A Novel Advanced Performance Ensemble-Based Model (APEM) Framework: A Case Study on Diabetes Prediction

Arda Yunianta

Department of Information Systems, Faculty of Computing and Information Technology Rabigh,
King Abdulaziz University, Jeddah, Kingdom of Saudi Arabia
Email: ayunianta@kau.edu.sa

***Abstract*—The use of machine learning algorithms for detection and prediction of diabetes diseases has been used widely in many studies. However, the accuracy of the prediction results in existing studies remained low. This research proposed a novel Advanced Performance Ensemble-based Model (APEM) framework as an enhancement method designed to enhance the accuracy level of predicting diabetes concerns compared with previous studies. Three main contributions are offered by APEM in this study as novel contributions. First, the paper discusses how to select the most appropriate preprocessing methods for the data, second, how to select and experiment with a number of machine learning algorithms as part of the ensemble learning process, and thirdly, how to Achieve the highest accuracy value compared to existing research. In general, there are three main stages in the APEM framework, the first stage is preprocessing data, the second stage is the ensemble method that uses five different machine learning algorithms, and the third stage is the second layer of the ensemble method with one machine learning algorithm. The result of this research produces better prediction results of diabetes prediction with an improvement in accuracy value of 99.06% compared with previous research, with a note that both this study and previous research utilized the same Pima Indian dataset and machine learning approach for their prediction.**

***Keywords*—ensemble method, diabetes, enhancement method, prediction, preprocessing methods, machine learning**

I. Introduction

Machine learning as a part of the Artificial Intelligence concept that is based on mathematical models has the purpose of processing the data with intelligence methods to enable computers to learn some problems and improve the achievement of the solutions [1]. There are many researchers used the machine-learning approach as a current solution for many cases such as in agriculture, air quality and polluted environment, energy poverty prediction, prediction for solar still performance, and prediction of specific diseases in healthcare [2–7]. However, this research focuses on the implementation of a machine-learning approach in the healthcare field to predict diabetes disease.

The prediction of diabetes diseases has been carried out using the implementation of machine learning. However, many studies still obtain low accuracy values to predict diabetes diseases from their experiment results. This issue poses a challenge in how to improve the accuracy level in diabetes prediction. There are several possible solutions to this issue. The first possible solution is how to preprocess the data to achieve high-quality data before processing in the next process. The selection of the best and appropriate data preprocessing methods is very important in machine learning implementation because if the data being processed is still dirty and contains many errors, it will have a big impact on the quality of prediction results [8–11]. The second possible solution is how to select the best and the proper machine learning methods to process the data and to get the best result on diabetes prediction. The selection of machine learning methods is crucial to maximize the data processing activities and implement the correct calculations and processing of the data from specific machine learning algorithms to produce better prediction results.

The selection of the ensemble model as the latest method in the machine learning approach is expected to give better results and the best result for diabetes prediction in this study. The aim of this research is to enhance the existing ensemble model to provide better results in diabetes prediction and improve the accuracy level of diabetes prediction. Finally, from the achievement results from this research, we compare the accuracy level between this research results and existing studies that used the same case study, dataset, and method on how to predict diabetes diseases.

The main contribution of this research is to propose a novel Advanced Performance Ensemble-based Model (APEM) Framework as an enhancement method to predict diabetes diseases. There are several achievements offered by APEM, the first achievement is the selection of the best methods namely Synthetic Minority Over-sampling Technique combined with Edited Nearest Neighbors (SMOTENN), Standard Scalar, and Optuna in the data preprocessing stage to provide a high-quality dataset with

Manuscript received April 12, 2024; revised July 15, 2024; accepted July 24, 2024; published October 23, 2024.

doi: 10.12720/jait.15.10.1193-1204

less/no error. The second achievement is to choose and experiment with the suitable and best machine learning algorithms namely Support Vector Machine, K-Nearest Neighbor, Gaussian naïve Bayes, Logistic Regression, Decision Tree, and Random Forest that are applied to APEM. Finally, the final goal as an achievement of this research is the enhancement of accuracy value compared with previous research that used the same Pima Indian dataset and machine learning approaches.

The representation of this research is divided into five parts. The first part is this section that explains the introduction of this research. The second part Section II presents all related studies that are similar to this research as information, knowledge, and comparison between existing studies and this research. The third part is the methodology presented in Section III to explain the proposed APEM framework and the explanation of every part of the APEM. The fourth part in Section IV is the implementation which explains the detail about experiment activities conducted in this research. The fifth part is a detailed explanation of the results and discussions of this research, and this is presented in Section V. The sixth part is conclusions and the possibility of future works, and this will be presented in Section VI.

II. Related Works

This paper will engage in discussions of relevant literature, with a focus on existing studies on diabetes prediction using machine learning methods. Emphasis will be placed on elucidating the review of existing studies and finding the gaps from existing studies to propose a solution offered by Advanced Performance Ensemble-based Model (APEM) Framework. However, the review will be limited to recently published articles, as advancements in performance accuracy within diabetes research, notably with the utilization of the Pima Indians Diabetes Dataset (Pima-IDD), have only recently emerged.

In 2022, Muhammad *et al.* [12] conducted a study to detect and predict diabetes diseases using supervised machine learning. This study used two different machine-learning algorithms namely Naïve Bayes and K-Nearest Neighbors. The background problem of this study was about there were many cases of diabetes in 2019 with the total number of cases being 463 million cases. The dataset used in this study was the Pima Indian dataset with nine different variables and 768 total number of records. There are several simple data preprocessing methods used in this study, such as merging, changing shapes, and transforming data to achieve an ideal dataset with several characteristics namely clean, reduce, integrate, and discretize. From several experimental activities, this study only achieved the highest accuracy value of 76.07% for the Naïve Bayes algorithm and 73.33% for the K-Nearest Neighbors algorithm.

Chollette *et al.* [13] conducted data preprocessing and machine learning approaches to detect and predict diabetes mellitus diseases in 2022. The idea of this study was to improve the existing machine learning performance, especially in how to increase the accuracy rate of diabetes prediction. This study used two different datasets, the first dataset is the Pima Indian dataset and the second dataset is from the laboratory of the Medical City Hospital (LMCH) diabetes dataset. In the preprocessing step, this study used missing value imputation and feature selection methods. The goal of the preprocessing step was to achieve the best dataset condition to process in the next step. The contribution of this study was to propose a framework that adopted polynomial regression and Spearman correlation for missing value imputation and feature selection activities in the data preprocessing step. For the prediction process, this study chooses three different supervised machine-learning models namely the support vector machine model, the random forest model, and their designed Twice-Growth Deep Neural Network (2GDNN) to produce better classification results. Based on the experiment results, this study achieved an accuracy level of 97.25% for the Pima Indian dataset and 97.33% for the LMCH diabetes dataset [13].

Another study that used single machine learning model is early prediction and diagnosis of diabetes diseases by Victor *et al.* [14] in 2022. The main objective of this study was to increase the quality of healthcare and patient outcomes using the implementation of a computer-based system in diabetes early detection and prediction. This study used a dataset from a subset of the Behavioral Risk Factor Surveillance System (BRFSS) dataset. There are four data preprocessing methods used in this study, the first method is Synthetic Minority Over-sampling Technique (SMOTE), the second method is Adaptive Synthetic Sampling (ADASYN), the third method is principal component analysis and the fourth method is T-distributed stochastic neighbor embedding. To achieve the detection and prediction result, this study used and compared five different machine learning classifiers. The first Machine Learning (ML) classifier is Decision Tree, the second ML classifier is Random Forest, the third ML classifier is K-Nearest Neighbors, the fourth ML classifier is Logistic Regression and the fifth ML classifier is Naïve Bayes. Based on several experiments with different machine learning classifiers, this study achieved the highest accuracy value of 82.26% for the Random Forest classifier [14].

