Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 4 (2025)
https://www.jneonatalsurg.com

ADR-W: A Novel Accuracy-Driven Weighting Approach for Accurate and Reliable Diabetic Diagnosis

V. Usha[1], Dr. N. R. Rajalakshmi[2]

[1,2]Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, India.

Email ID: v.ushavelusamy@gmail.com

.Cite this paper as: V. Usha1, Dr. N. R. Rajalakshmi, (2025) ADR-W: A Novel Accuracy-Driven Weighting Approach for Accurate and Reliable Diabetic Diagnosis. *Journal of Neonatal Surgery*, 14 (4), 141-157.

ABSTRACT

Metabolic diseases manifest as hyperglycemia. It occurs when the corpse's humulin production is insufficient. Glycemia may be fatal if not treated appropriately and detected on time since it threatens the eyes, kidneys, nerves, heart, and blood arteries, among other important bodily organs. Research in computational diabetes has shown that machine learning can accurately predict who will get diabetes. Nevertheless, the current accuracy rate indicates that there is enough opportunity for enhancement. Using the three datasets provided, develop a machine learning system capable of diabetes prediction and diagnosis. A sequential strategy to improving predictive modeling's categorization accuracy. In the first phase, we employ preprocessing techniques such as Hot Deck Imputation (HDeckImp) to efficiently deal with missing variables and reduce classification errors. In the second stage, K-fold cross-validation is employed to ensure that the model is durable and adaptable to new scenarios. The third stage uses four traditional machine learning models to make correct predictions. In the final stage, Accuracy-Driven Reinforced Weighting (ADR-W) is applied to increase ensemble performance depending on expected accuracy. The suggested methodology aims to improve the accuracy and dependability of predictions in classification tasks. With its 98.5% accuracy on the Frankfurt dataset, the proposed model proved to be valuable. ADR-W's model to reduce processing time further enhances the system's functionality.

Keywords: Diabetes; Hot Deck Imputation; Machine learning; Accuracy Driven; Reinforced Weighting

1. INTRODUCTION

The Indian Council of Medical Research (ICMR) estimates that this condition infects over one hundred million individuals in India. To put this in perspective, India is the fifteenth most populous nation in the world based on the number of individuals with diabetes. Elevated glycemia is a precursor to diabetes mellitus, a metabolic illness. When a person dies, their body either stops producing insulin or doesn't use the insulin it does make effectively. Cells in the human body take in sugar from the blood and either put it into storage or use it as fuel, all because of the hormone insulin. Diabetes may be the cause of any functional abnormalities. However, you can protect your health by learning more about diabetes and doing everything you can to keep it under control or prevent it. Roughly 20% of those with hyperglycemia have a lifetime risk of chronic renal disease. Diabetics are at increased risk of developing kidney problems and other complications due to the damage that can occur to the blood vessels in these vital tissues caused by consistently high blood pressure, fat, and insulin levels. Furthermore, being overweight or having diabetes may increase your likelihood of developing renal failure. Cigarettes accelerate the progression of preexisting vascular disorders and induce renal failure, making them especially dangerous for diabetics.

If a person's body blood glucose level is more than one hundred and twenty-six milligrams per deciliter, or if their oral glucose tolerance test results are more than two hundred milligrams per deciliter, then they are clinically diagnosed with diabetes. Nevertheless, there may be racial differences in the glycemic threshold values used to diagnose diabetes. This is because glycemic risk levels vary among ethnic groups. As a result, doctors face the contentious dilemma of how to establish a glycemic threshold for diabetes diagnosis that does not take into account patients' ethnic backgrounds, as well as whether this threshold can be accurate enough to warrant a battery of additional tests to confirm the diagnosis. Because many blood sugar samples are required earlier and afterwards meals, making a meaningful choice in a single clinical diagnosis lingers. But there is a computational way to make diagnosis easier.

Diabetes is an auto-resistive disease that develops when the body's protective organism attacks insulin-producing beta cells. Tissues rely on the pancreatic molecule [1], which aids in glucose regulation, to metabolize blood sugar for energy. Because of this, the body experiences high blood sugar levels just before treatment. Ordinary symptoms of this eminent blood sugar include frequent urination, excessive thirst, an enlarged urge to eat, a decreased appetite, and other serious issues.

Type 2 diabetes, a chronic disease, is considered by surplus glucose in the plasma due to a malfunction in the body's glucose regulation mechanisms. This condition has the potential to create difficulties with the immunological system, neurological system, and heart and circulatory systems. Inadequate insulin sensitivity and pancreatic insulin deficiency are the main issues. Although it disproportionately affects the elderly, type 2 diabetes, sometimes recognized as insulin resistance with a teenage commencement, may occur at every stage. Type 2 diabetes does not include healing.

Pregnant women who develop glucose intolerance may also have a pregnancy-related metabolic syndrome [23]. Elevated insulin levels characterize this disorder, and failure to receive treatment poses risks to both the unborn child and the mother. When the reproductive system isn't making enough insulin to meet its increasing demands, gestational diabetes usually manifests itself around the 24th or 28th week of pregnancy. Machine learning, through the use of data, enables computers and robots to imitate human intelligence and problem-solving techniques.

Reduced glucose tolerance is a symptom of diabetes mellitus type 1. Insulin levels increase more than in type 2 diabetes. Risk factors include cardiovascular disease, stroke, and diabetes, especially type 2 diabetes. The glucose concentration should be between 100 and 125 mg per deciliter. The fibrinogen A1C level value can range from 5.7% to 6.4%. Several symptoms are associated with prudence, such as hypertension, low HDL, hyperglycemia, obesity, a large waist circumference, and excess fat. The idea of obesity is a combination of the three.

The healthcare sector has the enormous challenge of combating the diabetes epidemic. Deploying cutting-edge information and communication technology tools and methodologies for early diabetes diagnosis may be a viable option to reduce a proportion of humanity's mortality. Algorithms for mechanism knowledge are among the most worn and accurate tools for medical diagnosis. Machine learning is a field that focuses on creating and studying ways for computers to learn new things. In this context, the tenure "learning" denotes to the procedure of building a data model, intending to utilize this model to generate predictions based on novel statistics that are believed to have originated from the same population as the initial facts. Making novel inferences from the initial data, with or without human intervention, is the real test when developing systems that take into account a collection of patterns.

Algorithms develop machine-learning models by receiving data. The programs then use the trends they have discovered to gain knowledge from this information and generate assertions as well as judgments [24]. Depending on the methodologies and acquisition styles used, we can broadly classify machine learning (ML) into four primary groups. These are supervised, unsupervised, reinforced, and semi-supervised. In the supervision-based machine learning process of classifying, the predictive algorithm makes an effort to forecast the label of the incoming data. Testing as well as training information are two distinct categories into which the data is divided. The test model not only uses the training facts to train the designated model with the input information, nevertheless too evaluates the facts. During a given input set of observations' supervision-based machine learning characterization process, it aims to determine the correct label. The classification framework extensively trains on the training set, evaluates it using test data, and then generates predictions on new, unobserved data. Binary categorization and multi-class classification are two different forms of classification. There are two projected outcomes in binary classification. Several classifications predict more than a pair of results. This script utilizes various datasets that exhibit the highest degree of similarity in features. The Random Forest, XGBoost, and AdaBoost algorithms were utilized for classification, employing Accuracy-Driven Reinforced Weighting (ADR-W) to improve performance and attain high accuracy.

The primary goal of providing the Accuracy-Driven Reinforced weighing (ADR-W) method is to overcome the inadequacies of traditional ensemble weighing systems. Conventional methods including simple average or uniform weighting do not adequately differentiate among models with different accuracy. This can lead to poor performance since lower-accuracy models contribute disproportionately and hence reduce general ensemble accuracy. Moreover, stationary weight assignment is less helpful in real-world applications where model performance fluctuates since it does not dynamically change to various datasets.

This model achieves better results than prior approaches by utilizing Accuracy-Driven Reinforced Weighting (ADR-W), a method that adjusts the value of features dynamically according to classification outcomes. Particularly in important uses like diabetic prediction in healthcare, ADR-W reduces the negative impact of weaker models by dynamically changing weights depending on accuracy performance, hence improving general classification efficacy. As opposed to static or manual weighting methods, ADR-W is constantly refining weights to make them more adaptive and less biased. Across several datasets, empirical results demonstrate improved accuracy, stability, and generalizability. It is the greatest option for complex machine learning problems since it can learn and change feature contributions in real-time, guaranteeing optimal classification.

These are the principal contributions of the work is specified as underneath:

- Hot Deck Imputation was implemented to resolve the absence of values in numerous datasets.
- The computational framework was assessed using K-fold cross-validation to ensure its robustness.
- Implemented four conventional machine learning models to assess the correctness of classification and identify the most effective predictive strategy.Utilized Accuracy-Driven Reinforced Weighting (ADR-W) to enhance the ensemble model's predictive capabilities by employing accuracy-based weighting.

