AI Based Diabetes Prediction System

TEAM MEMBER 311521106081 J SAIHARSHAVARADHAN

PHASE 3 DOCUMENT SUBMISSION

1. Introduction

Diabetes mellitus is a metabolic disorder characterized by hyperglycemia which results from the inadequacy of the body to secret and respond to insulin . Usually, it presents itself in different ways: prediabetes - a higher than normal glycemia, overt diabetes: type I and type II, or gestational diabetes, resulting from pregnancy. Diabetes has been medically proven to be linked with long-term impairment of vital organs, including the eyes, kidneys, nerves, heart, and blood vessels. More alarming is its effect on pregnancies – about 7% of pregnancies are affected by diabetes annually , which is a dual life-threatening risk to both mother and her unborn child. The number of people suffering from diabetes has been on the rise and it has been estimated that about 48% of the world population will be diabetic by the year 2045 .

The clinical detection of diabetes involves a fasting plasma glucose level greater than 126 mg/dl (7.0 mmol/l) or a 2h/3h oral glucose tolerance test resulting in plasma glucose greater than 200 mg/dl (11.1 mmol/l) . However, the glycemic threshold levels for detecting diabetes may vary with race. This is because different ethnic groups differ by their glycemic risk levels. As a result, clinicians are posed with the a controversial issue of determining a glycemic threshold for diagnosing diabetes irrespective of the ethnic group of individuals and with an impending question of whether there exists a threshold that can be precise without a series of backup tests to confirm the diagnosis . To reach a meaningful decision in a one-time clinical diagnosis is humanly exacting because several blood sugar test must be carried out both before and after a meal. However, the diagnostic process can be computationally simplified.

In the past years, there have been numerous computational efforts, mainly around the adoption of machine learning algorithms in diabetes research geared towards helping clinicians make a quick and meaningful diagnostic decision. These are the neural network (NN) based algorithms such as the multilayered perceptron (MLP), deep neural networks (DNN) and conventional machine learning models (CML) Also, with the growing development of tools for diabetes testing and , individuals can engage in personalized diabetes status assessment for better lifestyle adjustments.

Despite the body of research efforts in the prediction of the onset of diabetes, the accuracy rate to date suggests that there is still much room for improvement. This is necessitated by the fact that diabetes poses serious health challenges if not properly managed or diagnosed on time. In this paper, we propose a robust machine learning framework for building a diabetes prediction model to aid the clinical diagnosis of diabetes. The contributions of this paper are summarized as follows:

- 1.
  Considering that most real-world data do not satisfy normality assumptions: the Spearman correlation (SC) is used for feature selection while polynomial regression (PR) is used for missing value imputation. Both methods are approached in a way that their functionality is best utilized for any given data.
- 2.

We propose a CML-based classifier and design a DNN-based classifier that scales to the diabetes prediction problem. We in turn explore their hyperparameter optimization. Then, we compare with state-of-the-art classification algorithms.

- 3.
We relabel the PIMA Indian dataset to accommodate prediabetes prediction for a comprehensive clinical diagnosis of diabetes.

The remainder of the paper is organized as follows. Section 2 presents the literature on the state-of-the-art in diabetes prediction research and Section 3 introduces the proposed framework. Section 4 reports on the experiments, results, and discussion. This paper's limitations and future work are presented in Section 5, and finally, Section 6 concludes the paper.

2. Related work

The discussions on existing literature will be made from the points of view of data preprocessing, and classification in a way that highlights the contributions of this paper. However, we will limit our review to recently published articles because only recently has performance accuracy in diabetes research begun to improve.

The family of NN-based methods has continued to show improvements in accuracy in diabetes research. In they apply min-max normalization and a variational autoencoder sparse autoencoder to address data normalization, imbalance, and feature augmentation, respectively. MLP was subsequently used for classification to achieve a 92.31% accuracy. A further improvement in accuracy can be seen in, where their artificial backpropagation scaled conjugate gradient neural network (ABP-SCGNN) was reported to achieve 93% accuracy without data preprocessing. Another good performance recorded with NN-based models is apparent in the work of . In their work, the median value imputation, k-nearest neighbor (K-NN), and an iterative imputer were compared for missing value imputation. Then, MLP was used for classification to achieve an F1-score of 98%. Khanam and Foo 2021 applied Pearson correlation, and median value imputation for feature selection, and missing values imputation. They further normalized the data and removed outliers using interquartile ranges. Their DNN based classification model with different hidden layers achieved 88.6% accuracy. In, a deep neural network (DNN) model achieved an accuracy of 98.07%. Though the authors claim data cleaning was applied, the method used was not mentioned in the work. In the principal component analysis (PCA) and the median value were used for feature selection and missing value imputation, respectively. MLP was then adopted for classification to achieve an accuracy of 75.7%. Also, in PCA and minimum redundancy, maximum relevance (mRMR) was employed for feature selection and missing value imputation, respectively. Then, with an MLP, they achieved a classification accuracy of 73.90% accuracy.

Interestingly, CML-based methods show comparable performance accuracies to NN-based methods. In after preprocessing the data, the team evaluate the performance of different classifiers: RF, light gradient boosting machine (LGBM), linear regression (LR), and support vector machines (SVM), for their classification performances. The LGBM emerged as the best model with 86% accuracy. In a classification performances of decision tree (DT), RF, naïve Bayesian (NB), K-NN, Adaptive boosting (AB) were compared, with AB achieving the best accuracy of 79.42%. In , they applied what they termed a step forward and backward feature selection strategy with PCA and mean values for feature selection and missing values imputation methods, respectively. They then compared the classification performances of RF and SVM, of which the RF model emerged as the best with an accuracy of 83%. Gnanadass Iswaria applied the mean of each column of the data for addressing

missing values and then trained on different classification models: NB, linear regression (LR), RF, AB, gradient boosting machine (GBM), and extreme gradient boosting (XGBoost). The XGBoost emerged as the best model with an accuracy of 77.54%. In , they compared the performance of different classification models: SVM, K-NN, NB, Gradient boosting (GB), and RF. The RF ranked the highest with an accuracy of 98.48%. Hasan et al. applied Pearson correlation and mean value imputation for feature selection and missing value imputation, respectively. With the grid search method for hyperparameter tuning under the K-fold cross-validation setting, they experimented on the performance of different classification models: extreme boosting (XB), AB, RF, DT, and K-NN. The XB ranked the best with a 94.6% accuracy. Singh and Singh employed a stacked ensemble of Linear SVM, Radial Basis function SVM, DT, and K-NN for classification and achieved an accuracy of 83.8%. In they achieved an accuracy of 87.1% with a combination of methods: NB for missing value imputation, and RF classifier. Maniruzzaman et al employed the group median and median imputation method for addressing missing values and outliers and applied RF for feature selection. Then they compared the performance of SVM, NB, linear discriminant analysis (LDA), linear regression (LR), DT, RF, AB, gaussian process classification (GPC), quadratic discriminant analysis (QDA) of which the RF ranked the best with a 92.26% accuracy. In after preprocessing the data, the performances of DT and RF classifiers were compared, of which RF ranked the best with an accuracy of 76.04%

Table 1. Summary of literature review.