The Pima Indian dataset was also used by Reza *et al.* [15] in 2023 to propose a method for diabetes prediction. This study used an improved Support Vector Machines algorithm to detect diabetes diseases. The contribution of this study was about the improvement of non-linear kernels namely radial basis function and Radial Basis Function (RBF) city block kernels. The improvement of these two kernels can enhance the performance of the Support Vector Machines model to detect and classify diabetes diseases. These two kernels also can help the Support Vector Machine model to adapt and learn complex decision boundaries. There are several preprocessing methods used in this study, the first method is using the median to address outliers and missing values problems, and the second method is by leveraging a robust synthetic-based over-sampling approach to handle class imbalance problems. The final result of this study was achieved accuracy level of 85.5% [15].

Rashi and Mamta [16] in 2023 conducted a study to detect and predict diabetes diseases using data mining techniques. Rashi and Mamta used four different data mining techniques namely Support Vector Machine, Naïve Bayes, Logistic Regression, and Random Forest algorithms. The performance measurements used in this study were accuracy performance metrics, sensitivity, and confusion matrix mechanisms. After the experiment activities, this study achieved an accuracy value of 82.46% [16].

The study about machine learning based on a smart healthcare framework to detect and predict diabetes diseases in 2023 by Alain *et al.* used two different machine learning algorithms and several data preprocessing methods. The two different machine learning algorithms used in this study namely Logistic Regression and Random Forest conducted in several experiments with different combinations between machine learning algorithms and data preprocessing methods. There are two main data preprocessing methods were used in this study, the first method is the hyperparameter tunning method using Python's GridSearchCV library, and the second method uses the feature selection method and data balancing method using Synthetic Minority Oversampling Technique, Cross-Validated and Recursive Feature Elimination. From different combination experiments this study achieved the best accuracy value of 98% for the Sylhet dataset and 81% for the Pima Indian dataset [17].

The next study about diabetes prediction was conducted by Liangjun *et al.* [18] in 2023. The background problem of this study was about the relationships between the risk of diabetes diseases and key lifestyle indicators in community follow-up was still ambiguous. To finish this study, they used 252,176 of diabetes data from year of 2016 to 2023, and this data gathered from Haizhu District, Guangzhou, China. This study was using feature selection method in the data preprocessing step. The purpose of this method was to determine the key life characteristic indicators that affect the diabetes and to optimize the feature subset. There are four different machine learning algorithms used in this study, the first algorithm is Random Forest, the second algorithm is the eXtreme Gradient Boosting, the third algorithm is the K-Nearest Neighbors and the forth algorithm is Ensemble Learning From experiment activities, this study achieved the highest accuracy level of 95.15% for Random Forest algorithm, and all of the accuracy result gathered and tested using the original data [18].

In 2024, Zaiheng *et al.* [19] proposed a prediction model namely AHDHS-Stacking for diabetes diagnosis using harmony search and stacking ensemble method. The main idea of this study was to detect, diagnose, and treat diabetes diseases by utilizing machine learning implementation and algorithms from metaheuristic optimization. The way how this model works is by utilizing the Harmony Search (HS) algorithm and stacking ensemble model as the latest approach in machine learning implementation. Adaptive hyperparameters and feature selection were chosen as strategies to improve the performance of the proposed model. This study used two datasets from the Chinese and Western Medicine Diabetes (CWMD) dataset and the Pima Indians Diabetes (PID) dataset. At the end of this study, the proposed model can achieve an accuracy value of 93.09%, a recall value of 91.60%, an MCC value of 84.79%, an F-measure value of 92.25%, and a precision value of 93.22% [19].

The study for monitoring and detection of diabetes mellitus using a machine learning approach was conducted by Alain *et al.* in 2024 using three different machine learning algorithms and achieved the best result by comparing three different results from three machine learning algorithms. Three different machine learning algorithms were namely Support Vector Machine, Logistic Regression, and Random Forest. This study also uses IoT-edge Artificial Intelligence and blockchain systems to implement the proposed system. There are several contributions in this study, the first contribution was using three different datasets namely Sylhet, Medical Information Mart for Intensive Care III (MIMIC III), and Pima Indian dataset to compare and analyze the best result from the experiment activities. The second contribution of this study was providing comparative results from three different machine learning algorithms in three different datasets. The third contribution to this study was comparing the results from different medical sensors, devices, and methods. There are different results of accuracy value in this study, the one accuracy result value on the Pima dataset was achieved at 81% [20].

Another study related to diabetes prediction was performed by Prabhakar *et al.* [21] in 2024. The problem background of this study was the drawback of the current machine learning method on failing to classify diabetes disease in the initial stage. From this problem background, this study proposed a user-cloud-based using an ensemble model. In the ensemble model, this research used three different machine learning algorithms namely artificial neural network, support vector machine, and decision tree classifier. The proposed model also uses a missing-value method in the data preprocessing phase. Based on the experiment and simulation results from the Pima Indian Diabetes dataset, this study achieved an accuracy level value of 87.41% [21].

From the review process above and Table I below, there are several drawbacks to previous studies. Firstly, from several studies, we found that they didn't focus on the data preprocessing methods and they didn't mention in detail the preprocessing activities and methods that used in their studies. Secondly, several studies that used an ensemble machine learning approach showed low accuracy results compared to the single machine learning approach. This means the selection of machine learning models used in the ensemble model is still not optimal and needs to be improved and need to select the proper machine learning models to achieve the highest accuracy value. Thirdly, all studies reviewed above show that the accuracy level value for the Pima Indian dataset is lower than other datasets. This means because of Pima Indian dataset has only 768 total records of data and nine variables, consequently with less number of data and less number of variables, it is a challenge to choose the correct preprocessing methods and suitable machine learning models to manipulate and process the data to achieve the best and the highest accuracy result.

A Novel Advanced Performance Ensemble-based Model (APEM) Framework is proposed to provide solutions for several limitations and weaknesses mentioned above. The APEM focuses on the selection of the best preprocessing data methods to provide a high-quality dataset before being used in the next processes. The utilization of standardization data is also one of the solutions to provide balancing data in the training dataset and testing dataset which is expected to enhance the ability of machine learning to detect minority classes, thereby resulting in a solid and robust model performance. The hyperparameter optimization was also chosen as a solution to get an optimal parameter for every machine learning model used in the APEM. The last novelty in the APEM is the use of double layers of machine learning processing and ensemble-based approach that shown and explained in detail in the next section.