To choose the most excellent sculpt for diabetes prediction, it’s vital to understand how several algorithms, such as RandomForest, Adaboost, XGBoost, and Bagging, compare in terms of performance. Presented here is the remaining paper process flow: Section II gives a technical summary of the preprocessing and classification methods, and also explains the associated work. Section III lays out the methodology that will be used, and Section IV discusses performance and comparison analysis. Section V wraps up the research work by offering suggestions designed for potential investigations.

2. LITERATURE REVIEW

A. Related Work

María Teresa et al. [1] utilized deep learning approaches similar Variational Auto Encoders and Convolutional Neural Networks to improve diabetes diagnosis. Among other methods, they preprocess and classify diabetes data using SMOTE, GAN, and VAEs. Additionally, they use a Lacking Automatically generator for feature additions, and this adds more features to the dataset. By combining Sparse Auto encoders with Convolutional Classifiers, the proposed architecture achieves a higher accuracy of 92.31% compared to current techniques. In light of the limitations of the sample size, the study suggests further research using higher and extra diverse datasets to enhance generality. The results indicate that these approaches have promise as a means of diabetes detection.

Ram D. Joshi et al [2] used a logistic regression model and decision tree, an algorithm for machine learning. To forecast the occurrence of type 2 diabetes in women of Pima Indian dataset. People all across the world suffer from diabetes mellitus, a condition that has far-reaching consequences for their health and the economy. Patients might be able to take the necessary precautions and start treatment sooner if this condition could be predicted and diagnosed on time. Glucose, pregnancy, BMI, diabetes pedigree function, and age were the five primary determinants of insulin resistance that were discovered in the investigation. The analysis was supplemented and validated by exploring a classification tree. The ten-node tree found that age, glucose, BMI, pregnancy, diabetes pedigree function, and BMI were significant predictors, whereas the six-fold classification tree found that age and BMI were relevant variables. With a cross-validation error rate of 21.7%, the model had a prediction accuracy of 78.3%. If diabetes can be accurately predicted, then health policies and treatments can be developed to assist avoid the illness.

B. Shamreen Ahamed et al. [3] evaluated three ML classifiers—RF, GB, LGBM to provide a method for predicting the onset of diabetes mellitus. Oversampling and feature augmentation are two methods used in the study to deal with problems including short training datasets and unbalanced datasets. This analysis makes use of the Pima dataset housed in the UCI Repository. When pitted against the RF and GB classifiers, the LGBM algorithm proved to be the most accurate. Improving the dataset with sophisticated techniques and fine-tuning the classifiers can further increase the accuracy percentage. To variety more exact estimates, the study recommends calculating the likelihood of illness incidence.

Al Sadi and W Balachandran [4] proposed a model that routines an artificial neural network (ANN) in conjunction with six machine learning classifiers to predict which prediabetes patients in Oman would become type 2 diabetic mellitus (T2DM). Clinical data from Oman's Al Shifa health system and a prediabetes registry were utilized in the study. The Pima_Indian_Diabetes (PID) dataset, which is extensively employed, compared to this dataset. To predict type 2 diabetes, eleven clinical characteristics were taken into account. On the Oman dataset, the RF, DT model outperformed the PID dataset by 9.1% utilizing an accurateness of 98.4%. Precision, sensitivity, specificity, and accuracy were the criteria used to assess the outcomes. Using ANN and machine learning algorithms, the study found that the suggested strategy successfully distinguished between diabetic and non-diabetic individuals.

Kiran K P, Jaya Prakash et al. developed an automatic diabetes prediction utilizing 2 learning approaches is made possible by the novel data modeling framework [5] presented in the study article, which is based on correlation measurements between characteristics. The research intends to solve the problems of biomedical dataset’s lack of data and noise. The projected context employs a Deep Convolution Neural Network (CNN) in combination with a novel data modeling technique to predict diabetes accurately. To deal with anomalies, missing data, inconsistencies, and duplication, the framework contains a pre-processing step. To enhance the method's efficacy and reduce overfitting, a dependable training procedure, such 5-fold cross-validation is utilized. Experiments on the PIMA dataset show that deep CNN models achieve an accuracy of 96.13% utilizing the suggested data modeling approach, which enhances machine learning model accuracy by an average of 9%. Results show that the suggested method successfully predicts diabetes in its early stages.

Karthikeyan. R & Ramaraj et al. [6] introduces a model for early diabetes prediction that uses rule-based multi-class categorization. This research makes use of the PIMA Indians diabetes dataset and supervises mechanism learning methods. Logistic Regression (LR), Decision Tree (DT), and RepTree are the three machine learning classifiers cast off in the technique. The dataset is assessed using a binary class and contains 768 training data points. When the model's accuracy drops below 95%, rules are put in place to enhance it with multi-class classification training data and feature selection using the Information Gain Ranking Filter. The Decision Tree J48 classifier does quite well, with a 99% accuracy rate, according to the data. The model can distinguish between a healthy state, diabetes, and pre-diabetes. Finding people with pre-diabetes at an early stage is crucial in the fight against diabetes mellitus, according to the study. In sum, the study details a supervised machine learning–based Rule-driven Multi-Class Segmentation Framework that successfully predicts the start insulin resistance in its initial stages. In terms of forecasting various diabetic complications, the data show that it is quite accurate.

The research by *B. Shamreen Ahamed, Meenakshi, and others investigates* [7] the feasibility of using an ML algorithm for the forecast of kind 2 Glucose glycemia. The research evaluates many models on the PIMA Indian Dataset [19], as well as XGBoost, decision trees, gradient boosting, ExtraTrees, random forest, and LGBM. When compared to other algorithms, the LGBM algorithm's 95.20% accuracy is the best. To increase the prediction accuracy of an advanced LGBM algorithm, future research should evaluate diverse datasets and tweak their parameters.

Using machine learning methods, *C. Mallika, S. Selva muthukumaran, and colleagues* [8] developed a mixed interactive approach for rapid identification and classification of diabetes mellitus. The HOMED model recommends the use of Adaptive Principal Component Analysis (APCA) for missing value imputation, data clustering, and feature selection. For classification, an updated incremental support vector machine (ISVM) is used. The efficacy of HOMED is assessed using quality indicators such as sensitivity, accuracy, precision, particulars, good predictive value, as well as bad predictive value. Experimental results on the Pima Indian diabetes dataset show that HOMED significantly improves classification accuracy while lowering the level of processing difficulty when compared to offline methods. Healthcare providers can use the suggested system as a decision-support tool.

To estimate the beginning of diabetes, *Minakshi et.al*. [9] analyzed a wide range of physiological variables. It uses five different ML algorithms and compares how well they use various criteria. The results demonstrate that, when compared to other algorithms, decision trees and stochastic gradient boosting techniques achieve superior accuracy. The investigation also shows that those with high levels of both body mass index and hemoglobin A1c are much more likely to develop type 2 diabetes. Diabetic complications are more likely in people with a body mass index (BMI) greater than 23. They can reduce your risk of diabetes and other health problems by losing weight via exercise and a good diet.

Z. M. Alhakeem, H. Hakim, et al. [10] presented a Long-Short Term Memory (LSTM) neural network to classify Iraqi diabetics based on both physical features and medical testing. They employ the Binary Dragonfly Algorithm (BDA) to narrow down the prediction feature set to only the most important ones. The research found five key indicators of diabetes and claimed an accuracy rate of up to 98% in identifying diabetic, non-diabetic, or pre-diabetic individuals. With a 98% technique performance and a 3% accuracy rate, this method outperforms prior efforts.

B. The Research Deficit and the Suggested Solution

Handling missing data, poor generalizability, and underoptimization of ensemble learning methods are some of the issues that machine learning models encounter when predicting diabetes. To improve accuracy, a four-stage sequential technique is suggested. The first step handles missing data via Hot Deck Imputation (HDeckImp), while the second stage uses K-fold cross-validation to improve model generalizability across datasets. The third step employs four typical machine learning models to construct a robust prediction framework, while the fourth stage includes Accuracy-Driven Reinforced Weighting (ADR-W) to optimize ensemble learning and classification performance. This strategy seeks to fill gaps in existing algorithms and enhance diabetes prediction accuracy.

3. PROPOSED METHODOLOGY

To develop a machine learning-based system that can precisely predict and diagnose diabetes, the main goals of this research are to remove barriers such as missing data, enhance model generalizability, and optimize ensembles. The Proposed methodology implements preprocessing and classification techniques which are divided into 2 subsections. Initially first dataset is given to the input for the preprocessing techniques the same as exposed in Fig. 1.

