A Comparative Study to Predict Diabetes using Machine Learning Techniques

Arpita Das, Shanta Kumar Das, Dola Das and Kazi Md. Rokibul Alam
Department of Computer Science and Engineering
Khulna University of Engineering & Technology, Khulna-9203, Bangladesh
Email: das1607067@stud.kuet.ac.bd, das1607068@stud.kuet.ac.bd,
dola.das@cse.kuet.ac.bd, rokib@cse.kuet.ac.bd

2021 International Conference on Science & Contemporary Technologies (ICSCT) | 978-1-6654-2132-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICSCT53883.2021.9642493

***Abstract*—In the present world, diabetes can be a life-threatening disease. If it is not treated timely, it can lead to different diseases like kidney damage, stroke, nerve damage, etc. But diabetes can be easily controlled if it is predicted as early as possible. In this paper, we proposed a system that can predict if a person has diabetes or not. The experiment was done with a dataset collected from the UCI machine learning repository having 768 patients and 8 numeric features of each. Best features were selected using genetic algorithm (GA). k-fold cross-validation was used for splitting the dataset. After that, K-nearest neighbor (KNN), Multi-layer perceptron (MLP), Deep Neural Network (DNN), and Naive Bayes (NB) classifiers were applied on both selected datasets using GA and initial dataset (8 features) to find out which dataset gives the best results. Finally, we compared the accuracy values of these classifiers with each other. It showed that classifiers using selected features using GA showed better results than classifiers using initial features. Using the dataset selected by GA, KNN showed accuracy around 93.33%, DNN around 77.27%, MLP around 74.92%, and NB around 74.89%. We found that KNN gave comparatively better results than other classifiers.**

***Index Terms*—Diabetes, k-nearest neighbor (KNN), deep neural network (DNN), multi-layer perceptron (MLP), naive bayes (NB).**

I. Introduction

According to a report presented by the World Health Organization (WHO), around 422 million people worldwide have diabetes. Every year, approximately 2-5 million people die because of having diabetes and by 2045, this number can increase up to 629 million [1]. There are basically two types of diabetes, type1, and type2 where type1 represents only 5 to 10% of all cases and type2 represents 90% of all diabetes cases [2].

Diabetes is a disease that occurs due to the increment of blood glucose, mainly called blood sugar level in our body. Blood sugar is the main energy source of our body to conduct our daily life work and this sugar comes form the food we eat everyday. High level of sugar in our body shows different symptoms like frequent urination, feeling thirsty, increased hunger. If it is not medicated in the proper time, it will lead to many difficulties, even death [3]. The normal sugar level of the human body is less than 140 mg/dL or 7.8 mmol/L. If the range is between 140mg/dl and 199mg/dl, then it is called pre-diabetes and if is is more than 200mg/dl after two hours from taking foods then it is considered diabetes [4]. Prediction of diabetes in a human body in an early stage can reduce the risk by taking proper treatment.

Different machine learning algorithms and deep learning algorithms are very helpful and have already been used in the prediction of diabetes. In this research, we used a dataset named Pima Indians Diabetes Dataset (PIDD) from the UCI machine learning repository [5] containing 768 patients each having 8 features to predict diabetes in a human body. GA was used to select the best features from 8 features of PIDD and 4 features were found as the best features. After that, DNN, MLP, KNN, and NB classifier algorithms were applied on both the selected feature set and initial feature set from PIDD. Then the performance of the algorithms was compared with each other using accuracy values. We tried to find out the most effective algorithm which can predict the diabetes of a person properly. Also, we tried to find out if selected features worked better than the initial dataset of PIDD to predict if a patient has diabetes or not. Finally, we found KNN using the selected feature set by GA works much better than other algorithms we compared with.

So, our contribution to this paper is:

1) Using a Genetic algorithm to the PIDD to get the best features to predict whether a person having diabetes or not.

2) Comparing the results using the selected feature set and initial feature set.

3) Applying newly proposed models like KNN, DNN, MLP, and NB for prediction and comparing their performance with accuracy and receiver operating characteristic (ROC).

4) Comparing the results, we found that KNN works much better than other models.

The following sections give a proper explanation of our proposed system. Section II explains some of the related works in this field done previously. section III describes the overall flowchart of the system's working process, dataset pre-processing, used models in the dataset. Section IV analyzes the results gained from different models of the paper. Finally, section V discusses the overall system processes as well as the future plan of this study.

II. Related Works

There have been so many studies on this PIDD with the help of Machine Learning and it is found that no algorithm works exceptionally perfect.

