International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Comparative Analysis of Machine Learning Models and Majority Voting Ensemble for Diabetes Prediction

Sangita Baruah[1], Chandan Jyoti Kumar[2], Bipasha Bhattacharjee[3]

Department of Computer Science and Information Technology, Cotton University, India

Abstract

Diabetes is a rapidly growing chronic health condition affecting millions worldwide. Early detection is critical to managing and mitigating the long-term complications associated with the disease. In this study, we perform a comparative analysis of machine learning models for diabetes detection using the Pima Indians Diabetes Dataset. Model performance was evaluated across three experimental phases: (1) using all available features, (2) using selected features obtained through the Recursive Feature Elimination (RFE) algorithm, and (3) separating attributes into clinical and non-clinical groups to assess their individual predictive potential. To enhance robustness, a majority voting ensemble was implemented by combining Linear Discriminant Analysis, Decision Tree, and Random Forest classifiers. The results show that the full feature set consistently achieved better performance compared to RFE-selected subsets, while clinical features outperformed non-clinical features but did not surpass the complete feature set. Majority voting improved stability in most scenarios, with the best overall performance obtained using all features. This study highlights the importance of dataset-specific evaluation of feature selection and demonstrates the potential of ensemble learning as a reliable approach for non-invasive diabetes detection.

Keywords: Diabetes Prediction, Machine Learning, Feature Selection, Majority Voting, Ensemble Learning, Pima Indians Dataset

1. Introduction

Diabetes Mellitus, particularly Type 2 Diabetes, is a chronic metabolic disorder and a major global health concern. According to the International Diabetes Federation, approximately 537 million adults worldwide were living with diabetes in 2021, a number projected to rise sharply in the coming decades. Early and accurate identification of diabetes is essential to manage disease progression and reduce associated complications. However, in many low-resource settings, access to diagnostic tools remains limited.

In recent years, machine learning (ML) has emerged as a promising approach for non-invasive and data-driven diabetes detection. The Pima Indians Diabetes Dataset (PIDD), a widely studied benchmark dataset, provides clinical and lifestyle-related variables that enable researchers to evaluate predictive modelling approaches.

While numerous studies on PIDD have explored individual classifiers, fewer works have systematically compared the role of feature selection and the contribution of clinical vs. non-clinical attributes to classification performance. Moreover, the potential benefits of ensemble learning methods, such as majority voting, remain underexplored in this context.

In this study, we address these gaps by performing a phase-wise comparative analysis under three scenarios:

1. Using all original features.
2. Using statistically selected features via the Recursive Feature Elimination (RFE) algorithm.
3. Separating features into clinical and non-clinical groups to assess their individual predictive potential.

To enhance robustness, we employ a majority voting ensemble combining Logistic Regression, Decision Tree, and Random Forest classifiers.

The key objectives of this study are to:

- Evaluate the performance of individual classifiers (Linear Discriminant Analysis, Decision Tree, Random Forest) and their majority voting ensemble on the Pima Indians Diabetes Dataset.
- Compare classification performance using (a) the full feature set, (b) subsets of features selected using Recursive Feature Elimination (RFE), and (c) clinical vs. non-clinical feature groups.
- Analyse whether feature selection enhances or reduces performance compared to using all features.
- Provide practical insights into the effectiveness of feature selection strategies for diabetes detection and their implications in small, real-world datasets.

This research contributes to the growing body of work on ML for healthcare by offering new insights into how feature selection strategies and ensemble learning affect diabetes detection.

2. Related Work

The area of diabetes detection has received considerable research attention due to its prevalence on healthcare across the globe. Many machine learning and data-driven approaches have been developed to provide efficient early detection. This section reviews a subset of contributions in this area. Table 1 summarizes related studies across different datasets, while Table 2 focuses specifically on works using the Pima Indians Diabetes Dataset (PIDD) for fair comparison with the present study.

Kamble *et al.* [1] have used the Restricted Boltzmann Machine, which is based on Deep Learning, as a classifier and feature extractor for making a diabetes detection model.