The suggested technique, HDeckImp, uses various ensemble classifiers, including XGBoost, Adaboost, and Bagging, to forecast the likelihood of category 2 diabetes. At first, researchers obtained information from three different sources: Pima, the Iraqi Patient Dataset for Diabetes (IPDD), and the Frankfurt database. To clean up the dataset by filling in missing values, the data preparation approach HDeckImp was created (section 3.1). Various classifiers used to predict diabetes are then discussed in Section 3.2, followed by the remaining work added.

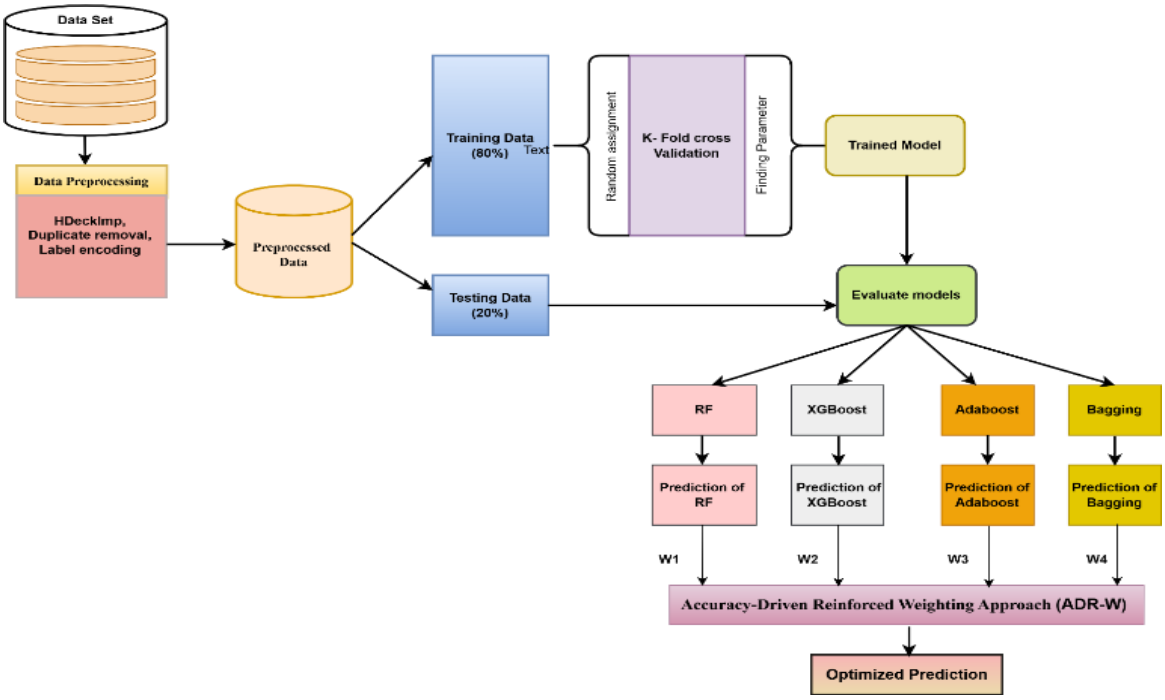


Fig. 1. Overview of the proposed work

A. Dataset

Since diabetes characterization relies on machine learning approaches and datasets, selecting an appropriate dataset has grown in importance during model training. Three datasets are used in this study. Most current data-driven diabetes diagnostic techniques are trained using the [20] Pima Indians Diabetes Database (PIDD), a publicly accessible diabetes dataset [25]. The 76 attributes of 768 female occurrences with a minimum age of 21 are documented in this 1988 dataset.

During in-hospital physical exams at Iraq's Specialized Centre for Endocrinology and Diabetes-Al-Kindy Teaching Hospital, 1,000 samples for the Iraqi Patient Dataset for Diabetes (IPDD) were collected [26]. The ages of the 565 men and 435 females included in the samples ranged from twenty-something to seventy-nine. Frankfrut dataset [27]is composed of 837 samples labeled as Diabetic (Y), 103 samples labeled as Non-Diabetic (N), and 53 samples labeled as Predicted Diabetic (P). This section highlights eleven indicators found during the physical assessment. Subtract Unrelated Data Points 173 samples were too similar to be included in the study after reviewing all 1000 data points.

B. Preprocessing

Data preparation is the first and most crucial stage in the suggested structure used for predicting diabetes illness. It improves facts quality, which in turn affects learning classification models. Remove Duplicate Samples, Convert Attributes, Fill in Missing or Null Values are all part of the preprocessing procedures in the suggested framework.

C. Casting Attributes

The class label Outcome (O) and gender (g) attributes requirements in this work data set are qualitative, not numerical. To make them usable in models, transform them to numerical values using Equation (1) and Equation (2). Category value and then assigning them a binary value of either 1 or 0. Every integer value is represented as a binary vector. A one represents the index, and all of the values are zero. The numbers 0 and 1, respectively, were assigned to the men and women. Yes, diabetic is set to 1, and no, to 0.

$$Gender\ (g) = \begin{cases} 1, if\ g = female \\ 0, if\ g = male \end{cases} \qquad (1)$$

$$Outcome\ (O) = \begin{cases} 1, if\ O = Y\ (Diabetic) \\ 0, if\ O = N\ (Non\ Diabetic) \end{cases} \qquad (2)$$

Dealing with Null or Missing Data For every class in the classification, there could be missing or null values that cause inaccurate inferences or predictions. In these data sets, there are no null values after HDeckImp. Normalization lessens the impact of characteristics with large numerical contributions on the learning process, making ensuring that all variables are properly treated. Training durations are reduced, numerical stability is improved, and meaningful feature comparisons are made simpler. Given the wide range of values for some continuous characteristics in the data, this can have a substantial impact on the classifier's performance. After removing duplicate values from this dataset2, the remaining process uses only the [827 rows x 13 columns] data.

D. HDeckImp

When datasets contain missing data, imputation is a method used to fill in the gaps. When a dataset has missing values, statisticians and data analysts often turn to hot deck imputation, a technique that involves stealing or copying values from nearby or comparable observations. Imputation utilizing data from a comparable "deck" of observations is known as a "hot deck" technique.

Finding the nearest neighbors and imputing missing values for the 'SkinThickness' variable in the PI Mellitus collection constitute the hot deck imputation procedure. Px, where 'SkinThickness' is absent from observation x, is a vector describing its characteristics. 'Pregnancies,' 'Glucose,' 'BloodPressure,' 'Insulin,' 'BMI,' 'Diabetes, Pedigree Function,' and 'Age' are all examples of characteristics that make up Q, which is utilized to detect similarities.

Using d(Px, Py) as a measure of similarity or distance between x and y data according to the characteristics Q. Definition of Sx as the collection of k-nearest neighbor indices of observation x. Mathematically, the procedure for hot deck imputation looks like this: Regarding any observation x that lacks the 'SkinThickness' represented in Eduation3.

$$S_x = argmin_{y \in all\ observations}\ d(P_x, P_y) \qquad (3)$$

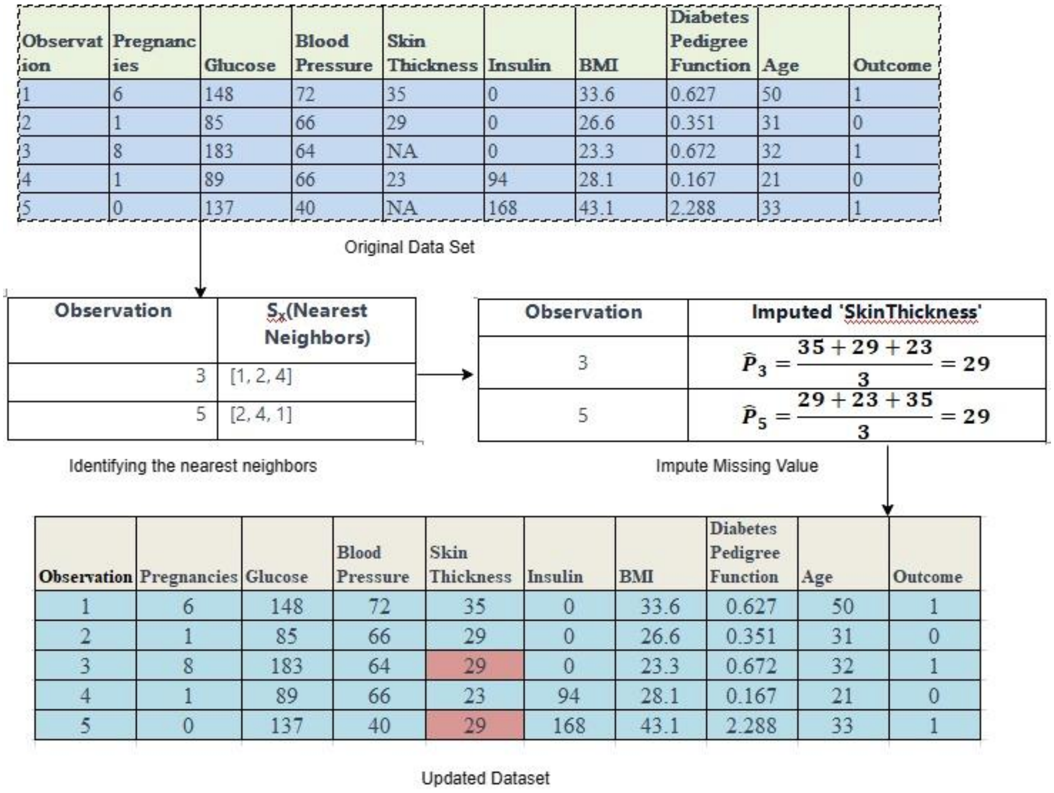
Use the values of the observations immediately around observation x to infer the 'SkinThickness' values that are missing in Equation (4).

