International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. , No. (Mon-20..)

http://dx.doi.org/10.12785/ijcds/XXXXXX

Comparison of Classification of Different Machine Learning Algorithms in the Diagnosis and Detect of Diabetes

Zainab N. Nemer[1], Sabreen Fawzi Raheem[2] and Maytham Alabbas[3]

[1]*College of Computer Science and Information Technology, University of Basrah, Iraq*
[2]*Basra Technical Institute, Southern Technical University, Basrah, Iraq*
[3]*College of Computer Science and Information Technology, University of Basrah, Iraq*

Received Mon. 20, Revised Mon. 20, Accepted Mon. 20, Published Mon. 20

Abstract: Diabetes, characterized by elevated blood glucose levels, can be detected through various instruments that analyze blood samples. Untreated diabetes can lead to serious complications, including heart attacks and kidney failure. Consequently, detecting and evaluating gestational diabetes requires more robust research and advanced learning models. The information system for detecting diabetes in this study is based on machine learning (ML) algorithms. Various machine learning techniques were explored, including Decision Trees (DT), Random Forest (RF), Logistic Regression (LR), Extreme Gradient Boosting (XGBoost), and K-Nearest Neighbors (KNN). The data was collected from the Iraqi society, primarily from the laboratory of Medical City Hospital and the Specializes Centre for Endocrinology and Diabetes-Al-Kindy Teaching Hospital. GridSearchCV, an effective tool for hyperparameter adjustment in machine learning, was utilized. It systematically explores various parameter value combinations, and cross-validating to identify the optimal parameter configuration. The features considered were patient ID, patient number, blood sugar level, age, gender, creatinine ratio, body mass index, urea, cholesterol, fasting lipid profile, and the patient's diabetes diagnosis (Diabetic, Non-Diabetic, or Predict-Diabetic). Research was conducted to enhance the prediction index using the Recursive Feature Elimination approach. The performance of all five algorithms was evaluated on various measures, including Precision, Accuracy, F-score, Recall, Cohen Kappa, and AUC. According to the performance statistics, XGBoost achieved the highest accuracy (98.5%), followed by RF (97.96%), KNN (91.2%), DT (97.99%), and LR (89.9%). The findings of this study can inform a program for screening potential diabetes patients.

Keywords: Machine Learning, Random Forest, XGBoost, Diabetes, GridSearchCV, KNN, Decision Tree.

1. INTRODUCTION

Diabetes is a chronic and incurable condition. Due to this disease, the enzyme responsible for transporting sugar into cells decreases, leading to elevated blood glucose levels and serious complications such as stroke, lung disease, vision loss, kidney failure, and death. Patients with diabetes often experience weight loss, blurred vision, infections, and frequent urination [1]. In 2019, 1.5 million diabetes-related deaths occurred, with 48% of these fatalities happening in adults under the age of 70 [2]. Machine learning (ML) techniques have shown promising results in addressing this health challenge.

The significance of a reference can be derived from summarizing and comparing the results of various classifiers when applied to their classification tasks. This study evaluates and compares five classic ML classifiers—Gaussian Mixture Models (GMM), Random Forest (RF), Support Vector Machines (SVM), Extreme Gradient Boosting (XGBoost), and Naive Bayes (NB)—to demonstrate how they perform [3]. The performance of the XGBoost model on the RNA-seq and GEO datasets, as well as a comparison of the findings with other models, are presented. Studies have shown that the XGBoost model outperforms the current D-GEX algorithm, Linear Regression (LnR), and KNN approaches in terms of overall error. The XGBoost method demonstrates superiority over existing models and significantly expands the toolkit for predicting gene expression values [4].

A multi-model combination forecasting approach based on XGBoost is proposed. This approach creates a novel time series as the set of features through the outputs of the forecasting model's fitting and forecasting. Researchers can choose from several models for feature reconstruction when utilizing the results of the forecasting model's predictions. The effectiveness of the features can subsequently be evaluated using the score of the reconstructed features in the

E-mail address: zainab.nemer@uobasrah.edu.iq, sabreen.fawzi@stu.edu.iq, ma@uobasrah.edu.iq

training set of the subsequent forecasting model [5].

The creation of XGBoost led to an effective implementable tree-boosting system that produces new results presumably in different fields. For instance, a novel sparsity-aware algorithm was proposed for how to handle sparse data, and a theoretical weighted quantile sketch for approximate learning was provided. An innovative method that incorporates data sparsity was proposed and a quantile sketch based on weighted approximation of the tree was proposed for the construction of approximation trees. To realize a tree-boosting system that is simple to scale and facilitate analysis of the cache usage patterns, data layout, and data sharing, XGBoost was able to beat the scalability of billions of samples to utilize fewer resources than the current systems. These were made possible by all those findings [6].

The classification of data of compact polarimetric (CP) RISAT-1 cFRS mode has been performed. The Mumbai area was investigated using the methods of Artificial Neural Networks (ANN) and XGBoost. Afterwards, the Raney decomposition approach was employed to split the R, G, and B channels following the preprocessing step. Hyperparameter tuning of ANN was also done to get the best classification parameters. In contrast, the two algorithms showed the same efficiency in terms of accuracy over the performed weather classification. However, the performance of the XGBoost classifier was somewhat less accurate specifically, 1% accuracy difference on both train and test sets but this is conceptually negligible. As the attachment of the ANN approach was tuned, computation took a longer time compared to the XGBoost algorithm which did not require tuning and still performed efficiently [7].

The XGBoost algorithm was utilized for the prediction of risk assessments in corporate finance. A data preparation technique was successfully applied to preprocess and classify the enterprise revenue information source. Subsequently, the XGBoost technique was used to assess the risk associated with the enterprise's financial data. Finally, a set of models for assessing enterprise risk in finance was established. The study's findings demonstrate the high reliability of the XGBoost model in forecasting enterprises' financial risk assessments, with an error rate of less than 3%. Most of the prediction errors can be attributed to the profit and loss of the business's income status, which amounted to only 2.68%. The error rate was deemed sufficiently trustworthy for corporate use, with a minimum error of 0.56% [8]. Due to advancements in data collection and efficient computation, which allowed it to address real-world challenges in terms of cost of usage, machine learning has gained popularity [9].

