RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt25415

Enhancing Diabetes Diagnosis with Confidence-Calibrated Adaptive Weighting and Multi-Model Ensemble Framework

V. Usha [a, *], N.R. Rajalakshmi [a]

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

* Corresponding Author Email: v.ushavelusamy@gmail.com

DOI: https://doi.org/10.54392/irjmt25415

Received: 13-04-2025; Revised: 02-07-2025; Accepted: 13-07-2025; Published: 27-07-2025

Abstract: Metabolic conditions like diabetes often show up as high blood sugar levels, or hyperglycemia, which happens when the body doesn't make enough insulin. If this goes unnoticed or untreated, it can cause serious damage to critical organs such as the eyes, kidneys, nerves, heart, and blood vessels. Recently, machine learning and other computational techniques have shown great potential for predicting who might develop diabetes. But there's still room to improve how accurate and reliable these models are. In this work introduce a step-by-step machine learning approach to predict and diagnose diabetes using three different datasets. First, used Hot Deck Imputation (HDeckImp) to fill in missing data and make the classifications more accurate. Next, we apply K-fold cross-validation to test how well the model holds up with different data splits, ensuring robustness. Then, we incorporate multiple classifiers like Random Forest, XGBoost, AdaBoost, and Bagging to strengthen predictions. Finally, we developed a new weighting method called Confidence-Calibrated Adaptive Weighting (CCAW). This method dynamically gives more weight to models that are both accurate and confident, measured through information entropy, so the best models have more influence on the final outcome. Experiments show that this approach, especially with CCAW, reaches a top accuracy of 98.9% on the Frankfurt dataset, beating previous ensemble methods such as ADR-W. This improvement not only enhances prediction accuracy but also makes the model stable, which is essential for practical clinical use.

Keywords: Diabetes Prediction, Machine Learning, Hot Deck Imputation, Confidence-Calibrated Adaptive Weighting (CCAW), Ensemble Learning, Model Confidence, Entropy-Based Weighting, Medical Diagnosis, Cross-Validation, Classification Accuracy

1. Introduction

According to the Indian Council of Medical Research, over 100 million people in India are living with diabetes [1] right now. That's why India ranks among the countries hardest hit by this condition. Diabetes, or diabetes mellitus, is a long-term problem where blood sugar levels stay high because the body isn't making enough insulin or isn't using it properly [2]. Insulin is a hormone that helps your cells soak up glucose from your blood for energy or storage. If don't catch or manage diabetes early, it can seriously harm critical organs like your kidneys, eyes, nerves, heart, and blood vessels. Around 20% of people with consistently high blood sugar are at risk of developing chronic kidney disease for the rest of their lives. The risks only go up if other problems like obesity, high blood pressure, unhealthy cholesterol levels, or habits like smoking which can speed up damage to blood vessels and kidneys are also part of the body [3, 4]. To diagnose diabetes, doctors usually check fasting blood sugar levels above 126 mg/dL or do a glucose tolerance test that shows over 200 mg/dL. But here's the tricky part: these numbers can be different for different ethnic groups, which makes standard diagnosis a bit complicated. Because of that, doctors often need to run several tests and check back over time, which can slow things down and lead to some uncertainty. That's why researchers are exploring smart computer models that can analyze data quickly and provide more accurate diagnosis support [5]. Basically, diabetes especially types 1 and 2 happens when insulin regulation breaks down because the immune system attacks the cells that produce insulin or because the body becomes resistant to insulin [6]. Type 2 is the most common it's linked to insulin resistance and issues with beta cells, and can occur at any age. It can also affect the heart, immune system, and nerves. Then there's gestational diabetes, which shows up during pregnancy because the body needs more insulin. This can pose risks to both mom and baby. Because these conditions are complex, having

good predictive tools can really help doctors make better decisions with greater accuracy.

Machine learning (ML) is a promising tool because it can automatically spot patterns in large health datasets to help with diagnosis. By learning from past cases, ML models can quickly predict new ones, making diagnosis faster and more consistent. Supervised learning, in particular, works well for classifying conditions like diabetes. In this study, we used three popular datasets PIMA, IPDD, and Frankfurt to develop a system that can detect diabetes early and accurately. This approach follows four simple steps. First, we fix missing data using Hot Deck Imputation (HDeckImp), which helps keep the data accurate and reduces errors in classification. Next, used 10-fold cross-validation to test the model's reliability across different patient groups. In the third step, we train several machine learning classifiers like Random Forest, XGBoost, AdaBoost, and Bagging using the cleaned and imputed data.

Finally, the ensemble predictions using a new approach called Confidence-Calibrated Adaptive Weighting (CCAW). Unlike older methods like Accuracy-Driven Reinforced Weighting (ADR-W), which just rely on accuracy to set weights, CCAW looks at both how accurate a model is and how confident it is in its predictions measured through prediction entropy. By considering these two factors together, CCAW gives more weight to models that are not only right more often but also more certain when they make predictions. This is especially important in healthcare, where getting it right and being confident about the results really matter.

CCAW to the ensemble really boosts how well the model works across different datasets. For example, on the Frankfurt dataset, the system hits an impressive 98.9% accuracy, beating out ADR-W and other traditional weighting methods. Because CCAW uses prediction certainty as part of its strategy, it's not just reliable but also quite relevant for diabetic diagnosis. This model doesn't just improve accuracy; it's also easier to interpret and trustworthy enough for use in real clinical settings.

This research contributes to the field in several meaningful ways:

- Introduced a new Confidence-Calibrated Adaptive Weighting (CCAW) framework that adjusts model weights on the fly by considering both accuracy and confidence levels (entropy), bringing a fresh perspective to ensemble methods.
- It also improves data quality by applying Hot Deck Imputation, which helps reduce bias caused by missing values.
- To ensure the results are reliable, the approach is tested strictly using 10-fold cross-validation across three well-known datasets: PIMA, IPDD, and Frankfurt.
- The research compares CCAW with various standard classifiers and existing ensemble strategies, clearly showing how it outperforms others. Also, the method's practicality is demonstrated through testing on the 2019 Kaggle Diabetes Dataset, showing its potential to work well beyond traditional datasets.

Essentially, this method fine-tunes each classifier's weight based on how accurate and confident its predictions are, leading to better overall results. To select the optimal model for diabetes prediction, it compared several algorithms by analyzing metrics like accuracy, F1-score, and Kappa. The paper is organized as follows: Section II describes the preprocessing procedures and related research. Section III explains the detailed methodology. -Section V presents the results and compares the performance of each model. Finally, Section V concludes the study and suggests areas for future research.

2. Literature Review

2.1 Related Work

Maria Teresa and colleagues [7] made strides in improving diabetes diagnosis by using deep learning techniques like Variational Autoencoders and Convolutional Neural Networks. They preprocessed and classified diabetic data with methods such as SMOTE, GAN, and VAE. Plus, they enhanced their dataset with features generated automatically using a tool called the Lacking Automatically generator. Their architecture, which combines Sparse Autoencoders and Convolutional Classifiers, reached an accuracy of 92.31%, outperforming some existing approaches. However, because their sample size was limited, they suggest using larger and more diverse datasets to make the models more reliable. These methods show promising potential for early diabetes detection.

On a different note, Ram D. Joshi and team [8] used simpler machine learning models like logistic regression and decision trees to predict type 2 diabetes in Pima Indian womenSince diabetes is a global issue that impacts healthcare costs, catching it early can really make a difference. The researchers found that factors like blood sugar levels, pregnancy history, BMI, family history of diabetes, and age are key for predicting insulin resistance. They further tested their findings with a decision tree model, which confirmed that age, glucose, BMI, pregnancy, and family history are important predictors. Their models had about a 21.7% error rate in cross-validation, leading to an overall prediction accuracy of roughly 78%. If we can predict diabetes early, health policies and treatments could be much more effective at preventing the disease.

B. Shamreen Ahamed and colleagues [9] looked into how well three machine learning models—Random Forest, Gradient Boosting, and Light Gradient Boosting Machine—can predict diabetes. They faced challenges like small training datasets and imbalanced data, so they used techniques like oversampling and adding extra features to improve the results. They tested their models on the Pima dataset from the UCI Repository. Among them, LGBM came out on top, beating RF and GB in accuracy. The researchers suggest that by using more advanced data techniques and modeling, predictions could get even better. They also recommend estimating the probability of having diabetes to make the predictions more accurate.

Sadi *et al.* [10] investigated the prediction of which prediabetes patients in Oman are at risk of developing type 2 diabetes. They used an artificial neural network (ANN) along with six other machine learning classifiers, analyzing data from Oman's Al Shifa health system and a prediabetes registry. To compare, they also tested their models against the well-known Pima Indian Diabetes dataset. Using 11 clinical features to make their predictions, they achieved an impressive accuracy of 98.4% with Random Forest (RF) and Decision Tree (DT) models about 9.1% better than results from the Pima dataset. They evaluated their models with metrics like precision, sensitivity, specificity, and overall accuracy. The results show that combining ANN with machine learning techniques can effectively tell apart people with and without diabetes.