978-1-6654-2132-4/21/$31.00 ©2021 IEEE

In paper [6] Naive Bayes was proposed as a classifier. They also used Genetic Algorithms for feature selection purposes. Their proposed system has ended up with almost 75.6% train accuracy and 78.6% test accuracy. In paper [7], they worked with a very general regression neural network for diabetes prediction with maximum accuracy of 86% for the J Rip algorithm. The authors of the paper [8] worked WEKA decision tree to predict type2 diabetes with the dataset. They neglected the other 7 features available in the dataset and considered only one feature their main attribute which is the Plasma Insulin attribute.

Aiswarya et al. [9] aimed at finding solutions by using Decision Tree and Naive Bayes to predict diabetes with investigation and examination with the patterns available in the data. It is stated that their Correctly Classified Instances is almost 76%, but it can be improved by using the feature selection algorithm, which we have incorporated in our research.

In paper [10], the authors developed modified radial basis functional neural networks (MRBF) as a classifier for the prediction of diabetes for patients. It is a supervised machine learning model. Normally, in the medical field, predictions must have much better accuracy levels and an accuracy level above 85% is considered as a good prediction for early detection of diabetes. They found maximum accuracy of 78.8%. In paper [11], they used discriminant analysis, Support Vector Machine with 10-fold cross-validation for classification and to calculate accuracy and it achieved 82.05%. The authors of the paper [12] worked with b-coloring technique which is mainly related to the field of clustering analysis. It is used as a supervised machine learning model to predict diabetes diseases.

Hence we proposed a system that is reliable, faster, and more accurate system to give a good probability analysis of a patient having diabetes. The accuracy using some algorithms both with applying Genetic algorithms and without applying Genetic algorithms on PIDD is shown in the paper.

III. Methodology

This section describes the data collection process, data preprocessing techniques, and model construction in brief. Our entire methodology consists of 5 steps -

1. Data Collection: The Pima Indians Diabetes Dataset was collected from the UCI machine learning repository.

2. Feature Selection: Among the eight features we used Genetic Algorithm (GA) to select the best features.

3. Missing values Imputation: We filled up the missing values with the mean of non-missing values.

4. Cross-Validation: Before training and testing the model, we used 20-fold cross-validation to split the dataset.

5. Models Construction: Here we have used 4 classifiers -

- K-Nearest Neighbour (KNN)
- Deep Neural Network (DNN)
- Multi-layer Perceptron (MLP)
- Naive Bayes(NB)

We tested as well as trained our model both for feature set selected with and without GA. Fig. 1 shows the overall proposed methodology of our system.

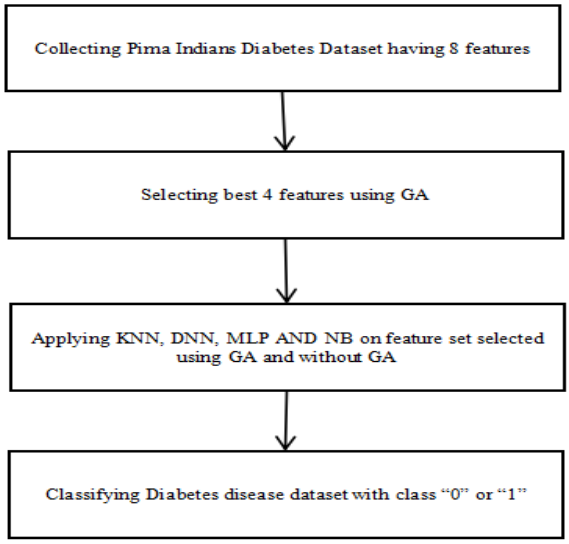

Fig. 1. Proposed Methodology of the System.

A. Data Collection

The Pima Indians diabetes dataset was downloaded from the UCI machine learning repository. The dataset has 768 samples data points each of which has 8 numeric attributes and a binary class for prediction. The collected dataset is plotted in 2D and 3D visualization by implementing Principal Component Analysis (PCA) in the dataset. The principal components(PC) of PCA are created by calculating the eigenvectors and eigenvalues for each of the PC from the dataset after standardization where the first PC contains most of the information or the largest amount of variance, the second one contains the most information after the first one and so on. Fig. 2 represents the 2D visualization with 2 PC and fig. 3 represents the 3D visualization with 3 PC and total explained variance of 97.59% which is calculated by summing up the variance of every PC and dividing by the total variance of the dataset. In both figures, blue color represents the data point for not having diabetes and yellow color represents the data point for having diabetes. Table I shows the attribute information called the features for the prediction of diabetes of the dataset [5].

B. Feature Selection:

Among the eight features of the dataset, we needed to select suitable features for better prediction. We used GA which is considered one of the best algorithms for feature selection. Fig. 4 depicts the states for the feature selection process with GA. Here, we assigned a fitness function to different feature subsets and done several processes like selection, crossover, and mutation to find the best feature set in an iteration. Finally, we gave a stopping criteria or threshold value to stop the iteration with best feature set.