$$\hat{P}_x = f\left(P_{y1,}, P_{y2}, \dots, P_{yn}\right) \qquad (4)$$

thus, f is an imputation function (such as the mean, median, etc.), and y1, y2,..., yk are the indices of the closest neighbors of observation x. Incorporate the imputed values into the initial dataset using Equation (5).

$$\mathrm{P}_x \rightarrow \hat{\mathrm{P}}_x \qquad (5)$$

Based on the characteristics that were provided, the imputer would locate nearby features and use an imputation approach to fill in missing values.



Observation	Pregnancies	Glucose	Blood Pressure	Skin Thickness	Insulin	BMI	Diabetes Pedigree Function	Age	Outcome
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	NA	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	NA	168	43.1	2.288	33	1

Original Data Set

Observation	S_x(Nearest Neighbors)
3	[1, 2, 4]
5	[2, 4, 1]

Identifying the nearest neighbors

Observation	Imputed 'SkinThickness'
3	$\hat{P}_3 = \frac{35+29+23}{3} = 29$
5	$\hat{P}_5 = \frac{29+23+35}{3} = 29$

Impute Missing Value

Observation	Pregnancies	Glucose	Blood Pressure	Skin Thickness	Insulin	BMI	Diabetes Pedigree Function	Age	Outcome
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	29	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	29	168	43.1	2.288	33	1

Updated Dataset

Fig. 2. Hot Deck Imputation Process

Fig. 2. illustrates the procedure for 'SkinThickness' variable imputation using a hot deck in the Pima Indian Diabetes dataset. It updates the original dataset with the imputed values after identifying the nearest neighbors and imputed the missing values.

Eighty percent of the data is utilized for training, according to Data Separation Equation 6, which is employed for this work's Test Training Split. Twenty percent of the data are used for evaluation, which helps to train the algorithm using real-world examples. Used to evaluate if it can be applied to new scenarios in the future.

E. k-fold Cross-Validation

Once imputation is complete, divide the data into k folds. Select a number with k that works with the information's quantity and processing restrictions; however, 5 or 10 are frequent possibilities but here given 10. One statistical method for comparing and contrasting the performance of classifiers in ML algorithms is k-fold cross-validation, or k-fCV. It partitions the data into two sets: one for model training and another for testing or validation. For k-fCV to work, the data must first be folded into k-equal segments. Following that, k-training and validation iterations are carried out, with each iteration utilizing a distinct data fold for validation and the remaining k-onefold for training. Within k-fold cross-validation, for every fold: Consider a single fold to be the control set. Train a machine learning model using the remaining k-1 folds. Imputed training data can be used to train the model. Used the imputed validation data to test a model.

F. Machine learning classifiers

For diabetes prediction, Machine Learning models were taken into account in this investigation. Here are the Machine Learning models used for the experiments. NaiveBayes, LogisticRegression, SupportVectorMachine, K-NearestNeighbors, GradientBoot, AdaBoost, XGBoost, Bagging, DecisionTree(DS), RandomForest.

1. The Random Forest (RF)

An ensemble of decision trees that have each been trained on a random portion of the data is used in the Random Forest method of diabetes classification. Randomization in feature selection reduces overfitting, enhancing model robustness. In the classification process, each tree "votes," and the majority decision determines the final prediction. RF effectively handles noisy data, outliers, and missing values, making it suitable for healthcare datasets. Feature importance analysis helps identify key contributors to predictions. Its balance between bias and variance ensures stable performance across diverse datasets. RF is scalable to large datasets and high-dimensional spaces. While individual trees are interpretable, the ensemble nature may reduce interpretability. RF often performs well with minimal hyperparameter tuning, making it a popular choice for diabetic prediction. A be the feature matrix. b be the target variable (class labels). DS is the dataset, where $\mathrm{DS} = \{(\mathrm{A}_1, \mathrm{b}_1), (\mathrm{A}_2, \mathrm{b}_2), \dots\dots, (\mathrm{A}_\mathrm{M}, \mathrm{b}_\mathrm{M})$. A_{train}, A_{test}, b_{train}, b_{test} be the training and testing splits of A and b. $RF_i(P)$ be the prediction of the i-th decision tree. The split the dataset

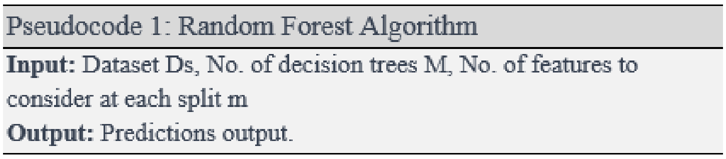
$$DS = DS_{train} \ \cup \ DS_{test} \qquad (6)$$

Here $DS_{train} = \{(A_{train,} b_{train})\}$, $DS_{test} = \{(A_{test,} b_{test})\}$

The Random Forest model consists of M decision trees in Equation (7).

$$Random\ Forest(RF(P)) = \{rf_1\,(P), rf_2(P), \dots\dots, rf_M\,(P)\} \qquad (7)$$

For each decision tree i. $RF_i(P)$ is trained on a bootstrapped subset of $DS_{train.}$



Pseudocode 1: Random Forest Algorithm

Input: Dataset Ds, No. of decision trees M, No. of features to consider at each split m

Output: Predictions output.

```
TrainDecisionTree($DS_{train}$, m):
Sample a bootstrap subset $DS_{traini}$ from $DS_{train}$.
   from the total features randomly select m features
   Train a decision tree $DT_i$ using $DS_{traini}$ and the selected features.
   Return $DT_i$
PredictDecisionTree(DT, P):
  prediction DT(P) using the trained decision tree DT, Return the prediction.
TrainRandomForest($DS_{train}$, m, M):
  Create a blank list RF at first.
  Regarding i = 1 through M:
Train decision tree $RF_i$ using TrainDecisionTree($DS_{train}$, m).
```

```
      Append $RF_i$ to the list RF and return.
PredictRandomForest(RF, $DS_{test}$):
    Initialize empty list predictions.
   For each instance ($A_{testi}$, $b_{testi}$) in $DS_{test}$:
      Initialize an empty list tree_predictions.
      For each decision tree $RF_i$ in RF:
    Make a prediction using PredictDecisionTree($RF_i$, $A_{testi}$).
          Append the prediction to tree_predictions.
   For classification, majority voting among tree_predictions.
           Final Prediction = mode (tree_predictions)
   For regression, the average of tree_predictions.
          Final Prediction = average(tree_predictions)
   Append the Final Prediction to predictions.
$DS_{train}$, $DS_{test}$ = split_dataset(DS)
RF = TrainRandomForest($DS_{train}$, m, M)
predictions = PredictRandomForest(RF, $DS_{test}$)
```

2. XG Boost

As a high-performance boosting method, eXtreme Gradient Boosting optimizes itself by minimizing the loss function, which can be achieved in several different ways. This technique iteratively adds models to a community using gradient boosting. Focusing on challenging cases when the model fails to provide an accurate prediction is the fundamental idea underlying boosting. By manipulating the distribution of observations in such a way that certain measures seem likely in a sample, we may highlight these cases even more. In light of this, the next weak student will put more effort into accurately predicting challenging occasions. To create XGBoost, a strong predictor, all of the basic prediction rules were combined into one holistic model.

E={e1,e2,e3…..e_n} Set of base learners

The final Prediction is $\widehat{x_\iota} = \sum_{n=1}^{N} fn(y_j)$ (8)

Equation (8) the final forecast is the result of adding together all the trees' predictions: where $fn(y_j)$ is the forecast of the n-th tree for the j-th run.