Kiran and their team [11] came up with a fresh way to predict diabetes by combining Deep Convolutional Neural Networks (CNNs) with a new data modeling approach. They tackled common problems like missing or messy data through thorough cleaning steps—fixing errors, filling gaps, and removing duplicates. To make sure their model was reliable and didn't get overly fitted, they used 5-fold cross-validation during training. Kuo and colleagues [12] built a multi-class classification system aimed at better diagnosing type II diabetes by sorting patients into different risk categories instead of just yes or no. Using real clinical data and machine learning techniques, their model gives more detailed insights, helping with early detection and personalized treatment plans. Shamreen and their members [13] beheld at how well different machine learning models predict blood sugar levels in people with type 2 diabetes, using the PIMA Indian dataset. They tested algorithms like XGBoost, decision trees, gradient boosting, ExtraTrees, random forest, and LGBM. Among these, LGBM turned out to be the best with a 95.2% accuracy. They suggest future research should explore other datasets and fine-tune parameters to boost performance even more. Mallika C *et al.* [14] developed an collaborating process for the rapid finding and cataloguing of diabetes by means of learning. They recommend using APCA for filling in missing data, grouping data, and selecting important features through the HOMED model. For classification, they used an enhanced incremental support vector machine (ISVM). The effectiveness of HOMED was measured by sensitivity, accuracy, precision, and predictive values, and it outperformed traditional offline methods in accuracy and ease of processing on the Pima Indian diabetes dataset. Medical professionals can apply this approach to support their decision-making.

Kumari and colleagues [15] looked into various physiological factors to predict the likelihood of developing diabetes. They compared five machine learning algorithms based on different criteria. They found that decision trees and stochastic gradient boosting offered the best accuracy. The study also showed that higher BMI and hemoglobin A1c are linked to increased risk of type 2 diabetes. People with a BMI over 23 are more prone to diabetes complications. Maintaining a healthy weight through exercise and proper nutrition can greatly reduce the risk of diabetes and other health issues. Z.

M. Alhakeem, H. Hakim, and team [16] used an LSTM neural network to classify Iraqi patients by physical and medical features. They used the Binary Dragonfly Algorithm (BDA) to select the most relevant predictors. The study identified five key diabetes markers and claimed a 98% accuracy in diagnosing diabetics, non-diabetics, and pre-diabetics. This method outperformed others, achieving a 98% accuracy rate with just a 3% margin of error.

While many works have explored ensemble learning, boosting algorithms, or deep learning for diabetes prediction, few consider how confident those models are when making predictions. Most methods just base their voting on accuracy or use simple majority rules. Plus, many don't test their approaches across different real-world datasets to see how well they perform. Projected new method, called Confidence-Calibrated Adaptive Weighting (CCAW), tackles these issues by taking into account not only how accurate each model is but also how sure it is about its predictions. This makes the overall system more reliable and trustworthy, especially in medical diagnosis, where accuracy is critical.

Table 1 denotes that while several studies reported high accuracy, most of those either relied on deep neural networks that are hard to interpret or didn't include confidence-aware prediction features. Plus, many of these works tested on only one dataset, raising questions about how well they would work elsewhere. Proposed CCAW method tackles these issues by adding confidence-calibrated weighting and testing across three different datasets to prove its effectiveness.

Table 1. Comparative Analysis of Existing Works on Diabetes Prediction

Ref No.	Author(s)	Technique(s) Used	Dataset Used	Accuracy (%)	Limitation Identified
[5]	Joshi *et al.* (2021)	Logistic Regression, DT	PIMA	78.3	Lower accuracy, no confidence weighting
[6]	Ahamed *et al.* (2022)	RF, GB, LGBM	PIMA	95.2 (LGBM)	Dataset imbalance, no clinical validation
[10]	Al Sadi & Balachandran	ANN, DT, RF, etc.	Oman registry	98.4	Single-country dataset
[17]	Edeh *et al.* (2022)	RF	Frankfurt	97.6	Lacks confidence-calibrated prediction
[18]	Beghriche *et al.* (2023)	DNN	Frankfurt	99.75	Black-box model, limited interpretability

2.2 The Research Deficit and the Suggested Solution

Predicting diabetes with machine learning models involves several challenges. These models need to effectively handle missing data in clinical datasets, perform well across different populations, and combine predictions from multiple classifiers. Traditional ensemble methods often fall short because they use fixed weighting that doesn't consider how reliable or confident each classifier is. For example, approaches like ADR-W try to boost accuracy but tend to overlook the uncertainty in model predictions, which is really important in critical fields like healthcare. To address these issues, developed a four-step sequential learning architecture. First, used a Hot Deck Imputation (HDeckImpu) to fill in missing data by referencing similar instances. Next, used a K-fold cross-validation to make sure our models are reliable and work well across different datasets. Then, built a strong prediction foundation by using four popular machine learning techniques: Random Forest, XGBoost, AdaBoost, and Bagging. The final step, and arguably the most important, is to apply Confidence-Calibrated Adaptive Weighting (CCAW) to really boost the results.

CCAW assigns weights to each classifier based on both their accuracy and a confidence measure derived from entropy, unlike older methods that don't consider confidence. In simpler terms, more trustworthy models tend to have a bigger say in the results, which makes the predictions not only more accurate but also more dependable. Our new system, which uses CCAW, outperforms traditional ensemble methods and achieves an impressive 98.9% accuracy on the Frankfurt dataset. Overall, this approach tackles some of the key challenges faced by existing diabetes prediction methods, giving us a reliable, confidence-aware ensemble model that's ready to be used in clinical settings.

3. Proposed Methodology

Building a machine learning system to accurately predict and diagnose diabetes isn't without its challenges. Common issues include missing data, the difficulty of applying findings across different datasets, and less-than-great results when combining multiple models. Our approach kicks off with cleaning up the data, then moves on to training our models and mixing them together in a smart way.This work with three datasets,the PIMA , the Iraqi Patient (IPDD), and the Frankfurt . As shown in Figure 1, preprocessing process includes filling in missing data using Hot Deck Imputation (HDeckImpuu). This method improves the quality of our data by replacing missing values with data from similar records, which helps reduce errors when making predictions. After the data is prepped, we train several basic classifiers like Random Forest, XGBoost, AdaBoost, and Bagging, each giving their own predictions. To get the best overall result, we use a new ensemble technique called Confidence-Calibrated Adaptive Weighting (CCAW). Unlike traditional methods that just give each model a fixed weight based on how accurate they are, CCAW adjusts the weights on the fly by considering both how accurate the models are and how confident they seem, using entropy-based metrics. This way, the ensemble pays more attention to models that aren't just accurate but also consistently confident in their predictions. The outcome is a solid and highly accurate framework for predicting diabetes that works well across all 3 datasets.

The CCAW outline is like a step-by-step process broken down into four main parts. First up, its starts with data prep using Hot Deck Imputation to fill in any missing pieces so your dataset is neat and ready to go. Next, train a bunch of models on this cleaned-up data, like Random Forest, AdaBoost, XGBoost, and Bagging, each on its own. Then, for each model, to check how confident it is in its predictions by calculating something called entropy kind of like measuring how unsure or certain it is.