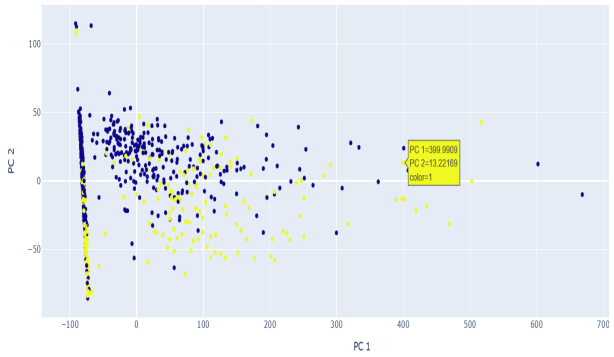

Fig. 2. 2D visualization of dataset using PCA.

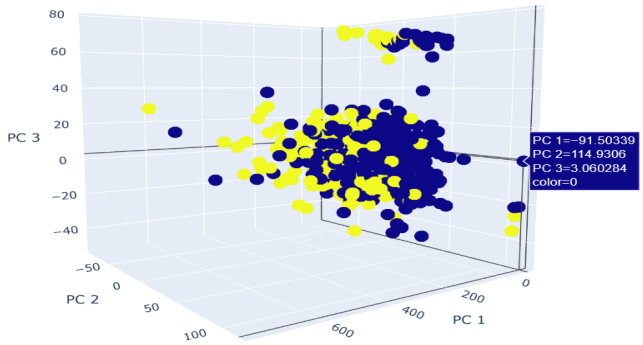

Fig. 3. 3D visualization of dataset using PCA.

C. Missing Values Imputation:

In the dataset, 763 missing values were found from 6,144 values in the feature set. We filled up the certain missing values of a particular feature column with the mean of non-missing values available in that column as missing values don't give us the correct result. This technique is called Missing Values Imputation. Finally, we standardized the dataset.

D. Cross Validation:

Before training and testing the model we used 20-fold cross-validation to split our dataset.

TABLE I
FEATURES OF THE PIDD DATASET.

Features of the dataset
1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function
8. Age (years)

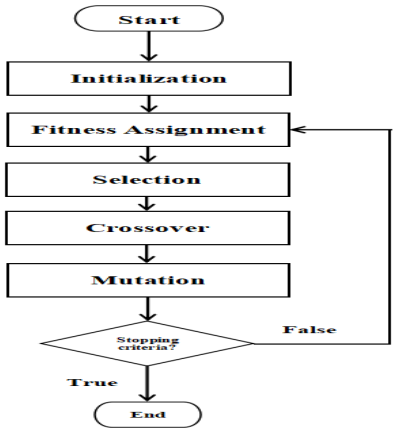

Fig. 4. Feature selection process with GA.

E. Models Construction:

1) KNN: KNN is a supervised machine learning algorithm. It is used for regression as well as classification prediction problems. To implement KNN, data points need to be transformed into feature vectors. First of all the algorithm finds the distance between the mathematical values of these points then works on it. We can find this distance with equation-

$$distance(q,p) = \sqrt{(q_1-p_1)^2+(q_2-p_2)^2+...+(q_n-p_n)^2} \quad (1)$$

$$= \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2} \quad (2)$$

2) DNN: DNN is mainly a Feed-Forward Networks(FFNNs) classifier. In DNN, data flows from the input layer to the next output layer. The paths between the layers go only in the forward direction, not in a backward direction. Any node doesn't touch other nodes again. Fig. 5 represents the architecture of DNN with our proposed layers.

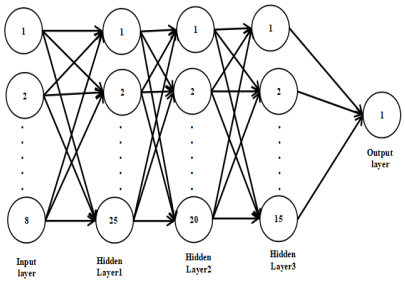

Fig. 5. DNN architecture with proposed layers.

3) MLP: MLP is also an FFNN classifier. But it uses back-propagation which is the same as a supervised learning technique. Nodes in the input layer are mainly the features of the input data. All other nodes map inputs to the output layer created by a linear combination using the input data with the node's weight values, bias values, and an activation function.

4) NB: Naive Bayes method is a supervised machine learning model which works based on the Bayes' theorem.

