In this paper, the impact of oversampling is explored not only on the training set but also on the test set. Specifically, in some experiments, oversampling techniques were applied before splitting the dataset into training and testing subsets. This approach simulates a scenario where data balancing might occur prior to the separation of data, as is sometimes observed in practice. The objective was to assess how this preprocessing step, applied uniformly to the entire dataset, influences the overall performance of the classification models. Furthermore, both traditional data-splitting methods and cross-validation techniques were implemented to ensure a comprehensive evaluation of model performance. By employing these methods, the developed framework aims to enhance the reliability of the results and provide insights into the effectiveness of various oversampling techniques in different scenarios.

Table 5. Summary of the classification algorithms [27]

| Clas. | Description | Pros | Cons |
|---|---|---|---|
| KNN | Uses the k-nearest neighbors for classification. | Simple and effective. | Computationally expensive. |
| NB | Assumes Gaussian distribution of features. | Fast and handles high-dimensional data. | The assumption of normality may not always hold. |
| SVM | Finds optimal hyperplane. | Effective for high-dimensional spaces. | Memory-intensive and sensitive to parameter initialization. |
| DT | Splits data based on feature values. | Intuitive. | Prone to overfitting. |
| Ripper | Generates rules for classification. | Interpretable. | May be less effective on complex datasets. |
| RF | Combines multiple decision trees. | Robust and reduces overfitting. | Slow with a large number of trees. |
| NN | Models non-linear relationships. | Captures non-linear patterns. | Requires large datasets and is sensitive to hyperparameters. |
| AdaBoost | Combines weak classifiers. | Robust and reduces bias. | Sensitive to noisy data and prone to overfitting. |
| XGBoost | Combines weak classifiers. | High accuracy. | Requires careful tuning and can be memory-intensive. |

Table 6. Metrics comparison and use cases

| | Formula | Pros | Cons |
|---|---|---|---|
| Accuracy | $= \frac{TP + TN}{TP + TN + FP + FN}$ | Indicates overall performance for balanced data. | Misleading for imbalanced datasets. |
| Preci | $= \frac{TP}{TP + FP}$ | Measures false positives. | Covers limited aspects. |
| Recal | $= \frac{TP}{TP + FN}$ | Measures false negatives. | Covers limited aspects. |
| F1- | $= \frac{2 * precision * reca}{precision + recall}$ | Balances the importance of precision and recall. | Does not reflect the overall accuracy. |

### 3.5 Evaluation

The commonly utilized classification metrics to evaluate the prediction process's performance include accuracy, precision, recall, and F-measure (also known as F1-score). Each of these metrics offers insights into different aspects of model performance. The accuracy is the most straightforward metric, representing the proportion of correctly classified samples (both true positives and true negatives) out of the total samples in the testing set. Precision, also known as the positive predictive value, measures the proportion of true positive predictions out of all the instances that were predicted as positive. Recall, also known as sensitivity or true positive rate, measures the proportion of actual positives that were correctly predicted. The F1 score combines precision and recall, providing a single metric that balances both aspects. A summary of metrics is given in Table 6.

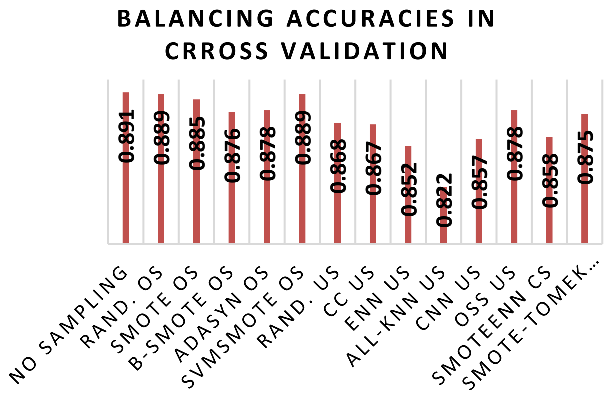

Figure. 4 Comparison results of the balancing techniques in cross-validation

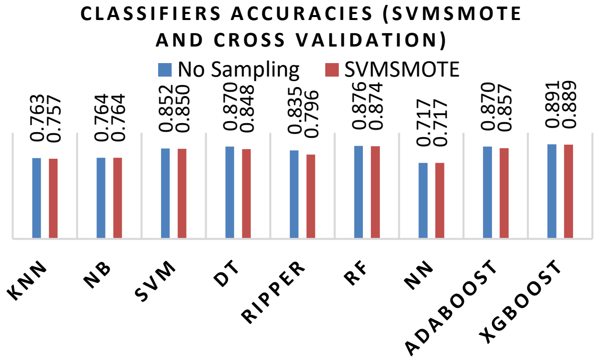

Figure. 5 Comparison results of the classifiers with SVMSMOTE in cross-validation

### 3.6 Experimental setup

Two experimental setups were implemented: (1) the training set balanced with the test set unaltered, reflecting real-world conditions, and (2) both training and test sets balanced for controlled comparative analysis. The first approach evaluates generalizability, while the second explores the full potential of the framework under ideal conditions.