Authors	**Year**	**Feature Selection (FS) & Missing Value Imputation (MVI)**	**Classification**	**Comments**
Neural network-based methods				
Garcia-Ordas et al.	2021	FS: none specified; MVI: removed missing values;	MLP	MLP achieved the best accuracy, 92.31%
Bukhari et al.	2021	FS: none specified; MVI: none specified	ANN trained with ABS conjugate gradient neural network (ABP-CGNN)	Achieved 93% accuracy
Roy et al.	2021	Median value, K-NN, and iterative imputer were used for missing value imputation.	ANN	ANN achieved 98% accuracy
Khanam et al.	2021	FS: Pearson correlation MVI: Median value for missing values imputation.	DNN run with different hidden layers	Achieved 86.26% accuracy with 2 hidden layers.
Naz and Ahuja	2020	Method not stated	MLP and DL with 2 hidden layers	DL achieved best accuracy of 98.07%
Alam et al.	2019	FS: PCA; MVI: Median value	MLP	Achieved 75.7% accuracy
Zou et al.	2018	FS: PCA; MVI: redundancy and minimum relevance	MLP	Achieved 73.90% accuracy

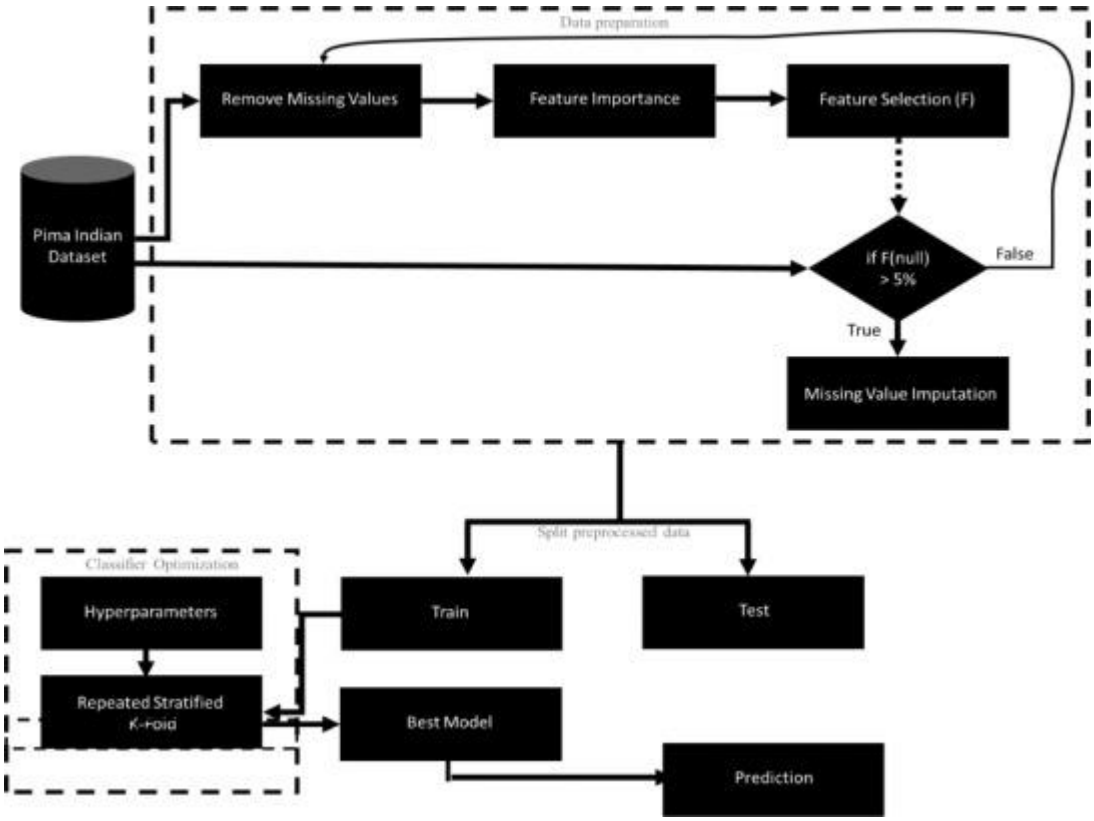
Authors	Year	Feature Selection (FS) & Missing Value Imputation (MVI)	Classification	Comments
Conventional machine learning-based methods				
Roy et al.	2021	Median value, K-NN, and iterative imputer were used for missing value imputation.	LR, SVM, RF, LGBM	LGBM achieved 86%
Khanam et al.	2021	FS: Pearson correlation; MVI: Median value for missing values imputation;	DT, RF, NB, K-NN, AB	Adaboost achieved 79.42%
Sivaranjani et al.	2021	FS: Step-forward + Backward FS + PCA; MVI: mean value	RF, SVM	RF achieved best accuracy of 83%
Gnanadass	2020	MVI: mean value	NB, LR, RF, AB, GBM, XGBoost	XGBoost achieved best accuracy of 77.54%
Reddy et al.	2020	FS: none specified; MVI: none specified	SVM, K-NN, NB, GB, RF, LR	RF achieved best accuracy of 98.48%
Hasan et al.	2020	FS: correlation; MVI: mean value	XB, AB, RF, DT, K-NN	XB achieved best accuracy of 94.6%
Singh et al.	2020	FS: none specified; MVI: none specified	Ensemble models Radial basis SVM, DT, linear SVM, K-NN	Achieved 83.8% accuracy
Wang et al.	2019	FS: none specified; MVI: NB for predictive imputation	RF	Achieved 87.1% accuracy
Maniruzzaman et al.	2018	FS: RF for predictive feature selection; MVI: median value	SVM, NB, LDA, LR, DT, RF, Adaboost, GPC, QDA, j48	RF achieved best accuracy of 92.26%
Zou et al.	2018	FS: PCA; MVI: redundancy and minimum relevance	DT, and RF	RF achieved best accuracy of 76.04%

In all, feature selection and missing value imputation data preprocessing approaches have proven to substantially to be relevant to classification performance in diabetes prediction. However, most of the methods adopted for data preprocessing have been shown to perform best when the distribution of the data is normal. In a case where the data violates normality assumptions, nonlinear methods will be better suited to the problem and are expected to contribute highly to the performance gains of a classifier. Hence, nonlinear preprocessing methods and classifiers will be explored in this paper for data preprocessing.

3. Methodology

Our proposed framework comprises two stages: data preprocessing and classification. This is designed to address what we presume to affect accuracy in the early diagnosis of diabetes

mellitus. They are: (1) not all the attributes are important features for prediction, (2) there are numerous missing values, (3) is there a classifier that better fits the data? the proposed framework is diagrammatically illustrated. In what follows, each stage will be discussed in detail. Our discussions will begin with the algorithms that make up each component of the framework.