In 2017, Gujral [2] published a review paper on early diabetes detection systems based on ML techniques. The paper also looks into the detection of diseases like Diabetes Retinopathy and Diabetes Neuropathy that frequently occur to diabetic patients. A similar survey paper was published by Indoria *et al.* [3] that delved into various diabetes Detection and Prediction Models based on Machine Learning that have been developed recently.

Warke *et al.* [4] used four Machine Learning algorithms including Logistic Regression, Support Vector Machine, Naïve Bayes and K Nearest Neighbours to make a diabetes prediction model. Majumdar &

Vaidehi [5] proposed a diabetes prediction model with a dataset containing some external factors in addition to the clinical and non-clinical factors usually used in diabetes prediction models.

Kopitar *et al.* [6] published a study to compare four machine learning models to regression models for early prediction of type 2 diabetes. Rani [7] put forward a project in 2020 with the aim of a model that aids early diabetes detection by employing various machine learning models and combining their results to achieve higher accuracy.

Llaha & Rista [8] used various data mining techniques to analyse the dataset and then testing it on four machine learning algorithms including Artificial Neural Network, to find the most efficient one. Farajollahi *et al.* [9] performed a study to analyse the performances of six machine learning classifiers for detection of diabetes mellitus. Abdulhadi & Mousa [10] put forward research with the objective to correctly predict diabetes early, especially in women, so that they can make necessary changes in their lifestyle to prevent diabetes and other diseases that are associated with it. Sharma & Shah [11] also published a comprehensive review on various diabetes detection models that use machine learning classifiers such as support vector machine, decision tree, etc. and also deep learning models like artificial neural network and recurrent neural network.

Wee *et al.* (2023) [12] published a review paper to show how a data-driven diabetes classification model can be constructed based on datasets with non-lab-invasive medical features. Uddin et al. [13] in 2023 proposed a new pipeline for diabetes prediction that uses SMOTE to balance data and predicts diabetes using machine learning algorithms. Harnal *et al.* [14] put forward research that aimed at employing eight machine learning models and finding the most efficient model for prediction of diabetes.

Table 1: Summary of Related Works on Diabetes Prediction (General Datasets)

Sl. No.	*Authors*	*Approach*	*Dataset Used*	*Attribute*	*FS Algorithm Used*	*Classification Algorithm Used*	*Performance Matrix*
1	Kamble et al.	Deep learning	Not Mentioned	5: Glucose, insulin, BMI, age, DPF	Not Mentioned	Restricted Boltzmann machine, Decision Tree	Pre: 80.64% (RBM) Pre: 92.10% (DT)
2	Warke et al.	Machine Learning	Dataset from National institute of Diabetes and digestive and kidney diseases.	8: age, BMI, DPF, Pregnancies, Glucose, Insulin, BP, skin thickness	Not mentioned	NB, DT, VM, KNN, Logistic Regression	Acc: 72%(NB)

				, outcome			
3	Majumdar et al.	Machine Learning	Curated with 800 records and 10 attributes	8: age, BMI, Pregnancies, Glucose, Insulin, BP, skin thickness, job type	K-Mean clustering	Logistic regression. Adaboost classifier for pipelining.	Acc: 96% (LR) Acc: 98.8% (Adaboost)
4	KM. Jyoti Rani	Machine Learning	Diabetes Dataset taken from the hospital. Frankfurt, Germany	8: age, BMI, DPF, Pregnancies, Glucose, Insulin, BP, skin thickness, outcome	Not Mentioned	Logistic regression, DT, RF, SVM	Acc: 99% (DT)
5	Llaha et al.	Machine Learning	From Public Health Institute	8: age, BMI, etc.	Not mentioned	NB, VM, DT, Artificial Neural Netork	Acc: 79%(DT)
6	Harnal et al.	Machine Learning	Gathered from Kaggle Public Repository	17	Not Mentioned	SVM, NB, LR, DT, Extra Tree Classifier (ETC), RF, KNN, XGB	Acc: 98.55%(ETC) 98.32%(RF)