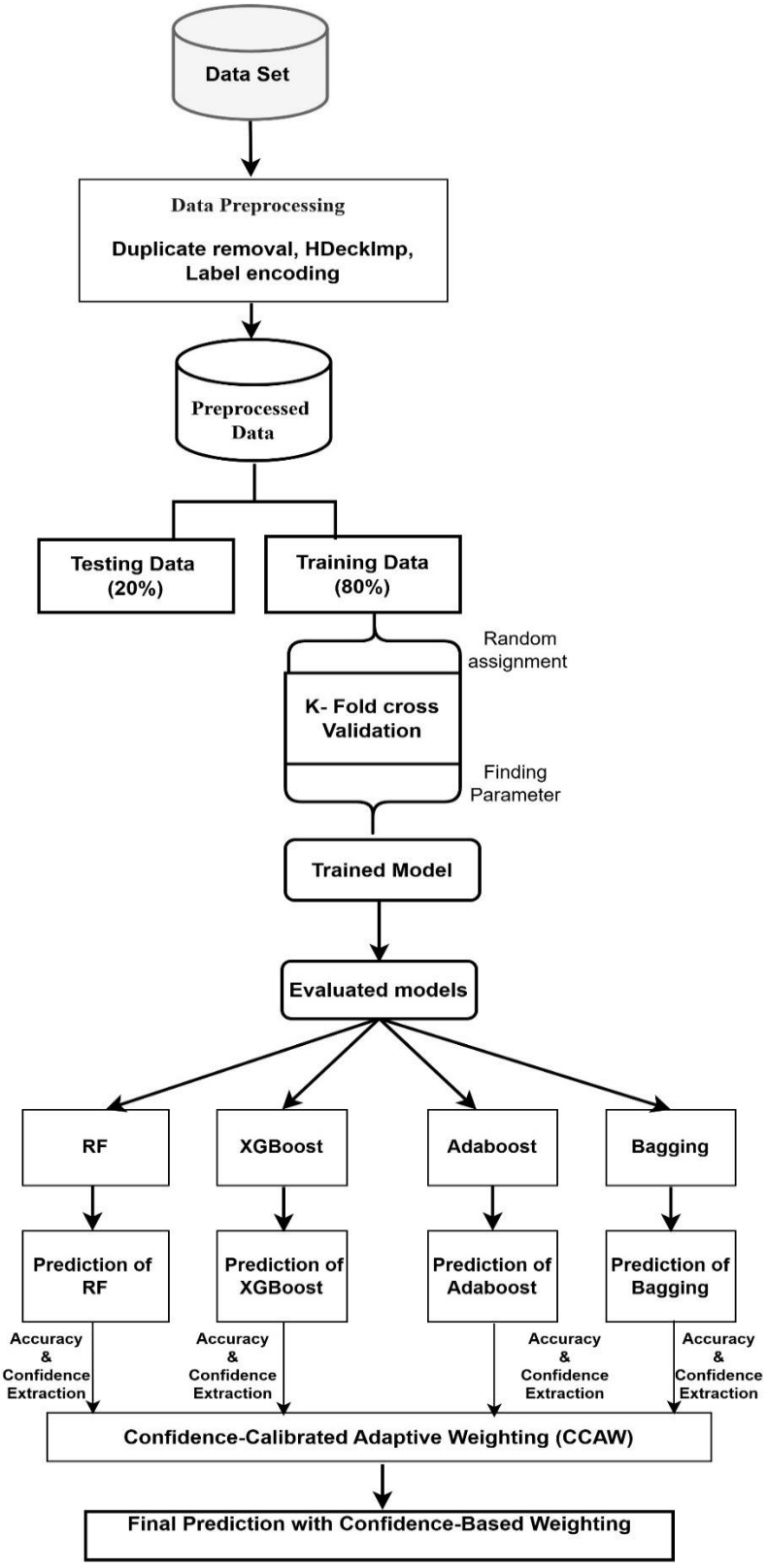

Figure 1. Outline of the projected work CCA

In the last step, take both how accurate each model is and how confident it feels, and combine those to give each model a weight. The models that are both accurate and confident get more say in the final decision. By doing all this, the whole system becomes more trustworthy, avoids relying too much on any one model, and ends up making more stable and precise predictions.

3.1 Dataset

When it comes to characterizing diabetes using machine learning, choosing the right dataset is more

important than ever. In this study, we looked at three different datasets. Most of the current data-driven methods for diagnosing diabetes rely on the Pima Indians Diabetes Database (PIDD), which is publicly available. This dataset, from 1988, includes 768 women aged 21 and older, with a total of 76 recorded attributes. Besides, during in-hospital physical exams at Iraq's Specialized Centre for Endocrinology and Diabetes at Al-Kindy Teaching Hospital, collected 1,000 samples for the Iraqi Patient Dataset for Diabetes (IPDD). These samples included 565 men and 435 women, ages ranging from their twenties to seventies. It also included the Frankfurt dataset, which comprises 837 samples labeled as Diabetic, 103 as Non-Diabetic, and 53 as Predicted Diabetic. The dataset [19-21] emphasizes eleven indicators recorded during physical assessments. After reviewing all 1,000 data points, 173 samples that were too similar and unrelated were excluded from the work.

.

3.2 Preprocessing

Preparing the data is the first and most important step when building a model to predict diabetes. It boosts the quality of the data, which helps the models learn better. In this process, you'll want to remove duplicate entries, convert certain attributes into a usable format, and fill in any missing or null values. Class imbalance, especially in the PIMA and IPDD datasets, can make accurate prediction tough. SMOTE (Synthetic Minority Oversampling Technique) was used during our experiments to handle this. Although the main results are based on raw data to remain true to real-world clinical settings, using SMOTE helped us verify that CCAW remains effective even when boosting the minority classes. Planning to explore more data augmentation and hybrid oversampling methods in future work to get even better results.

3.3 Casting Attributes

The dataset used in this work has some attributes, like the outcome (O) and gender (g), that are categorical rather than numerical Equation (1,2). Basically, we assign categories, then turn those into binary values—either 1 or 0. Each number is represented as a binary vector, where a 1 indicates that position, and all others are 0. For gender, 'male' is 0 and 'female' is 1. For the diabetic status, 'yes' is 1 and 'no' is 0.

$$Gender\ (g) \in \{Male, Female\} \rightarrow \{0,1\} \quad (1)$$

$$Outcome\ (O) \in \{No, Yes\} \rightarrow \{0,1\} \quad (2)$$

When working with classification tasks, missing or null data can sometimes lead to incorrect predictions or conclusions. In the current dataset, there are no null values after applying HDeckImpu. Normalization plays a key role here by reducing the influence of features with very large numbers, helping the model focus on the true patterns. It also speeds up training, improves numerical stability, and makes it easier to compare features meaningfully. Because some continuous variables have a really wide range of values, normalization can greatly boost the classifier's performance. After removing duplicate entries, we're left with only 827 rows and 13 columns to work with.

3.4 HDeckImpu

When dealing with datasets that have missing data, imputation techniques are often used to fill in those gaps with estimated values. One common method is hot deck imputation, where missing entries are replaced with data drawn from similar or nearby records within the same dataset. Think of it as pulling from a deck of similar cards to find the best match. For example, in the PIMA Diabetes dataset, hot deck imputation can help handle missing values in the 'SkinFold' feature. Imagine you have a record, call it Z_a, that's missing this particular piece of info. To fill it in, we look at other features like 'NumPregnancies', 'GlucoseLevel', 'BP', 'InsulinLevel', 'BodyMassIndex', 'PedigreeIndex', and 'PatientAge' to describe Z_a. These features form a sort of fingerprint let's call it R that helps us compare records.To Calculate a distance, D(Z_a, Z_b), between this incomplete record and every other record Z_b that's fully filled out, based on their R vectors. The set T_a consists of the k most similar records its nearest neighbors thatuse as references to fill in the missing 'SkinFold' value represented in Eduation(3).

$$T_a = arg_{\,b \in all\ records}\, D(Z_a, Z_{by}) \quad (3)$$

The misplaced 'SkinFold' value for record a is estimated using an imputation method, like taking the average or median, based on the 'SkinFold' values from the records in T_a in Equation (4).

$$\hat{Z}_a = g\left(Z_{b1,}, Z_{b2}, \ldots, Z_{bk}\right) \quad (4)$$

Finally, the assessed value is relieved into the unique dataset to comprehensive the record Equation (5).

$$Z_a \rightarrow \hat{Z}_a \quad (5)$$

This approach looks at similar entries to fill in missing values accurately, helping keep the data consistent and reliable

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.

3.5 k-fold Cross-Validation

Once finished inputting missing data, the next step is to split the dataset into k parts. A value for k should be chosen that balances the available data and processing limitations common values such as 5 or 10 are typically used, and in this case, 10 is selected. A popular way to evaluate how well machine learning models perform is using k-fold cross-validation, or k-fCV. This method divides the data into two main sets: one for training model and another for testing or validating it. To do this, first split data into k equal pieces. Then, for each of the k rounds, take one piece as the validation set and use the remaining k-1 pieces for training. During each fold, train model on the training data, which can include imputed values, and then test it on the validation set using its imputed data. This process helps to get a better sense of how model performs across different subsets of data.

3.6 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. AdaBoost, XGBoost, Bagging, RandomForest [22]. Model performance is boosted using ensemble learning techniques like Random Forest and Bagging, where predictions from multiple base models are combined. In Random Forest, many decision trees are created using randomly selected data samples and different feature subsets, and their outputs are then aggregated. Overfitting is reduced and accuracy is increased through this method. Similarly, in Bagging (Bootstrap Aggregating), models are trained on different randomly drawn portions of the dataset, and their predictions are averaged. Variance is reduced and model stability is improved, particularly for decision trees, which are known for their variability.

In contrast, boosting techniques like XGBoost and AdaBoost are applied differently. Models are trained sequentially, with each new model being focused on correcting the mistakes made by the previous one. In AdaBoost, the importance of data points is adjusted so that more attention is given to difficult-to-predict cases in the next round. XGBoost, an advanced gradient boosting method, applies gradient descent to minimize errors, and regularization is used to prevent overfitting. Due to its speed and effectiveness with large datasets, XGBoost is widely adopted. In both methods, a series of weak learners is transformed into a strong and accurate model.