TABLE I. THE COMPARISON OF SEVERAL STUDIES RELATED TO THE DIABETES PREDICTION

Reference	Machine Learning Algorithms	Dataset	Accuracy Value
Febrian *et al.* [12]	Naïve Bayes and K-Nearest Neighbors	Pima Indian dataset	76.07% for the Naïve Bayes algorithm and 73.33% for the K-Nearest Neighbors algorithm
Olisah *et al.* [13]	Support Vector Machine Model, Random Forest Model, and their designed Twice-Growth Deep Neural Network (2GDNN)	Pima Indian dataset and laboratory of the Medical City Hospital (LMCH) diabetes dataset	97.25% for the Pima Indian dataset and 97.33% for the LMCH diabetes datase
Chang *et al.* [14]	Decision Tree, Random Forest, K-Nearest Neighbors, Logistic Regression and Naïve Bayes	Behavioral Risk Factor Surveillance System (BRFSS) dataset	81.02% for Decision Tree, 82.26% for Random Forest, 80.55% for K-Nearest Neighbors, 72.64% for Logistic Regression and 70.56% for Naïve Bayes
Reza *et al.* [15]	improved Support Vector Machines algorithm	Pima Indian dataset	Accuracy level of 85.5% for improved Support Vector Machines algorithm
Rastogi and Bansal [16]	Support Vector Machine, Naïve Bayes, Logistic Regression, and Random Forest algorithms.	Their own dataset	79.22% for Support Vector Machine, 79.22% for Naïve Bayes, 82.46% for Logistic Regression, and 81.81% for Random Forest algorithms.
Hennebelle *et al.* [17]	Logistic Regression and Random Forest algorithms	Pima Indian dataset and the Sylhet dataset	98% for the Sylhet dataset and 81% for the Pima Indian dataset
Jiang *et al.* [18]	Random Forest, the eXtreme Gradient Boosting, the K-Nearest Neighbors and Ensemble Learning (VC)	They used 252,176 of diabetes data from year of 2016 to 2023, and this data gathered from Haizhu District, Guangzhou, China	95.15% for Random Forest, 67.98% for the eXtreme Gradient Boosting, 74.91% for the K-Nearest Neighbors and 85.56% for Ensemble Learning (VC)
Z. Zhang *et al.* [19]	the Harmony Search (HS) algorithm and stacking ensemble model	the Chinese and Western Medicine Diabetes (CWMD) dataset and the Pima Indians Diabetes (PID) dataset	Accuracy value of 93.09%
Hennebelle *et al.* [20]	Support Vector Machine, Logistic Regression, and Random Forest	namely Sylhet, MIMIC III, and Pima Indian dataset	76% for Support Vector Machine in Pima Indian Dataset, 77.34% for Logistic Regression in MIMIC III dataset, and 98% for Random Forest in Sylhet dataset
GPrabhakar *et al.* [21]	The ensemble model with three different machine learning algorithms namely Artificial neural network, support vector machine, and decision tree classifier	Pima Indian dataset	Accuracy level value of 87.41%

III. PROPOSED ADVANCED PERFORMANCE ENSEMBLE-BASED MODEL (APEM) FRAMEWORK

A. Research Framework

The focus of this research is to find the optimal solution for addressing diabetes prediction by employing a more robust machine-learning approach that yields superior performance. Machine learning is preferred for its ability to achieve optimal performance and its capability to work with limited datasets. Nevertheless, native machine learning with improved approaches can be further enhanced to attain superior results.

This research concerned with improvement through the integration of multiple approaches including data preprocessing, data resampling, and ensemble modeling. Moreover, utilization of hyperparameter optimization to acquire the most effective hyperparameters for each machine-learning model, thus attaining optimal model performance. Fig. 1 depicts the general pipeline of architecture for Advanced Performance Ensemble-based Model (APEM) Framework.

The first stage of our pipeline encompasses data preprocessing. Within this stage, we undertake standard data preprocessing procedures, including the identification and rectification of any errors or inconsistencies in the data, such as missing values, duplicates, or outliers.

This research also involve data resampling. Similar to numerous public datasets, the Pima Indians Diabetes Dataset exhibits class imbalance, a prevalent challenge in real-world datasets that can substantially affect the efficacy of machine learning models. Therefore, the utilization of the Synthetic Minority Over-sampling Technique combined with Edited Nearest Neighbors (SMOTENN) [22] in the Pima-IDD dataset is to provide an efficient implementation of the neighborhoods related to the minority samples of data and produce a more balanced and representative dataset, fostering robust machine learning models and producing more dependable predictions.

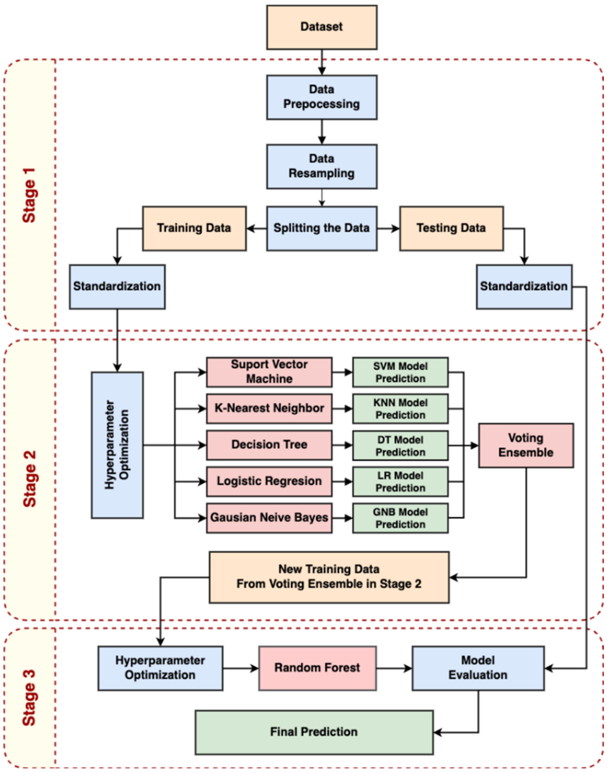

Fig. 1. A novel Advanced Performance Ensemble-based Model (APEM) framework.

Once a balanced dataset is obtained, the next step involves partitioning the dataset into two subsets: a training dataset and a testing dataset. This division is carried out with a ratio of 70% for the training data and 30% for the testing data. The training set is utilized to train the machine-learning model, while the testing set is employed for evaluating the performance of the machine learning models. The final step in stage one is to normalize the dataset. We integrate standardization, specifically employing a standard scaler, as a preprocessing step for our dataset to ensure the robust and reliable performance of our machine learning models.

Following the data preprocessing phase, the second stage involves modeling using machine learning techniques. Our approach entails an ensemble voting model, employing several base machine learning such as Support Vector Machine, k-Nearest Neighbor, Logistic Regression, and Gaussian Naïve Bayes.

Before training process in the stage two, hyperparameter optimization is performed to enhance the models' effectiveness. To optimize machine-learning hyperparameters, we utilized the Optuna framework [22]. Then, by employing a voting ensemble model, we generated predictions for the training dataset. These predictions were then combined with the corresponding class labels to construct a new dataset. It is expected that the new dataset will offer more robust insights into the problem domain concerning the classification of diabetes diseases.

In the third stage, we employ the new training dataset to train a Random Forest, which serves as the final classifier. Analogous to the preceding training step, during this process, we utilize Optuna for hyperparameter optimization. Subsequently, utilizing the testing dataset, we conduct predictions to discern the underlying patterns and relationships between the input features and the target classes.