Classification was done using various algorithms and methods. Some of these approaches have demonstrated greater proficiency in specific tasks. Classical methods, such as KNN, NB, SVM, the Recchio method, and ANN, were employed. Each of these methods possesses its unique advantages. For example, KNN is simpler to implement, while SVM, although more sophisticated, is more resilient and adaptable[10].

The study proposed classification models as an electronic diagnostic system. These models were evaluated to determine the presence of a positive diagnosis of diabetes mellitus, based solely on eight attributes. Three ML classifiers—J48 DT, RF, and NB—were employed to train the models [11].

The incorporation of three convolutional neural networks (CNNs)—Inception V3, ResNet50, and DenseNet121—into a meta-learning framework enabled both generalization and achieved an accuracy rate of 90%. This was particularly evident in the detection of cancerous tumors. The research demonstrated the potential of meta-learning and ensemble methods to enhance the accuracy and efficiency of breast cancer diagnoses. Applying the same concept to medical imaging datasets could lead to advancements in the treatment of several types of cancer or illnesses. The study's dataset was limited by both a small sample size and a lack of diversity in the cases [12].

ML methods were harnessed for the diagnosis of diabetes, where four models came into play: ANN, SVM, NB, and LR. This study aimed to find the ideal method and evaluate muscle strength for detecting the Chinese population with previously undiagnosed diabetes. The ANN algorithm showed better performance in identifying new diabetes cases compared to SVM, NB, and LR. Such difference observed could be explained by the different research scopes (diabetes compared with undiagnosed diabetes) and target areas (developed countries as opposed to undeveloped regions) [13].

The study assessed the efficiency of using machine learning algorithms to predict type 2 diabetes mellitus (T2DM) in the early stages. Classification model: This is a dataset containing patients with T2DM and normal controls used to build the classification model. To select important features, they used some new methods in the model. The model was trained and tested with LR, KNN, DT, RF, and SVM algorithms based on machine learning. The results showed that in 98% of T2DM diagnoses; the RF model had an excellent accuracy. Formal validation in larger and more diverse populations through research is needed to establish the strength of the model, establishing generalizability across settings. Further research should examine the extent to which characteristics such as body size, height, and body mass index (BMI) on the early detection of diabetes. This might potentially enhance the precision of the diagnostic model [14].

A decentralized privacy-aware learning technique was proposed for the accurate prediction of melanoma skin cancer. Researchers examined federated learning using the skin cancer database. A hybrid CNN and SVM approach

was employed to predict and categorize skin malignant melanoma. The model utilized the ABCD rule, which rates skin lesions according to specific standards, to assist the SVM classification technique. The performance of SVM was superior. The outcomes revealed an accuracy rate of 92% for the Async-Fed-CNN-SVM model[15].

The research has great findings on the identification and categorization of diabetes through the analysis and review of state-of-the-art ML and deep learning (DL) techniques. One of the limitations that must be pointed out is the lack of sufficient data relating to diabetes. Pen database measurement includes lab-based and invasive test measurements. In particular, the accuracy of diabetes detection predictors, which include non-lab tests and non-invasive measurements should be further examined to determine possible ways of decreasing the cost and time invested in the detection and treatment of diabetes. For this, it is crucial to use a dataset of higher quality which includes more recorded features and samples and thus the absence or presence of no abnormal values [16].

To predict diseases, the research revealed the need to incorporate the ML model. The model comprises of LnR, LR, KNN, NB, RF, SVM, and DT ML methods according to the nomenclature used here. The investigation involved comparing the aforementioned model across two datasets. The pre-processing of data was done before the data was presented to the ML model for assessment. When using the diabetes dataset, the RF algorithm provided the best performance which was at 97% accuracy [17].

The focus of this research is therefore to design and analyze an information system for diagnosing diabetes employing diverse ML algorithms. Therefore, the application is designed to improve the process of diagnosis of diabetes, especially gestational diabetes, with the help of artificial intelligence and machine learning models. The purpose of this study is to improve the accuracy of the ML algorithms under study, using Python's GridSearchCV hyperparameter tuning technique as well as the Recursive Feature Elimination (RFE) feature selecting technique. The result of this paper is to employ and evaluate five different forms of ML techniques: DT, RF, LR, XGBoost, and KNN to predict diabetes. The dataset utilized was obtained from the Iraqi society, specifically from two medical institutes, providing local data for the worldwide field of diabetes detection study. An in-depth evaluation of several performance measures, such as Precision, Accuracy, F-Score, Recall, Cohen Kappa, and AUC, was presented for each model, offering valuable insights into their efficacy in diagnosing diabetes. The study demonstrated that XGBoost achieved the maximum level of accuracy (98.5%), establishing it as the most effective model for detecting diabetes in the specific dataset employed. This approach improved the prediction index through the optimization of the feature set using RFE, enhancing model performance. These optimizations can provide valuable insights for future screening programs targeting diabetes patients. The present study contributes to the field by introducing a resilient ML methodology aimed at enhancing the early identification and diagnosis of diabetes. This can alleviate severe consequences such as hypertension and renal failure.

The preceding section provides an introduction and literature review of classification algorithms relevant to diabetes prediction. The subsequent sections of the paper are structured as follows: Section 2 outlines the entire procedures. Section 3 introduces the present work and algorithm for comparative analysis. Section 4 details the performance indicators employed to assess the models. Section 5 provides experimental data and discussion. Section 6 presents forthcoming research and resulting conclusions.

2. METHODOLOGY

The current work methodology is structured into the following phases:

A. Data Preprocessing

ML algorithms can effectively map the nonlinear association between patient data and output diagnoses. The Iraqi Patient Dataset for Diabetes, obtained from the Specialized Centre for Endocrinology and Diabetes-Al-Kindy Teaching Hospital in Iraq, comprises 1000 samples, including 565 males and 435 females between the ages of 20 and 79. This information exhibits a complex relationship but can also have a significant impact and aid in diagnosis. When the diagnosis is performed accurately, it can assist both the doctor and the patient in avoiding the disease. A series of procedures were implemented to preprocess the dataset in this work, which facilitated the attainment of satisfactory findings. These procedural stages include:

- Data filtering is the process of eliminating or deleting variables or observations from a dataset, which aids in concentrating the analysis on pertinent data. It involves extracting data based on predetermined criteria and is used for filtering data, erasing undesirable values, or extracting data that meets specific criteria. Improving data quality before analyzing or using it in machine learning models is crucial. Patients who meet the inclusion criteria are adults between 25 and 65 and have a body mass index that aligns with normal, overweight, or obese (BMI greater than or equal to 18.5). Consequently, we do not consider patients who are either young or elderly, nor do we consider patients who have a body mass index that is equivalent to being underweight.