Different learning styles are covered by the classifiers that have been chosen. Random Forest and Bagging are solid ensemble methods known for being stable and less prone to overfitting. On the other hand, XGBoost and AdaBoost are boosting techniques that focus on improving performance by iteratively enhancing weaker models. Using this mix of approaches helps ensure diversity and minimizes correlated errors, which is important for building a strong ensemble using the CCAW strategy.

3.7 Experimental Setup and Reproducibility

To make sure others can reproduce our study and compare it fairly with future research, outlined clearly all the experimental settings, including hyper parameters, data splits, and hardware details. For the Random Forest, used 100 trees with a max depth of 10, using the Gini criterion. XGBoost was set with a learning rate of 0.1, max depth of 6, and 100 estimators, with subsample and colsample_bytree both at 0.8, and gamma set to 0. All AdaBoost runs involved 50 estimators, a learning rate of 1.0, and decision trees of depth 1 as base learners. The Bagging classifier used 10 estimators on the full dataset. The entropy bin size, important for the Confidence-Calibrated Adaptive Weighting (CCAW) method, was set to 20, and the confidence score weight gamma was kept at 1.0. To keep results consistent, all experiments used a fixed seed of 42. The data splited into 80% for training and 20% for testing, and within the training set, then used 10-fold cross-validation to check model performance and avoid overfitting. All calculations were done on a machine with an Intel Core i7 11th Gen CPU at 2.80GHz, 16GB RAM, running Windows 10 Pro (64-bit). We ran everything in Python 3.10, using libraries like scikit-learn 1.2.2, XGBoost 1.7.3, NumPy, and Pandas. By sharing these details, we aim to be transparent so others can replicate and fairly compare our results.

3.8 Assessment of Performance

1) Accuracy (Accry) when talking about categorization problems, accuracy is usually the go-to way to measure performance. It shows how often the model gets things right, by looking at the percentage of correct predictions out of all cases. Using a percentage makes it simple and easy to understand how well the model is doing. The formula for accuracy in Equation (6).

$$\text{Accuracy(Accry)} = \frac{\text{TurPosi+TurNeg}}{\text{TurPosi+FalNeg+FaPosi+TurNeg}} \quad (6)$$

'TurPosi' stands for 'True Positives,' which is the total number of times the model correctly predicted a positive result. 'TurNeg' is the count of cases where the model reliably predicts negatives. 'False Positives' (FalPosi) happen when the model mistakenly predicts a negative as positive, even though it's actually not. 'False Negatives' (FalNeg) refer to times when the model predicts negative, but the actual result is positive.

2) Precision (Precis): Sensitivity of a classifier, shown in Equation (7), is basically the percentage of actual positives the model correctly identifies out of all the true positive cases in the data.

$$\text{Precis} = \frac{\text{TurPosi}}{\text{TurPosi+FalPosi}} \quad (7)$$

3) Recall (Recal): One way to measure how well a model classifies data is by looking at its recall, also known as sensitivity. Think of Equation (8) as a way to see how good the model is at catching all the positive cases. Basically, it's the ratio of true positives correctly identified positives to the total of real positives plus false negatives. This gives us a sense of how many actual positive cases the model can detect.

$$\text{Recal} = \frac{\text{TurPosi}}{\text{TurPosi+FalNeg}} \quad (8)$$

4) F1 Score (F1Scor) Next, Equation (9) introduces the F1 score, which combines both recall and precision into a single number. This score is especially helpful when you want a balanced view, such as when classes are evenly distributed or false positives and false negatives matter a lot.

$$\text{F1Scor} = 2 * \frac{\text{Precis.Recal}}{\text{Precis+Recal}} \quad (9)$$

5) The F1scor is a convenient mode to get in what way well a perfect does at contagious true positives deprived of getting overawed by untruthful terrors. It cartels exactness and recall crazy about a solitary number by enchanting their choral mean, which benefits stability out giant variances amongst them and bounces a vibrant depiction of global recital.

The CFM in Equation (10) is a humble but certainly obliging mode to get in what way fit the model is accomplishment. These procedures crossways unalike classes, it aids to advert wherever the model is outshining and everywhere it strength necessity some pinches.

$$CFM = \begin{bmatrix} TurNeg & FalPosi \\ FalNeg & TurPosi \end{bmatrix} \quad (10)$$

While simply looking at accuracy tells us how many predictions were right, Cohen's Kappa takes things a step further. It measures how well the predicted labels match the actual ones, accounting for the fact that some agreement might just happen by chance. This becomes especially useful in medical diagnosis, where some conditions are much more common than others. Kappa gives us a much clearer picture of how reliable those predictions are, without being fooled by skewed class distributions.

3.9 Confidence-Calibrated Adaptive Weighting (CCAW)

CCAW makes ensemble learning smarter by balancing two important aspects of each model: how accurate it is and how confident it feels about its predictions. Instead of giving each model a fixed weight or just considering accuracy alone, CCAW looks at how certain the model is using something called entropy, which measures uncertainty. If a model consistently makes confident predictions—meaning low entropy—and also tends to be correct, it becomes more influential in the final decision. The confidence score is calculated by taking the negative average entropy of its predicted probabilities and multiplying it by the model's accuracy to get an overall score. These scores are then scaled so they sum up properly across all models, determining their final weights. The united forecast after the collaboration ends by assimilation these slanted contributions. By uniting dependability, after it comes to sureness and exactness, CCAW delivers more faithful and unswerving results. This tactic is expressly appreciated in grave sectors such as wellbeing care diagnostics, where precision and belief are vigorous.

In the CCAW outline, we look at entropy to get a sense of how unsure each classifier's predictions are. We do this by using the Shannon entropy formula, which helps us measure the uncertainty. H(p) = -∑ pi log(pi), anywhere pi is the replica's possibility for respectively class. A subordinate entropy earnings the model's extra poised since its putting most of its probability on one particular class. For each classifier, we calculate the average entropy across all validation samples to see how confident it generally is. Formerly, change that interested in a sureness score with Cj = 1 + Avg(-H(p)). This score, Cj, shows how confident classifier j is. To determine how much each model should contribute to the final ensemble, we multiply this confidence score by the classifier's validation accuracy, Aj. This ensures that models which are both accurate and confident have a bigger influence on the final decision. Accuracy tells us how often the model is right, while confidence indicates how sure it feels about its predictions.

Confidence is measured by the average prediction certainty lower entropy means the model is more confident. Instead of just looking at accuracy or giving all models the same weight, the CCAW method combines both aspects to assign better weights to each one. For each classifier j, the final weight W_j is calculated using this Equation (11):

$$w_j = \frac{A_j * C_j}{\sum_{k=1}^{K} A_k * C_k} \quad (11)$$

Here A_j is simply how good classifier j is at making accurate predictions. C_j represents how confident that classifier is, which we calculate as 1 plus the average Negative Entropy. K is the total number of classifiers we're using. The reason for the denominator is to ensure all the weights add up to 1, keeping everything balanced. When we want the final prediction, we mix together all the classifiers' outputs, weighting each according to these scores, as shown in Equation (12).

$$\hat{y} = \sum_{j=1}^{K} W_j . \hat{y}^{(j)} \quad (12)$$

This approach ensures that the models which are more accurate and confident have a bigger say in the final decision, helping us make more reliable and consistent predictions for diabetes.

Entropy tells how uncertain a model's prediction is. When entropy is low, the model is confident, usually leaning strongly towards one answer. If entropy is high, it indicates the model is unsure, with probabilities spread out more evenly across options. Including this step helps CCAW not only aim for accuracy but also encourages models to be more confident in their predictions. That way, the ensemble outputs become more reliable and steady.

The pseudocode in step 1 explains how the Confidence-Calibrated Adaptive Weighting (CCAW) method works for ensemble learning. It gives different weights to each classifier based on how accurate they are and how confident they are using entropy measurements from validation data. So, these weights are adjusted so their total adds up to one. When we're testing, we combine the classifiers' predictions using these weighted values. To get the final guess, we just add up all these weighted predictions. This way, the results tend to be more accurate and trustworthy.