Algorithm 1: A novel Advanced Performance Ensemble-based Model (APEM)

```
STEP1: START
STEP2: Import a module named pandas as pd
STEP3: data is equal to pd.read_csv("data_cln.csv")
STEP4: X is equal to data.iloc[:,:-1]
STEP5: y is equal to data.iloc[:, -1]
STEP6: from imblearn.combine Import a module named SMOTEENN
STEP7: smotenn is equal to SMOTEENN (random_state=42)
STEP8: X_resampled, y_resampled is equal to smotenn.fit_resample(X, y)
STEP9: Display the following in the console ("number of data beInitiate a for loop with variablee undersampling:", X.shape[0])
STEP10: Display the following in the console ("number of data after undersampling:", X_resampled.shape[0])
STEP11: from sklearn.model_selection Import a module named train_test_split
STEP12: rs is equal to 100
STEP13: X_train, X_test, y_train, y_test is equal to train_test_split(X_resampled, y_resampled, test_size=0.3, stratCheck whether y=y_resampled, random_state=rs)
STEP14: from sklearn.model_selection Import a module named RepeatedKFold
STEP15: cv is equal to RepeatedKFold(n_splits=10 , n_repeats=5, random_state=rs)
STEP16: from sklearn.preprocessing Import a module named StandardScaler
STEP17: sc is equal to StandardScaler()
STEP18: X_train is equal to sc.fit_transInitiate a for loop with variablem(X_train)
STEP19: X_test is equal to sc.transInitiate a for loop with variablem(X_test)
STEP20: from sklearn.svm Import a module named SVC
STEP21: from sklearn.neighbors Import a module named KNeighborsClassCheck whether ier
STEP22: from sklearn.naive_bayes Import a module named GaussianNB
STEP23: from sklearn.linear_model Import a module named LogisticRegression
STEP24: from sklearn.tree Import a module named DecisionTreeClassCheck whether ier
STEP25: from sklearn.ensemble Import a module named VotingClassCheck whether ier
STEP26: from sklearn.metrics Import a module named accuracy_score
STEP27: model_1 is equal to SVC (C= 14.999257923224931, gamma= 0.6329913798108597, probability=True)
STEP28: model_2 is equal to KNeighborsClassCheck whether ier(n_neighbors= 10, weights= 'distance', metric= 'manhattan', p= 28, algorithm= 'kd_tree', leaf_size= 86)
STEP29: model_3 is equal to GaussianNB(var_smoothing= 0.0006962434993094332)
STEP30: model_4 is equal to LogisticRegression(C= 6.313285120853204, solver= 'liblinear', max_iter= 6389)
STEP31: model_5 is equal to DecisionTreeClassCheck whether ier(max_depth= 54, ccp_alpha= 0.007525972152059758,
max_features= ‘log2’, min_samples_split= 6, min_samples_leaf= 2, criterion= ‘entropy’)
    STEP32: ensemble_model is equal to VotingClassCheck whether ier(estimators=[(‘SVC’, model_1),
                            (‘KNeighborsClassCheck whether ier’, model_2),
                            (‘GaussianNB’, model_3),
                            (‘LogisticRegression’, model_4),
                            (‘DecisionTreeClassCheck whether ier’, model_5)], voting=‘hard’)
    STEP33: ensemble_model.fit(X_train, y_train)
    STEP34: y_pred is equal to ensemble_model.predict(X_test)
    STEP35: accuracy is equal to accuracy_score(y_test, y_pred)
    STEP36: Display the following in the console (“Accuracy:”, accuracy)
    STEP37: from sklearn.metrics Import a module named confusion_matrix
    STEP38: Import a module named matplotlib.pyplot as plt
    STEP39: Import a module named seaborn as sns
    STEP40: cm is equal to confusion_matrix(y_test, y_pred)
    STEP41: plt.figure()
    STEP42: sns.heatmap(cm, annot=True, fmt=‘d’, cmap=‘Blues’)
    STEP43: plt.xlabel(‘Pred’)
    STEP44: plt.ylabel(‘Akt’)
    STEP45: plt.title(‘Confusion Matrix’)
    STEP46: plt.show()
    STEP47: from sklearn.metrics Import a module named  Define a classCheck whether ication_report
    STEP48: Display the following in the console (Define a classCheck whether ication_report(y_test, y_pred, zero_division=0))
    STEP49: Import a module named  numpy as np
    STEP50: Import a module named  pandas as pd
    STEP51: predicted_labels is equal to ensemble_model.predict(X_train)
    STEP52: X_train_labeled is equal to np.column_stack((X_train, predicted_labels))
    STEP53: column_names is equal to [‘Pregnancies’,’Glucose’,’BloodPressure’,’SkinThickness’,’Insulin’, ’BMI’,’DiabetesPedigreeFunction’,’Age’,’Outcome’]
    STEP54: df is equal to pd.DataFrame(X_train_labeled, columns=column_names)
    STEP55: df.to_csv(‘labellingresult.csv’, index=False)
    STEP56: dataset is equal to pd.read_csv(“labellingresult.csv”)
    STEP57: dataset
    STEP58: X1 is equal to dataset.iloc[:,:-1]
    STEP59: y1 is equal to dataset.iloc[:, -1]
    STEP60: from sklearn.model_selection Import a module named cross_val_score
    STEP61: from sklearn.ensemble Import a module named RandomForestClassCheck whether ier
    STEP62: Define a funtion objective_rf(trial)

            param_rf is equal to {‘max_depth’: trial.suggest_int(“max_depth”, 2, 64),
                    ‘max_features’: trial.suggest_categorical(‘max_features’,[‘sqrt’, ‘log2’]),
                    ‘n_estimators’: trial.suggest_int(“n_estimators”,10, 200),
                    ‘min_samples_split’: trial.suggest_int(“min_samples_split”, 2, 30),
                    ‘min_samples_leaf’: trial.suggest_int(“min_samples_leaf”, 1, 30),
                    ‘criterion’: trial.suggest_categorical(“criterion”, [“gini”, “entropy”])}

            rafo is equal to RandomForestClassCheck whether ier(**param_rf,random_state=rs)
            rafo.fit(X1, y1)

            score is equal to cross_val_score(rafo, X1, y1, cv=cv, scoring=“accuracy”).mean()
            Return score
    STEP63: Import a module named optuna
    STEP64: study_dect is equal to optuna.create_study(direction=‘maximize’,study_name is equal to “rafo”)
    STEP65: study_dect.optimize(objective_rf, n_trials=100)
    STEP66: Display the following in the console (“Best trial:”, study_dect.best_trial.number)
    STEP67: Display the following in the console (“Best accuracy:”, study_dect.best_trial.value)
    STEP68: Display the following in the console (“Best hyperparameters:”, study_dect.best_params)
    STEP69: from sklearn.metrics Import a module named confusion_matrix
    STEP70: Import a module named matplotlib.pyplot as plt
    STEP71: Import a module named seaborn as sns
    STEP72: from sklearn.metrics Import a module named  Define a classCheck whether ication_report
    STEP73: best_param_dect is equal to study_dect.best_params
    STEP74: rafo is equal to RandomForestClassCheck whether ier(**best_param_dect, random_state=rs).fit(X1, y1)
    STEP75: y_pred_raf is equal to rafo.predict(X_test)
    STEP76: Define a funtion display_results(y_test, y_pred_raf, cm_title)
            cm is equal to confusion_matrix(y_test,y_pred_raf)
            sns.heatmap(cm, annot=True, fmt=‘d’).set_title(cm_title)
            Display the following in the console (Define a classCheck whether ication_report(y_test,y_pred_raf))
    STEP77: display_results(y_test, y_pred_raf, cm_title is equal to “Confusion matrix (rafo Model)”)
STEP78: STOP
```

B. Data Resampling

The performance of classifiers hinges not only on the selected algorithm but also on the quality of the input data. However, the actual dataset suffers from skewness. In such instances, an imbalanced dataset often introduces bias in predictions favoring the majority class [23].