3.1. Spearman correlation

The Spearman correlation (SC) is a nonparametric estimate of the strength and direction of monotonic associations between two variables calculated based on ranks. The SC coefficient can be calculated from the following relation where r is the sample correlation coefficient, d is the difference between ranks, $\sum d_i^2$ is the sum of d squared values and n is the number of samples.

Since the SC coefficient focuses on differences in rank orders of data rather than differences in means, it is appropriate for non-normally distributed continuous data and for data with outliers. Usually, the coefficients are scaled in the range $[-1, +1]$ where ($p = +1, \ -1$) describes a perfect monotonic association and $r = 0$ describes a lack of association.

To evaluate the significance of the statistical test, the hypothesis test is widely used [23] to estimate the strength of the relationship in the population from which the data were sampled. There are two ways to approach the significance of the test: using the correlation coefficient, or the p-value. If the value of r is not between the positive and negative critical values, then the statistical test is significant, otherwise, it is not significant. For the p-value, the decision to reject or accept the null hypothesis lies in the strength of the evidence of the p-value compared against the significance value. Usually, the significance value can be set to 0.05 or 0.01 which are: statistically significant and highly statistically significant, respectively. Similarly, if p-value <= 0.05, or p-value <= 0.01, there is strong, or very strong evidence, respectively, to reject the null hypothesis in favor of the alternative.

3.2. Polynomial regression

Polynomial regression (PR) is a special case of linear regression that models a curvilinear relationship between the predictor variable and the outcome variable. The PR suffices with a goal to fit the regression line to a curved set of points, that is, the non-linear patterns between the predictor variable and outcome variable when linearity is not satisfied. Polynomial models can approximate continuous functions with precision which makes them more powerful at handling nonlinearity.
The Eq. (2) is a *k*th-order polynomial model in one variable; where β_0 is the bias term, $\beta_1,\beta_2,\cdots,\ \beta_k$ are the coefficients to be determined and x the predictor variable with additional variables x^2, …, x^k created by raising x to an exponent.
Summary of literature reviewIt is always possible to fit a polynomial model of order n-1 perfectly to a data set of n points. However, this will almost certainly result in overfitting. Therefore, a low-order model should be preferred to a high-order model as long as the model provides a "good" fit to the data. A typical low-order polynomial such as the 2-order degree polynomial can be expressed as:(3) With a 2-order degree polynomial, only one new variable is added. For instance, using a given predictor vector, x, where a system of linear equations can be created as:(4)

3.3. Machine learning classification models

This paper only considers nonlinear classifiers for the classification problem.

3.3.1. Random forest model

Random forest is a supervised learning algorithm for building a predictor ensemble of decision trees, usually trained with the "bagging" method, that grows in randomly selected subspaces of data. By bagging, it is meant that multiple decision tree learning models are combined for accurate and stable prediction. As proposed by Breiman , each tree in the ensemble of trees outputs a prediction, however, only the class with the most votes is considered the model's prediction . However, the model's prediction can take two forms: if the output is a mean value, then the RF solves a regression problem, while if the output is a mode of the classes, then the RF solves a classification problem. Essentially, RF prediction stability is formed from weakly-correlated classifiers/regressors.
For the RF to be capable of identifying and responding to the best features among a random subset of features, it has to be insensitive to noisy variables and be stable in the presence of small amounts of data . However, the ability of the RF to be insensitive to noisy variables might not always be the case. Therefore, by ensuring the best features are fed in, a higher performance is more likely.

3.3.2. Support vector machines

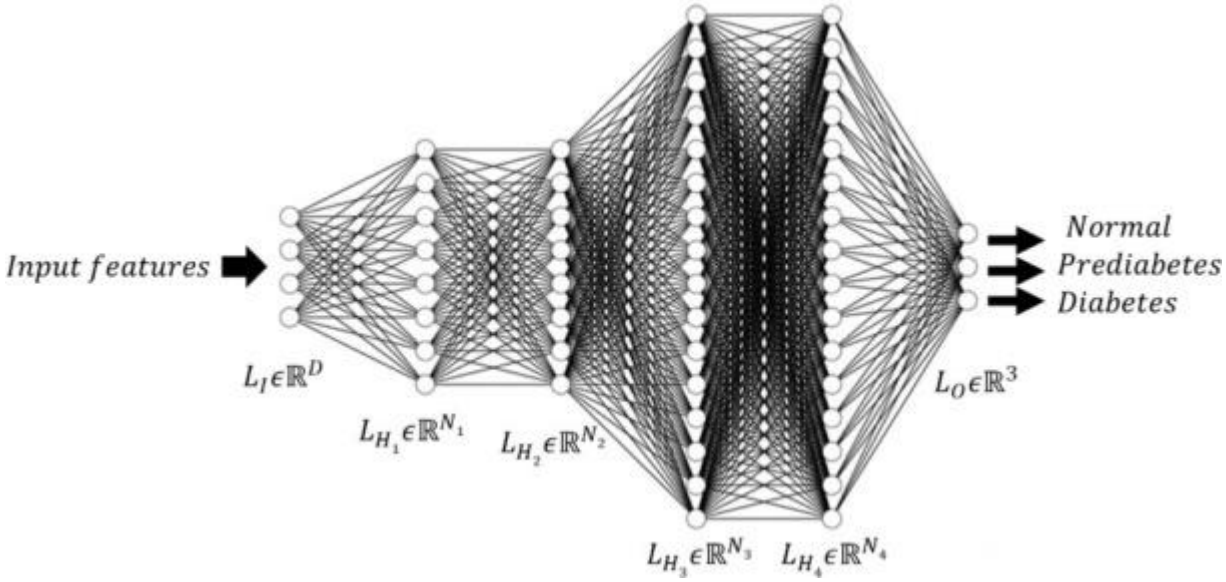
The SVM is a supervised machine learning algorithm that aims to find the optimal boundary between data points in feature space. Traditionally, the SVM tries to find the best fit line, a hyperplane, that maximizes the separation margin between two classes. However, most real-world data are mostly nonlinear. Nonlinear problems in SVM are solved by mapping *n*-dimensional input space to a high dimensional feature space where the SVM can still operate linearly. Similarly, in a multi-classification problem, SVM generates multiple binary classifiers to linearly separate data points of pairs of classes within a high dimensional feature

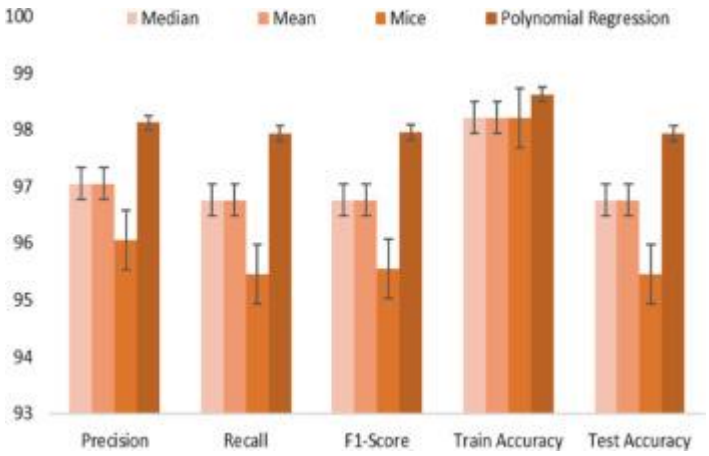
space. This is achieved using what is popularly termed the kernel trick . SVM has been popularly investigated for binary classification problems in diabetes research and .