- Visualization and descriptive statistical analysis involve using illustrations and summary statistics to understand the dataset's main characteristics, thereby allowing the ability to identify patterns, trends, and potential outliers.

- Outlier management and detection: The technique of outlier management and identification was employed

to discover any data flaws, specifically biomarker values that are not attainable by humans. Outliers were detected using two methods: isolation forest and DBSCAN. Observations identified as outliers using both approaches were finally determined to be outliers.

- Error detection: Upon examining the major statistics of the numeric variables, it is evident that some minimum and maximum values are significantly higher or lower than those observed in a healthy population. Although these values are physiologically conceivable, they are more likely to occur in a cardiometabolically impaired sample such as this one.
  A similar phenomenon occurs with the discovered outlier findings, which are nonetheless within the human possible range. Therefore, the observed outliers are not considered errors and will not be eliminated.

- Errors were identified through the presence of duplicates. Duplicated values were deleted as it is improbable that multiple patients would have identical features and biomarker levels, and duplicated data provides no additional insights to the models.

- Missing data management includes addressing missing values in the dataset by either imputing values or removing incomplete observations.

- Data reduction is achieved through Principal Component Analysis (PCA), which simplifies analysis by reducing the dimensionality of the dataset. PCA could reveal that patient age, BMI, and LDL have a significant effect on the first principal component, which covers general health factors. The second principal component, which focuses on cardiovascular health, maybe more closely related to VLDL and other issues.

- Data reduction and feature relevance determination involve improving the dataset by identifying notable features of the patient and eliminating redundancy, leading to the construction of more efficient models. Feature selection reduces the dimensionality of the data. The features are selected, and their relevance is determined using the ReliefF method, which is a feature selection algorithm particularly well-suited for datasets containing both categorical and numerical attributes. In multi-class scenarios, it identifies the nearest neighbors from each class to a randomly selected sample, x. The algorithm then assigns higher weights to features that effectively discriminate between within-class and between-class instances. This process is iterated across all features to determine the optimal feature weighting vector.

- Data scaling, a common preprocessing technique, involves standardizing or normalizing the numeric values of a dataset. This step is essential to ensure that variables with disparate units or scales have an equal impact on the model's training and analysis, preventing any single variable from dominating the results [18], [19].

B. Machine Learning Techniques

The present work utilized the following five models as classifiers:

1) XGBoost Classifier

XGBoost is an ML algorithm that uses DTs for data structure and gradient boosting for learning. Boosting is a type of technique in Supervised learning, it uses the base learners and builds a sequence of DTs with all the trees more directed towards minimizing the overall error rates of the previous model. Such, an iterative process results in constructing a highly performing ensemble model for further dealing with intricate patterns in the data. XGBoost offers the user numerous hyperparameters which enable the solution to be further tuned depending on the specific type of task at hand [20].

This strategy shows a factor of five increase in speed when compared with conventional ML and DL models due to parallel, distributed, out-of-core, and cache-aware computing. It is also very versatile and can effectively handle large amounts of data. It is specially developed to solve issues that are associated with insufficient data, missing values, overwhelmingly amplitudinous zeros, and aspects of feature transformation. This is in agreement with the rule of ensemble methodology where models are added in a step-wise manner until marginal gains in performance can no longer be observed[21].

Gradient boosting is an enhanced learning process that focuses on constructing numerous DTs in a step-by-step process while it attempts to minimize the loss function. This research work employs gradient boosting known as G-Boost in this context. The DTs which are the building blocks of G-Boost have their decision-making processes thus resulting in interpretability. So, ensemble models, which combine the results of several models, are good at detecting complicated relations between the features and can be applied to large-scale data too [22], [23].

2) Logistic Regression (LR) Classifier

This model employs a single multinomial LR model for class prediction. LR determines class boundaries and calculates probabilities based on distance from these boundaries. As the dataset grows, the predicted probabilities tend to converge towards 0 and 1, providing more nuanced predictions than binary classifiers. However, these probabilistic outputs can occasionally be misleading. Like Ordinary Least Squares (OLS) regression, LR is a predictive modeling technique [24]. Unlike OLS, LR produces binary outcomes, making it suitable for classification tasks. LR is a widely used tool in the field of statistics for analyzing non-continuous data and incorporates linear interpolation [25], [26].

3) K-Nearest Neighbor (KNN) Classifier

KNN can be determined using Euclidean distance, a widely used metric. While other distance measures exist, Euclidean distance often provides a favorable balance of simplicity, effectiveness, and computational efficiency [27], [28].

KNN can be viewed as a form of analogical learning, where a test instance is classified based on its similarity to nearby training instances. The class of the nearest neighbors is used to determine the classification. KNN often considers multiple neighbors, hence the name 'K-Nearest Neighbor,' where 'K' represents the number of neighbors used in the classification. KNN is often referred to as a 'lazy learner' because it simply stores training data and performs generalization only when presented with a test instance [29].

4) Decision Tree (DT) Classifier

DTs represent each instance as a collection of attributes (features) associated with a single class label. The tree's leaf nodes indicate the predicted class. The DT algorithm constructs a hierarchical structure of features to effectively predict class labels using a training set with labeled instances. Each instance is mapped to a point in the feature space, and features serve as decision points. The DT partitions the feature space into regions associated with different classes, enabling the prediction of class labels for new instances based on their attribute values. A classification tree is a hierarchical structure of nodes. The topmost node is called the root node, and the subsequent nodes are called internal nodes. Nodes in a network or graph are referred to as vertices. Internal nodes in a DT represent tests used to categorize instances. Each test outcome is represented by a child node. For categorical attributes, the potential outcomes are discrete values (e.g., A = d1, d2, ..., dh). For continuous attributes, outcomes are binary (e.g., A ≤ t or A > t). Leaf nodes at the bottom of the tree determine the predicted class. DTs offer interpretability and are suitable for various applications due to their lack of assumption regarding attribute independence. Previous studies have demonstrated the effectiveness of DTs in traffic management, healthcare, marketing, gene identification, and medical diagnostics[30], [31], [32].