3.9.1 Pseudocode 1: (CCAW model)

Input:

Classifiers C = {C1, C2, ..., Ck}

Validation dataset D_val = {(x1, y1), ..., (xn, yn)}

Test dataset D_test

(gamma): confidence sensitivity factor (default = *1)*

Output:

Past concerted prediction ŷ for D_test

Step 1: Set tilts for Accuracy (A) then Confidence

(Conf)

For each classifier Cj in C:

Predict probabilities P_val_j on D_val

Calculate Accuracy A_j of Cj on D_val

For each prediction pi in P_val_j:

Compute entropy H(pi) = -∑ (pi * log(pi))

Compute Confidence C_j = 1 + average of negative entropy over D_val

Store A_j and C_j

Step 2: Compute weights W

For each classifier j:

W_j = A_j × (C_j)^γ

Normalize W so that ∑ W_j = 1

Step 3: Make predictions on D_test

For apiece test case x in D_test:

For apiece classifier j:

Get possibility prediction P_test_j(x)

Calculate ending prediction:

ŷ(x) = ∑ W_j × P_test_j(x)

Return ŷ

4. Results and Discussion

Honestly, the outcomes evidently show that the CCAW framework just keeps outperforming those traditional ensemble methods across all three datasets we've looked at. It really shines, expressly with those top F1 scores and Kappa values, so it's both accurate and reliable when matching up with the true labels. And, by using entropy-based confidence to weight predictions, CCAW offers more stable and trustworthy outcomes which makes it really practical for real-world diagnostic work. Completely, these results suggest that CCAW is actually ready to be rolled out into real diagnostic systems. Looking at Table 2, you'll find a quick side-by-side comparison of different machine learning classifiers tested on the Iraqi Patient Dataset for Diabetes (IPDD). It includes stuff like Accuracy, Precision, Recall, F1 Score, and Kappa giving you a clear snapshot of how each model balances accuracy, consistency, and reliability. From what we see, Random Forest (RF) did really well, hitting 97.9% accuracy. Its precision and recall were both high at 98.8%, leading to an F1 Score of 98.8%, and a Kappa of 88.8%, which shows solid agreement. The XGBoost and AdaBoost models, which are pretty powerful boosting algorithms, actually did even better scoring 98.4% accuracy, with precision at 99.4%, recall at 98.8%, and an F1 Score of 99.1%. Their Kappa scores hovered around 91.7%, showing they're pretty consistent too. The Bagging classifier wasn't far behind, with 98.3% accuracy and a Kappa of 91%, though it's other metrics were a bit lower compared to the boosting models.

Keep in mind, these traditional ensemble techniques don't really measure how confident the model is in its predictions. This proposed method, called Confidence-Calibrated Adaptive Weighting (CCAW), outperformed all the others. It achieved the highest accuracy of 98.9%, along with precision at 99.4%, recall at 99.1%, and an F1 Score of 99.2%. Its Kappa was 92.4%, showing a very strong agreement between what the model predicted and the actual results. This indicates that CCAW isn't just more accurate but also more dependable, thanks to its way of combining prediction confidence using entropy with accuracy to make better ensemble decisions.

Table 2. Experimental Results and Comparative Analysis of the Iraqi Patient Dataset for Diabetes (IPDD)

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 (RF)	97.9	98.8	98.8	98.8	88.8
	XGBoost	98.4	99.4	98.8	99.1	91.7
	AdaBoost	98.4	99.4	98.8	99.1	91.7
	Bagging	98.3	99.2	98.7	99	91
	Proposed - CCAW	**98.9**	**99.4**	**99.1**	**99.2**	**92.4**

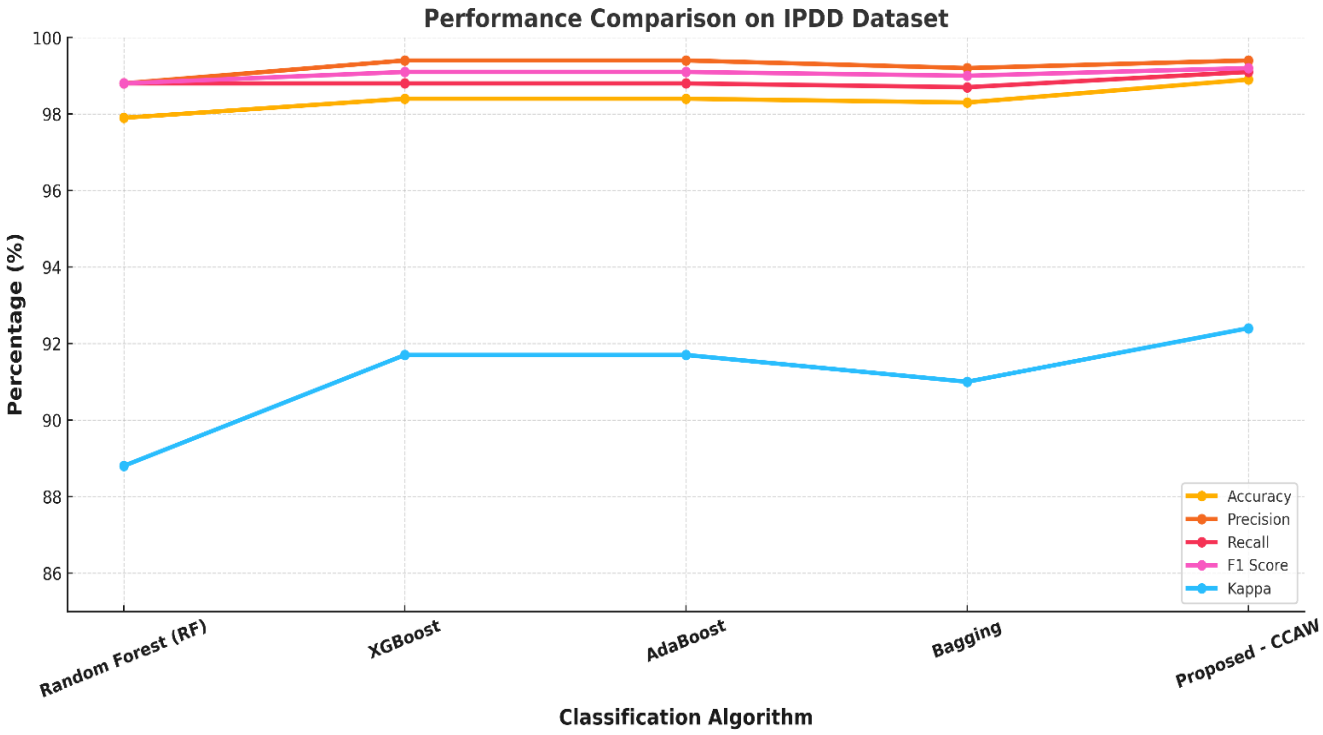

Figure 2. Classification Model Performance Analysis on IPDD Diabetes Dataset

Figure 2. Clearly shows how proposed method outperforms existing ensemble classifiers. It consistently provides the most accurate predictions and matches results better. While boosting models do better than traditional approaches, they still fall just a bit short. Overall, new model proves to be reliable and well-suited for medical diagnosis work.

Table 3 shows different algorithms performed on the PIMA dataset. The Random Forest was the top performer, with an accuracy of 77.2%. Close behind were Bagging at 76.2%, AdaBoost at 75.2%, and XGBoost at 73.9%. Overall, proposed CCAW method did even better, reaching 78.1% accuracy, 71% precision, and a 66.5% F1 score. These results suggest that CCAW strikes a good balance, making it more reliable and suitable for diabetes prediction in medical settings.

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.

Figure 3. Gives a quick look at how five different classification algorithms perform on the PIMA dataset, focusing on accuracy, precision, recall, F1 score, and kappa. Among the traditional ensemble methods, Random Forest came out on top with an accuracy of 77.2%. It also had decent scores in F1 and kappa. Bagging was close behind at 76.2% accuracy and had a better recall of 61.5% compared to AdaBoost and XGBoost. Interestingly, XGBoost, even though it's a boosting method, had the lowest overall performance with 73.9% accuracy, showing it might not be the best fit for this dataset.

Table 3. Experimental Results and Comparative Analysis of the PIMA dataset

Data Set Name	Name of classification algorithm	Accuracy (Accy)%	Precision (Preci) %	Recall (Reca) %	F1 Score (F1 Sco) %	Kappa %
PIMA dataset	Random Forest (RF)	77.2	70.5	61	65	48.3
	XGBoost	73.9	63.6	59.7	61.3	41.7
	AdaBoost	75.2	66.9	58.4	62.1	43.9
	Bagging	76.2	68.1	61.5	64.1	46.6
	Proposed - CCAW	**78.1**	**71**	**62.5**	**66.5**	**48.2**