3.3.3. Deep neural network

The deep neural network (DNN) of interest in this paper is a class of feed-forward DNN that extracts a feature and transforms it using nonlinear activation functions. DNN layers comprise the input, hidden, and output layers. The connection between these layers begins from the input layer with associated weights to the hidden layers and then to the output layer. For any neuron in each layer to pass data to the next layer, the output of that node has to be above a specified threshold value defined by an activation function. During training, the weight of a neuron is updated using back-propagation to minimize the error of the network to generalize to unseen samples. The added capability of the DNNs comes from the depth of the hidden layers. Depending on the problem, the more the depth, the better the generalizability of the network . This is especially true for image-based problems. DNNs have been adopted for diabetes prediction

In this paper, our deep learning model is designed to be double, or twice in size, and is twice repeated. As such, we term it twice growth deep neural network (2GDNN). In essence, the hidden layers grow by two in size of the input and are repeated twice. Our 2GDNN architecture is described in and consists of an input layer, four hidden layers, and an output layer. The decision to pass a neuron from one layer to another is dependent on the function, *f*, acting on a neuron, *x*, to either pass or not pass the neuron. This is expressed as: where *x, w, b*, ϕ, are the inputs, weights, bias, and activation function, respectively. During learning, the network updates its w_i and *b* using the backpropagation method to minimize the difference between a target output of a problem and the network's predicted output.





3.4. The proposed framework

3.4.1. Data

The datasets used in this paper are the publicly available PIMA Indian diabetes mellitus dataset and the publicly available diabetes dataset from the Laboratory of Medical City Hospital (LMCH). The former consists of 768 instances: 268 patients belong to the diabetic class and 500 patients belong to the non-diabetic class. The diabetes data is sampled from the

Pima Indian population near Phoenix, Arizona . Each patient is described by the following attributes: pregnancies, glucose, blood pressure, skin thickness, insulin, body mass index (BMI), diabetes pedigree function, and age. A description of the PIMA Indian dataset is provided in . The latter consists of data from 1000 patients of Iraqi nationals collected from LMCH . In all, about 103, 53, and 844 patients belong to the normal, prediabetes, and diabetes class, respectively. Each patient is described using the following attributes: the number of patients, sugar level blood, age, gender, creatinine ratio (Cr), BMI, urea, cholesterol (Chol), Fasting lipid profile, including total, LDL, VLDL, Triglycerides (TG) and HDL Cholesterol, HBA1C. There are no available descriptions for these attributes.

Table 2. Description of the PIMA Indian diabetes dataset.

S/N	Feature	Description	Missing Value
1	Pregnancies	Number of pregnancies	110
2	Glucose	Glucose concentration (2h oral test)	5
3	Blood Pressure (BP)	Diastolic blood pressure	35
4	Skin Thickness (ST)	Skin fold thickness in mm	227
5	Insulin	2h insulin serum (mm u/ml)	374
6	BMI	Body mass index = weight in kg / height in m^2	11
7	Diabetes Pedigree Function (DPF)	Likelihood value computed from the relationship between the patient and the genetic history of the patient's relative	0
8	Age	Age in years	0

3.4.2. Data preprocessing

As the base dataset, the PIMA Indian dataset is analyzed for normality using a whisker plot. This is a statistical tool often used in explanatory data analysis. From , it can be observed that the values of some features are skewed which is an indication of a violation of normality assumptions. Considering that the PIMA Indian dataset does not satisfy normality assumptions, we approached data preprocessing differently. As an initial step to the preprocessing, the dataset is relabeled to include the prediabetes class. This is because the existing diabetes research is limited to only the prediction of normal and diabetes classes. However, if the research is targeted towards diagnosing diabetes, then there is a need for the prediabetes class. As a result, the dataset is relabeled based on the levels of glucose to conform to medical charts provided online[1] on clinical practices in the diagnosis of diabetes. Then, SC and PR methods are employed for feature importance selection, and missing value imputation, respectively.

3.4.3. Feature importance and selection

Since the PIMA Indian dataset consists of numerous missing values, as shown in , which is likely to bias feature selection, the dataset is replicated to be used for feature selection. With the replicated dataset, each row entry with missing values for all features is removed to eliminate bias the zero entries will present to the feature selection process. Next, the SC is applied to the non-zero entries of the dataset to generate the p-values. A p-value measures the probability of significance of the correlation between each predictor variable and the outcome variable - that means the less the p-value, the higher the feature importance. The significance threshold, *T*, is set to 0.01 for a confidence rate of 99%. To address the problem of features competing for importance, the p-values are scaled. This is to amplify the importance of one feature over another. The scaled p-values are presented in . Then, the most important features are selected by evaluating the scaled p-value over *T*. If a scaled p-value is less than *T*, the null hypothesis, H_0, is rejected in favor of the alternative hypothesis, H_1, otherwise, it is accepted. The algorithmic structure of these steps is presented in Algorithm 1 and the hypothesis is given as follows:

Table 3. Statistical significance of the scaled p-value of predictor and outcome variables for feature selection.

Variables	Preg	Glucose	BP	ST	Insulin	BMI	DPF	Age	Output
Preg	0	0.022	0.061	>0.1	>0.1	>0.1	>0.1	<0.0001	0.043
Glucose	0.022	0	0.086	>0.1	<0.0001	>0.1	>0.1	<0.0001	**<0.0001**
BP	0.061	0.086	0	>0.1	>0.1	0.0007	>0.1	<0.0001	**0.007**
ST	>0.1	>0.1	>0.1	0	>0.1	<0.0001	>0.1	0.062	>0.1
Insulin	>0.1	<0.0001	>0.1	>0.1	0	<0.0001	>0.1	0.003	**<0.0001**
BMI	>0.1	>0.1	0.0007	<0.0001	<0.0001	0	>0.1	>0.1	>0.1
DPF	>0.1	>0.1	>0.100	>0.1	>0.1	>0.1	0	>0.1	>0.1
Age	<0.0001	<0.0001	<0.0001	0.062	0.003	>0.1	>0.1	0	**<0.0001**
Output	0.043	**<0.0001**	**0.007**	>0.1	**<0.0001**	>0.1	>0.1	**<0.0001**	**0**

ST - Skin Thickness, DPF – Diabetes Pedigree Function, Preg – Pregnancy.

Algorithm 1. Feature Importance Determination.