5) Random Forest (RF) Classifier

RF is an ensemble model that constructs DTs based on random subsets of data. Each tree casts a vote in the final prediction, resulting in a robust ensemble. RF is well-suited for handling sparse data, missing values, noise, and errors. LR is often used for text categorization. LR models the relationship between dependent and independent variables and uses statistical estimation to predict the most likely class[33], [34].

3. Current work

Initially, correlation was used as a pre-processing step because it effectively imputes missing values in the dataset. Additionally, correlation can be employed to estimate the causal relationships within the available data.

The dataset is split into a training set (70%) and a testing set (30%). The training set is utilized to build the model, and the testing set is employed for its evaluation. Preprocessing is critical prior to training any model on a dataset. For instance, in training ANNs with gradient descent, standardization is vital to avoid overshooting the minimum and to guarantee convergence. It also streamlines data processing and analysis. This approach enables companies to make better-informed decisions and acquire significant insights by facilitating data comparison and analysis.

Standardization of data may be very critical in assisting the organization to avoid the implications of making decisions based on faulty or incomplete information. In essence, given that organizations have an assurance that their data will be accurate and complete, there is a likelihood that enhancing key data-influenced decisions may lead to higher profitability. The most general standardization method defines the standardized value as a function, containing a z-score as the result, calculated from the mean and standard deviation of a dataset. It specifies how many standard deviations the given value falls from the mean. In a perfectly normal distribution, the sum of all z-scores equals zero. A negative z-score indicates a value that falls below the mean, while a positive z-score indicates a value that is above the mean.

Next, the classification method identifies and prioritizes the key features using XGBoost, DT, RF, SVM, and KNN. Figure 1 presents the outline of the proposed work, while its detailed steps are explained below:

- Import required libraries by going to import essential libraries.
- Download the dataset by using the function load_data ('path_to_dataset'). X = data.Features y equals desired output
- Divide the dataset into testing and training sets:
  - train_test_split(X,y,test_size=0.2,random_state=42)
  - X_train,X_test,y_train,y_test=train_test_split (X,y,test_size=0.2, random_state=42)
- Set up the models
  - Decision Tree () as dt_model;
  - Random Forest () as rf_model;
  - KNeighbors() as knn_model;
  - XGBoost() as xgb_model;
  - and Logistic Regression () as lr_model
- Train models
  - dt_model. train (X_train, y_train)
  - rf_model. train (X_train, y_train)
  - knn_model. train (X_train, y_train)
  - xgb_model. train (X_train, y_train)
  - lr_model. train (X_train, y_train)

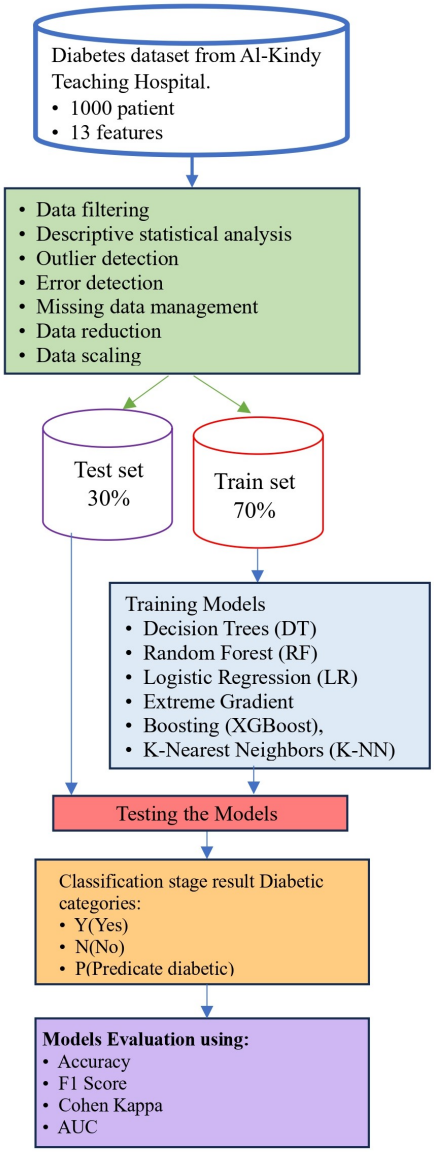


Figure 1. Proposed system methodology

- Prediction
  - Model=dt_predictions.prediction (X_test)
  - rf_predictions=rf_model.(X_test)
  - predict(knn_predictions)=knn_model.anticipate (X_test)
  - xgb_predictions=xgb_model.(X_test)
  - predict(lr_predictions)=lr_model.Forecast (X_test)
- Evaluate models using the metrices

4. MODEL EVALUATION METRICS

A range of performance metrics, such as accuracy, precision, the integral of the receiver operating characteristic curve (ROC), and detection rates, are utilized to assess the proposed model. These metrics include:

1) Accuracy
Accuracy, as defined in Equation 1, refers to the fraction of correctly detected observations [14], determining the effectiveness of the classification algorithm under test.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (1)$$

2) F-Measure (F1-Score)
The F-score balances precision and recall. It, ranging from 0 (worst score) to 1 (best score), is calculated using Equation 2.

$$F1 - score = 2(precisionrecall)/(precision + recall) \quad (2)$$

Where, precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall is the ratio of correctly predicted positive observations to all actual positive observations. The equations for precision and recall are given by Equations 3 and 4, respectively formulates precision.

$$Precision = TP/(TP + FP) \quad (3)$$

$$Recall = TP/(TP + FN) \quad (4)$$

TP, FP, and FN denote the number of true positives, false positives, and false negatives, respectively, for a given category.

3) Cohen Kappa
Cohen's interpretation of Kappa coefficients is as follows: values below 0 suggest inconsistency, 0.01 to 0.20 imply slight to no agreement, 0.21 to 0.40 denote fair agreement, 0.41 to 0.60 reflect moderate agreement, 0.61 to 0.80 represent substantial agreement, and 0.81 to 1.00 signify almost perfect agreement[35].

4) Area Under the Curve (AUC)
AUC represents the classifier's ability to detect classes. A perfect classification is indicated by an AUC of 1, while an AUC of 0.5 signifies random selection. Previous studies have demonstrated the insensitivity of AUC to imbalanced datasets [36], [37], [38].