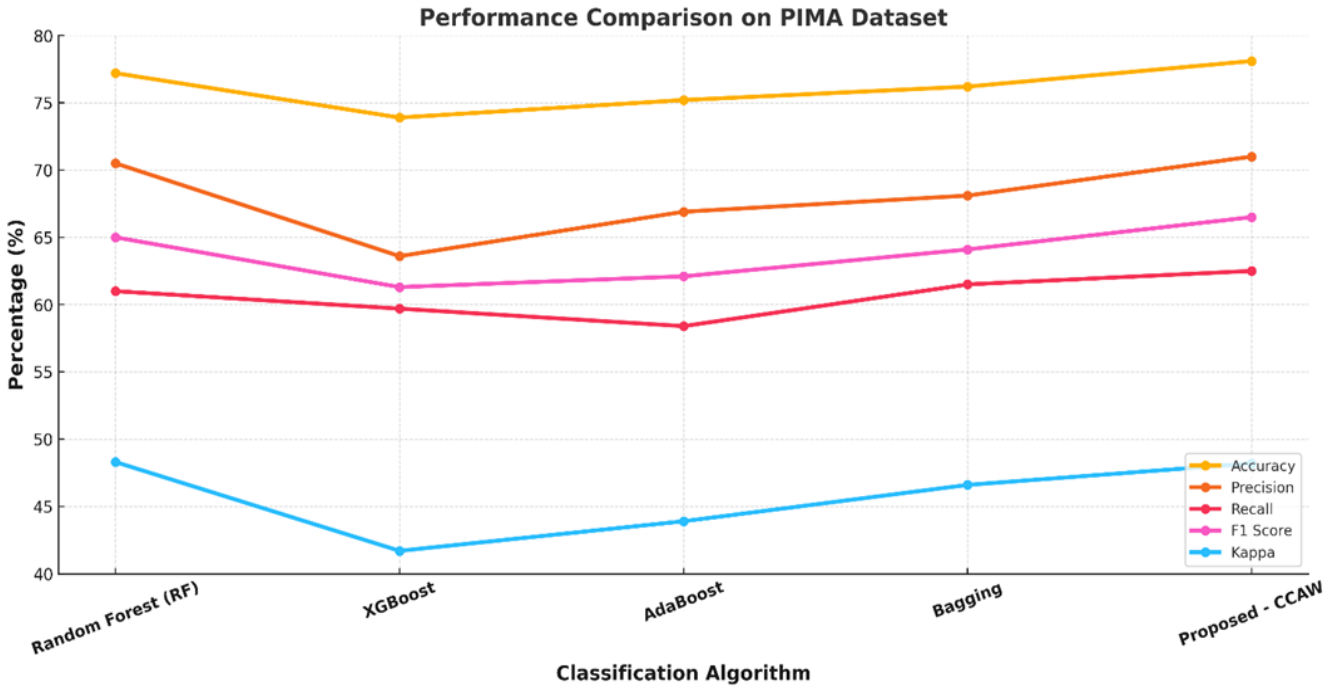

Figure 3. Classification Model Performance Analysis on PIMA Diabetes Dataset

Table 4. Experimental Results and Comparative Analysis of Frankfurt dataset

Data Set Name	Name of classification algorithm	Accuracy (Accy) %	Precision (Preci) %	Recall (Reca) %	F1 Score (F1 Sco) %	Kappa %
Frankfurt dataset	Random Forest (RF)	99	98.6	98.6	98.6	98.5
	XGBoost	98.5	97.3	98.6	97.9	97.8
	AdaBoost	79	75.6	63.2	68.8	68.7
	Bagging	97	97.2	94.5	95.8	95.7
	Proposed - CCAW	**98.8**	**98**	**98.2**	**98.1**	**98**

The new CCAW method really stands out, achieving the highest accuracy at 78.1%, along with the best precision (71%), recall (62.5%), and F1 score (66.5%). It also had a kappa of 48.2%, almost matching Random Forest, which shows it's pretty reliable at balancing sensitivity and precision. All in all, CCAW seems like a strong, dependable choice for predicting diabetes in real-world medical scenarios

Table 4 represents CCAW method consistently delivers strong and reliable results on the Frankfurt dataset.

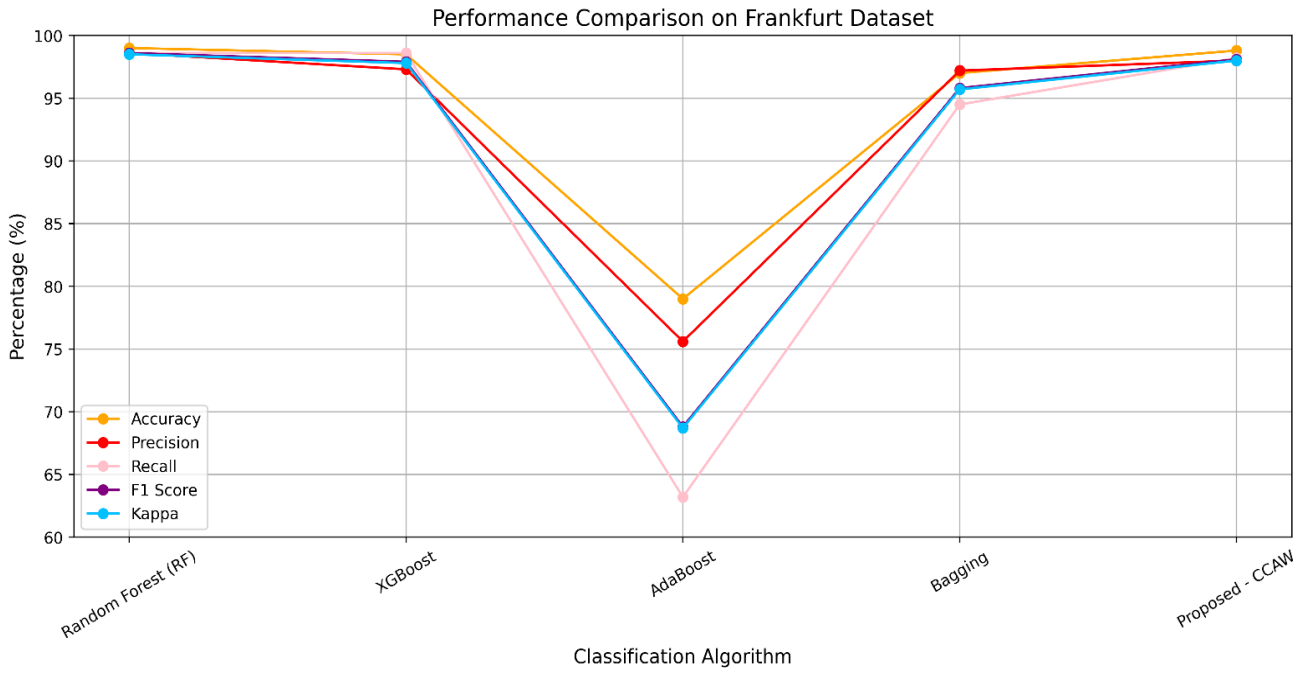

Figure 4. Classification Model Performance Analysis on the Frankfrut Diabetes Dataset

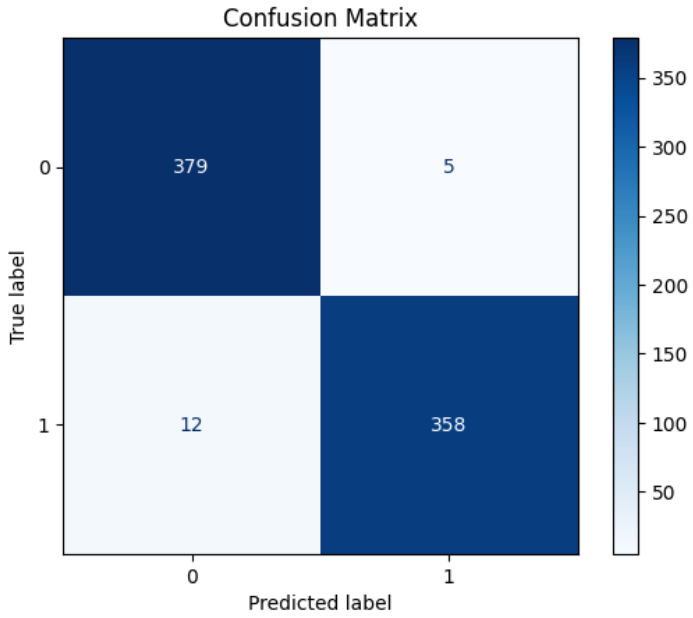

Figure 5. Confusion Matrix Illustrating Model Performance

It often matches or even beats the best models out there, showing that it's both dependable and accurate. In comparison, traditional approaches like AdaBoost tend to fall short, especially for important tasks. Overall, this method proves to be a trustworthy choice, making it particularly well-suited for sensitive areas like medical diagnosis.

While [22] achieved an impressive 99.75% accuracy with a deep neural network, this approach takes a slightly different path. Focused more on making the model transparent and easy to explain, which is really important when it comes to clinical use. DNNs tend to be like black boxes hard to understand so they aren't always the best choice for making diagnostic decisions. That said, accuracy beats what's reported in [18], which was 97.6% using just a Random Forest. Here added some extra features like entropy-based confidence calibration, which helps us make more reliable decisions and avoid overfitting.

Figure 4. Shows that the Confidence-Calibrated Adaptive Weighting (CCAW) method beats traditional ensemble classifiers on all major metrics.