Table 2: Comparative Studies Using Pima Indians Diabetes Dataset (PIDD)

Sl. No.	*Authors*	*FS Algorithm Used*	*Classification Algorithm(s) Used*	*Best Performance Reported*
1	Farajollahi et al.	PCA	Adaboost, Decision Tree, RF, SVM	Acc: 83.76% (Adaboost)

2	Abdulhadi et al.	Heat Map	Random Forest	Acc: 82%
3	Wee et al.	PCA / PCC	SVM, LR, RF, CNN, DBN, DNN	Acc: 81.9% (RF), 98% (DNN)
4	Md. Ashraf Uddin	SMOTE applied	LR, SVM, RF, NB, DT, KNN	Acc: 97.54% (RF)
5	Our Work (2025)	RFE (Recursive Feature Elimination)	LR, DT, RF, Majority Voting Ensemble	Acc: 77.99%, Pre: 71.43%, Rec: 61.57%, F1: 66.13%, ROC AUC: 74.18%

From Tables 1 and 2, it is clear that studies using additional or curated datasets often report very high accuracies (96–99%). However, results on the standard PIDD dataset are generally lower due to its inherent challenges: small size, imbalanced classes, and noisy attributes. Within this context, the present study achieves competitive results while also providing novel comparative insights into:

- The role of feature selection (RFE) on classification outcomes,
- Performance differences between clinical and non-clinical features,
- The effectiveness of majority voting ensembles compared to individual classifiers.

This strengthens the contribution of the present work as a comparative analysis study rather than purely performance-maximization.

3. Methodology

This section presents the dataset description, preprocessing steps, model selection, feature selection techniques, feature engineering strategies, ensemble method, and evaluation metrics used in the study.

3.1 Dataset Description

The study uses the Pima Indians Diabetes Dataset (PIDD) obtained from the Kaggle Dataset Repository. The dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. It comprises 768 female patients of Pima Indian heritage, aged 21 years or older, and includes eight numerical input features relevant to diabetes risk assessment. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes. The following table (Table 3) represents the feature descriptions of the PIDD dataset.

Table 3: Feature descriptions of the PIDD dataset.

Feature	*Description*
Pregnancies	Number of times pregnant
Glucose	Plasma glucose concentration (2 hours in OGTT)

BloodPressure	Diastolic blood pressure (mm Hg)
SkinThickness	Triceps skin fold thickness (mm)
Insulin	2-Hour serum insulin (mu U/ml)
BMI	Body mass index (kg/m²)
DiabetesPedigreeFunction	Genetic diabetes likelihood score
Age	Age in years
Outcome	Diabetes status (1 = diabetic, 0 = non-diabetic)

3.2 Data Preprocessing

Data preprocessing refers to the preparation and transformation of raw data into a format suitable for training machine learning models. It is an essential step in an AI or ML pipeline because it directly impacts, the performance and accuracy of the models. It consists of cleaning the data for handling missing values, removing outliers, scaling features, encoding categorical variables and splitting the data into training and test sets. The following steps were performed:

- Handling Missing Values: Biological implausible zero values in Glucose, BloodPressure, SkinThickness, Insulin, and BMI were replaced with NaN to indicate missingness. Missing values were subsequently imputed using median imputation.
- Feature Scaling: All input features were standardized using the StandardScaler. It transforms data so that the mean becomes 0 and standard deviation becomes 1. It subtracts mean of the data and divides by the standard deviation [15].
- Stratified K-Fold is a variation of the standard K-Fold cross-validation. In binary or multiclass classification tasks, it ensures that each fold maintains approximately the same percentage of samples for each class as in the full dataset. This method generates train/test indices to split the data while preserving class distribution [16].

3.3 System workflow

Figure 1 shows the system workflow of our methodology. First, the dataset is preprocessed so that the data can be sent as input to the feature selection algorithm. The selected features are then applied to the classification algorithms. The classification results of all the classifiers are combined using the Majority Voting Ensemble algorithm, which produces the final classification result.