## 4. Results and discussion

The evaluation was conducted in three sets of experiments using 10-fold cross-validation, 80-20 percentage split, and prior balancing. In cross-validation, the whole dataset was split into 10 folds; in each run, only the training data underwent oversampling (OS), undersampling (US), and combined sampling (CS), and the testing set remained as it was. In the percentage split, similarly, the training set was subject to balancing techniques, which are implemented, as only one run is required. In prior balancing, the whole dataset was sampled first; then, 10-fold cross-validation was conducted. As such, in the last experiments, both training and testing sets undergo the balancing techniques.

### 4.1 PIMA results

Fig. 4 shows a comparison between the different balancing techniques in cross-validation experiments with reference to no-sampling results, which was 89.1%. The results showed that balancing techniques decrease the accuracy, especially the undersampling. The accuracy for undersampling using the best classifier ranges from 82.2% with All KNN to 87.8% with OSS. Oversampling, on the other hand, achieved better results, ranging from 87.6% using Borderline SMOTE to 88.9% with SVMSMOTE, which is very close to the results obtained without balancing. Fig. 5 compares the results obtained without balancing and those using balancing techniques, highlighting the best performance achieved with SVMSMOTE.

As noted in Fig. 5, the results of classifier accuracy without any sampling techniques show that different models perform variably, with the highest accuracy achieved by XGBoost (89.1%), followed by RF (87.6%), AdaBoost (87%), and DT (87%). These tree-based models perform better because they handle complex data distributions and capture intricate patterns. SVM also performs well, with an accuracy of 85.2%. However, KNN, NB, and NN perform lower, with accuracies of 76.3%, 76.4%, and 71.7 respectively. The Ripper algorithm, which is rule-based, achieves a moderate accuracy of 83.5%, which is competitive but slightly lower than the tree-based methods.

When oversampling is applied to balance the dataset, a general trend of performance changes is observed. The XGBoost classifier exhibits a slight decrease in accuracy to 88.9%. Similarly, the accuracy of RF decreases from 87.6% to 87.4%, the accuracy of AdaBoost decreases from 87% to 85.7%, the accuracy of DT shows a significant reduction from 87% to 0.848, the accuracy of SVM experiences a slight decline from 85.2% to 85%, and the accuracy of KNN exhibits a minor decrease from 76.3% to 75.7%. On the other hand, NB remains consistent with an accuracy of 76.4%, and NN shows no change in accuracy, remaining at 71.7%. The Ripper algorithm exhibits a notable decrease in accuracy from 83.5% to 79.6%, possibly due to oversampling introducing more complexity into the rule-based learning process.

Fig. 6 compares the different balancing techniques in percentage split experiments. The classification accuracy without any sampling was 89.6%. The results showed that balancing techniques,