Input: data: nonzero entries of the original PIMA Indian diabetes dataset

Output: sorted feature: list of features sorted in the order of importance based on the probability value

Initialization *p* = ← [] // p-value list for all features

Initialization *label* ← [] // feature labels list

***p* = *tdist*(*t, f, k*) // probability values**

where *r* is the correlation coefficient, *n* is the sample size, and *p* is the associated *p-value* given *t*-statistics with degrees of freedom, *f,* and a number of tails, which is usually 2.

for i, j in data do check

set k to the index of the response variable

set the significance threshold, *T*

scale *j*[*k*]

if *scaled j*[*k*] ≤ *T* then

add *j*[*k*] *to p*

add i to *label*

end if

end for

H_0: There is no significant correlation between each feature (f_1, f_2, …, f_n) and the outcome variable.
H_1: There is a significant correlation between each feature (f_1, f_2, …, f_n) and the outcome variable
shows that glucose, blood pressure, insulin, and age, are significantly correlated to the outcome variable. Therefore, H_1, is accepted and leads to the selection of features considered to be important and will make up a subset of the original PIMA Indian dataset. Though there exist multicollinearities among the selected predictor features as evidenced from they are negligible since they do not affect predictions of new observations]. The same feature selection algorithm is applied to the LMCH dataset and can be adapted to any other data.

3.4.4. Missing value imputation

A common practice in diabetes research is the use of mean, and median for imputing missing values. However, these methods are highly likely to increase data bias . Another method is that of multiple imputations of missing values (MICE) . This method is known to surpass the mean and median approaches; however, MICE suffer from performance degradation in the presence of nonlinearities in predictor variables . In this paper, we employ a predictive approach to missing value imputation using PR which is a nonlinear regressor. The input to the missing value imputation process is the subset of the PIMA Indian dataset with only selected features. The steps are as follows:

- i.
  Firstly, the percentage of missing values for each selected feature is checked over a decision threshold of 5%. The decision is: if the number of zero entries in the subset data is greater than 5%, PR is applied, otherwise, the entry is removed. From the PIMA diabetes dataset, Insulin is observed to have above 5% zero entries.

- ii.

Secondly, the feature from the subset data that highly correlates to Insulin becomes the predictor variable for predicting Insulin. The variable is Glucose.

- iii.
Lastly, the data points are divided into nonzero and zero sets, where the nonzero set is used for training and testing while the zero set is predicted. The resulting output is combined with the nonzero set to form the final dataset.

3.4.5. Classifier optimization

We hypothesize that: with the reduced feature space, the best hyperparameters of each classifier, RF, SVM can be optimized. We define the space of the hyperparameters for RF, SVM, as: $\Lambda_1,\Lambda_2,\cdots, \Lambda_n$, which is integer valued. For the hyperparameter setting $\lambda \in \Lambda$, the best possible hyperparameter value combinations can be obtained by:(where the objective function $f(\lambda)$ is to maximize accuracy with combinations of hyperparameters λ.
In this paper, the hyperparameters used for each of the classifiers, RF, SVM and 2DGNN are briefly described in . However, RF and SVM parameters were optimized because only a fraction of the hyperparameters contribute to the classification performance [39]. To find the best configuration of $\lambda \in \Lambda$, an exhaustive simple search mechanism, the grid search method is adopted, particularly because the hyperparameter is of reduced space. As a result, the problem of the *curse of dimensionality* can be avoided. We specify a finite set of values for the hyperparameters, to evaluate $\Lambda = \Lambda_1 \times \Lambda_2 \times \cdots \Lambda_n$, the Cartesian product of the sets. Then, the grid search for hyperparameter tuning follows repeated cross-validation with stratification. Stratification implies the sorting of data into smaller sub-groups known as strata, such that each group is a good representative of the whole. The output variable is stratified, and the dataset is pseudo-randomly split into k-folds to ensure that different strata are proportionally in each fold. Then the number of times the cross-validation is repeated is minimized. This is to avoid redundancy . Finally, the hyperparameter tuning results in a model, considered to be the best model obtained from the combination of the hyperparameters with the highest cross-validation accuracy. However, finding the best combination of hyperparameters is not a trivial task and as such might not always present the best accuracy.

Table 4. Experimental setup and parameters for the classifiers with and without optimization.

Items		**Description**	**Without Optimization**	**With Optimization**
RF	Max-Depth	Controls how specialized each tree is to the training dataset. The more the value the more likely overfitting.	2	3
	Max-Features	The maximum allowable number of trees the RF will consider for each split.	3	4
	n-Estimators	The number of trees you want the algorithm to build.	50	50
SVM	C	A regularization parameter that controls the error of the misclassification of SVC to data.	100	1000

Items		Description	Without Optimization	With Optimization
	Kernel	A non-linear transformation function to map data to a high-dimensional space	rbf	rbf
	Gamma	A nonlinear parameter that represents the separation line or decision region between classes.	0.0001	0.001
	Optimizer	An algorithm that minimizes the loss function of the network during training.	Adam	RMSProb
2GDNN	Epoch	Defines the number of passes made to the entire training dataset during training.	100	200
	Batch_size	The number of samples utilized in one iteration.	1	5
K-Fold	n-Splits	The number of different validations set to create from the given train data.	10	10
	n-repeats	The Number of times cross-validation is repeated.	–	3
PIMA (#728)	Train	Percentage of the dataset for training	582	582
	Test	Percentage of the dataset for testing	146	146
MCH	Train		700	700
	Test		150	150

3.4.6. Evaluation

The experimental settings for evaluating the proposed machine learning framework for diabetes prediction are (1) evaluation of the proposed data preprocessing methods within the ML framework, (2) performance evaluation of different machine learning classifiers with and without optimization and severity assessment of the best model in diabetes prediction, (3) performance evaluation of the proposed deep learning model performance across datasets, and (4) comparison with the state-of-the-art. Performance in each setting is evaluated using the following metrics: sensitivity, precision, F1-score, specificity, and accuracy which are briefly discussed in the following subsections.

3.4.6.1. Specificity

This is the proportion of patients with no diabetes, the negative instances, who are identified as being non-diabetic and it is computed as the ratio of true negatives (TN) to the sum of TN and false positives (FP).(8)Specificity=TNTN+FP

3.4.6.2. Sensitivity

This is the proportion of patients with diabetes, the positive instances, who are correctly identified as being diabetic and it is computed as the ratio of true positives (TP) to the sum of TP and false negatives (FN).(9)Sensitivity=TPTP+FN

3.4.6.3. Precision

This is the proportion of patients with diabetes, the positive instances, who are correctly identified as being diabetic out of all the diabetic patients and is computed as the ratio of TP to the sum of TP and false positives (FP).(10)Precision=TPTP+FP

3.4.6.4. F1-Score

This is the weighted average of precision and recall. As a result, this score considers both false positives and false negatives.(11)F1−score=2*[(Recall*Precision)Recall+Precision]

3.4.6.5. Accuracy

This is the ratio of the total number of correct predictions and the total number of predictions, Unlike commonly practiced in diabetes research, this paper will also report each model's training and testing accuracy for each experiment.

4. Results

The result of the experiments with its supporting discussions will be presented in the order of the experimental scenarios.