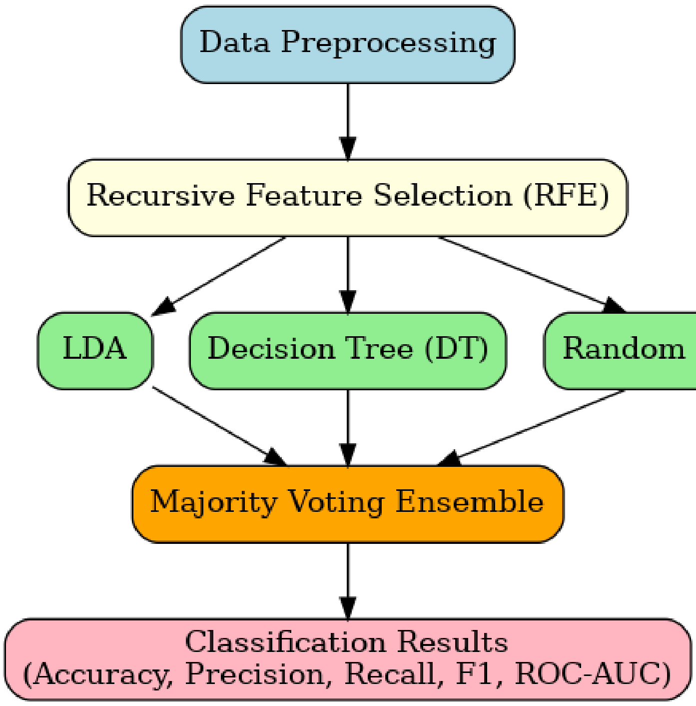

Figure 1: System workflow

3.4 Experimental Phases

The study was divided into three experimental phases to evaluate model performance under different feature scenarios and to get the optimal values for the parameters of each model.

Phase 1: Feature Selection using Recursive Feature Elimination (RFE):

RFE is an FS technique that repeatedly discards features from the dataset based on their importance to the performance of the model [17]. The model is initially trained with all the features, and their importance scores are estimated. Then the least important feature or features are discarded, and the process is continued with the rest of the features until the required number of features is reached [18]. RFE comes in handy especially when datasets have a high number of features, where identification of the most relevant subset is required for effective model training and inference procedures.

3.4.1 Parameter tuning of RFE:

The RFE algorithm's key parameters are listed in Table 4 along with their optimal values [19]. The estimator parameter specifies an ML model that is used to evaluate feature importance and eliminate features recursively; in our experiments, the estimator is Decision Tree. The step parameter indicates how many features to remove at each iteration. If the default value of 1 is used, then one feature will be removed at each iteration. The n_features_to_select parameter indicates the number of features to be selected by the algorithm. The default value none selects half of the features, which in our case is 4; thus, we have experimented with values ranging from 2 to 4 listed in Table 5.

Table 4: Key parameters and their optimal values of RFE algorithm

Parameter Name	*Optimal Value*
Estimator	DecisionTreeClassifier
step	1
n_features_to_select	2, 3, None

Table 5: Features selected by Recursive Feature Elimination.

n_features_to_select	*No. of features selected*	*Features*
None	4	Glucose, BMI, DiabetesPedigreeFunction, Age
2	2	Glucose, BMI
3	3	Glucose, BMI, Age

Phase 2: Classification Algorithms

For classification, we used three classifiers: Linear Discriminant Analysis, Decision Tree, and Random Forest. Furthermore, we applied a Majority Voting Ensemble algorithm, in which these three classifiers served as the base learners.

- *Linear Discriminant Analysis (LDA):* LDA is a supervised machine learning technique used for classification. It is useful when the intent is to find a linear combination of features that best separates two or more categories. LDA is particularly adept at optimizing class separability where the classes have a similar covariance structure. It is widely used in many fields probably because it is simple and easy to interpret. LDA operates under the assumption that features are normally distributed and that the covariance of the features is the same (homoscedasticity) for each class. LDA's goal is to find a hyperplane that maximizes the distance between classes while minimizing variance among classes [20].