5. EXPERIMENTAL RESULTS

A. Dataset Description

The dataset was compiled from Iraqi society, specifically from the laboratory of Medical City Hospital and the Specialists Centre for Endocrinology and Diabetes-Al-Kindy

Teaching Hospital. Patient data was collected and extracted, subsequently being entered into a database to establish the diabetes dataset. This dataset encompasses medical information, laboratory analysis, and other pertinent details of 1000 patients. The initial data input into the system included: patient ID, patient number, blood sugar level, age, gender, creatinine ratio (Cr), body mass index (BMI), urea, cholesterol (Chol), fasting lipid profile (including total, LDL, VLDL, triglycerides (TG), and HDL cholesterol), HBA1C, and class (corresponding to the patient's diabetes diagnosis as Diabetic, Non-Diabetic, or Predict-Diabetic). A comprehensive list of dataset features is provided in Table I. To evaluate model performance, a train-test split was implemented, dividing the dataset into two subsets: 70% for model training and 30% for testing.

TABLE I. Dataset description (1000 patients ,13 features)

Dataset features	Features description
N0_patient	patient number
Sex	F or M
Age	Patient age
Urea	the amount of urea nitrogen in blood
Cr	creatinine ratio
HbAlc	blood test that is used to diagnose type 2 diabetes
Chol	Cholesterol
TG	triglyceride
HDL	High_density lipoprotein
LDL	Low_density lipoprotein
VLDL	very low_density lipoprotein
BMI	body mass index
Class	Diabetic categories: Y(Yes), N(No), and P (predicate diabetic)

Figure 2 presents a correlation plot depicting the interrelationships among various patient metrics within the dataset. The Pearson correlation coefficient was employed to quantify the relative strength of the association between each pair of patient data variables. Correlation values range from -1 to 1. A strong positive correlation signifies that both variables demonstrate a concurrent movement in the same direction. Conversely, a strong negative correlation indicates that the variables exhibit opposite movements. A value of zero denotes the absence of correlation between the two variables. Color intensity signifies the strength of the relationship, with deeper hues (blue) indicating a robust association and lighter hues (yellow to green) suggesting a negligible or absent association. For example, the HbA1c and BMI variables exhibit a moderate positive correlation with a coefficient of 0.41. The correlation coefficient between Chol and Diagnosis is 0.54, implying a moderately favorable relationship. Quantitative factors such as No_Patient and LDL demonstrate minimal or negligible correlation with other variables.

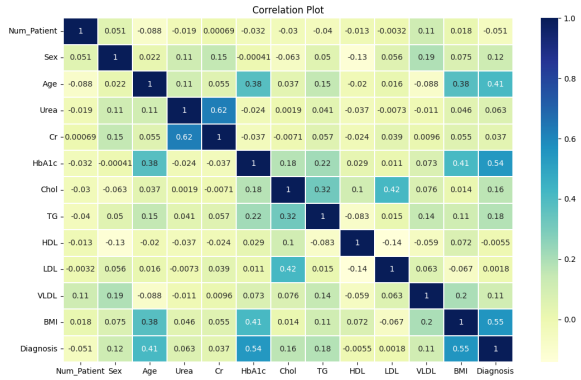


Figure 2. Correlation plot.

B. *Parameters Setting*

Hyperparameters are the parameters that dictate the behavior of an ML model. These parameters are not learned during training; instead, they must be established beforehand. The hyperparameter optimization process involves identifying the optimal values for these parameters, which is a vital step in developing an ML model. Hyperparameter optimization can be done in various ways, with grid search and randomized search being the most popular methods.

To optimize the model's hyperparameters, a grid search method was employed. This involves constructing a list of potential values for each hyperparameter and subsequently training the model with every possible combination of these values. For instance, if the goal were to optimize the hyperparameters of a KNN model, a list of values for the parameter "neighbors" might be specified, such as (3, 5, 7, 9). The grid search algorithm would then utilize these values to train multiple models, evaluating their performance. The hyperparameter values that yield the best-performing model are ultimately selected.

Cross-validation (CV) was also employed by the models. In CV, the training data was partitioned into multiple subsets, and the model was iteratively trained with a different subset used for validation each time. This approach helped to mitigate overfitting and provide a more accurate assessment of the model's performance. GridSearchCV was utilized to optimize LR parameters, systematically exploring a grid of values for each hyperparameter to identify the optimal combination. This optimization aimed to balance model complexity and generalization.

Some variables that influence the structure and behavior of a DT include the maximum depth and the minimum number of samples required for a leaf node. These parameters can be adjusted based on the dataset and the performance of the model.

To optimize RF parameters, consider adjusting the minimum number of samples required to split a node, the minimum number of samples required at a leaf node,

TABLE II. The hyperparameters for knn, logistic regression, decision trees, random forest, and xgboost

Model	Hyperparameter	Description	Example Values
KNN	n_neighbors	The number of neighbors that will be used	3, 5, 7, 9
	weights	The prediction's weight function	'uniform', 'distance'
	algorithm	The algorithm that calculates the closest neighbors	'auto', '$ball_tree$', 'kd_tree', 'brute'
Logistic Regression	C	the opposite of the regularization strength. Greater regularization is indicated by smaller values.	0.1, 1, 10
	solver	The optimization problem's algorithm to apply	'newtoncg', 'lbfgs', 'liblinear', 'sag', 'saga'
	multi_class	For each label, a binary problem is appropriate if the selected option is "ovr."	'auto', 'ovr', 'multinomial'
Decision Trees	Max_depth	Tree's maximum depth.	5, 10, 15
	min_samples_split	Minimum sample count needed to divide an internal node.	10,5,2
	min_samples_leaf	A leaf node must have a minimum of samples.	1,2,4
Random Forest	n_estimators	The count of trees within the forest.	100,200,500
	Max_depth	Maximum tree depth	5, 10, 15
	min_samples_split	Minimum sample count needed to divide an internal node.	2, 5, 10
	min_samples_leaf	A leaf node must have a minimum amount of samples.	1,2,4
	bootstrap	Whether constructing trees requires the use of bootstrap samples	True, False
XGBoost	learning_rate (eta)	Step size shrinking is used to prevent overfitting.	0.01, 0.1, 0.2
	n_estimators	How many rounds of boosting or treebuilding	100, 200, 500
	max_depth	A tree's maximum depth. A higher value adds complexity to the model.	3, 5, 7
	subsample	The proportion of samples that will be utilized for training the individual base learners.	0.5, 0.7, 1
	colsample_bytree	Proportion of characteristics to be utilized for training the individual foundational learners.	0.5, 0.7, 1
	gamma	The minimum reduction in loss necessary to create an additional partition on a leaf node of the tree.	0, 0.1, 0.2
	random_state	Regulates the level of unpredictability in the process of creating samples for developing trees.	0, 42, 100

and the maximum tree depth. Through experimentation, determine the parameter combination that maximizes the model's impact. Bootstrap sampling can assess the influence of different parameter values on the model's generalization. Key XGBoost parameters include the learning rate (eta). A lower learning rate generally results in a more robust model but requires huge trees. The number of trees represents the total number of boosting rounds or trees to constructed. The following parameters were employed to achieve the results:

The "learning rate" parameter was set to mitigate overfitting issues by adjusting the step size for feature weight updates. The "max_depth" parameter determined the maximum depth of each decision tree in the ensemble, with higher values leading to more complex models. The number of boosting rounds or trees was also specified.