4.1. Performance evaluation of the proposed data preprocessing methods

We will begin the evaluation of the data preprocessing methods in the proposed ML framework from the feature selection point of view. To analyze the contribution of the feature selection method to the performance of the ML framework, the LMCH dataset is used. This is because it originally came with a prediabetes class and there is no missing value. Therefore, it will be easy to equate a difference in accuracy to the feature selection method across different experimental settings with and without feature selection. shows that feature selection improves the performance of the classifiers by 0.68%, 4%, and 0.67% for the 2GDNN, RF, and SVM models, respectively. This result shows that the performance of the RF model is greatly enhanced after feature selection. Subsequently, all analyses will use the subset of the datasets after feature selection as input.

Table 5. Evaluation of the performance of feature selection within the ML framework.

Set	Model	Precision(%)	Recall(%)	F1-Score(%)	Train Acc.(%)	TestAcc.(%)
No FS	SVM	94.385	94.000	93.714	95.429	94.000
	RF	88.651	92.500	90.432	92.000	92.500
With FS	2GDNN	96.212	96.000	96.051	100	95.999
	SVM	94.116	94.667	94.272	95.286	94.667
	RF	96.653	96.500	95.976	97.375	96.500
	2GDNN	**97.348**	**96.667**	**96.965**	**98.714**	**96.667**

FS – feature selection.

Next, we evaluate the performance of the proposed missing value imputation method. This is achieved by using the selected feature subset of the PIMA Indian dataset which becomes the input to the ML framework. This experiment compares the performance of the mean, median, MICE missing value imputation methods to the proposed PR method. This is to determine the best method that addresses missing value in the proposed ML framework. For simplicity of the experiment, only the RF model was used as the base classifier for this experiment and without optimization.

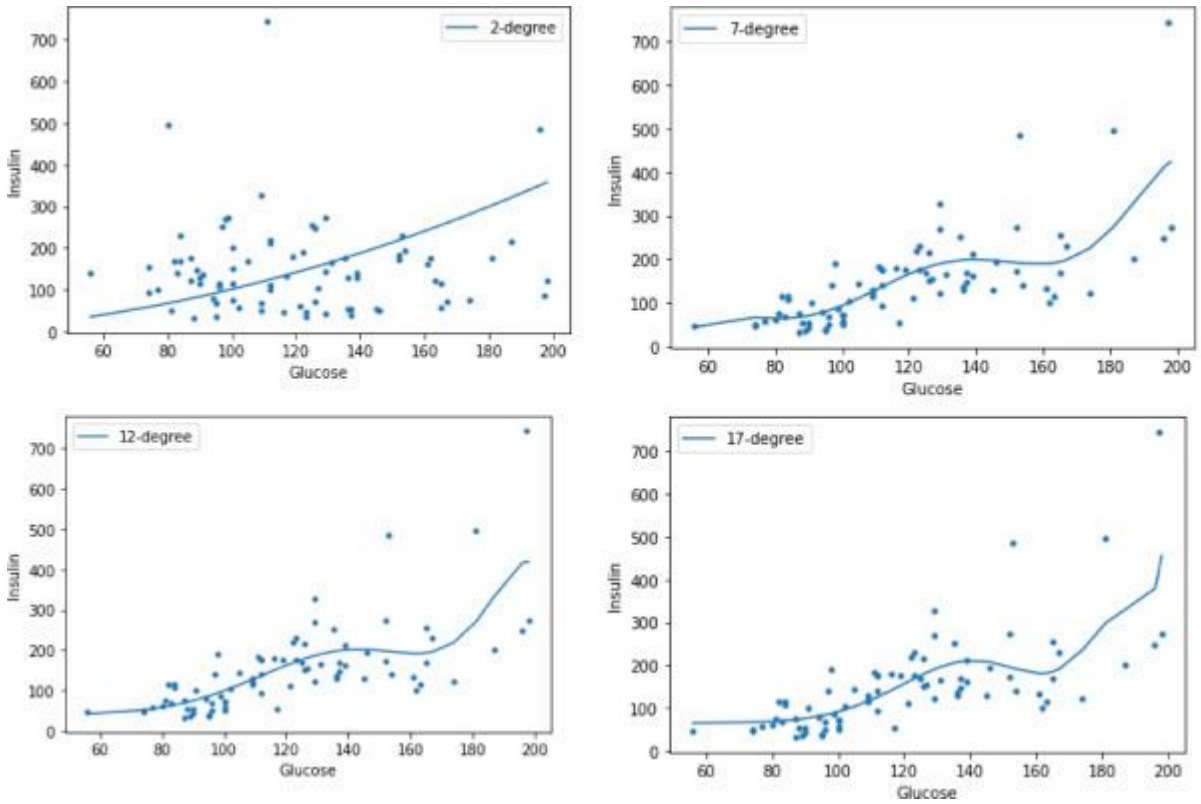
show that the PR predictive imputation method is better at addressing missing values than the mean, median methods commonly used in the literature by 1.2% difference in accuracy. As expected, the MICE fell short in performance with a difference of 2.5% in accuracy compared to the PR. This is likely because of its sensitivity to nonlinearities in the data. Further, the results show that the mean and median performed equally across the evaluation metrics which indicates that both methods are affected equally when the data distribution is nonlinear. On the other hand, the MICE is observed to be more affected by the nonlinearities of the data given its accuracy of 98.2% on train data and 95.5% on the test data which is a difference of 2.7%. However, with a difference of 0.69% between the train and test accuracies, the predictive imputation method, PR, show that it is a better method for addressing missing values for nonlinearly distributed data.

Table 6. Evaluation of the performance of missing value imputation methods.

Data Preprocessor	**Precision(%)**	**Recall(%)**	**F1-Score(%)**	**Train Accuracy(%)**	**Test Accuracy(%)**
FS+Mean	97.045	96.753	96.761	98.208	96.753
FS+Median	97.045	96.753	96.761	98.208	96.753
FS+Mice	96.054	95.455	95.552	98.208	95.455
FS+PR	**98.119**	**97.931**	**97.954**	**98.618**	**97.931**

FS – feature selection, MVI – missing value imputation method, PR – polynomial regression.

On a deeper note, the predictive power of the PR depended on the experimental evaluation of the best *n*th-order degree polynomial to find a fit for the insulin data. Using the root mean square error (RMSE) and R-squared (R2) errors, it can be inferred from Fig. 4 that the *7*th-order degree polynomial is a better fit to the data. Hence, the PR is generated with the *7*th-order degree polynomial.



N-Degree	RMSE
2	80.91
7	**76.99**
12	77.19
17	77.29

1. Download : Download high-res image (398KB)
2. Download : Download full-size image

Fig. 4. The plot of fit of the polynomial regression line of the predictor data (Glucose) to the predicted values of Insulin. An nth degree polynomial of 2, 7,12, and 17 are plotted. The performance summary shows that a 7-degree polynomial is a better fit.