3. Adaboost

AdaBoost, which stands for "Adaptive Boosting," is a method for ensemble learning that strengthens classifiers by combining the predictions of weaker learners, most often decision trees. One of the main ideas underlying AdaBoost is to use weights to categorize the training cases. The weights may be adjusted in each iteration to target the instances that are more challenging to distinguish accurately. The AdaBoost prediction mechanism is mathematically represented here: The AdaBoost model's prediction for a new input p is supplied by a collection of weak classifiers $g_i(P)$ where i= 1, 2…I their corresponding weights α_i.

$G(p) = Sign(\sum_{i=1}^{I} \alpha_i \; .g_i(P))$ (9)

(G(p)): The AdaBoost model's final forecast for input p. When the argument is positive, the value returned by the sign (.) in Equation (9) returns 1, when it's low, -1, and 0 when it is zero. I: The overall count of students who struggle to learn.

4. Bagging

Bagging, which stands for Bootstrap Aggregating, is an ensemble method in machine knowledge used to boost model accuracy plus stability. Machine knowledge combines the predictions of many copies of a basic model that have been trained on separate subsets of the training data. Since it presumably has Ba basic models S_1, S_2,…….S_{Ba} trained on bootstrap samples from the dataset DS_1, DS2,…..,$DS_{Ba.}$ To forecast the next occurrence y. For classification tasks, every base model S_i generates a prediction for a class label given in Equation (10).

$Prediction = mode(S_1(\mathrm{y}), S_2(\mathrm{y}), \ldots, S_{\mathrm{Ba}}(\mathrm{y}))$ (10)

In which the class that appears most often in the predictions is represented by the mode.

G. Assessment of Performance

1) Accuracy (Accy): When discussing categorization issues, accuracy is a frequent performance measure. By determining the proportion of cases for which predictions were accurate about the whole numeral of occurrences, it evaluates a model's overall accuracy. Percentage is a straightforward and easy-to-understand way to measure a model's accuracy. The accuracy is represented by the Equation (11).

$$\text{Accuracy(Accy)} = \frac{\text{TrPosi+TrNeg}}{\text{TrPosi+FaNeg+FaPosi+TrNeg}} \qquad (11)$$

TrPosi stands for "True Positives," which is the total amount of positive predictions that were accurate.

TrNeg is the number of cases that can be reliably anticipated to be negative.

The term "False Positives" (FaPosi) refers to the number of times a negative result is anticipated based on a good one.

The term FaNeg (FalseNegatives) refers to the quantity of events that are predicted as negative but turn out to be positive.

2) Precision (Preci): The sensitivity of a classifier is defined in Equation (12) as the proportion of genuine positive observations to all observations made in the actual class.

$$\text{Preci} = \frac{\text{TrPosi}}{\text{TrPosi+FaPosi}} \qquad (12)$$

3)Recall (Reca): One measure of a model's performance in classifying data is its recall, which is sometimes called sensitivity. Equation (13) it evaluates how well the model can detect all occurrences that belong to a positive class. To get a sense of how well a model can catch all real positive cases, the ratio of true positives to the sum of real positivity and false negatives is how it is calculated.

$$\text{Reca} = \frac{\text{TrPosi}}{\text{TrPosi+FaNeg}} \qquad (13)$$

4) F1 Score (F1 Sco): Equation (14) F1 Sco offers a single estimate of a model routine, integrating recall and accuracy, particularly useful in uniform class distributions or high false positive/negative importance.

$$\text{F1 Sco} = 2 * \frac{\text{Preci.Reca}}{\text{Preci+Reca}} \qquad (14)$$

The score of F1 is a useful statistic for addressing the conflict between fake positives and fake negatives. It is computed as the mean of the harmonics of accuracy and memory. It penalizes excessive levels of precision or recall, ensuring a balanced evaluation.

5) *ConFusion Matrix (CFM)*: To measure how well a machine learning model performs in a classification task, a confusion matrix is a useful table represented in Equation (15). When applied to other classes, these measures illuminate the model's relative merits and shortcomings.

$$CFM = \begin{bmatrix} TrNeg & FaPosi \\ FaNeg & TrPosi \end{bmatrix} \qquad (15)$$

6) Matthews Correlation Coefficient (Matthews Corr Coef): When dealing with imbalanced datasets, Mattews Corr Coef is particularly useful since it equalizes the influence of different components in the confusion matrix, painting a more comprehensive view of a model's performance in Equation (16). The range of Mattews Corr Coef is -1 to 1, specifically: The forecast is spot on if the value is 1. When the result is 0, the forecast is essentially random. While a value of -1 means that there is a complete discrepancy between the two.

$$\text{Matthews Corr Coef} = \frac{Tr\,Pos * Tr\,Neg - Fa\,Pos * Fa\,Neg}{\sqrt{(Tr\,Pos + Fa\,Pos)(Tr\,Pos + Fa\,Neg)(Tr\,Neg + Fa\,Pos)(Tr\,Neg + Fa\,Neg)}}$$

(16)

H. Accuracy-Driven Reinforced Weighting (ADR-W)

One ensemble learning method that uses the relative accuracy of various classifiers to enhance model predictions is Accuracy-Driven Reinforced Weighting (ADR-W), which essentially gives more weight to classifiers with better accuracy and less weight to models with lower accuracy. This approach ensures that less robust models do not diminish overall performance.

This approach ensures that less robust models do not diminish overall performance. By using a non-linear, exponential weighting scheme, the ADR-W approach can bolster ensembles of high-accuracy models. By giving more weight to models with better performance, ADR-W improves the final classification accuracy, in contrast to typical averaging methods.

Where Acc_i denotes the accuracy of the i-th classifier in the ensemble. In ADR-W, there are two main processes for assigning weights:

Accuracy Scaling Using Exponential Reinforcement: An exponential transformation is used to compute the raw weight for each model, ensuring that high-accuracy classifiers are given higher emphasis:

$$W_i = e^{\beta . Acc_i} \qquad (17)$$

The initial weight to the ith model, denoted as W_i calculated using Equation (17). The accuracy of the model is $Acc_{i,}$ β is a reinforcement factor that regulates the prominence given to higher accuracies value is 49.9.

Normalize the weighted total must be one. Computed the raw weights using Equation (18).

$$w_i = \frac{W_i}{\sum_{j=1}^{N} W_j} \qquad (18)$$

Where the final optimized weight for the i^{th} classifier is $w_{i.}$ Several models are termed as N.

The Scaling of Accuracy To find the best weights for several classifiers according to their accuracy, the Exponential Reinforcement approach was used. This method guarantees that the final prediction gives more weight to the models with the highest accuracy. The equation (17) was used to calculate the raw weight (W_i) for every model through an exponential transformation with a reinforcement factor (β) set to 49.9. Because of this change, the weighting procedure gives greater weight to numbers with higher precision. The final optimized weights were obtained by normalizing the computed raw weights using Equation (18) to make sure that the entire sum of weights was one. Following this procedure, the following classifier weights were obtained: Bagging (0.1716), AdaBoost (0.00002), Random Forest (0.4656), and XGBoost (0.3628). Based on these weights, it is clear that Random Forest and XGBoost were the most influential and accurate, while AdaBoost had very no weight at all. The total performance was maximized by computing the final ensemble accuracy using the weighted sum of individual model accuracies.

Different models in an ensemble classification scenario to validate the ADR-W approach. The corresponding accuracies of these models are presented in Table 1. To determine the final ensemble accuracy, in Equation (19) which is denoted by the acronym $Acc_{ensemble}$, the weighted total of the individual model accuracies is utilized.

$$Acc_{\text{ensemble}} = \sum_{i=1}^{N} w_i * Acc_i \qquad (19)$$

Where the final ensemble accuracy is represented as Acc_{final} .The total number of the models is represented as N. ith model accuracy and the weight is represented as Acc_i and w_i. This ensures that the final prediction is more heavily weighted by the models with good accuracy. Same as when only the remaining performance metrics are calculated.

Table 1 value metric values are applied to Equation (17) and Equation (18). After getting values are applied in Equation (19).