The "random state" parameter, sometimes referred to as "seed," is a learning parameter that randomizes the dataset into k sections.

The aforementioned tree booster parameters were used to calculate the results presented below. While numerous settings can be configured, the model primarily determines these settings. However, parameters can be defined according to the desired model behavior.

The "Tree Maximum Depth" parameter specifies the maximum depth for each decision tree in the ensemble,

while the "Minimum Child Weight" parameter sets the minimum sum of instance weights required for a child node. This parameter can be used to manage overfitting. The parameters adjusted in the five models to achieve satisfactory results are presented in Table II.

Table II presents an overview of the hyperparameters for five ML models. Key hyperparameters for KNN include the number of neighbors, the prediction weighting method, and the distance metric used to identify nearest neighbors. In LR, hyperparameters primarily address regularization strength, optimization algorithm, and multiclass classification handling. DTs are characterized by factors such as maximum tree depth, minimum samples required for node splitting, and minimum samples required at leaf nodes. RF shares similar hyperparameters with DTs but also includes the number of trees in the forest and bootstrap sampling options. As a more sophisticated model, XGBoost incorporates hyperparameters like learning rate, number of boosting rounds, tree depth, gamma, and subsample size to manage overfitting. Each hyperparameter plays a crucial role in model optimization and requires careful fine-tuning to achieve optimal results.

C. Results

The five models were trained and evaluated using Google Colab, a cloud-based GPU environment. Python 3.10 was employed to implement the proposed techniques on a Windows 10 system equipped with an Intel Core i7 CPU at 7 GHz and 8.00 GB of total RAM. Before training, all data within the dataset were normalized. Subsequently, each dataset was partitioned into training and testing sets, with 30% of the data allocated for testing and the remaining 70% for training.

This study employed four widely recognized metrics: accuracy, F1 score, Cohen's kappa, and ROC AUC using the one-versus-rest approach. Table 3 offers a comparative analysis of the performance of each model. As shown in Table 3, XGBoost consistently outperforms the others in accuracy, F1 score, Cohen's kappa, and AUC. The analysis includes the mean and standard deviation (SD) for the five models that were tested.

1) The mean AUC values, indicative of perfect discriminative performance across models, ranged from 0.937953 to 0.997889. The standard deviation demonstrated low-performance variability within each model.
2) The mean Kappa values, indicating varying levels of agreement beyond chance, ranged from 0.631183 to 0.953714. The standard deviation values illustrated variations in the agreement among models.
3) The mean F1 score values ranged from 0.626668 to 0.971192, suggesting the models' effectiveness in balancing precision and recall. The standard deviation values indicated variability in the harmonic mean of recall and precision.

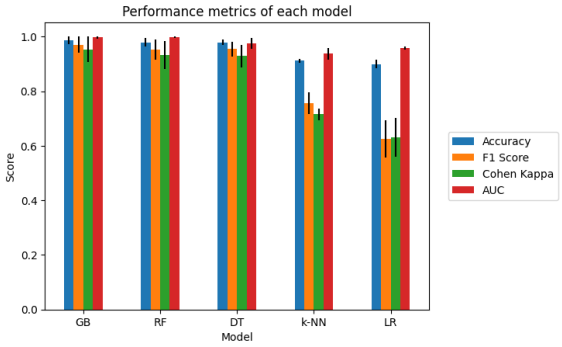


Figure 3. Performance metrics

4) The mean accuracy values ranged from 0.985925 to 0.899828, representing the percentage of instances accurately classified. The standard deviation values reflected variability in classification accuracy among models. By comparing and contrasting the distributions of various performance metrics among models, these findings shed light on the consistency and reliability of these measurements. This result aligns with our state-of-the-art study, confirming that XGBoost outperforms DT, KNN, RF, and LR for diabetes detection. As depicted in Figure 3, among the proposed models, including interaction terms, XGBoost achieved the best performance with 98.5% accuracy, 97.1% F1 Score, 95.4% Cohen Kappa, and a 99.7% AUC

While the proposed ML approach for diabetic condition prediction demonstrates promising findings and potential, it is important to acknowledge several limitations:

1) Dataset Availability and Quality: The accuracy and performance of any machine learning model are significantly influenced by the quality and availability of the training and testing dataset. This study utilized multiple datasets to achieve a high level of representation, quality, and variance. However, limitations in dataset availability may impact the generalizability of the proposed models.
2) Algorithm Selection: This study employed various ML classification algorithms to identify the most optimal option for diabetes prediction. However, the selection of algorithms is inherently subjective and can influence the results. It is possible that alternative algorithms not considered in this work could potentially achieve superior accuracy or offer different trade-offs between performance metrics. Therefore, the choice of ML algorithms warrants careful consideration and evaluation in future studies.

6. CONCLUSION

The research has been dedicated to the development of computerized decision-making tools to assist healthcare providers in various aspects of patient care. Developers

TABLE III. The effectiveness of the five machine learning modeling methods for classifying diabetics

Models	Accuracy		F1 Score		Cohen Kappa		AUC	
	sd	mean	sd	mean	sd	mean	sd	mean
GB	**0.985925**	**1.1703E 16**	**0.971192**	**0**	**0.953714**	**0**	**0.99761**	**1.1703E-16**
RF	0.979652	0.001473	0.952213	0.00454	0.932438	0.005095	0.997889	0.000147
DT	0.97998	0.000663	0.955136	0.001786	0.930079	0.002323	0.978092	0.00265
kNN	0.912365	0	0.756725	1.17028E-16	0.715168	1.17028E-16	0.937953	0
LR	0.899828	0	1.17028E-16	0.626668	0.631183	0	0.958123	0

of these systems often assert that such tools enhance the accuracy of healthcare diagnosis and lead to improved patient outcomes.