- *Decision Tree (DT):* The decision tree (DT) classifier is a well-known supervised machine learning algorithm suitable for classification and regression tasks. Its appeal lies with its simplicity and speed, making it a reasonable option when transparency in decisions is needed. The algorithm forms a tree-like diagram, where the internal nodes represent decisions based on a particular feature, branches represent the outcome of the decision and leaves represent the final class label or value. DTs are created by repeatedly partitioning the training data into smaller groups according to your most distinctive features. When new data arrives, the algorithm picks a branch to follow starting from the root node and makes a series of decisions at each node based on the attributes of the data until reaching a leaf node which then tells us the classification [20,21].

- *Random Forest (RF):* An RF classifier is an ensemble learning approach that utilizes the combined power of multiple decision trees (DTs) to provide more reliable and accurate classifications. RF is popular because of its flexibility and ability to handle complex data, and as a result, it is commonly used in classification or regression in a variety of domains. The algorithm consists of a collection of DTs, and each tree is trained on a randomly selected subset of the training data. Using a method known as bagging, RF trains each tree separately on a different bootstrap sample (with replacement) of the original dataset. Additionally, at each node of a DT, a random sample of features is evaluated for generating a split, which adds variability amongst the individual trees [20, 21].

- *Ensemble learning algorithm (majority voting):* Ensemble learning is a machine learning method that improves overall accuracy and generalisation by combining predictions from multiple models. Majority voting is a simple and effective ensemble method for classification problems consisting of many base models trained on different sets of data. Each base model independently makes a prediction about the class of a given instance. Then, the ensemble makes a prediction based on a majority vote from the base models: which ever class receives the most votes is the prediction output by the ensemble classifier [19].

3.4.2 Parameter tuning of classification algorithms

Parameter tuning is the process of finding the optimal set of hyperparameters for a machine learning model. In this study, the optimal hyperparameters for all three models, as well as for the majority voting ensemble, were obtained using Python’s RandomizedSearchCV algorithm. The key parameters and their corresponding optimal values for each classifier are presented in Tables 6, 7, 8, and 9.

Table 6: Key parameters and their optimal values of DT algorithm

Parameter Name	*Optimal Values*
splitter	best
random_state	42
min_samples_split	15
min_samples_leaf	17
max_features	None
max_depth	5
criterion	gini

Table 7: Key parameters and optimal values of RF algorithm

Parameter Name	*Optimal Values*
n_estimators	40
min_samples_split	2
min_samples_leaf	8
max_features	log2
max_depth	None
bootstrap	True

Table 8: Key parameters and their optimal values of LDA algorithm.

Parameter Name	*Optimal values*
tol	0.001
store_covariance	True
solver	eigen
shrinkage	None
priors	None
n_components	1

Table 9: Optimal weights for soft voting of DT, RF, LDA algorithms respectively.

Optimal Weights	*Best Accuracy*
[1, 1, 2]	74.10%

3.5 Evaluation Metrics

Model performance was evaluated using the following metrics:

- **Accuracy:** Proportion of correctly classified instances.
- **Precision:** Proportion of true positive predictions among all positive predictions.
- **Recall (Sensitivity):** Proportion of true positives correctly identified.
- **F1-Score:** Harmonic mean of precision and recall.
- **ROC-AUC:** Area under the Receiver Operating Characteristic (ROC) curve, used to assess the model’s overall discrimination capability.

Each model was evaluated on the held-out test set, and cross-validation was also performed, where applicable, to ensure the stability and robustness of results.

4. Results and Discussion

In this section, the classification results are presented. The experiments were conducted in three stages. First, all features of the dataset were used. Next, feature subsets were generated using the Recursive Feature Elimination (RFE) algorithm. By default, when the RFE parameter n_features_to_select=None, the algorithm selects half of the features — in this case, four. Therefore, experiments were carried out using the full feature set, as well as subsets containing four, three, and two features.