I. Final Weighted Ensemble Calculation

To compute the ensemble values for each metric are given below:

Ensemble Accuracy

(0.20886937×0.979)+(0.26805952×0.984)+(0.26805952×0.984)+(0.2550116×0.983)

=0.2045+0.2637+0.2637+0.2507=**0.9826**

Ensemble Precision

=(0.20886937×0.988)+(0.26805952×0.994)+(0.26805952×0.994)+(0.2550116×0.992)

=0.2065+0.2664+0.2664+0.2530=**0.9923**

Ensemble Recall

(0.20886937×0.988)+(0.26805952×0.988)+(0.26805952×0.988)+(0.2550116×0.987)

=0.2065+0.2650+0.2650+0.2517=**0.9882**

Ensemble F1 Score

(0.20886937×0.988)+(0.26805952×0.991)+(0.26805952×0.991)+(0.2550116×0.990)

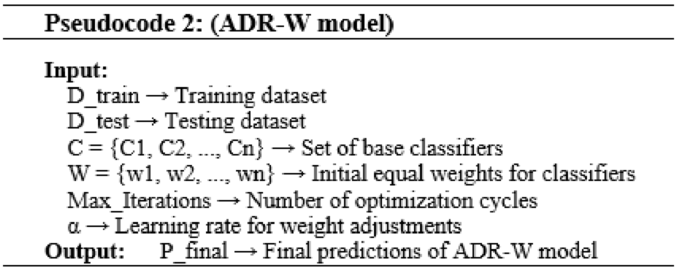
=0.2065+0.2657+0.2657+0.2525=**0.9904**

Ensemble Kappa

(0.20886937×0.888)+(0.26805952×0.917)+(0.26805952×0.917)+(0.2550116×0.910)

=0.1854+0.2458+0.2458+0.2320=**0.9090**

The pseudocode 2 demonstrates the Accuracy-Driven Reinforced Weighting (ADR-W) approach, which improves ensemble classification by dynamically modifying classifier weights based on accuracy. Initially, classifiers are trained and assigned weights based on their accuracy. These weights are iteratively adjusted using a reinforcement process, which prioritizes high-performing classifiers while minimizing the influence of weaker ones. The final prediction is based on a weighted aggregation of classifier results. This method enhances classification accuracy while decreasing processing time by maximizing ensemble contributions.



Pseudocode 2: (ADR-W model)

```
Input:
   D_train → Training dataset
   D_test → Testing dataset
   C = {C1, C2, ..., Cn} → Set of base classifiers
   W = {w1, w2, ..., wn} → Initial equal weights for classifiers
   Max_Iterations → Number of optimization cycles
   α → Learning rate for weight adjustments
Output:    P_final → Final predictions of ADR-W model
```

```
Step 1: Train Classifiers
   For each classifier Ci in C:
      Train Ci on D_train
Step 2: Compute Initial Accuracy
   For each classifier Ci in C:
      Predict on D_test → Get predictions Pi
      Compute Accuracy Ai using ground truth labels
Step 3: Initialize Weights
   For each classifier Ci in C:
      wi = Ai / sum(A) → Normalize weights based on accuracy
Step 4: Reinforce Weights Iteratively
   For iteration = 1 to Max_Iterations:
      For each classifier Ci in C:
         Predict on D_test → Get updated predictions Pi
         Compute new accuracy Ai
         Update weight wi using:
            wi = wi + α * (Ai - wi)
      Normalize W such that sum(W) = 1
Step 5: Compute Final Ensemble Prediction
   For each sample x in D_test:
      Weighted prediction:
         P_final(x) = sum (wi * Pi(x)) for all i
      Assign the final class label using majority voting or thresholding.
```

4. RESULT AND DISCUSSION

TABLE I. illuminates that the XGBoost and AdaBoost classifiers had the best accuracy of 98.4%, together with precision (99.4%) and F1 Score (99.1%). These models also showed a great deal in classification results with a Kappa score of 91.7%. With an accuracy of 97.9%, the Random Forest model did somewhat poorly; Bagging achieved an accuracy of 98.3%, equivalent to the recommended ADR-W approach.

An analysis of the ML framework's efficiency with ADR-W using the IPDD dataset

Data Set Name	Name of classification algorithm	Accuracy (Accy) %	Precision (Preci) %	Recall (Reca) %	F1 Score (F1 Sco) %	Kappa %
Iraqi Patient Dataset for Diabetes (IPDD)	Random Forest (RanF)	0.979	0.988	0.988	0.988	0.888
	XGBoost	0.984	0.994	0.988	0.991	0.917
	AdapBoost	0.984	0.994	0.988	0.991	0.917
	Bagging	0.983	0.992	0.987	0.990	0.910
	Proposed method (ADR-W)	0.983	0.992	0.988	0.990	0.909

A 99.0% F1 score, 98.3% accuracy, 99.2% precision, and 98.8% recall were obtained by the ADR-W approach, which gives more weight to more accurate models. Though its accuracy is on par with Bagging, it guarantees that every model has equal weight in the ensemble approach. Although

XGBoost and AdaBoost beat all the other models overall, the ADR-W method provides a more balanced ensemble approach over several evaluation criteria.

The following Fig. 3. depicts the methodology applied to assess the performance of each model:

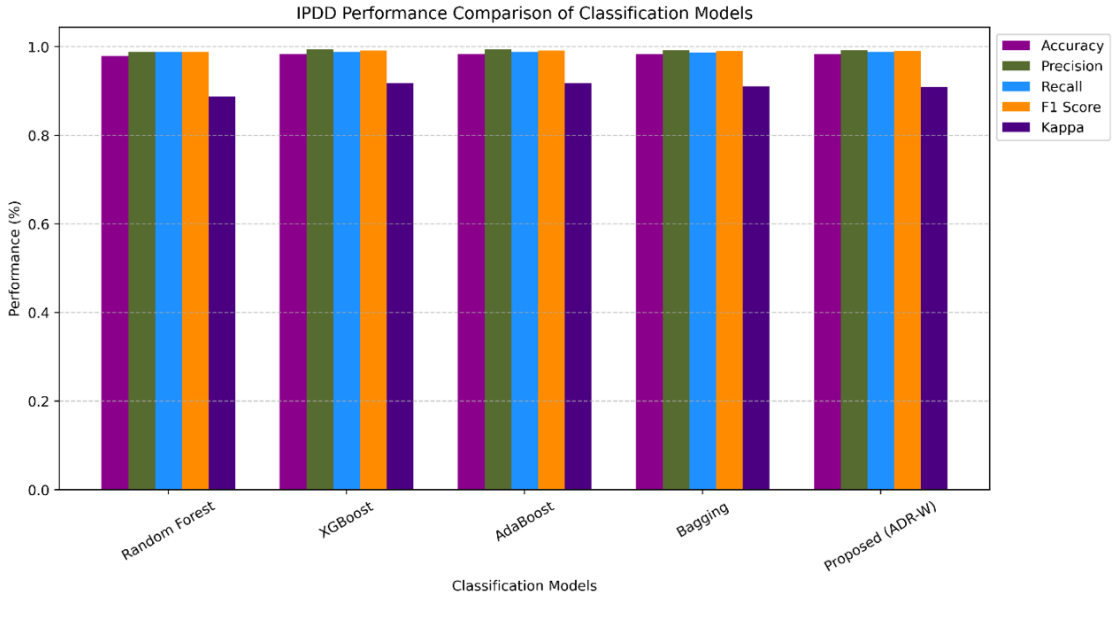


Fig. 3. Classification Model Performance Analysis on IPDD Diabetes Dataset

The top classifiers, according to the results, are XGBoost and AdaBoost, which reach a combined accuracy and precision of almost 98.4 and almost 99.4 percent, respectively. When compared to other models, including Bagging, the ADR-W (Proposed Method) performs competitively. Despite Random Forest's somewhat lower accuracy (about 97.9%), it continues to demonstrate strong performance in other criteria. Notably, Kappa values show larger variation, reflecting disparities in agreement levels among classifiers.

TABLE II. presents an analysis of the accuracy, precision, recall, F1 score, and kappa values of various classification algorithms applied to the PIMA dataset. Among the individual classifiers, Random Forest achieved the highest accuracy of

77.2%, followed by Bagging at 76.2%, AdaBoost at 75.2%, and XGBoost at 73.9%.

The proposed ADR-W method demonstrated a slight improvement over Bagging and AdaBoost, achieving an accuracy of 76.3%. The ADR-W approach was implemented using exponential reinforcement, where weights were dynamically assigned to classifiers based on accuracy. The highest precision (68.6%), recall (60.6%), and F1-score (63.9%) were obtained through this method, ensuring a well-balanced handling of false positives and false negatives.

Table II: Proposed ML framework performance on PIMA dataset

Data Set Name	Name of classification algorithm	Accuracy (Accy)%	Precision (Preci) %	Recall (Reca) %	F1 Score (F1 Sco) %	Kappa %
PIMA dataset	Random Forest (RanF)	0.772	0.705	0.610	0.650	0.483
	XGBoost	0.739	0.636	0.597	0.613	0.417
	AdapBoost	0.752	0.669	0.584	0.621	0.439
	Bagging	0.762	0.681	0.615	0.641	0.466
	Proposed method (ADR-W)	0.763	0.686	0.606	0.639	0.464

However, the kappa statistic (46.4%) indicates moderate agreement between the predicted and actual values. Due to its ability to emphasize high-performing models, the ADR-W technique proves to be effective in diabetic classification tasks, leading to a slight increase in accuracy and precision when compared to individual classifiers.