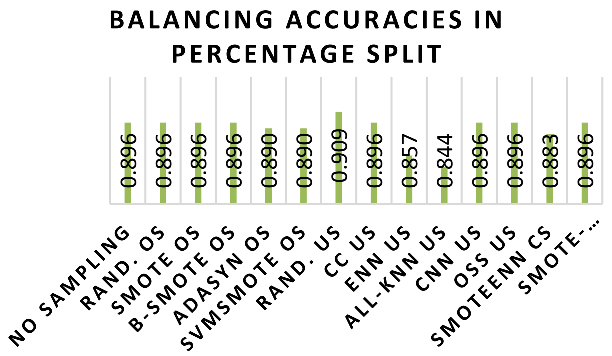

Figure. 6 Comparison results of the balancing techniques in percentage-split

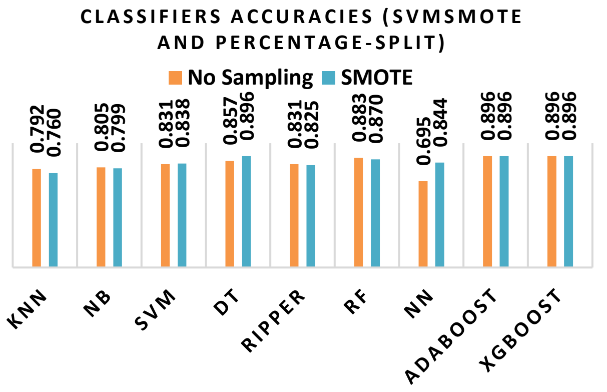

Figure. 7 Comparison results of the classifiers in percentage-split

specifically random, SMOTE, and SMOTE-B, maintain the same accuracy, while the rest decrease the accuracy slightly. The results of undersampling and combined sampling also exhibit the same trends, while the random undersampling slightly improves the results with an accuracy of 90.9%. Fig. 7 compares the results obtained without balancing and those with balancing techniques, highlighting the best performance achieved with SMOTE.

As noted in Figs. 6 and 7, the same results and conclusions apply as discussed in cross-validation experiments, with the exception of SVM, DT, and NN, which show slight improvements. However, in this case, the results depend on the selected training and testing subsets, meaning they cannot be generalized as reliably as those from the cross-validation.

The results of the prior balancing are quite notable, as they exhibit a different pattern compared to the earlier experiments. Fig. 8 shows a comparison of the different balancing techniques in prior balancing experiments, with no-sampling results used as reference, which was 89.1%. The results indicate that balancing techniques generally increase accuracy, with SMOTEENN combined sampling standing out, achieving an accuracy of 97.5%. The oversampling techniques also show improvements, with accuracies ranging from 88.7% using ADASYN to 93.6% using Random oversampling. The undersampling techniques present a varied range, with accuracies ranging from 80.8% using CNN to 94.1% using All KNN. Fig. 9 compares the results obtained with no balancing against those with balancing techniques, highlighting SMOTEENN as the top performer.

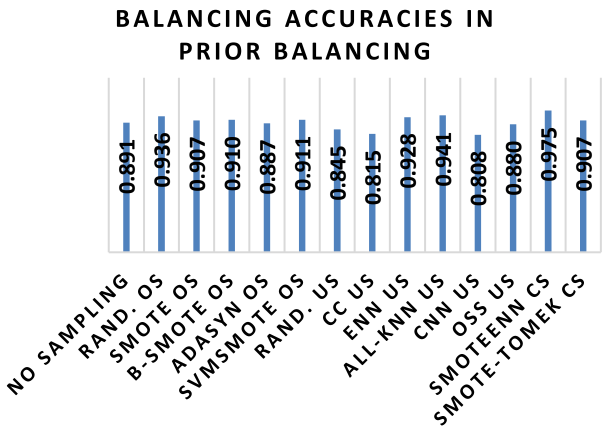

Figure. 8 Comparison results of the balancing techniques in prior balancing

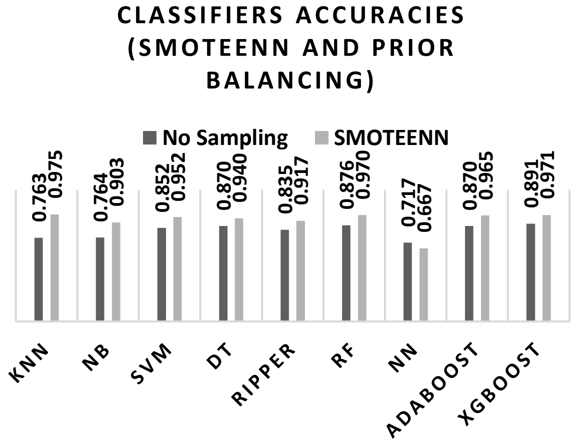

Figure. 9 Comparison results of the classifiers in prior balancing

As noted in Figs.8 and 9, the accuracies improved using SMOTEENN for all classifiers except for the NN. The best results were achieved by the KNN classifier, with an accuracy of 97.5%.

The variation in results across different experiments can be attributed to the quality of the utilized data. In both cross-validation and percentage split experiments, accuracy generally decreased with the application of balancing techniques. However, in the prior balancing experiment, the results improved significantly. This variation is likely due to the quality of the samples, particularly rows with missing values, which may not contribute effectively to the classification process. Balancing the training set alone can sometimes lead to overfitting, especially if synthetic samples fail to represent real-world data. In such cases, the model may learn patterns specific to the balanced training data but perform poorly on the imbalanced test set. Moreover, the original dataset may not be fully representative, which justifies the observed increase in accuracy with undersampling and combined sampling techniques.