4.1 Model Performance with All Features

The following table (Table 9) shows the results of three classifiers (LDA, DT and RF) and Majority Voting Ensemble algorithm with all features (8 features) of the dataset.

Table 9: Performance of classifiers (LDA, DT, RF) and the Majority Voting Ensemble using all features of the dataset.

Classifier Used	Accuracy	Precision	Recall	F1 Score	ROC AUC
LDA	76.30%	71.29%	53.73%	61.28%	71.07%
DT	72.53%	60.84%	59.70%	60.26%	69.55%
RF	74.22%	64.96%	56.72%	60.56%	70.16%
Majority Voting	77.99%	71.43%	61.57%	66.13%	74.18%

4.2 Model Performance with Selected Features

Table 10 presents the performance of three classifiers (LDA, DT, and RF) along with the Majority Voting Ensemble using the feature subset selected through RFE. In this case, the number of selected features is two: *Glucose* and *BMI*.

Table 10: Performance of classifiers (LDA, DT, RF) and the Majority Voting Ensemble using RFE-selected features (n = 2: Glucose, BMI).

Classifier Used	Accuracy	Precision	Recall	F1 Score	ROC AUC
LDA	75.26%	69.50%	51.87%	59.40%	69.83%
DT	72.40%	62.39%	52.61%	57.09%	67.81%
RF	73.05%	63.32%	54.10%	58.35%	68.65%

Majority Voting	73.83%	63.45%	58.96%	61.12%	70.38%

Table 11 presents the performance of three classifiers (LDA, DT, and RF) along with the Majority Voting Ensemble using the feature subset selected through RFE. In this case, the number of selected features is three: *Glucose*, *BMI and Age*.

Table 11: Performance of classifiers (LDA, DT, RF) and the Majority Voting Ensemble using RFE-selected features (n = 3: Glucose, BMI, Age).

Classifier Used	Accuracy	Precision	Recall	F1 Score	ROC AUC
LDA	74.74%	68.14%	51.87%	58.90%	69.43%
DT	73.05%	61.87%	59.33%	60.57%	69.86%
RF	74.48%	64.17%	60.82%	62.45%	71.31%
Majority Voting	73.31%	62.45%	58.96%	60.65%	69.98%

Table 12 presents the performance of three classifiers (LDA, DT, and RF) along with the Majority Voting Ensemble using the feature subset selected through RFE. In this case, the number of selected features is four: *Glucose*, *BMI, Age and* DiabetesPedigreeFunction.

Table 12: Performance of classifiers (LDA, DT, RF) and the Majority Voting Ensemble using RFE-selected features (n = 4: Glucose, BMI, Age, DiabetesPedigreeFunction).

Classifier Used	Accuracy	Precision	Recall	F1 Score	ROC AUC
LDA	76.17%	71.14%	53.36%	60.98%	70.88%
DT	75.39%	66.67%	58.96%	62.57%	71.58%
RF	74.61%	64.66%	60.07%	62.28%	71.24%
Majority Voting	75.13%	64.98%	62.31%	63.62%	72.16%

4.3 Model Performance with Clinical and Non-Clinical Features

To further investigate the contribution of different feature groups, the dataset was divided into clinical (Glucose, Blood Pressure, Skin Thickness, Insulin) and non-clinical (Pregnancies, BMI, Diabetes Pedigree

Function, Age) attributes. Classification models (LDA, DT, RF) and the Majority Voting Ensemble were trained and evaluated separately on these two subsets.

Table 13: Performance of classifiers (LDA, DT, RF) and the Majority Voting Ensemble using only clinical features.

Classifier Used	Accuracy	Precision	Recall	F1 Score	ROC AUC
LDA	74.61%	68.16%	51.12%	58.42%	69.16%
DT	73.57%	68.16%	45.52%	54.59%	67.06%
RF	73.31%	63.64%	49.63%	55.77%	67.21%
Majority Voting	71.61%	60.25%	54.85%	57.42%	67.73%

Table 14: Performance of classifiers (LDA, DT, RF) and the Majority Voting Ensemble using only non-clinical features.