In this study, multiple ML techniques, including DTs, RF, LR, XGBoost, and KNN, have been explored. The distinguishing feature of this work lies in the in-depth research conducted to identify results that guide decision-making and determine efficiency.

The performance results have demonstrated a high accuracy value, with XGBoost achieving a particularly impressive 98.5% accuracy compared to other models used in this research. Future refinements to the model could involve incorporating more data from diverse sources and considering alternative ML techniques.

ML models offer numerous practical applications that can enhance the quality of patient care, improve diagnostic accuracy, and optimize treatment strategies. These models can analyze vast quantities of patient data, including electronic health records, genetic information, and lifestyle characteristics, to identify individuals at high risk of developing diabetes. Early identification can enable prompt action, potentially preventing the onset of diabetes or mitigating its severity.

ML algorithms can also predict the probability of diabetes-related complications, such as cardiovascular disease, renal failure, and diabetic retinopathy. By identifying individuals at increased risk, healthcare professionals can implement preventive measures and monitor these patients closely, potentially reducing the occurrence and severity of complications. Additionally, algorithms can generate personalized recommendations for insulin dosage, dietary modifications, and exercise routines based on real-time data from continuous glucose monitors and other wearable devices.

ML has the potential to significantly reduce healthcare costs associated with diabetes by improving early detection, tailoring treatment programs, and preventing complications. This includes reducing hospital readmission rates, decreasing the need for emergency care, and mitigating long-term consequences that require expensive therapies.

Future research directions include the development of mobile applications and digital health technologies using ML to empower individuals in managing their diabetes. These systems can provide real-time feedback, educational materials, and personalized recommendations, enhancing patient engagement and self-management capabilities. The dynamic nature of ML models ensures their continued relevance and effectiveness in the evolving field of diabetes management.

While there are notable advantages, it is essential to consider challenges and limitations such as data privacy and security, and the need for interpretability. Safeguarding patient data and ensuring its ethical use is paramount. Additionally, clinicians must understand and trust the recommendations generated by ML algorithms.

The XGBoost, KNN, LR, DT, and RF models can be applied in real-world clinical settings to improve diabetes treatment. The implementation process typically involves several sequential steps: data collection, model training, validation, and patient engagement.

References

[1] A. S. Mosa and Z. N. Nemer, "Covid-19 diagnosis based on chest x-ray using deep convolution neural network and testing the software complexity using halstead metrics and artificial neural network||," *Advances in Mechanics*, vol. 9, no. 3, pp. 780–802, 2021.

[2] H. Q. Jaleel, J. J. Stephan, and S. A. Naji, "Textual dataset classification using supervised machine learning techniques," *Eng. Technol. J*, vol. 40, pp. 527–538, 2022.

[3] H. Tan, "Machine learning algorithm for classification," in *Journal of Physics: Conference Series*, vol. 1994, no. 1. IOP Publishing, 2021, p. 012016, https://doi.org/10.1088/1742-6596/1994/1/012016.

[4] K. K. Chari, M. C. Babu, and S. Kodati, "Classification of diabetes using random forest with feature selection algorithm," *Int. J. Innov. Technol. Explor. Eng*, vol. 9, no. 1, pp. 1295–1300, 2019.

[5] M. Phongying and S. Hiriote, "Diabetes classification using machine learning techniques," *Computation*, vol. 11, no. 5, p. 96, 2023, https://doi.org/10.3390/computation11050096.

[6] N. Memon, S. B. Patel, and D. P. Patel, "Comparative analysis of artificial neural network and xgboost algorithm for polsar image classification," in *International conference on pattern recognition and machine intelligence*. Springer, 2019, pp. 452–460, https://doi.org/10.1007/978-3-030-34869-4$_4$9.

[7] R. Qin, "The construction of corporate financial management risk model based on xgboost algorithm," *Journal of Mathematics*, vol. 2022, no. 1, p. 2043369, 2022, https://doi.org/10.1155/2022/2043369.

[8] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794, https://doi.org/10.1145/2939672.2939785.

[9] W. Li, Y. Yin, X. Quan, and H. Zhang, "Gene expression value prediction based on xgboost algorithm," *Frontiers in genetics*, vol. 10, p. 1077, 2019.

[10] Z. Li, T. Lu, X. He, J.-P. Montillet, and R. Tao, "An improved cyclic multi model-extreme gradient boosting (cmm-xgboost) forecasting algorithm on the gnss vertical time series," *Advances in Space Research*, vol. 71, no. 1, pp. 912–935, 2023.

[11] V. Chang, J. Bailey, Q. A. Xu, and Z. Sun, "Pima indians diabetes mellitus classification based on machine learning (ml) algorithms," *Neural Computing and Applications*, vol. 35, no. 22, pp. 16 157–16 173, 2023, https://doi.org/10.1007/s00521-022-07049-z.

[12] M. D. Ali, A. Saleem, H. Elahi, M. A. Khan, M. I. Khan, M. M. Yaqoob, U. Farooq Khattak, and A. Al-Rasheed, "Breast cancer classification through meta-learning ensemble technique using convolution neural networks," *Diagnostics*, vol. 13, no. 13, p. 2242, 2023, https://doi.org/10.3390/diagnostics13132242.

[13] Y. Xu, S. Qiu, J. Ye, D. Chen, D. Wang, X. Zhou, and Z. Sun, "Performance of different machine learning algorithms in identifying undiagnosed diabetes based on nonlaboratory parameters and the influence of muscle strength: A cross-sectional study," *Journal of Diabetes Investigation*, vol. 15, no. 6, pp. 743–750, 2024, https://doi.org/10.1111/jdi.14166.

[14] S. Gowthami, R. V. S. Reddy, and M. R. Ahmed, "Exploring the effectiveness of machine learning algorithms for early detection of type 2 diabetes mellitus," *Measurement: Sensors*, vol. 31, p. 100983, 2024, https://doi.org/10.1016/j.measen.2023.100983.