The results indicate that oversampling the test set can artificially inflate model performance by altering the natural distribution of the data. While this approach may be suitable in specific applications where balanced datasets are prioritized even during testing, it generally does not reflect real-world conditions where data imbalance is common. This scenario was tested to illustrate the impact of preprocessing decisions on model accuracy and to caution against oversampling test data unless the objective is to measure performance under fully balanced conditions.

The findings also reveal that oversampling did not consistently lead to improved accuracy. This underscores the potential risks of overfitting when synthetic samples are introduced into the dataset. Oversampling can sometimes degrade performance, particularly when the synthetic data does not accurately represent real-world conditions or when decision boundaries are noisy or unclear. This overfitting can result in models performing well on a balanced training set but poorly on an imbalanced test set. Therefore, while oversampling can be advantageous in some cases, it should be used carefully to prevent the introduction of noise and to ensure the model maintains its generalization ability.

The results demonstrate that balancing the test set leads to marginally higher recall but compromises real-world applicability. The unbalanced test set results, which better reflect deployment scenarios, still show competitive performance, validating the framework's robustness.

The results show that XGBoost, which can handle both high-dimensional data and imbalanced datasets effectively, performs the best in classifying diabetes cases. XGBoost, when paired with techniques like SMOTE, SMOTE-TOMEK, and SVMSMOTE, which generate synthetic samples to balance the dataset, not only enhances the minority class representation but also reduces noise and overlapping between classes, which XGBoost can better exploit due to its tree-based structure. SMOTE-TOMEK, for example, combines oversampling and under-sampling, improving recall by reducing the impact of outliers and noise while increasing the decision boundary clarity for the classifier. In contrast, simpler classifiers such as SVM or AdaBoost, though benefiting from balancing techniques like SMOTE and ADASYN, struggle to match XGBoost's performance due to their sensitivity to noise and overfitting. For instance, while SVM shows an improvement with methods like SVMSMOTE, it still lags behind XGBoost in terms of recall and accuracy. This is because SVM relies on margin maximization, which can be compromised in the presence of noisy or overlapping synthetic data. AdaBoost, being a boosting method as well, performs well with techniques like SMOTE and B-SMOTE but tends to be less stable compared to XGBoost when handling imbalanced datasets, as it may not always generalize as effectively, especially with small or noisy datasets.

### 4.2 Results comparison

In comparison to the state-of-the-art, the results from this study reveal notable improvements in classification accuracy when using the SMOTEENN balancing technique, achieving an impressive accuracy of 97.5% for the KNN classifier, compared to state-of-the-art results such as 91% with SMOTE and SMOTESVM in a recent study ([15], 2024) and 90% with SMOTE in another ([10], 2022). In contrast, traditional balancing methods showed varied effectiveness, with oversampling methods yielding slight declines in accuracy for some classifiers. This highlights the necessity of selecting appropriate balancing techniques based on the specific dataset and model characteristics. However, the potential limitations of this study must be acknowledged. While balancing techniques can enhance accuracy, they may also lead to overfitting, particularly if synthetic samples are not representative of real-world distributions.

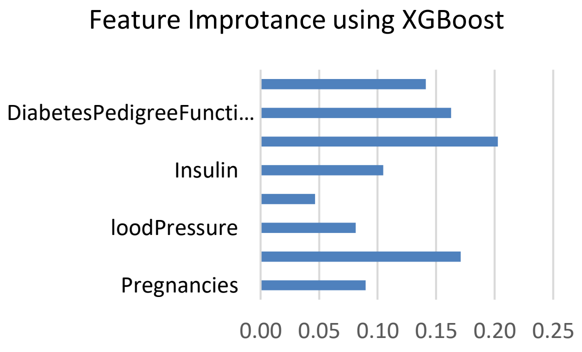

Figure. 10 Diabetes feature significance

Table 7. A Summary of the best-performing interacted techniques using the accuracy measure

| Balancing Technique | Accuracy | | |
|---|---|---|---|
| | **RF** | **AdaBoost** | **XGBoost** |
| No Sampl. | **0.876 ± 0.005** | **0.870 ± 0.007** | **0.891 ± 0.006** |
| **SMOTE** | **0.874 ± 0.006** | **0.865 ± 0.008** | **0.885 ± 0.005** |
| **B-SMOTE** | **0.862 ± 0.007** | **0.874 ± 0.006** | **0.876 ± 0.004** |
| **ADASYN** | **0.866 ± 0.006** | **0.862 ± 0.005** | **0.878 ± 0.004** |
| **SVM-SMOTE** | **0.874 ± 0.005** | **0.857 ± 0.008** | **0.889 ± 0.006** |
| **OSS** | **0.872 ± 0.006** | **0.861 ± 0.007** | **0.878 ± 0.004** |
| **SMOTE-TOMEK** | **0.867 ± 0.005** | **0.866 ± 0.006** | **0.875 ± 0.005** |