To address the class imbalance issue, one of the prevailing strategies involves resampling the dataset to achieve better balance. Synthetic Minority Over-sampling Technique combined with Edited Nearest Neighbors (SMOTENN) [24], offers the capability to rebalance datasets using two concurrent approaches. It addresses this imbalance by generating synthetic instances for the minority class using Synthetic Minority Over-sampling Technique (SMOTE), thereby augmenting its representation in the dataset. Additionally, the integration of Edited Nearest Neighbors (ENN) works to refine the synthetic samples by eliminating potentially noisy instances, thus enhancing the overall dataset quality.

C. Data Normalization

By standardizing the dataset using a standard scaler, we center the data on zero and scale it to have a standard deviation of one, thus simplifying the optimization process and enhancing the convergence of machine learning algorithms. Furthermore, standardization not only aids in the interpretation of model coefficients but also improves model performance. Hence, the utilization of a standard scaler guarantees that our machine learning models accurately capture the inherent data patterns while mitigating the effects of feature scaling inconsistencies. The representation of the standard scaler is depicted in Eq. (1).

$$x_{std} = \frac{x - mean(x)}{std.dev(x)} \quad (1)$$

where, x denotes an individual data point from the dataset, x_{std} represents the standardized value of a particular data point x. The term $mean(x)$ signifies the average value of all the data points and $std.dev(x)$ represents the standard deviation of the dataset, which measures the dispersion or spread of the data points around the mean.

D. Hyperparameter Optimization

In every classification and prediction tasks, hyperparameter optimization play an important role to provide the best accuracy result by performing the parameter selection in many machine learning algorithms [25, 26]. By making changes and modifications to the hyperparameter values of each machine learning algorithm, it has been proven that it can improve the performance of the machine learning model and can also reduce training time very significantly. The optimization process is usually an iterative process by constantly changing all parameter values to find optimal values and produce greater accuracy in the results of the machine learning process [27].

Exploring hyperparameters stands out as one of the most challenging tasks within machine learning endeavors. With the increasing complexity of machine and deep learning methodologies, there emerges a pressing need for a proficient framework capable of automatically adjusting hyperparameters. The Optuna framework [22] presents a systematic approach to tackle hyperparameter exploration, methodically assessing various combinations to identify configurations that optimize the model's performance metrics.

Utilizing the Optuna for hyperparameter optimization enables us to finely tune our machine-learning models, thus elevating their predictive accuracy and robustness while minimizing the necessity for extensive manual tuning endeavors. Moreover, leveraging Optuna as the framework for hyperparameter tuning facilitates the automated exploration of hyperparameter spaces. This process assists in refining pivotal parameters such as learning rates, regularization strengths, and network architectures, thereby ultimately enhancing the predictive accuracy and generalization capabilities of the models.

E. Machine Learning Methods

This section aims to elaborate machine learning models that employed in the APEM (Advanced Performance Ensemble-based Model) framework, including Support Vector Classifier (SVC), k-Nearest Neighbor (KNN), Random Forest (RF), Gaussian Naive Bayes (GNB), Logistic Regression (LR), and Decision Tree (DT) as base models in the stage 2 (Fig. 1). Additionally, we will discuss the Voting Ensemble and Random Forest as the final models utilized in this paper.

1) Support vector classifier

As a primary framework in machine learning, Support Vector Machines (SVM) [28, 29] are founded upon the principles of the Vapnik–Chervonenkis theory and the concept of structural risk minimization. The SVM seeks to strike a balance between minimizing the error on the training set and maximizing the margin to attain optimal generalization capabilities while guarding against overfitting.

SVM demonstrates exceptional utility in scenarios involving high-dimensional datasets or those where linear separation is not feasible. Renowned for their capacity to generalize effectively to novel data and manage intricate decision boundaries, SVMs stand as a formidable tool in machine learning applications.

2) K-Nearest neighbor

The K-Nearest Neighbor (KNN) [30] algorithm is a classification method employed in data mining to forecast outcomes for unclassified data (test data) by leveraging the nearest neighbor information. This approach involves identifying a subset of nearest neighbors to the test data and assigning its class based on the predominant class among the identified nearest neighbors. Euclidean distance is utilized for estimation, following Eq. (2).

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

where, d represents the Euclidean distance between two points in a two-dimensional space, where x_1 and y_1 denote the coordinates of the first point, and x_2 and $y_2$denote the coordinates of the second point.

3) Random forest

Random Forest (RF) is characterized by its methodology of generating multiple decision trees, with each tree contributing to the decision-making process. Usually, a subset of n data points is randomly selected from the dataset, and their amalgamation yields a robust decision. In cases where multiple predictions are made, the average of these predictions is utilized. Both classification and regression challenges are addressed employing the RF technique [14, 31] The architecture of RF comprises numerous trees, each offering a distinct choice [13]. By averaging all the choices, the most recent prediction is determined.

4) Gaussian naive bayes

The Gaussian Naive Bayes classifier is a fundamental probabilistic classification model that utilizes the Bayesian theorem under the assumption of strong (naive) independence [16]. Also referred to as Simple Bayes and Independence Bayes, Naïve Bayes computes the probability for each class and chooses the one with the highest likelihood. In contrast, Bayes' Theorem elucidates the probability of an event given prior knowledge of conditions potentially associated with that event. The Naïve Bayes model follows Eq. (3) [32].

$$P(H|E) = \frac{P(E|H) \times P(H)}{P(E)} \quad (3)$$

where, $P(H|E)$ represents the posterior probability of the hypothesis, $P(E|H)$ denotes the likelihood of the evidence given the hypothesis is true, (H) signifies the prior probability of the hypothesis, and $P(E)$ denotes the prior probability that the evidence is true.

5) *Logistic regression*

Logistic Regression employs a logistic function for estimating the likelihood of a binary outcome based on input features. In contrast to linear regression, which forecasts continuous values, Logistic Regression deals with the likelihood of an event happening, usually within the range of 0 to 1. Through adjusting the logistic function to the training data using optimization techniques like gradient descent, the model acquires an understanding of the connection between input features and the binary target variable.

This method forecasts the probability of an observation belonging to the binary class by utilizing a sigmoid function, as specified in Eq. (4) [20].

$$P(A) = \frac{e^{\beta_0} + \sum_{i=1}^{n} \beta_i R_i}{1 + e^{\beta_0 + \sum_{i=1}^{n} \beta_i R_i}} \tag{4}$$

In this context, $P(A)$ denotes the probability of belonging to class A, where R stands for the set of risk factors, and β_0 and β_i signify the regression coefficients, representing the intercept and the slope respectively. The regression coefficient values are determined through maximum likelihood estimation, ensuring that the value of Eq. (5) is maximized.

$$l(\beta_0, \ldots, \beta_1) = \prod_{i,y_1=1} P(A) \prod_{i,y_i=0} \big(1 - p(A)\big) \tag{5}$$

6) *Decision tree*

Decision Tree (DT) is a supervised Machine Learning (ML) algorithm extensively utilized for addressing classification and regression tasks. In a decision tree, each leaf node signifies the classification outcomes, while each internal node represents attribute judgments [33].

The algorithm is employed to construct the decision tree, employing a top-down learning approach. The process of building a decision tree involves several steps: firstly, selecting the most appropriate attribute for the root node; secondly, dividing instances into subsets where each subset's instances possess identical attribute values; finally, recursively repeating this process for each subset until all instances within them share identical classes.