Classifier Used	Accuracy	Precision	Recall	F1 Score	ROC AUC
LDA	65.62%	51.27%	30.22%	38.03%	57.14%
DT	67.45%	54.74%	38.81%	45.41%	60.80%
RF	68.49%	55.56%	48.51%	51.79%	63.85%
Majority Voting	68.62%	55.95%	47.39%	51.31%	63.69%

The results (Tables 13 and 14) indicate that models trained on clinical features achieved higher performance than those trained on non-clinical features, particularly in terms of accuracy and ROC-AUC. However, neither group alone outperformed the full feature set, suggesting that both clinical and non-clinical variables contribute complementary information to the detection of diabetes.

4.4 Comparative Analysis Across Phases

To provide an overall perspective, the results from all experimental phases were consolidated into a comparative summary table (Table 15). The analysis highlights several key observations:

1. All features vs. Feature Selection (RFE):

   Models trained on the full feature set consistently outperformed those trained on subsets generated by RFE (2, 3, or 4 features). This indicates that, for the PIDD dataset, feature elimination reduced useful information rather than improving generalization.

2. Clinical vs. Non-Clinical Features:

Clinical features achieved better performance than non-clinical features, but neither group alone matched the accuracy or robustness of the full feature set. This suggests that both feature types contribute complementary information to diabetes detection.

3. Majority Voting vs. Individual Classifiers:

Majority voting generally improved performance stability compared to individual classifiers, though in some cases the gain was modest. The ensemble was most effective when applied to the full feature set, but less beneficial with reduced subsets or non-clinical features.

4. Trade-offs Between Complexity and Performance:

While majority voting ensembles and Random Forests add computational complexity compared to simpler models like LDA, they consistently offered better or more reliable performance. However, the performance gap was not large enough in all cases to justify the added complexity, particularly when feature subsets were used.

Overall, the comparative analysis shows that the best performance was achieved using all features in combination with majority voting, emphasizing the importance of retaining the full set of clinically relevant attributes in small biomedical datasets such as PIDD.

5. Conclusion

Our experiments on the Pima Indians Diabetes Dataset (PIDD) demonstrate that while feature selection techniques such as Recursive Feature Elimination (RFE) are widely employed to reduce dimensionality, they did not yield performance improvements in this study. In fact, the highest accuracy (77.99%) and balanced evaluation metrics were obtained when using all original features, whereas subsets of selected features produced comparatively lower results. This suggests that in small-scale biomedical datasets like PIDD, where the total number of features is already limited, removing variables may inadvertently discard clinically meaningful information. Therefore, our findings highlight the importance of carefully evaluating the impact of feature selection, rather than assuming its universal benefit, and emphasize that retaining all features can sometimes provide more reliable results for diabetes detection.

References

1. Kamble, M. T., & Patil, S. T. (2016). Diabetes detection using deep learning approach. *International Journal for Innovative Research in Science & Technology*, *2*(12), 342-349.
2. Gujral, S. (2017). Early diabetes detection using machine learning: a review. *Int. J. Innov. Res. Sci. Technol*, *3*(10), 57-62.
3. Indoria, P., & Rathore, Y. K. (2018). A survey: detection and prediction of diabetes using machine learning techniques. *International Journal of Engineering Research & Technology (IJERT)*, *7*(3), 287-291.
4. Warke, M., Kumar, V., Tarale, S., Galgat, P., & Chaudhari, D. J. (2019). Diabetes diagnosis using machine learning algorithms. *Diabetes*, *6*(03), 1470-1476.
5. Mujumdar, A., & Vaidehi, V. (2019). Diabetes prediction using machine learning algorithms. *Procedia Computer Science*, *165*, 292-299.