Table 8. A Summary of the best-performing interacted techniques using the precision measure

| Balancing Technique | Precision | | |
|---|---|---|---|
| | **SVM** | **AdaBoost** | **XGBoost** |
| No Sampl. | **0.847 ± 0.004** | **0.839 ± 0.005** | **0.854 ± 0.004** |
| **SVM-SMOTE** | **0.776 ± 0.005** | **0.776 ± 0.006** | **0.823 ± 0.005** |
| **Cluster Centroids** | **0.812 ± 0.004** | **0.784 ± 0.005** | **0.792 ± 0.004** |
| **OSS** | **0.801 ± 0.005** | **0.799 ± 0.005** | **0.813 ± 0.004** |
| **SMOTE-TOMEK** | **0.786 ± 0.005** | **0.785 ± 0.005** | **0.807 ± 0.004** |

Table 9. A Summary of the best-performing interacted techniques using the recall measure

| Balancing Technique | Recall | | | |
|---|---|---|---|---|
| | **Ripper** | **RF** | **AdaBoost** | **XGBoost** |
| No Sampl. | **0.914 ± 0.004** | **0.810 ± 0.005** | **0.776 ± 0.005** | **0.828 ± 0.004** |
| **SMOTE** | **0.925 ± 0.004** | **0.840 ± 0.005** | **0.851 ± 0.004** | **0.862 ± 0.004** |
| **B-SMOTE** | **0.948 ± 0.004** | **0.858 ± 0.004** | **0.869 ± 0.004** | **0.851 ± 0.004** |
| **ADASYN** | **0.963 ± 0.003** | **0.851 ± 0.004** | **0.858 ± 0.004** | **0.862 ± 0.004** |
| **CSS** | **0.940 ± 0.004** | **0.854 ± 0.004** | **0.825 ± 0.004** | **0.825 ± 0.004** |
| **ENN** | **0.944 ± 0.004** | **0.925 ± 0.004** | **0.937 ± 0.004** | **0.922 ± 0.004** |
| **All-KNN** | **0.951 ± 0.004** | **0.929 ± 0.004** | **0.925 ± 0.004** | **0.937 ± 0.004** |
| **SMOTE-TOMEK** | **0.978 ± 0.003** | **0.896 ± 0.004** | **0.881 ± 0.004** | **0.888 ± 0.004** |

### 4.3 Feature significance

The significance of each feature, as analyzed using the XGBoost, is given in Fig. 10. The results showed that BMI and glucose are among the most significant features in diabetes predictions.

### 4.4 Interaction analysis and validations

In cross-validation, it was shown that oversampling methods, particularly SMOTE and ADASYN, improve recall, particularly for Ripper