7) *Voting ensemble*

The ensemble method integrates multiple classifiers to enhance the performance of individual classifiers. Ensemble algorithms are designed to yield more robust, precise, and accurate results [34]. A voting classifier employs two types of voting techniques: hard and soft. In hard voting, the final prediction is determined by a majority vote, wherein the aggregator selects the class prediction most frequently occurring among the base models. Conversely, soft voting requires the base models to possess the predict_proba method. The voting classifier demonstrates superior overall performance compared to other base models, as it amalgamates predictions from diverse models [35].

In the proposed framework, the ensemble comprises Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree (DT), Logistic Regression (LR) and Gaussian Naïve Bayes (GNB) classifiers. A soft voting classifier is employed, utilizing the predict_proba attribute column, which provides the probability of each target variable. Subsequently, the training data and data points are shuffled, and these points are then fed into logistic regression, Naïve Bayes, and Random Forest models. Each model computes individual predictions, and through the voting aggregator and soft voting technique, a majority voting process determines the final prediction.

F. The New Training Data

In Stage 2 (Fig. 1), the ensemble learning process was performed using several machine learning algorithms. This process used the original training data to produce a machine learning model, and then using that model to produce a new prediction result. Furthermore, from predict model produced a new data namely the new training data. In details, the new training data produced from every feature in the training data with a new labelling from prediction model on the ensemble learning process.

IV. Implementation

A. Systems

This paper presents the development and training of the APEM model, undertaken using an AMD Ryzen 5 5600G, coupled with a 12 GB NVIDIA RTX 3060 Graphics Processing Unit and 80 GB of Random Access Memory (RAM). The development process of the model involved the utilization of Python programming and modules such as Scikit-learn, pandas, and NumPy.

B. Dataset

This study used the Pima Indians Diabetes Dataset (Pima-IDD) [36] to evaluate the proposed framework. This dataset is publicly available and originates from the National Institute of Diabetes and Digestive and Kidney Diseases. The main aim is to predict diagnostically whether a patient has diabetes, utilizing specific diagnostic measurements from the dataset. Instances selected from a database were conducted by various constraints, notably including the requirement that all patients be females of Pima Indian descent aged at least 21 years. The dataset encompasses multiple medical predictor variables in addition to a single target variable, referred to as "Outcome". These predictor variables encompass factors such as the number of pregnancies, BMI (Body Mass Index), insulin levels, and age, among others [37]. Table II presents the correlation matrix values concerning the target class outcome, while Fig. 2 illustrates the correlation matrix of all features.

TABLE II. The Correlation Matrix of All Features Pima-IDD to the "Outcome" Class

Features	Value
Glucose	0.4903
Insulin	0.4812
SkinThickness	0.3113
BMI (Body Mass Index)	0.3061
Age	0.2556
Pregnancies	0.2318
DiabetesPedigreeFunction	0.1915
BloodPressure	0.1721

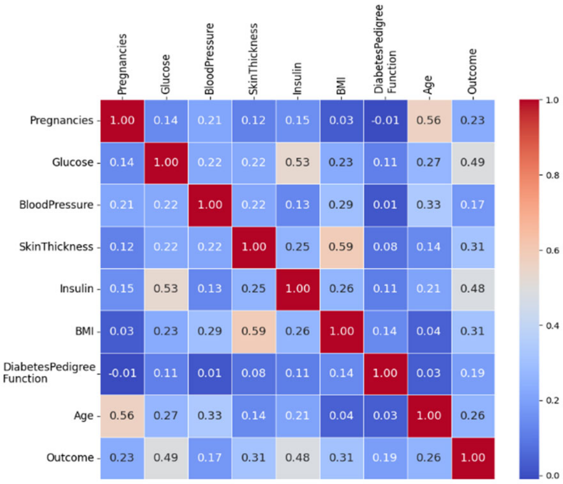

Fig. 2. Correlation matrix Pima-IDD (the Pima Indians Diabetes Dataset).

C. *Evaluation Metrics*

In evaluating the performance of classification models, various common metrics are employed to offer a comprehensive understanding of their effectiveness. Accuracy measures the ratio of correctly classified instances to the total number of instances. Precision captures the proportion of true positive predictions among all positive predictions made by the model. Similarly, recall quantifies the proportion of true positive instances correctly identified by the model among all actual positive instances. F1-Score provides a balanced assessment of a model's performance, particularly valuable in addressing class imbalances. The accuracy, precision, recall, and F1-Score are represented in Eqs. (6)–(9).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

Furthermore, the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) are used to illustrate the evaluation of the performance of classifiers across various threshold settings. The ROC curve plots the True Positive Rate (TPR) versus the False Positive Rate (FPR), while the AUC serves to summarize the classifier's overall performance, with a higher AUC indicating superior discrimination between positive and negative instances [38]. The TPR and FPR are represented by Eqs. (10) and (11).

$$TPR = \frac{TP}{TP + FN} \tag{10}$$

$$FPR = \frac{FP}{FP + TN} \tag{11}$$

where, TP represents true positives, TN denotes true negatives, FP signifies false positives, and FN indicates false negatives.

V. Results and Discussions

In this section will focusing on the outlining the results of the implementation of the proposed framework. Initially, we will elucidate the methodology employed in data preprocessing, which includes data resampling and optimization of machine learning parameters. Furthermore, to illustrate the substantial enhancements in performance achieved by the proposed framework in the context of Pima-IDD, a comparative analysis is conducted between the proposed framework and native machine learning models serving as benchmarks.

A. *Data Resampling Analysis*

One of the essential strategies in this research is to enhance machine-learning performance involves careful handling of the dataset. Pima-IDD inherently exhibits skewness, marked by a pronounced class imbalance. To address this, we have incorporated data resampling techniques as a fundamental aspect of our methodology. Specifically, SMOTENN is utilized to rebalance the dataset by removing instances from the majority classes and augmenting instances of the minority classes.

Through the utilization of data resampling, a more balanced dataset is obtained. It is expected that this balanced dataset will enhance the ability of machine learning to detect minority classes, thereby resulting in a solid and robust model performance. Table III presents a comparison between the original and resampled datasets.

TABLE III. Comparison of Instances between the Original and Resampled Datasets

Dataset	Normal	Diabetes	Total
Original Dataset	481	252	733
Resampling Dataset	382	327	709

B. *Hyperparameter Optimization Analysis*

In order to ensure equitable comparison, we trained both our proposed model and the benchmark machine-learning model with optimal parameters. Employing Optuna, we conducted experiments aimed at acquiring the most effective hyperparameters for each machine-learning model. The optimal hyperparameters identified through the utilization of Optuna are presented in Table IV.