4.2. Performance evaluation of machine learning classifiers with and without optimization

Using a subset of the feature selected PIMA Indian dataset and addressing missing values with the PR method, the result of the supervised classification algorithms: SVM, PR, and the proposed 2GDNN are evaluated with and without optimization and compared to determine the best fit model to the given problem. To decide on the classification algorithm for a given classification problem, it is important to select a model that generalizes best to unseen probe samples. The farther the test accuracy is from the training accuracy, the less the generalizability of the model to unseen data points. Table 7 shows the classifier's performance in the order from best to least: ORF, RF, O2GDNN, SVM, OSVM, 2GDNN in terms of accuracy for both scenarios of the classifier with and without optimization. The optimized RF (ORF) and RF did not only achieve higher performance accuracies than the other classifiers, but they both are also able to generalize best to unseen data points. They achieved a 0% and 0.68% difference between the train and test accuracies, respectively. The 2GDNN also had a better chance of generalizing to unseen data by its 1.76% difference between the train and test accuracies after its best parameters for the given data were sorted.

Table 7. Performance evaluation of classification algorithms within the proposed ML framework.

Data Preprocessor	Classifier	Precision (%)	Recall (%)	F1-Score (%)	Train Accuracy(%)	Test Accuracy(%)
FS+MVI	SVM	96.668	96.330	96.333	99.407	96.330
	OSVM	95.605	95.412	95.421	100	95.412
	RF	98.119	97.931	97.954	98.620	97.931
	ORF	**100**	**100**	**100**	**100**	**100**
	2GDNN	95.156	94.495	94.504	99.802	94.495
	O2GDNN	97.342	97.245	97.255	99.012	97.248

FS – feature selection, MVI – missing value imputation.

Further, we investigate the severity of a patients' diabetes prediction using the ORF and O2GDNN models. From , it can be observed that the 2GDNN model shows a higher probability of determining the severity of a diagnosis than ORF. However, the 2GDNN failed at one point to make a correct diagnosis which makes the ORF modestly better at handling an accurate diagnosis of diabetes severity. In a broader context, the O2GDNN is better off used when the number of data points is large, and in such a scenario, the ORF is expected to fail because it is only known to be stable with small amounts of data [28,29].

Table 8. Determining the severity of a prediction model for diabetes diagnosis.

S/N	Patient State	Glucose	Insulin	Blood Pressure	Age	Predicted	Probability of Diabetes Severity (%)					
							N		P		D	
							ORF (%)	O2GDNN (%)	ORF (%)	O2GDNN (%)	ORF (%)	O2GDN (%)
1	N	80	232	75	45	0	**97.13**	94.96	0.71	0.48	2.16	4.56
2	D	126	34	35	38	2	0.00	0.00	4.54	82.74	**95.46**	17.26
3	P	100	190	80	45	1	3.22	4.59	94.39	**94.84**	2.40	0.57
4	D	130	20	100	50	2	0	0.00	3.07	0.00	96.93	**100**
5	N	90	210	72	25	0	97.13	**99.68**	0.71	0.00	2.16	0.31
6	P	121	181	76	30	1	3.22	0.00	94.39	**99.99**	2.40	0.00

N - Normal, D - Diabetes, P – Prediabetes.

4.3. Performance evaluation of the proposed deep learning model performance across datasets

For this experiment, the performance of the proposed 2GNN model is evaluated across the PIMA Indian dataset and the LMCH dataset. The result of this experiment is shown in The datasets were preprocessed before classification based on the need of the dataset. The performance of the proposed 2GDNN model across the datasets shows that it is a promising classifier that fits well to the proposed machine learning diabetes prediction and diagnosis framework. Precisely, the 2GDNN model achieved a 97.248% test accuracy on the PIMA Indian dataset and achieved a 97.333% accuracy on the LMCH dataset. Further comparison is made to compare the 2GDNN with the only work in literature , as far as our search, where the LMCH dataset was used. In they achieved a 98.95% accuracy with 392 randomly sampled

data points of the LMCH dataset. In comparison to our proposed 2GDNN model which used the entire 1000 data points of the LMCH, a 1.617% difference in accuracy is observed.

Table 9. Performance evaluation of the proposed ML framework on different datasets.

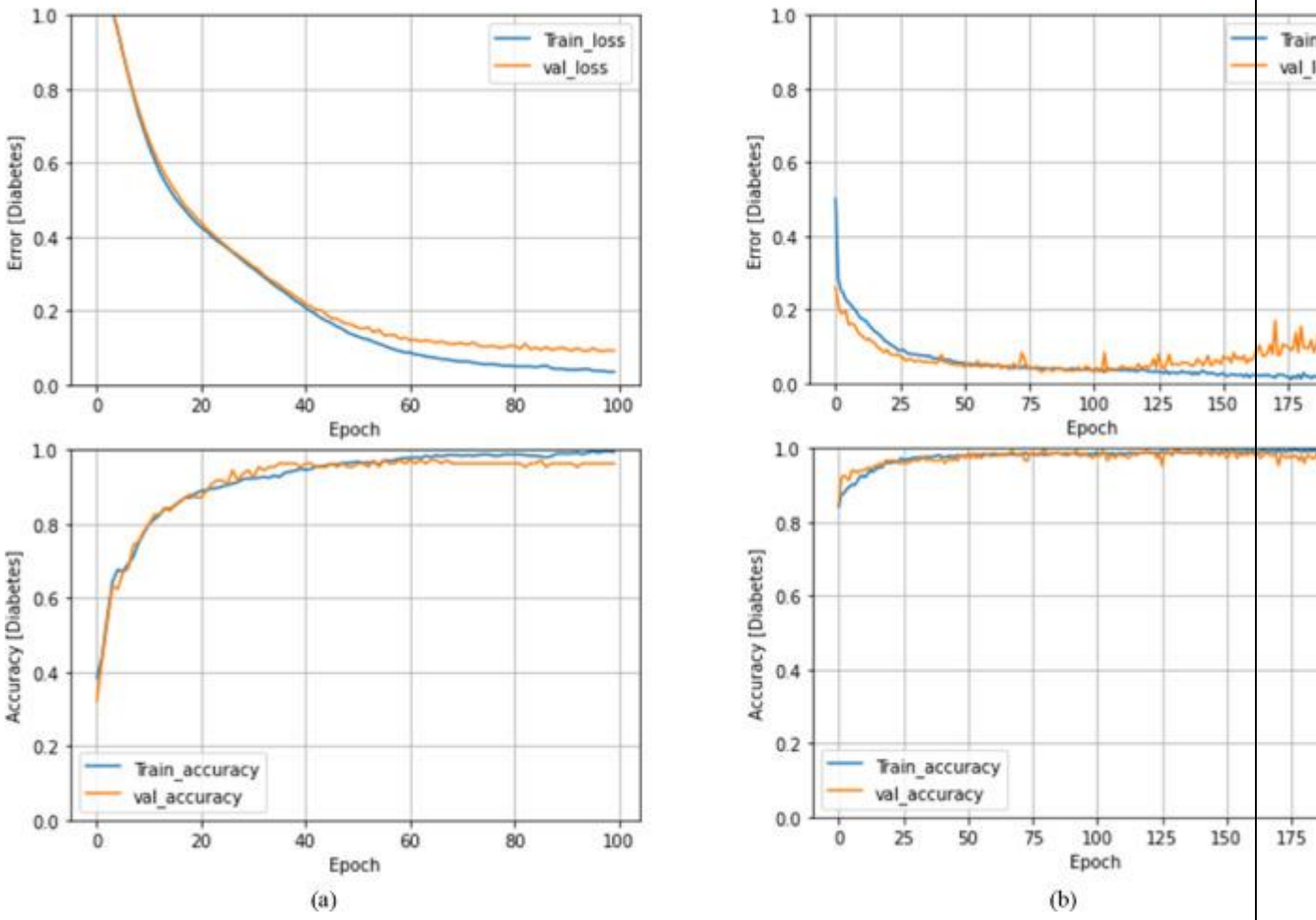
Data	Model	Precision	Recall	F1-Score	Train Acc(%)	Train Loss	TestAcc	TestLoss
PIMA	FS+MVI+2GDNN	95.156	94.495	94.504	99.802	1.000	94.495	0.152
	FS+MVI+O2GDNN	**97.342**	**97.245**	**97.255**	**99.012**	**0.018**	**97.248**	**0.042**
LMCH	FS+2GDNN	97.348	96.667	96.965	98.714	3.600	96.667	2.151
	FS+O2GDNN	**97.281**	**97.333**	**97.265**	**99.571**	**0.788**	**97.333**	**13.781**