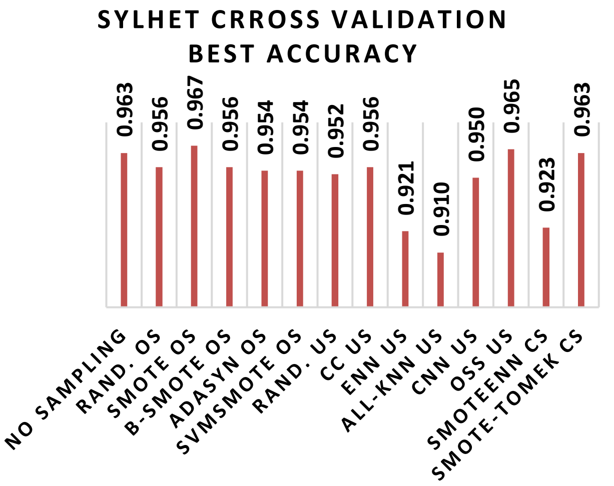

Figure. 11 The Sylhet dataset cross-validation best results

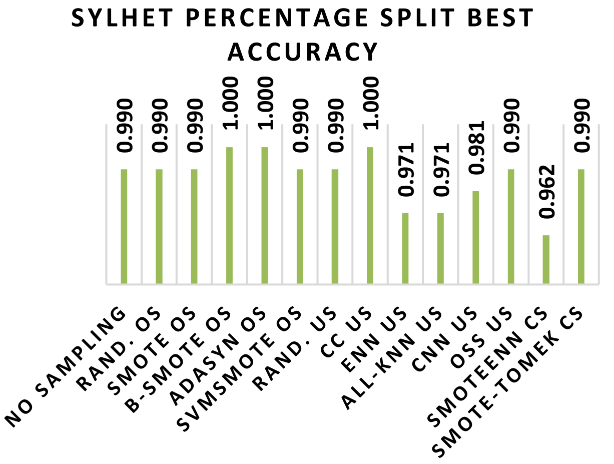

Figure. 12 The Sylhet dataset percentage-split best results

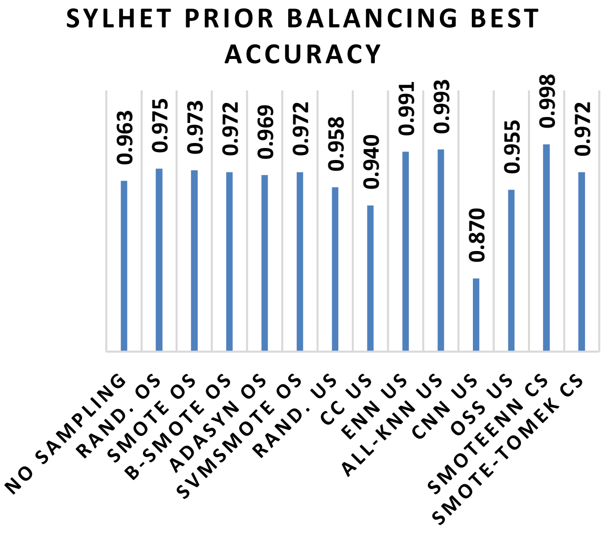

Figure. 13 The Sylhet dataset prior-balancing best results

(e.g., ADASYN achieves 96.3% recall) but slightly reduces precision. Undersampling ENN and All-KNN methods benefit recall but often lower accuracy and precision. Combined Techniques, the SMOTEENN and SMOTE-TOMEK balance recall and F1 scores effectively, especially for RF and XGBoost. The results were confirmed in percentage split and prior balancing. It was also noted that XGBoost and AdaBoost, together with SMOTEENN and SMOTE-TOMEK, achieve balanced improvements in precision and recall. NN, together with ADASYN, showed performance improvement. A summary of the best-performing balancing techniques for accuracy, precision, and recall is given in Tables 7-9, respectively. The **Confidence Intervals (CI) f**or each performance metric (accuracy, precision, recall, F1-score) are calculated based on 95% confidence intervals. For example, for XGBoost with "No Sampling," the accuracy is 0.891 with a margin of error of ± 0.006. The calculated confidence intervals indicate the margin of error for the mean performance of each balancing technique applied to each classifier.

For significant comparisons, the paired t-test results for accuracy comparisons across all sampling methods are reported. Table 10 lists only significant results among all pair-wise comparisons. Significant improvements in accuracy were observed in all comparisons involving SMOTE, B-SMOTE, and ADASYN against No Sampling and Random oversampling. The results indicate that the oversampling methods helped significantly improve the accuracy of classifiers. The precision improvements were most noticeable between Random and SMOTE and also between No Sampling and SMOTE. SMOTE consistently showed better performance than others, particularly for the SVM and RF classifiers. SMOTE and B-SMOTE provided notable improvements in recall compared to No Sampling and Random oversampling. This suggests that the oversampling techniques helped capture more of the positive class, especially for SVM, RF, and AdaBoost classifiers. The F1 score showed consistent improvements with SMOTE and B-SMOTE, indicating these methods balanced precision and recall well, leading to higher F1 values. Again, SVM and XGBoost benefited significantly from these sampling methods.

Overall, oversampling methods like SMOTE, B-SMOTE, and ADASYN generally led to significant improvements across the performance metrics, particularly in recall and F1 score, indicating they helped improve the classifier's ability to correctly identify the minority class while maintaining the overall balance between precision and recall.

### 4.5 Syllhat dataset results

The Sylhet Diabetes dataset [28] is used to generalize the previous results. The dataset consists of 520 records, with each record containing various features related to demographic, medical, and lifestyle data, along with a target variable indicating the presence or absence of diabetes. The results are given in Figs 11 to 13. The results confirmed the findings obtained for the PIMA dataset and showed the effect of each balancing technique on the classification results accordingly [29-30].