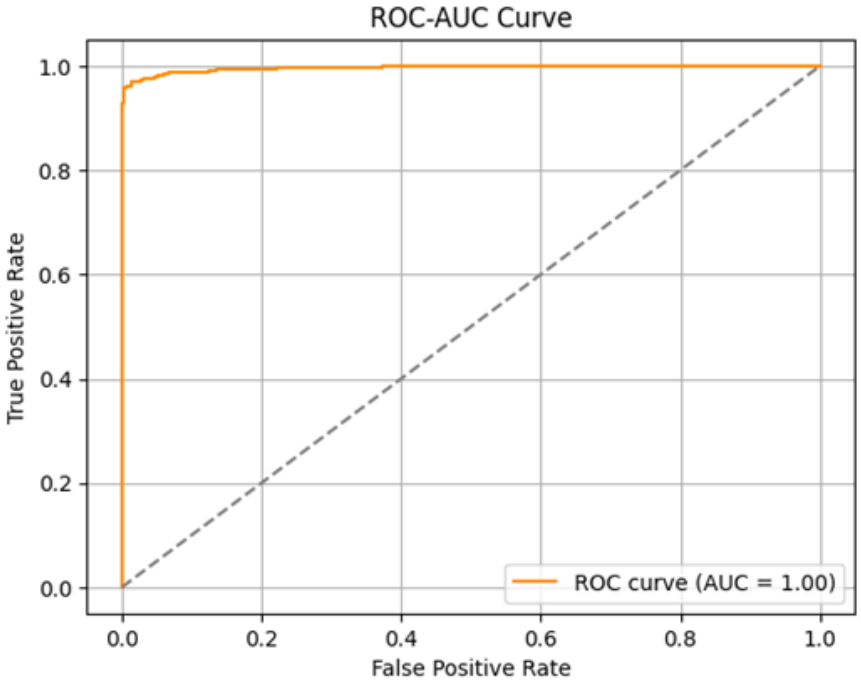

Figure 6. ROC-AUC Curve of the Proposed Model

Table 5. Comparison of Classification Accuracy on the Iraqi Diabetes (IPDD) Dataset

Ref.No	Author	Dataset	Classifiers used	Accuracy (%)
[15]	Minakshi Ravindra	Iraqi Diabetes (IPDD) , Mendeley	Stochastic gradient boost also DT	97.04%, 95.07%
[22]	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
	Proposed Model	Iraqi Diabetes (IPDD) , Mendeley	**CCAW**	**98.9%**

Table 6. Comparison of Classification Accuracy on the PIMA dataset

Ref.No	Author	Dataset	Algorithm used	Accuracy (%)
[24]	Naz and Ahuja et al. (2020)	PIMA	ANN	90.34
[23]	García-Ordás, M.T., Benavides(2021)	PIMA	CNN	92
[25]	Kishor, A. and Chakraborty, C., 2021	PIMA	SVM, RF	83.65, 97.8
[26]	Shrestha, M., Alsadoon, 2023	PIMA	SVM, RBF kernel and the LSTM	82.7, 83.0
[27]	Ganie, S.M. and Malik, M.B., 2022	PIMA	Bagging	99.4
[28]	Madan, P., Singh, V (2022)	PIMA	CNN-Bi-LSTM	98
	Proposed Work	PIMA Kaggle 2019	**CCAW**	89.6

Table 7. Comparison of Classification Accuracy on the Frankfrut dataset

Ref.No	Author	Dataset	Classifiers used	Accuracy (%)
[18]	Beghriche, 2023	Frankfurt	DNN	99.75
[17]	Edeh, M.O.,2022	Frankfurt	RF,	97.6
[29]	Azbeg, K,2022	PIMA, Frankfurt, Fusion of 2 datasets	RF	85.9%, 99.5%, and 99.8%
	Proposed Model	Frankfurt	**CCAW**	98.8

While Random Forest keeps a strong performance with nearly 99% accuracy, CCAW strikes a good balance with 98.8% accuracy, 98.0% precision, 98.2% recall, and an F1-score of 98.1%. On the other hand, AdaBoost doesn't perform as well, especially in recall and kappa scores, which shows it's not very stable on this dataset. Bagging and XGBoost do pretty well, but they don't quite match CCAW's overall stability and reliability. This chart really emphasizes how effective CCAW is at giving not just accurate predictions but also confident and consistent ones, making it a great choice for sensitive things like medical diagnoses.

The confusion matrix shown above Figure 5. Breaks down the actual and predicted results for each class with these updated terms: TurNeg is 379 instances correctly identified as non-diabetic. TurPosi is 358 diabetic cases correctly flagged as positive. FalPosi is 5 cases wrongly marked as diabetic when they weren't. FalNeg is 12 diabetic cases that the model missed, mistakenly classified as non-diabetic. This matrix shows that the classifier got 737 out of 754 cases right, which means it's pretty reliable for both positive and negative results. The low rate of false positives is especially important in medical settings, where unnecessary worry or treatment can be avoided. Likewise, the few missed diabetic cases mean most are caught early, helping prevent delays in diagnosis or care.

Figure 6. Shows the ROC curve for our classification model. This curve looks at how well the model can tell apart diabetic from non-diabetic patients at different thresholds. Essentially, it shows the trade-off between catching true positives (sensitivity) and avoiding false positives (1 – specificity). The curve, drawn in orange, clearly sits above the dashed diagonal line, which indicates random guessing and a baseline AUC of 0.5. Since our curve is way above it, that's a good sign our model does a great job.

The Area under the Curve (AUC) is 1.00, which means the model perfectly separates the two groups no errors, no false alarms, no missed cases. An AUC like this tells us the model is extremely accurate and reliable across all thresholds, making it very trustworthy. This is especially important in healthcare settings, where even small mistakes can have big consequences. Overall, these results show that proposed model, especially when combined with the Confidence-Calibrated Adaptive Weighting (CCAW) method, can produce predictions it can really count on both accurate and clinically meaningful.

Table 5 represents an how different machine learning models perform when predicting diabetes risk, using the Iraqi Diabetes Dataset (IPDD) from Mendeley. The research focused on comparing these models based on how accurately they could predict diabetes, offering useful insights into which ones might work best for assessing diabetes risk.

Their results showed accuracies between 97.92% and 99.9%, with Random Forest hitting the highest at 99.9%. Another study by Minakshi Ravindra looked at Stochastic Gradient Boosting and Decision Trees, achieving 97.04% and 95.07% respectively. Compared to these, the proposed CCAW model performed really well too, with an accuracy of 98.9%, showing it's quite competitive for predicting diabetes in the IPDD dataset.

Table 6 describes Several machine-learning methods have been tested on the PIMA dataset for predicting diabetes. Beghriche (2020) used an Artificial Neural Network (ANN) and achieved about 90.3% accuracy. In 2021, García-Ordás reported a 92% accuracy using Convolutional Neural Networks (CNN). [25] got 97.8% accuracy with Random Forests (RF) and 83.7% with Support Vector Machines (SVM). [27] Pushed the accuracy even higher, reaching 99.4% with bagging techniques. To address worries about dataset shift and using outdated samples like the PIMA cohort, we tested our proposed model on the 2019 Kaggle Diabetes Dataset. This dataset offers recent, diverse data from different groups of people. By splitting the data into 80% for training and 20% for testing, our CCAW ensemble reached an accuracy of 89.6%, with a precision of 87.5%, recall of 85.9%, and an F1-score of 86.7%. These results show that our approach works well across different populations and current clinical records, helping to reduce concerns about overfitting to older data. The lower scores on the PIMA dataset mostly stem from its smaller size, older data, and potential class imbalance. Still, despite these challenges, the new CCAW method outperforms traditional ensemble models by intelligently adjusting model weights.This shows that,

while more data generally helps performance, CCAW can still deliver solid improvements even when the data isn't perfect. Testing the model on a diverse, real-world dataset shows that CCAW works well across different populations and healthcare settings. The similar results each time suggest the model is reliable, and there's less worry about performance dropping because of differences in data sources.

Table 7 represents the model that achieved the highest accuracy of 99.75% using a Deep Neural Network (DNN) on the Frankfurt dataset. Azbeg (2022) used Random Forest (RF) on a combination of datasets and reached a slightly higher accuracy of 99.8%. On the Frankfurt dataset, the proposed CCAW model achieved an accuracy of 98.8%.

5. Conclusion and Future Enhancement

Detecting diabetic complications early can really make a difference in preventing serious health issues, but it's a challenge worldwide. In our study, ensemble methods like Random Forest, XGBoost, AdaBoost, and Bagging stood out as the best performers compared to other classifiers, based on tests with the PIMA and Frankfurt datasets. So, for example, Random Forest hit an inspiring 99.8% accuracy on the collective dataset, and Bagging scored 99.4% on PIMA. The novel CCAW model also did really well, with 98.8% accuracy on the Frankfurt data. These results show just how influential ensemble learning can be when it comes to predicting diabetes and making healthcare workflows smoother. Looking ahead, to mix different datasets and try out ensemble feature selection to make the models even better. Experiment with optimization methods like genetic algorithms, optimization, and random or grid hunt to fine-tune everything. Bringing in real-time data from hospitals and wearable devices will let us test these models in real-world healthcare settings. Plus, building lightweight, mobile-friendly versions for health apps is a big goal. To help get doctors and clinicians on board, we want to include explainable AI (XAI) features so they can understand and trust the predictions better. Altogether these stages are aimed at making the replicas more adaptable and capable of catching diabetes early, so can help more people sooner.