[15] Q. u. Ain, M. A. Khan, M. M. Yaqoob, U. F. Khattak, Z. Sajid, M. I. Khan, and A. Al-Rasheed, "Privacy-aware collaborative learning for skin cancer prediction," *Diagnostics*, vol. 13, no. 13, p. 2264, 2023, https://doi.org/10.3390/diagnostics13132264.

[16] B. F. Wee, S. Sivakumar, K. H. Lim, W. K. Wong, and F. H. Juwono, "Diabetes detection based on machine learning and deep learning approaches," *Multimedia Tools and Applications*, vol. 83, no. 8, pp. 24 153–24 185, 2024, https://doi.org/10.1007/s11042-023-16407-5.

[17] M. A. Uddin, M. M. Islam, M. A. Talukder, M. A. A. Hossain, A. Akhter, S. Aryal, and M. Muntaha, "Machine learning based diabetes detection model for false negative reduction," *Biomedical Materials & Devices*, vol. 2, no. 1, pp. 427–443, 2024, https://doi.org/10.1007/s44174-023-00104-w.

[18] A. Saleem, K. H. Asif, A. Ali, S. M. Awan, and M. A. Alghamdi, "Preprocessing methods of data mining," in *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*. IEEE, 2014, pp. 451–456.

[19] A. Tawakuli, B. Havers, V. Gulisano, D. Kaiser, and T. Engel, "Survey: time-series data preprocessing: a survey and an empirical analysis," *Journal of Engineering Research*, 2024.

[20] S. S. Dhaliwal, A.-A. Nahid, and R. Abbas, "Effective intrusion detection system using xgboost," *Information*, vol. 9, no. 7, p. 149, 2018.

[21] A. Paleczek, D. Grochala, and A. Rydosz, "Artificial breath classification using xgboost algorithm for diabetes detection," *Sensors*, vol. 21, no. 12, p. 4187, 2021.

[22] H. Raipal, M. Sas, C. Lockwood, R. Joakim, N. S. Peters, and M. Falkenberg, "Interpretable xgboost based classification of 12-lead ecgs applying information theory measures from neuroscience," in *2020 Computing in Cardiology*. IEEE, 2020, pp. 1–4.

[23] M. E. Narayanan *et al.*, "Malware classification using xgboost with vote based backward feature elimination technique," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 10, pp. 5915–5923, 2021.

[24] J. Hallman, "A comparative study on linear regression and neural networks for estimating order quantities of powder blends," 2019.

[25] D. M. Abdullah and A. M. Abdulazeez, "Machine learning applications based on svm classification a review," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 81–90, 2021.

[26] M. Surdeanu and M. A. Valenzuela-Escárcega, *Deep learning for natural language processing: a gentle introduction*. Cambridge University Press, 2024, https://doi.org/10.1017/9781009026222.004.

[27] D. Jurafsky, "Speech and language processing," 2000.

[28] H. Sain, H. Kuswanto, S. Purnami, and S. Rahayu, "Classification of rainfall data using support vector machine," in *Journal of Physics: Conference Series*, vol. 1763, no. 1. IOP Publishing, 2021, p. 012048, https://doi.org/10.1088/1742-6596/1763/1/012048.

[29] A. Kataria and M. Singh, "A review of data classification using k-nearest neighbour algorithm," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 6, pp. 354–360, 2013.

[30] D. A. Adeniyi, Z. Wei, and Y. Yongquan, "Automated web usage data mining and recommendation system using k-nearest neighbor (knn) classification method," *Applied Computing and Informatics*, vol. 12, no. 1, pp. 90–108, 2016.

[31] A. Awaysheh, J. Wilcke, F. Elvinger, L. Rees, W. Fan, and K. L. Zimmerman, "Review of medical decision support and machine-learning methods," *Veterinary pathology*, vol. 56, no. 4, pp. 512–525, 2019, https://doi.org/10.1177/0300985819829524.

[32] A. J. Albert, R. Murugan, and T. Sripriya, "Diagnosis of heart disease using oversampling methods and decision tree classifier in cardiology," *Research on Biomedical Engineering*, vol. 39, no. 1, pp. 99–113, 2023.

[33] B. Permana, R. Ahmad, H. Bahtiar, A. Sudianto, and I. Gunawan, "Classification of diabetes disease using decision tree algorithm (c4. 5)," in *Journal of Physics: Conference Series*, vol. 1869, no. 1. IOP Publishing, 2021, p. 012082, https://doi.org/10.1088/1742-6596/1869/1/012082.

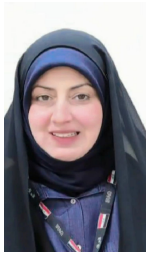
[34] M. Imran Molla, J. J. Jui, B. S. Bari, M. Rashid, and M. J. Hasan, "Cardiotocogram data classification using random forest based machine learning algorithm," in *Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019: NUSYS'19*. Springer, 2021, pp. 357–369.

[35] N. Jalal, A. Mehmood, G. S. Choi, and I. Ashraf, "A novel improved random forest for text classification using feature ranking and optimal number of trees," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp. 2733–2742, 2022.

[36] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012.

[37] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[38] S. F. Raheem and M. Alabbas, "A modified spider monkey optimization algorithm based on good-point set and enhancing position update," *Informatica*, vol. 47, no. 4, 2023, https://doi.org/10.31449/inf.v47i4.4531.



Dr. Zainab N. Nemer is currently an Assistant prof in the Department of Computer Science at the University of Basrah where she has been a faculty member since 2003. She received her PhD degree (2009) in Computer Science from the Basrah University, Iraq. She received her MSc (2000) and BSc (1996) in Computer Science from the University of Basrah, Iraq. Her current research concerns are Artificial Intelligence and soft computing. She published more than 11 journal papers. She can be contacted at email: zainab.nemer@uobasrah.edu.iq.

Sabreen Fawzi Raheem is currently an Assistant Lecturer at the Basra Technical Institute at Southern Technical University –Iraq. I got an M.Sc in Computer Science in the field of Artificial Intelligence from the University of Basrah at the College of Computer Science and Information Technology. She can be contacted at email: sabreen.fawzi@stu.edu.iq.

Dr. Maytham Alabbas , Professor of Computer Science at the University of Basrah since 2003, holds a PhD (2013) from the University of Manchester (UK). His research focuses on AI, NLP, Machine Learning, and related fields. He has published over 45 journals and conference papers. (Contact: ma@uobasrah.edu.iq).