## 5. Conclusion

In conclusion, this study analyzed the effects of balancing techniques on the PIMA Indian Diabetes dataset. A comprehensive framework for diabetes prediction was implemented, incorporating the balancing process alongside preprocessing, classification, and evaluation. The experiments demonstrate that the impact of sampling techniques varies across different evaluation methods. Balancing the training data in cross-validation and percentage split experiments often led to a slight decrease in accuracy, particularly with undersampling. However, prior balancing, where the entire dataset is balanced before splitting, resulted in significant improvements, especially with combined sampling methods like SMOTEENN, which achieved the highest accuracy. The study demonstrates that the choice of data balancing techniques significantly affects classification performance. SMOTEENN consistently outperformed other methods, achieving a peak accuracy of 97.5%. Statistical tests confirmed the significance of these improvements, with p-values indicating meaningful performance gains.

These findings suggest that balancing both training and testing sets can enhance classifier performance and provide more reliable and generalizable results. Cross-validation with balanced data remains essential for consistent model evaluation across different subsets. This study highlights the critical role of selecting appropriate balancing techniques for specific classifiers. Models like XGBoost exhibit robustness across all techniques, while SVM, RF, and Ripper benefit from different balancing techniques. The superior performance of SMOTE-TOMEK underscores the importance of combined balancing to improve the recall.

Future research will focus on exploring more robust synthetic sample generation methods, such as Generative Adversarial Networks (GANs), and employing stratified sampling to ensure balanced

Received: January 30, 2025. Revised: February 16, 2025.

representations in both training and testing datasets. Besides, future research will explore the integration of deep learning approaches and alternative feature selection methods. Finally, feature research will focus on using different feature selection techniques and various methods to handle missing data.

## Conflicts of Interest

The authors declare no conflict of interest.

## Author Contributions

Conceptualization, Ahmad Adel Abu-Shareha; methodology, Abdelrahman Hussein; software, Mosleh Abualhaj; validation, Adeeb Al-Saaidah; formal analysis, Anusha Achuthan; resources, Ahmad Adel AbuShareha; writing—original draft preparation, Ahmad Adel Abu-Shareha; writing—review and editing, Anusha Achuthan; visualization, Adeeb Al-Saaidah; supervision, Ahmad Adel Abu-Shareha; project administration, Mosleh Abualhaj; funding acquisition, Ahmad Adel Abu-Shareha".

## Acknowledgments

This work was supported by Al-Ahliyya Amman University–Jordan research grant (17/18/2023-2024).

## References

[1] R. Zolfaghari, "Diagnosis of diabetes in female population of pima indian heritage with ensemble of bp neural network and svm", *The International Journal of Computational Engineering and Management,* Vol. 15, No. 4, pp. 2230-7893, 2012.

[2] S. Yadu, R. Chandra, and V. K. Sinha, "Comparing Different Machine Learning Techniques in Predicting Diabetes on Early Stage", *Engineering Proceedings,* Vol. 62, No. 1, pp. 20, 2024.

[3] M. S. Reza, U. Hafsha, R. Amin, R. Yasmin, and S. Ruhi, "Improving SVM performance for type II diabetes prediction with an improved non-linear kernel: Insights from the PIMA dataset", *Computer Methods and Programs in Biomedicine Update,* Vol. 4, pp. 100118, 2023.

[4] D. J. Magliano and E. J. Boyko, "IDF Diabetes Atlas", *International Diabetes Federation Brussels,* 2022.

[5] O. N. Ergün and H. O. İlhan, "Early stage diabetes prediction using machine learning methods", *Avrupa Bilim ve Teknoloji Dergisi,* Vol. 2021, No. 29, pp. 52-57, 2021.

[6] V. Chang, J. Bailey, Q. A. Xu, and Z. Sun, "Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms", *Neural Computing and Applications,* Vol. 35, No. 22, pp. 16157-16173, 2023.

[7] A. Perdana, A. Hermawan, and D. Avianto, "Analyze Important Features of PIMA Indian Database For Diabetes Prediction Using KNN", *Jurnal Sisfokom (Sistem Informasi dan Komputer),* Vol. 12, No. 1, pp. 70-75, 2023.

[8] A. G. Karegowda, A. Manjunath, and M. Jayaram, "Application of genetic algorithm optimized neural network connection weights for medical diagnosis of pima Indians diabetes", *International Journal on Soft Computing,* Vol. 2, No. 2, pp. 15-23, 2011.