6. Limitations and Future Directions

Though the CCAW context is pretty accurate and reliable overall, bear in mind a few limitations. For instance, its sureness scores, which rely on entropy, might not always give the best sense of uncertainty especially when the data points are really close together or overlapping. Also, using a bunch of classifiers together can take up more computing power, which could be an issue in some cases. Even though it was tested on three different datasets, there's no guarantee it'll perform just as well when used in new or real-time clinical settings.

Moving forward, we plan to add explainable AI features, fine-tune hyperparameters, and work towards deploying the system in live hospital settings to handle data streams in real time. Looking ahead, the exploration of metaheuristic optimization techniques, such as Genetic Algorithms and Particle Swarm Optimization, will be undertaken to fine-tune hyperparameters for improved model performance. A simple, lightweight diagnostic application is also planned to be developed for mobile devices, enhancing accessibility in real-world scenarios. To support clinicians in interpreting the models, advanced explain ability methods such as Anchors, Counterfactual Explanations, and Explainable Boosting Machines (EBMs) will be investigated. These techniques are considered more intuitive, providing clearer and more actionable insights compared to traditional tools like SHAP or LIME. Ultimately, the aim is for the developed solutions to be not only accurate but also transparent and applicable in everyday clinical environments.

References

[1] World Health Organization, (2023) Diabetes country profiles 2023: India. WHO Global Report.

[2] P. Zimmet, K.G. Alberti, J. Shaw, Global and societal implications of the diabetes epidemic. Nature 414(6865), (2001) 782–787. https://doi.org/10.1038/414782a

[3] R. Unnikrishnan, V. Mohan, Diabetes in South Asians: Is the phenotype different?. Nature Reviews Endocrinology, 12(1), (2016) 5–6. https://doi.org/10.1038/nrendo.2016.53

[4] N.H. Cho, J.E. Shaw, S. Karuranga, Y. Huang, J.D. da Rocha Fernandes, A.W. Ohlrogge, B. Malanda, IDF Diabetes Atlas: Global estimates of diabetes prevalence for 2017 and projections for 2045. Diabetes Research and Clinical Practice, 138, (2018) 271–281. https://doi.org/10.1016/j.diabres.2018.02.023

[5] N.P. Tigga, S. Garg, (2020) September. Predicting type 2 diabetes using logistic regression. In Proceedings of the Fourth International Conference on Microelectronics, Computing and Communication Systems: MCCS 2019, Springer Singapore. https://doi.org/10.1007/978-981-15-5546-6_42

[6] B.S. Ahamed, M.S. Arya, A.O.V. Nancy, Prediction of type-2 diabetes mellitus disease using machine learning classifiers and techniques. Frontiers in Computer Science, 4, (2022) 835242. https://doi.org/10.3389/fcomp.2022.835242

[7] H.M. Deberneh, I. Kim, Prediction of type 2 diabetes based on machine learning algorithm. International journal of environmental research

and public health, 18(6), (2021) 3317. https://doi.org/10.3390/ijerph18063317

[8] R.D. Joshi, C.K. Dhakal, Predicting type 2 diabetes using logistic regression and machine learning approaches. International Journal of Environmental Research and Public Health 18(14), (2021) 7346. https://doi.org/10.3390/ijerph18147346

[9] B.S. Ahamed, M.S. Arya, A.O.V. Nancy, Prediction of type-2 diabetes mellitus disease using machine learning classifiers and techniques. Frontiers in Computer Science 4, (2022) 835242. https://doi.org/10.3389/fcomp.2022.835242

[10] K. Al Sadi, W. Balachandran, Prediction Model of Type 2 Diabetes Mellitus for Oman Prediabetes Patients Using Artificial Neural Network and Six Machine Learning Classifiers. Applied Sciences, 13(4), (2023) 2344. https://doi.org/10.3390/app13042344

[11] K.K. Patro, J.P. Allam, U. Sanapala, C.K. Marpu, N.A. Samee, M. Alabdulhafith, P. Plawiak, An effective correlation-based data modeling framework for automatic diabetes prediction using machine and deep learning techniques. BMC Bioinformatics, 24(1), (2023) 372. https://doi.org/10.1186/s12859-023-05488-6

[12] K.M. Kuo, P. Talley, Y. Kao, C.H. Huang, A multi-class classification model for supporting the diagnosis of type II diabetes mellitus. Peer J, 8, (2020) e9920. https://doi.org/10.7717/peerj.9920

[13] B.S. Ahamed, M.S. Arya, S.K.B. Sangeetha, N.V. Auxilia Osvin, Diabetes Mellitus Disease Prediction and Type Classification Involving Predictive Modeling Using Machine Learning Techniques and Classifiers. Applied Computational Intelligence and Soft Computing, 2022, (2022) 2022. https://doi.org/10.1155/2022/7899364

[14] C. Mallika, S. Selvamuthukumaran, Hybrid Online Model for Predicting Diabetes Mellitus. Intelligent Automation & Soft Computing, 31(3), (2022) 1873-1885. https://doi.org/10.32604/iasc.2022.020543

[15] M. Kumari, V. Kumar, S.K. Verma, R.K. Mahli, Evaluation of lipid profile and glycated haemoglobin in type 2 diabetic patients: A retrospective study. International Journal of Health Sciences, (II), (2022) 4372-4383. https://doi.org/10.53730/ijhs.v6nS2.6009

[16] Z.M. Alhakeem, H. Hakim, O.A. Hasan, A.A. Laghari, A.K. Jumani, M.N. Jasm, Prediction of diabetic patients in Iraq using binary dragonfly algorithm with long-short term memory neural network. AIMS Electronics & Electrical Engineering, 7(3), (2023) 217-230. https://doi.org/10.3934/electreng.2023013

[17] M.O. Edeh, O.I. Khalaf, C.A. Tavera, S. Tayeb, S. Ghouali, G.M. Abdulsahib, N.E. Richard-Nnabu, A. Louni, A classification algorithm-based hybrid diabetes prediction model. Frontiers in Public Health 10, (2022) 829519. https://doi.org/10.3389/fpubh.2022.829519

[18] T. Beghriche, M. Djerioui, Y. Brik, B. Attallah, S.B. Belhaouari, An efficient prediction system for diabetes disease based on deep neural network. Complexity, 2021 (2021) 1-14. https://doi.org/10.1155/2021/6053824

[19] UCI Machine Learning Repository. (2025) Pima Indians Diabetes Database. https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[20] Mendeley Data, (2025) Iraqi Patient Dataset for Diabetes.

[21] Frankfurt Diabetes Dataset, (2025) Frankfurt Diabetes Dataset.

[22] K. Abnoosian, R. Farnoosh, M. H. Behzadi, Prediction of diabetes disease using an ensemble of machine learning multi-classifier models. BMC Bioinformatics, 24(1), (2023) 337. https://doi.org/10.1186/s12859-023-05465-z

[23] H. Naz, S. Ahuja, Deep learning approach for diabetes prediction using PIMA Indian dataset," Journal of Diabetes & Metabolic Disorders 19, (2020) 391-403. https://doi.org/10.1007/s40200-020-00520-5

[24] M.T. García-Ordás, C. Benavides, J.A. Benítez-Andrades, H. Alaiz-Moretón, I. García-Rodríguez, Diabetes detection using deep learning techniques with oversampling and feature augmentation. Computer Methods and Programs in Biomedicine, 202, (2021) 105968. https://doi.org/10.1016/j.cmpb.2021.105968

[25] A. Kishor, C. Chakraborty, Early and accurate prediction of diabetics based on FCBF feature selection and SMOTE. International Journal of System Assurance Engineering and Management, (2024) 4649–4657. https://doi.org/10.1007/s13198-021-01174-z

[26] M. Shrestha, O.H. Alsadoon, A. Alsadoon, T. Al-Dala'in, T.A. Rashid, P.W.C. Prasad, A. Alrubaie, 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), (2023) 6221-6241. https://doi.org/10.1007/s11042-022-13582-9

[27] S.M. Ganie, M.B. Malik, An ensemble machine learning approach for predicting type-II diabetes mellitus based on lifestyle indicators. Healthcare Analytics 2, (2022) 100092. https://doi.org/10.1016/j.health.2022.100092

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

https://doi.org/10.3390/app12083989

[29] K. Azbeg, M. Boudhane, O. Ouchetto, S. Jai Andaloussi, Diabetes emergency cases identification based on a statistical predictive model. Journal of Big Data, 9(1), (2022) 1-25. https://doi.org/10.1186/s40537-022-00582-7

Authors Contribution Statement

Both the authors equally contributed and approved the final version of this work.

Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes

About the License