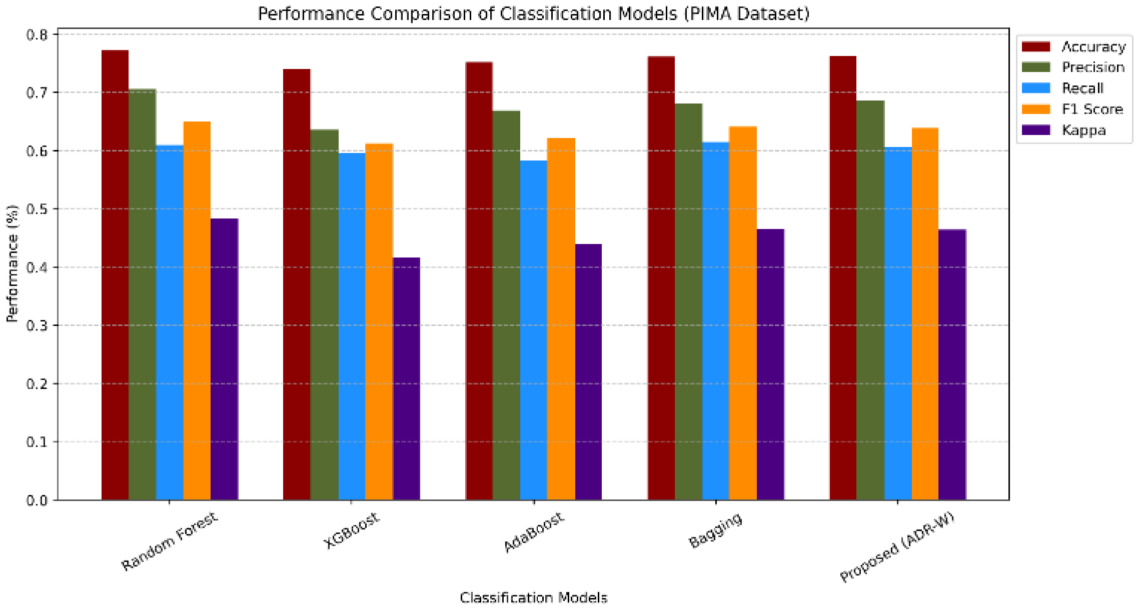


Fig. 4. Classification Model Performance Analysis on PIMA Diabetes Dataset

Fig. 4. depicts that while Random Forest and ADR-W demonstrate high accuracy, XGBoost performs comparatively lower in certain metrics. Though the Kappa values are lower and exhibit variation in agreement, the Precision and F1 Score are always greater across models. ADR-W is a good classification candidate since it performs competitively when compared to Bagging and AdaBoost.

TABLE III. represents the performance of various classification algorithms on the Frankfurt dataset is presented, with their effectiveness measured using different metrics. Among all the classifiers, the highest accuracy of 99.0% was achieved by Random Forest, followed by XGBoost (98.5%), Bagging (97.0%), and AdaBoost (79.0%

TABLE III: Proposed ML framework performance on Frankfurt dataset

Data Set Name	Name of classification algorithm	Accuracy (Accy) %	Precision (Preci) %	Recall (Reca) %	F1 Score (F1 Sco) %	Kappa %
Frankfurt dataset	Random Forest (RanF)	0.990	0.986	0.986	0.986	0.985
	XGBoost	0.985	0.973	0.986	0.979	0.978
	AdapBoost	0.790	0.756	0.632	0.688	0.687
	Bagging	0.970	0.972	0.945	0.958	0.957
	Proposed method (ADR-W)	0.985	0.979	0.979	0.979	0.978

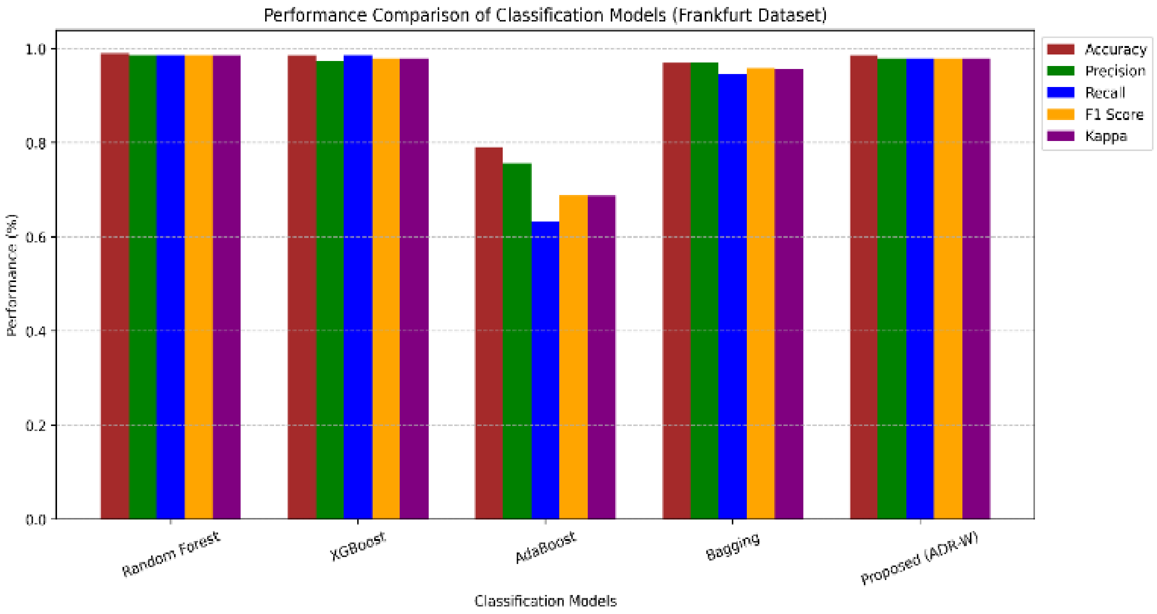


Fig. 5. Classification Model Performance Analysis on Frankfrut Diabetes Dataset

Fig. 5. depicts the proposed ADR-W method attained an accuracy of 98.5%, which is comparable to that of XGBoost, by dynamically assigning weights to classifiers based on their performance. In terms of precision (97.9%) and recall (97.9%),

ADR-W exhibited a balanced performance, resulting in an F1 score of 97.9%, reflecting consistency in both false positive and false negative rates. The kappa statistic (97.8%) indicates a high level of agreement between predicted and actual values, demonstrating the reliability of the ADR-W technique.

TABLE IV. Comparison of Classification Accuracy on the Iraqi Diabetes (IPDD) Dataset

Ref.No	Author	Dataset	Classifiers used	Accuracy(%)
[11]	Abnoosian, K, 2023	Iraqi Diabetes (IPDD) ,Mendeley	k-NN, AdapBoost, DT, SVM, RF, and GNaiveB	98.87, 98.61, 97.92, 98.51, and 99.9
[9]	Minakshi Ravindra	Iraqi Diabetes (IPDD) ,Mendeley	Stochastic gradient boost also DT	97.04%, 95.07%
	Proposed Model	Iraqi Diabetes (IPDD) ,Mendeley	ADR-W	98.3%

TABLE IV. summarizes the results of comparing the accuracy of several categorization methods on the Iraqi Diabetes (IPDD) dataset. In terms of accuracy, k-NN(98.87%), SVM(98.61%), and AdaBoost (98.51%) were the top three, whereas Random Forest (RF) from Abnoosian et al. (2023) [11] came in second and third, respectively, with 99.9%. With an accuracy rate of 98.3%, the suggested ADR-W model is a

viable substitute for conventional classifiers. In comparison to Stochastic Gradient Boosting (97.04%) and Decision Tree (95.07%), the ADR-W method guarantees dynamic weight adjustments, offering a balanced and flexible approach to prediction.

TABLE V. Comparison of Classification Accuracy on the PIMA dataset.

Ref.No	Author	Dataset	Algorithm used	Accuracy(%)
[12]	Naz and Ahuja et al. (2020)	PIMA	ANN	90.34%
[13]	García-Ordás, M.T., Benavides(2021)	PIMA	CNN	92%
[14]	Kishor, A. and Chakraborty, C., 2021	PIMA	SVM,RF	83.65,97.8
[15]	Shrestha, M., Alsadoon, 2023	PIMA	SVM,RBF kernel and the LSTM	82.7,83.0
[16]	Ganie, S.M. and Malik, M.B., 2022	PIMA	Bagging	99.4
[17]	Madan, P., Singh, V (2022)	PIMA	CNN-Bi-LSTM	98
	Proposed Work	PIMA	ADR-W	76.3

TABLE V. lays out numerous machine-learning techniques to the PIMA dataset for diabetes prediction. Beghriche (2020) attained 90.34% accuracy with ANN [12], García-Ordás

(2021) reached 92% with CNN [13], and Kishor (2021) obtained 97.8% with RF and 83.65% with SVM [14]. Ganie (2022) achieved the maximum accuracy of 99.4% using Bagging [16]. The proposed ADR-W model achieved 76.3%.