[9] S. Wei, X. Zhao, and C. Miao, "A comprehensive exploration to the machine learning techniques for diabetes identification", In: *Proc. of the IEEE 4th World Forum on Internet of Things (WF-IoT),* Singapore, 5-8, 2018.

[10] H. B. Kibria, M. Nahiduzzaman, M. O. F. Goni, M. Ahsan, and J. Haider, "An ensemble approach for the prediction of diabetes mellitus using a soft voting classifier with an explainable AI", *Sensors,* Vol. 22, No. 19, pp. 7268, 2022.

[11] S. Simaiya *et al.*, "A novel multistage ensemble approach for prediction and classification of diabetes", *Frontiers in Physiology,* Vol. 13, pp. 1085240, 2022.

[12] M. O. Edeh *et al.*, "A classification algorithm-based hybrid diabetes prediction model", *Frontiers in Public Health,* Vol. 10, pp. 829519, 2022.

[13] R. Marzouk, A. S. Alluhaidan, and S. A. El_Rahman, "An analytical predictive models and secure web-based personalized diabetes monitoring system", *IEEE Access,* Vol. 10, pp. 105657-105673, 2022.

[14] V. K. Yadav and Nilam, "Comparison of machine learning techniques for precision in measurement of glucose level in artificial pancreas", *Mathematical Methods in the Applied Sciences,* 2022.

[15] L. Al-Dabbas, & Abu-Shareha, A. (2024). "Early Detection of Female Type-2 Diabetes using Machine Learning and Oversampling Techniques", *Journal of Applied Data Sciences,* Vol. 5, No. 3, pp. 1237-1245, 2024.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique", *Journal of artificial intelligence research,* Vol. 16, pp. 321-357, 2002.

[17] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning", In: *Proc. of International conference on intelligent computing,* pp. 878-887, 2005.

[18] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning", In: *Proc. of 2008 IEEE international joint conference on neural networks,* pp. 1322-1328, 2008.

[19] Q. Wang, Z. Luo, J. Huang, Y. Feng, and Z. Liu, "A Novel Ensemble Method for Imbalanced Data Learning: Bagging of Extrapolation-SMOTE SVM", *Computational intelligence and neuroscience,* Vol. 2017, No. 1, pp. 1827016, 2017.

[20] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, “Clustering-based undersampling in class-imbalanced data”, *Information Sciences,* Vol. 409, pp. 17-26, 2017.

[21] D. L. Wilson, “Asymptotic properties of nearest neighbor rules using edited data”, *IEEE Transactions on Systems, Man, and Cybernetics,* No. 3, pp. 408-421, 1972.

[22] I. Tomek, “An Experiment with the Edited Nearest-Neighbor Rule”, *EEE Transactions on Systems, Man, and Cybernetics,,* Vol. 6, No. 6, pp. 448-452, 1976.

[23] P. Hart, “The condensed nearest neighbor rule”, *IEEE transactions on information theory,* Vol. 14, No. 3, pp. 515-516, 1968.

[24] M. Kubat and S. Matwin, “Addressing the curse of imbalanced training sets: one-sided selection”, *ICML,* Vol. 97, No. 1, pp. 179-186, 1997.

[25] G. E. Batista, R. C. Prati, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data”, *ACM SIGKDD explorations newsletter,* Vol. 6, No. 1, pp. 20-29, 2004.

[26] G. E. Batista, A. L. Bazzan, and M. C. Monard, “Balancing training data for automated annotation of keywords: a case study”, *Wob,* Vol. 3, pp. 10-18, 2003.

[27] C. Zhang, C. Liu, X. Zhang, and G. Almpanidis, “An up-to-date comparison of state-of-the-art classification algorithms”, *Expert Systems with Applications,* Vol. 82, pp. 128-150, 2017.

[28] M. F. Islam, R. Ferdousi, S. Rahman, and H. Y. Bushra, “Likelihood prediction of diabetes at early stage using data mining techniques”, *Computer Vision and Machine Intelligence in Medical Image Analysis,* pp. 113-125, 2020.

[29] S. Nasim et al., “Novel meta learning approach for detecting postpartum depression disorder using questionnaire data”, *IEEE Access*, Vol. 12, pp. 101247-101259, 2024.

[30] Q. Shambour, N. Qandeel, Y. Alrabanah, A. Abumariam, and M. K. Shambour, “Artificial Intelligence Techniques for Early Autism Detection in Toddlers: A Comparative Analysis”, *Journal of Applied Data Sciences*, Vol. 5, No. 4, pp. 1754-1764, 2024.