TABLE IV. The Optimum Hyperparameters for Each Machine Learning Method

Model	Optimum Hyperparameter

SVC	C = 2.335073123358093,
	Kernel = rbf,

KNN	n_neighbors = 9,
	p = 1

RF	n_estimators = 160, max_depth = 26,
	min_samples_split = 0.1757711067111652,
	min_samples_leaf = 0.10013205156939448

GNB	-
LR	C = 0.22493480225945017, penalty = l2

DT	max_depth = 18,
	min_samples_split = 0.529501941797138,
	min_samples_leaf = 0.432488258731745,

APEM	max_depth = 44, max_features = log2, n_estimators = 74, min_samples_split = 5, min_samples_leaf = 1, criterion = entropy

C. Model Performance Analysis

This section aims to evaluate the performance of the proposed framework by comparing the proposed framework with various machine learning and ensemble models for the Pima-IDD dataset. Among the individual models, the Support Vector Classifier (SVC) and K-Nearest Neighbors (KNN) exhibited the highest accuracy at 98.12%, with precision, recall, and F1-Score metrics also reaching 98.19% and 98.12% respectively. Notably, the Gaussian Naive Bayes (GNB) and Logistic Regression (LR) models performed slightly less effectively, achieving accuracy scores of 93.43% and 94.37% respectively. Conversely, the Stacked Ensemble Model (SEM) achieved a commendable accuracy of 94.17%. However, the proposed framework surpassed all others, achieving an accuracy of 99.06% and maintaining consistency across precision, recall, and F1-Score metrics, surpassing both traditional machine learning and SEM models. Our results underscore the effectiveness of our proposed methods in enhancing predictive performance for the given task. Table V provides a comparison of the proposed framework with other machine learning models, while Fig. 3 illustrates the comparison performance using the accuracy, precision, recall, and F1-Score metrics of the proposed framework. All models are trained using a similar preprocessing approach.

TABLE V. PERFORMANCE ACCURACY, PRECISION, RECALL, AND F1-SCORE COMPARISON BETWEEN APEM AND BENCHMARK MODELS

Model		Accuracy	Precision	Recall	F1-Score
Single Machine Learning Model	SVC	98.12	98.19	98.12	98.12
	KNN	98.12	98.19	98.12	98.12
	RF	97.18	97.20	97.18	97.18
	GNB	93.43	93.54	93.43	93.41
	LR	94.37	94.41	94.37	94.36
	DT	97.18	97.26	97.18	97.19
Ensemble Model	SEM [39]	94.17	95.92	94.95	95.43
	APEM [Proposed]	**99.06**	**99.06**	**99.06**	**99.06**

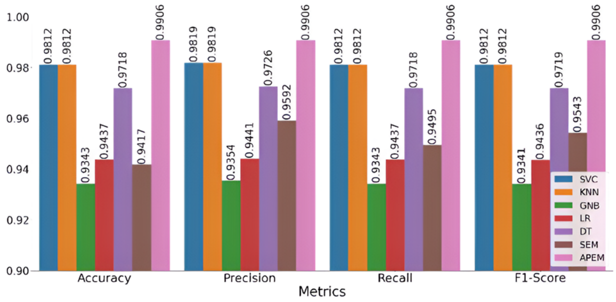

Fig. 3. The comparison results between the proposed model and several machine-learning models.

Receiver Operating Characteristic (ROC) curve is the one of the machine learning models performance metric evaluation [40, 41]. In relation with ROC, the value of the Area Under the Curve is also important to know the performance of machine learning models. Performance can be rated as good if the score reaches a maximum score of 1 and rated as bad if the score is at a minimum score of 0. In terms of the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve metrics, our proposed model also demonstrates superior performance compared to native machine learning models. Specifically, APEM yielded an Area Under the Curve (AUC) of 0.9995, surpassing other traditional machine-learning approaches. This indicates that APEM exhibits greater ability to discriminate between classes than benchmark models. The visualization of the AUC is presented in Fig. 4.

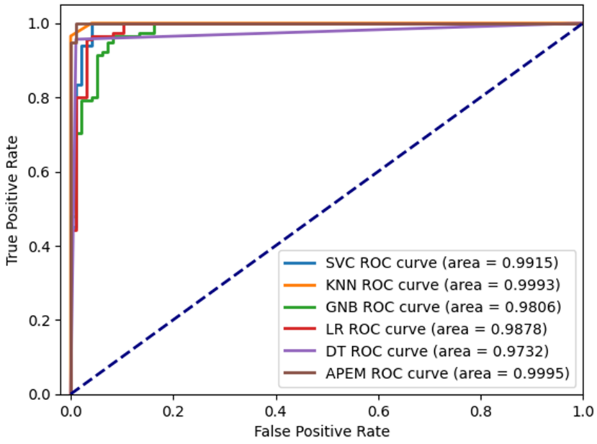

Fig. 4. The comparison AUC/ROC: APEM vs Native Machine Learning Model.

D. Comparison with Existing Works

The effectiveness of the proposed approach in creating a proficient NIDS was verified through a comparison with prior studies. This evaluation centered on various metrics including accuracy, precision, recall, and F1-Score, providing a thorough evaluation of the advantages of the proposed model. Table VI show the comparison of accuracy level between previous studies and the proposed method APEM (Advanced Performance Ensemble-based Model).

TABLE VI. COMPARISON OF PROPOSED MODEL AND EXISTING WORKS ON PIMA-IDD DATASET

Existing Works	Single or Ensemble Machine Learning Approach	Accuracy (%)
Hennebelle *et al.* [20]	Support Vector Machine, Logistic Regression, and Random Forest	81
Reza *et al.* [15]	Support Vector Machines	85.5
Rastogi and Bansal [16]	Support Vector Machine, Naïve Bayes, Logistic Regression, and Random Forest	82.46
Hennebelle *et al.* [17]	Logistic Regression and Random Forest	81
Febrian *et al.* [12]	Naïve Bayes and K-Nearest Neighbors	73.33
Olisah *et al.* [13]	Support Vector Machine model, Random Forest model, and their designed Twice-Growth Deep Neural Network (2GDNN)	97.33
Chang *et al.* [14]	Decision Tree, Random Forest, K-Nearest Neighbors, Logistic Regression, and Naïve Bayes	82.26
Zhang *et al.* [19]	Harmony Search and Stacking Ensemble method	93.09
Prabhakar *et al.* [21]	Ensemble model with three different machine learning algorithms namely Artificial Neural Network, Support Vector Machine, and Decision Tree classifier	87.41
Jiang *et al.* [18]	Random Forest, eXtreme Gradient Boosting, K-Nearest Neighbors and Ensemble Learning (VC).	95.15
Rahim *et al.* [39]	Stacked ensemble model, Support Vector Machine (SVM), K Nearest Neighbor (KNN), Naïve Bayes (NB), Random Forest (RF), and Logistic Regression	94.17
APEM [Proposed]	**Ensemble ML with Support Vector Machine, K-Nearest Neighbor, Gaussian naïve Bayes, Logistic Regression, Decision Tree, and Random Forest**	**99.06**

VI. CONCLUSIONS AND FUTURE WORKS

This study presented enhancement method namely APEM (Advanced Performance Ensemble-based Model) Framework, and designed for diabetes prediction. This study approach incorporates a pipeline for classifying the Pima Indians Diabetes Dataset (Pima-IDD) dataset. Our findings illustrate that APEM surpasses benchmark models, highlighting substantial enhancements over native machine learning techniques. The integration of key strategies, such as data resampling, hyperparameter optimization, and ensemble modeling, contributes to the superior performance of the final model.

The possibilities of the future works of this study are trying to implement this framework with different datasets and case studies to know accuracy level achievement in different datasets or case studies. The consideration of change and choosing different methods such as deep learning methods also interesting to try in this framework to achieve better accuracy results

CONFLICT OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] R. F. d. Mello and M. A. Ponti, *Machine Learning: A Practical Approach on the Statistical Learning Theory*, Springer, 2018.

[2] U. Ali, S. Bano, M. H. Shamsi *et al.*, "Urban building energy performance prediction and retrofit analysis using data-driven machine learning approach," *Energy and Buildings*, vol. 303, 113768, 2024.