6. Kopitar, L., Kocbek, P., Cilar, L., Sheikh, A., & Stiglic, G. (2020). Early detection of type 2 diabetes mellitus using machine learning-based prediction models. *Scientific reports*, *10*(1), 11981.
7. Rani, K. J. (2020). Diabetes prediction using machine learning. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *6*(4), 294-305.
8. Llaha, O., & Rista, A. (2021, May). Prediction and Detection of Diabetes using Machine Learning. In *RTA-CSIT* (pp. 94-102).
9. Farajollahi, B., Mehmannavaz, M., Mehrjoo, H., Moghbeli, F., & Sayadi, M. J. (2021). Diabetes diagnosis using machine learning. *Frontiers in Health Informatics*, *10*(1), 65.
10. Abdulhadi, N., & Al-Mousa, A. (2021, July). Diabetes detection using machine learning classification methods. In *2021 international conference on information technology (ICIT)* (pp. 350-354). IEEE.
11. Sharma, T., & Shah, M. (2021). A comprehensive review of machine learning techniques on diabetes detection. *Visual Computing for Industry, Biomedicine, and Art*, *4*(1), 30.
12. Wee, B. F., Sivakumar, S., Lim, K. H., Wong, W. K., & Juwono, F. H. (2024). Diabetes detection based on machine learning and deep learning approaches. *Multimedia Tools and Applications*, *83*(8), 24153-24185.
13. Uddin, M. A., Islam, M. M., Talukder, M. A., Hossain, M. A. A., Akhter, A., Aryal, S., & Muntaha, M. (2024). Machine learning based diabetes detection model for false negative reduction. *Biomedical Materials & Devices*, *2*(1), 427-443.
14. Harnal, S., Jain, A., Rathore, A. S., Baggan, V., Kaur, G., & Bala, R. (2023, March). Comparative approach for early diabetes detection with machine learning. In *2023 International conference on emerging smart computing and informatics (ESCI)* (pp. 1-6). IEEE.
15. scikit-learn developers. (2025, July). *StandardScaler*. Scikit-learn. Retrieved July 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
16. scikit-learn developers. (2025, July). *StratifiedKFold*. Scikit-learn. Retrieved July 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html
17. Chen, X. W., & Jeong, J. C. (2007, December). Enhanced recursive feature elimination. In *Sixth international conference on machine learning and applications (ICMLA 2007)* (pp. 429-435). IEEE.
18. Misra, P., & Yadav, A. S. (2020). Improving the classification accuracy using recursive feature elimination with cross-validation. *Int. J. Emerg. Technol*, *11*(3), 659-665.
19. Baruah, S., & Deka, V. (2025). Performance comparison of feature selection algorithms in context of P2P botnet detection. *International Journal of Information and Computer Security*, *27*(3), 379-404.
20. Baruah, S., Borah, D. J., & Deka, V. (2023). Detection of peer-to-peer botnet using machine learning techniques and ensemble learning algorithm. *International Journal of Information Security and Privacy (IJISP)*, *17*(1), 1-16.
21. Miller, S., & Busby-Earle, C. (2016, December). The role of machine learning in botnet detection. In *2016 11th international conference for internet technology and secured transactions (icitst)* (pp. 359-364). IEEE.
22. scikit-learn developers. (2025, July 15). *Recursive Feature Elimination (RFE)*. Scikit-learn. Retrieved July 15, 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html
23. scikit-learn developers. (2025, July). *DecisionTreeClassifier*. Scikit-learn. Retrieved July 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

24. scikit-learn developers. (2025, July 20). *LinearDiscriminantAnalysis*. Scikit-learn. Retrieved July 20, 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html
25. scikit-learn developers. (2025, July). *RandomForestClassifier*. Scikit-learn. Retrieved July 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
26. scikit-learn developers. (2025, July). *VotingClassifier*. Scikit-learn. Retrieved July 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html
27. Kaggle. (n.d.). *PIMA Indians Diabetes Dataset*. Kaggle. Retrieved July 2025, from https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database