FS – feature selection, MVI – missing value imputation.

Table 10. Comparison of the proposed methods with the state-of-the-art.

Empty Cell	Refs.	Year	FS	MVIM	Classifier	Precision (%)	Recall (SN) (%)	SP (%)	F1-S (%)	Train Acc (%)	Test Acc (%)
NN based Models	[10]	2018	PCA + mRMR	Remove missing values	NN	–	79.42	75.08	–	–	77.25
	[9]	2019	PCA Median Value		ANN	–	75.00	29.00	–	–	75.7
					Deep Learning	95.22	98.46	99.29	96.81	–	**98.07**
	[6]	2021	–	Median value	ANN	**98.00**	**98.00**	**99.00**	**98.00**	–	–
	[7]	2021	Pearson Correlation	Mean Values	MLP	–	–	–	–	78.96	88.57
	Our proposed (2GDNN+O2GDNN)					97.342	97.245	97.255	97.351	**99.012**	97.248
CCML based Models	[10]	2018	PCA + mRMR	Remove missing values	RF	–	74.58	79.85	–	–	77.21
	[17]	2018	RF	Group Median	RF	–	95.96	79.72	–	–	92.26
	[16]	2019	–	NB	RF	80.60	85.40	–	83.00	–	87.10
	[9]	2019	PCA	Median Value	RF	–	74.00	29.00	–	–	75.70
	[15]	2019	–	Median	Stacked models	–	96.10	79.90	88.80	–	83.80
	[14]	2020	Correlation Based	Mean Value	Ensemble Methods	84.20	78.90	93.40	–	–	–
	[12]	2020	–	Mean Value	RF	90.00	79.41	79.07	–	–	76.54
					XGBoost	90.53	84.31	79.06	–	–	77.54
	[13]	2020	–	–	RF & Others	98.00	95.57	–	97.73	–	**98.48**

Empty Cell	Refs.	Year	FS	MVIM	Classifier	Precision (%)	Recall (SN) (%)	SP (%)	F1-S (%)	Train Acc (%)	Test Acc (%)
	[11]	2020	Step Forward + PCA	Mean Values	RF & Others	83	82	–	–	–	77.61
	[7]	2021	Pearson Correlation	Mean Values	RF	77.90	77.10	–	77.40		
					KNN	80.40	79.40	–	79.80	–	79.42
	[6]	2021	–	Median	RF	85.00	85.00	81.00	85.00	–	–
					GB	86.00	87.00	79.00	87.00	–	–
	Our proposed (RF + ORF)					**98.119**	**97.931**	**97.238**	**97.954**	**98.620**	97.931
Empty Cell						**100**	**100**	**100**	**100**	**100**	**100**



(a) (b)

4.4. Comparison with the state-of-the-art

The comparison of our work with the state-of-the-art focuses solely on the most recent literature and particularly where data pre-processing methods were a substantial component of the reported work. The comparison will be for the NN-based and the CML-based diabetes prediction methods with the PIMA Indian dataset. From , the NN-based models show interesting performance gain in accuracy from 75.70% in 2018 to 98.07% in 2020 with the PIMA Indian dataset. A close comparison shows that our proposed ML framework with

2GDNN achieved a performance gain of 21.99%, 21.35%, −0.82%, 8.68% when compared to the work and respectively. Also, the proposed ML framework with RF achieved a performance gain of 20.71%, 5.67%, 10.83%, 22.2%, 14.13%, 21.4%, 20.4%, −0.55%, 20.32% and 18.54% in comparison to the work in and . Interestingly, the difference between our proposed ML framework with RF model test accuracy and the training accuracy is 1.04% which suggests the model is not overfitting. Though the method in performed better in terms of accuracy, ours outperformed in terms of F1-score by 0.224%.

5. Limitation and future work

The PIMA Indian diabetes dataset contains information of 768 women from a population near Phoenix, Arizona, in the USA. The dataset can be assumed to yield gestational diabetes information because there are pregnant women represented. On the other hand, the LMCH dataset comprises 1000 patients of Iraqi national patients, and though a more recent dataset it still does not address some of the limitations of the PIMA Indian dataset as only adult male and female patients information are presented. As such, there is a need for a representation that cuts across men, women (who are either pregnant or not), as well as children, and especially people of African descent, who are more at risk of developing diabetes. This is to enable the proposed model to generalize well to a wider diabetes population. To address this limitation, we propose to expand the scope of the study beyond the PIMA Indian and LMCH datasets and engage in unbiased diabetes data collection. Then, investigate the generalizability of the proposed ML framework in comparison to other machine learning algorithms on the collected data. The goal will be to develop a healthcare solution that meets the individualized diabetes diagnosis need of patients, irrespective of gender, age, or race.

6. Conclusion

In this paper, we proposed a robust machine learning framework to improve the performance of diabetes prediction using the PIMA Indian and LMCH datasets. The framework incorporated data preprocessing approaches, Spearman correlation, and polynomial regression, from a perspective that strengthens their performance. The proposed framework works for the SVM, RF, and our proposed 2GDNN model and shows well to- address the diabetes classification problem. This was demonstrated by the outstanding classification accuracy of 97.931% and 100% achieved on the PIMA Indian dataset. Similarly, a 97.333% accuracy was achieved on the LMCH dataset. These performances rank comparably to the state-of-the-art performance for the NN-based models and best for CML-based models. Therefore, it can be stated that the proposed framework presented a robust model for diabetes mellitus prediction and diagnosis