TABLE VI. Comparison of Classification Accuracy on the Frankfrut dataset.

Ref.No	Author	Dataset	Classifiers used	Accuracy(%)
[21]	Beghriche, T2023	Frankfurt	DNN	99.75
[18]	Edeh, M.O,2022	Frankfurt	RF,	97.6
[22]	Azbeg, K,2022	PIMA, Frankfurt, Fusion of 2 dataset	RF	85.9%, 99.5%, and 99.8%
	Proposed Model	Frankfurt	ADR-W	98.5

TABLE V. provides the [21] achieved maximum accuracy (99.75%) using a Deep Neural Network (DNN) on the Frankfurt dataset, according to Table 6. Azbeg (2022) used RF on a fusion of datasets and reached 99.8% accuracy. On the Frankfurt dataset, the suggested ADR-W model attained a 98.5% accuracy rate.

5. CONCLUSION AND FUTURE ENHANCEMENT

Diabetic complications can be avoided with prompt diagnosis, which is still a big concern in global health. Based on their performance on the PIMA and Frankfurt datasets, ensemble methods such as RF, XGBoost, AdaBoost, and Bagging outperform other classification algorithms in this research work. With RF, accuracy was 99.8% on the fusion dataset and 99.4% on PIMA for bagging. The suggested ADR-W model performed admirably, with an accuracy rate of 98.5% on the Frankfurt dataset. The results show that ensemble learning methods work great for predicting diabetes, which is great news for healthcare automation. Improving model accuracy through the integration of three datasets and the application of ensemble-based feature selection will be the focus of future research. To determine which classifiers are most effective for diabetes prediction, optimization methods will be employed. The integration of real-time clinical datasets will further ensure the practical validation of the model's performance. Better and more efficient diabetes prediction systems are the goal of this technique, which seeks to enhance model generalization.

REFERENCES

[1] Teresa Valenzuela, M., Urquidi, C., Rodriguez, N., Castillo, L., Fernandez, J. and Ramirez, E., 2021. Development of neutralizing antibody responses against SARS-CoV-2 in COVID-19 patients. Journal of Medical Virology, 93(7), pp.4334-4341.

[2] Joshi, R.D. and Dhakal, C.K., 2021. Predicting type 2 diabetes using logistic regression and machine learning approaches. International journal of environmental research and public health, 18(14), p.7346

[3] Ahamed, B.S., Arya, M.S. and Nancy V, A.O., 2022. Prediction of type-2 diabetes mellitus disease using machine learning classifiers and techniques. Frontiers in Computer Science, 4, p.835242

[4] Al Sadi, K. and Balachandran, W., 2023. Prediction Model of Type 2 Diabetes Mellitus for Oman Prediabetes Patients Using Artificial Neural Network and Six Machine Learning Classifiers. Applied Sciences, 13(4), p.2344

[5] Patro, K.K., Allam, J.P., Sanapala, U., Marpu, C.K., Samee, N.A., Alabdulhafith, M. and Plawiak, P., 2023. An effective correlation-based data modeling framework for automatic diabetes prediction using machine and deep learning techniques. BMC bioinformatics, 24(1), p.372.

[6] Karthikeyan .R,P Geetha,Karthikeyan Ar,2022. The Rule-based Multi-class Classification Model Predicts Early Diabetes Using Supervised Machine Learning Techniques, Dongbei Daxue Xuebao/Journal of Northeastern University

[7] Ahamed, B.S., Arya, M.S., Sangeetha, S.K.B. and Auxilia Osvin, N.V., 2022. Diabetes Mellitus Disease Prediction and Type Classification Involving Predictive Modeling Using Machine Learning Techniques and Classifiers. Applied Computational Intelligence and Soft Computing, 2022.

[8] Mallika, C. and Selvamuthukumaran, S., 2022. Hybrid Online Model for Predicting Diabetes Mellitus. Intelligent Automation & Soft Computing, 31(3).

[9] Kumari, Minakshi, et al. "Evaluation of Lipid Profile and Glycated Haemoglobin in Type 2 Diabetic Patients: A Retrospective Study." International Journal of Health Sciences, no. II, 13 Apr. 2022, pp. 4372-4383, doi:10.53730/ijhs.v6nS2.6009.

[10] Alhakeem, Z.M., Hakim, H., Hasan, O.A., Laghari, A.A., Jumani, A.K. and Jasm, M.N., 2023. Prediction of diabetic patients in Iraq using binary dragonfly algorithm with long-short term memory neural network. AIMS Electronics & Electrical Engineering, 7(3).

[11] Abnoosian, K., Farnoosh, R. and Behzadi, M.H., 2023. Prediction of diabetes disease using an ensemble of machine learning multi-classifier models. BMC bioinformatics, 24(1), p.337.

[12] Naz, H. and Ahuja, S., 2020. Deep learning approach for diabetes prediction using PIMA Indian dataset. Journal of Diabetes & Metabolic Disorders, 19, pp.391-403.

[13] García-Ordás, M.T., Benavides, C., Benítez-Andrades, J.A., Alaiz-Moretón, H. and García-Rodríguez, I., 2021. Diabetes detection using deep learning techniques with oversampling and feature augmentation. Computer Methods and Programs in Biomedicine, 202, p.105968.

[14] Kishor, A. and Chakraborty, C., 2021. Early and accurate prediction of diabetics based on FCBF feature selection and SMOTE. International Journal of System Assurance Engineering and Management, pp.1-9.

[15] Shrestha, M., Alsadoon, O.H., Alsadoon, A., Al-Dala'in, T., Rashid, T.A., Prasad, P.W.C. and Alrubaie, A., 2023. A novel solution of deep learning for enhanced support vector machine for predicting the onset of type 2 diabetes. Multimedia Tools and Applications, 82(4), pp.6221-6241.

[16] Ganie, S.M. and Malik, M.B., 2022. An ensemble machine learning approach for predicting type-II diabetes mellitus based on lifestyle indicators. Healthcare Analytics, 2, p.100092.

[17] Madan, P., Singh, V., Chaudhari, V., Albagory, Y., Dumka, A., Singh, R., Gehlot, A., Rashid, M., Alshamrani, S.S. and AlGhamdi, A.S., 2022. An optimization-based diabetes prediction model using CNN and Bi-directional LSTM in real-time environment. Applied Sciences, 12(8), p.3989.

[18] Edeh, M.O., Khalaf, O.I., Tavera, C.A., Tayeb, S., Ghouali, S., Abdulsahib, G.M., Richard-Nnabu, N.E. and Louni, A., 2022. A classification algorithm-based hybrid diabetes prediction model. Frontiers in Public Health, 10, p.829519.

[19] Christopher, S.Z., Siswantining, T., Sarwinda, D. and Bustaman, A., 2019, October. Missing value analysis of numerical data using fractional hot deck imputation. In 2019 3rd International Conference on Informatics and

Computational Sciences (ICICoS) (pp. 1-6). IEEE.

[20] Usha, V. and Rajalakshmi, N.R., 2023, September. Insights into Diabetes Prediction: A Multi-Algorithm Machine Learning Analysis. In *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)* (pp. 1207-1212). IEEE.

[21] Beghriche, T., Djerioui, M., Brik, Y., Attallah, B. and Belhaouari, S.B., 2021. An efficient prediction system for diabetes disease based on deep neural network. *Complexity*, *2021*, pp.1-14.

[22] Azbeg, K., Boudhane, M., Ouchetto, O. and Jai Andaloussi, S., 2022. Diabetes emergency cases identification based on a statistical predictive model. *Journal of Big Data*, *9*(1), pp.1-25.

[23] Usha, V. and Rajalakshmi, N.R., 2023, September. Insights into Diabetes Prediction: A Multi-Algorithm Machine Learning Analysis. In *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)* (pp. 1207-1212). IEEE.

[24] Usha, V., Balaji, K.S., Singh, K.P.N., Prasad, A.G. and Reddy, D.V.D., 2022, November. Social Distance Alerting System Using Machine Learning. In *2022 1st International Conference on Computational Science and Technology (ICCST)* (pp. 441-446). IEEE.

[25] Pima dataset link: Pima Indians Diabetes Database

[26] Iraqi Diabetes dataset link: Diabetes Dataset - Mendeley Data

[27] Frankfurt data set link: diabetes