[3] D. Al Kez, A. Foley, Z. K. Abdul, and D. F. del Rio, "Energy poverty prediction in the United Kingdom: A machine learning approach," *Energy Policy*, vol. 184, 113909, 2024.

[4] A. S. Abdullah, A. Joseph, A. W. Kandeal *et al.*, "Application of machine learning modeling in prediction of solar still performance: A comprehensive survey," *Results in Engineering*, vol. 21, 101800, 2024.

[5] G. Ravindiran, G. Hayder, K. Kanagarathinam, A. Alagumalai, and C. Sonne, "Air quality prediction by machine learning models: A predictive study on the Indian coastal city of Visakhapatnam," *Chemosphere*, vol. 338, 139518, 2023.

[6] Z. Kuang, Y. Zhao, and X. Yang, "Machine learning approaches for plant miRNA prediction: Challenges, advancements, and future directions," *Agriculture Communications*, vol. 1, no. 2, 100014, 2023.

[7] S. S. Bhat, M. Banu, G. A. Ansari, and V. Selvam, "A risk assessment and prediction framework for diabetes mellitus using machine learning algorithms," *Healthcare Analytics*, vol. 4, 100273, 2023.

[8] A. Ahmad, X. Xiao, H. Mo, and D. Dong, "Tuning data preprocessing techniques for improved wind speed prediction," *Energy Reports*, vol. 11, pp. 287–303, 2024.

[9] D. Nuñez-Ramirez, D. Mata-Mendoza, and M. Cedillo-Hernandez, "Improving preprocessing in reversible data hiding based on contrast enhancement," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 5468–5477, 2022.

[10] K. Graff, R. Tansey, A. Ip *et al.*, "Benchmarking common preprocessing strategies in early childhood functional connectivity and intersubject correlation fMRI," *Developmental Cognitive Neuroscience*, vol. 54, 101087, 2022.

[11] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data," *Frontiers in Energy Research*, vol. 9, 2021.

[12] M. E. Febrian, F. X. Ferdinan, G. P. Sendani, K. M. Suryanigrum, and R. Yunanda, "Diabetes prediction using supervised machine learning," *Procedia Computer Science*, vol. 216, pp. 21–30, 2023.

[13] C. C. Olisah, L. Smith, and M. Smith, "Diabetes mellitus prediction and diagnosis from a data preprocessing and machine learning perspective," *Computer Methods and Programs in Biomedicine*, vol. 220, 106773, 2022.

[14] V. Chang, M. A. Ganatra, K. Hall, L. Golightly, and Q. A. Xu, "An assessment of machine learning models and algorithms for early prediction and diagnosis of diabetes using health indicators," *Healthcare Analytics*, vol. 2, 100118, 2022.

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

[16] R. Rastogi and M. Bansal, "Diabetes prediction model using data mining techniques," *Measurement: Sensors*, vol. 25, 100605, 2023.

[17] A. Hennebelle, H. Materwala, and L. Ismail, "HealthEdge: A machine learning-based smart healthcare framework for prediction of type 2 diabetes in an integrated IoT, edge, and cloud computing system," *Procedia Computer Science*, vol. 220, pp. 331–338, 2023.

[18] L. Jiang, Z. Xia, R. Zhu *et al.*, "Diabetes risk prediction model based on community follow-up data using machine learning," *Preventive Medicine Reports*, vol. 35, 102358, 2023.

[19] Z. Zhang, Y. Lu, M. Ye *et al.*, "A novel evolutionary ensemble prediction model using harmony search and stacking for diabetes diagnosis," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 1, 101873, 2024.

[20] A. Hennebelle, L. Ismail, H. Materwala, J. Al Kaabi, P. Ranjan, and R. Janardhanan, "Secure and privacy-preserving automated machine learning operations into end-to-end integrated IoT-edge-artificial intelligence-blockchain monitoring system for diabetes

mellitus prediction," *Computational and Structural Biotechnology Journal*, vol. 23, pp. 212–233, 2024.

[21] G. Prabhakar, V. R. Chintala, T. Reddy, and T. Ruchitha, "User-cloud-based ensemble framework for type-2 diabetes prediction with diet plan suggestion," *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 7, 100423, 2024.

[22] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2623–2631.

[23] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Information Sciences*, vol. 513, pp. 429–441, 2020.

[24] C. Vairetti, J. L. Assadi, and S. Maldonado, "Efficient hybrid oversampling and intelligent undersampling for imbalanced big data classification," *Expert Systems with Applications*, vol. 246, 123149, 2024.

[25] Ö. İnik, "CNN hyper-parameter optimization for environmental sound classification," *Applied Acoustics*, vol. 202, 109168, 2023.

[26] W.-Y. Lee, S.-M. Park, and K.-B. Sim, "Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm," *Optik*, vol. 172, pp. 359–367, 2018.

[27] M. A. K. Raiaan *et al.*, "A systematic review of hyperparameter optimization techniques in convolutional neural networks," *Decision Analytics Journal*, vol. 11, 100470, 2024.

[28] L. Shen *et al.*, "Evolving support vector machines using fruit fly optimization for medical data classification," *Knowledge-Based Systems*, vol. 96, pp. 61–75, 2016.

[29] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[30] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[31] J. W. Lee, J. B. Lee, M. Park, and S. H. Song, "An extensive comparison of recent classification tools applied to microarray data," *Computational Statistics & Data Analysis*, vol. 48, no. 4, pp. 869–885, 2005.

[32] N. P. Tigga and S. Garg, "Prediction of type 2 diabetes using machine learning classification methods," *Procedia Computer Science*, vol. 167, pp. 706–716, 2020.

[33] V. Chang, J. Bailey, Q. A. Xu, and Z. Sun, "Pima Indians diabetes mellitus classification based on Machine Learning (ML) algorithms," *Neural Computing & Applications*, pp. 1–17, 2022.

[34] S. B. Kotsianti, and D. Kanellopoulos, "Combining bagging, boosting and dagging for classification problems," in *Proc. Knowledge-Based Intelligent Information and Engineering Systems: 11th International Conference, KES 2007, XVII Italian Workshop on Neural Networks,* Vietri sul Mare, Italy, 2007, pp. 493–500.

[35] S. Kumari, D. Kumar, and M. Mittal, "An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier," *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 40–46, 2021.

[36] Pima Indians Diabetes Database. [Online]. Available: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[37] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus," in *Proc. the Annual Symposium on Computer Application in Medical Care*, American Medical Informatics Association, 1988, pp. 261–265.

[38] M. Mulyanto, J.-S. Leu, M. Faisal, and W. Yunanto, "Weight embedding autoencoder as feature representation learning in an intrusion detection systems," *Computers and Electrical Engineering*, vol. 111, 108949, 2023.

[39] M. A. Rahim, M. A. Hossain, M. N. Hossain, J. Shin, and K. S. Yun, "Stacked ensemble-based type-2 diabetes prediction using machine learning techniques," *Annals of Emerging Technologies in Computing (AETiC)*, vol. 7, no. 1, pp. 30–39, 2023.

[40] Y. Ali, F. Hussain, and M. M. Haque, "Advances, challenges, and future research needs in machine learning-based crash prediction models: A systematic review," *Accident Analysis & Prevention*, vol. 194, 107378, 2024.

[41] S. Zhang, A. Khattak, C. M. Matara, A. Hussain, and A. Farooq, "Hybrid feature selection-based machine learning classification system for the prediction of injury severity in single and multiple-vehicle accidents," *PLoS One*, vol. 17, no. 2, e0262941, 2022.