IV. Experimental Results

As we said previously, we worked with 4 machine learning models that are applied to the dataset. Then, GA is applied to each of the algorithms for selecting the best features from the dataset. The results of the algorithms are compared for both test and train datasets. These results are compared before applying GA and after applying GA. Among the algorithms, we compared the performance using Accuracy and ROC score of each of them with the result of test dataset after applying 20-fold cross-validation. We also calculated precision, recall, F-score, kappa statistics, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Relative Absolute Error (RAE), Root Relative Square Error (RRSE) values for performance analysis with confusion matrix, and ROC curve. The ROC curve mainly shows the true positive rate with the false positive rate and ROC score is mainly the area under the ROC curve. It's values normally lie between 0.5 and 1.0. Here, 0.5 means a bad classifier, and 1.0 means a very good classifier. The straight line in the ROC curve denotes random performance by a random classifier with an equal error rate. The correct detection (accuracy) of a person having diabetes using the dataset feature values follows the equation [13]-

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

Here,
TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative .
The precision, recall and f-score is also calculated using the following equations-

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6}$$

A. Result of K-Nearest Neighbors model (KNN)

Among the four models, KNN gave a better result for the system. The results of our system, each time applied with 20-fold cross-validation for both test set and train set are shown using Table II with the comparison before applying GA and after applying GA. The confusion matrices for both the training set and test set with GA as well as without GA are also shown in Table III. The ROC graph for test set is also displayed in Fig 6.

B. Result of Deep Neural Network (DNN)

DNN is applied in PIDD dataset with 20-fold cross-validation. The results of our system using DNN model for both test set and training set are shown using Table IV with the comparison before applying GA and after applying GA. The confusion matrices for both the training set and test set with GA as well as without GA are also shown in Table V. The ROC graph for test set is also displayed in Fig 7.

TABLE II
Results of KNN with as well as without GA for PIDD dataset.

	KNN without GA		KNN with GA	
	Test	Train	Test	Train
Accuracy	86.67%	80.61%	93.33%	79.55%
Precision	0.857	0.745	.953	0.728
Recall	0.857	0.670	0.857	0.655
F-score	0.857	0.706	0.923	0.689
Kappa statistics	0.732	0.562	0.865	0.538
ROC score	0.866	0.774	0.929	0.763
MAE	0.133	0.194	0.067	0.205
RMSE	0.365	0.440	0.258	0.452
RAE	0.268	0.428	0.134	0.451
RRSE	0.732	0.925	0.518	0.950

TABLE III
Confusion matrix for KNN without GA and with GA.

	Test		Train		Classified as
	a	b	a	b	
KNN without GA	7	1	432	60	a = tested_negative
	1	6	86	175	
KNN with GA	8	0	428	64	b = tested_positive
	1	6	90	171	

C. Result of Multi-layer Perceptron (MLP)

MLP is an important type of classifier for prediction. It works mostly in same way as DNN but there is one difference, that is, with increasing layer its accuracy or performance decreases. It works best between 3/5 layers. In our system, we worked with a 3-layer model where we took 20-fold cross-validation for splitting. The results of our system using MLP model for both test set and training set are shown using Table

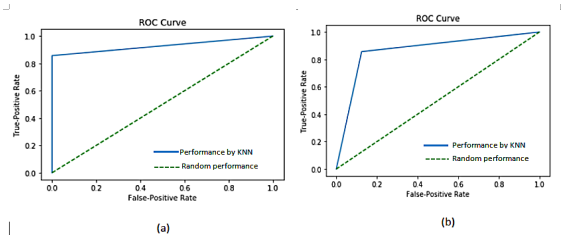

Fig. 6. ROC graph for KNN (a)with GA and (b)without GA.

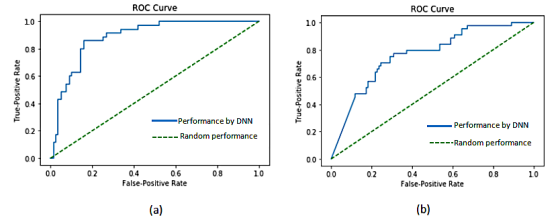

Fig. 7. ROC graph for DNN (a)with GA and (b)without GA.

TABLE IV
RESULTS OF DNN WITH AS WELL AS WITHOUT GA FOR PIDD DATASET.

	DNN without GA		DNN with GA	
	Test	Train	Test	Train
Accuracy	72.07%	88.59%	77.27%	87.95%
Precision	0.596	0.821	0.731	0.803
Recall	0.585	0.845	0.644	0.883
F-score	0.857	0.833	0.685	0.841
Kappa statistics	0.732	0.865	0.865	0.562
ROC score	0.866	0.929	0.748	0.774
MAE	0.133	0.067	0.106	0.121
RMSE	0.365	0.258	0.174	0.440
RAE	0.268	0.134	0.003	0.428
RRSE	0.732	0.518	0.005	0.925

TABLE V
CONFUSION MATRIX FOR DNN WITHOUT GA AND WITH GA.

	Test		Train		Classified as
	a	b	a	b	
DNN without GA	80	21	185	19	a = tested_negative
	22	31	16	87	
DNN with GA	81	14	172	24	b = tested_positive
	21	38	13	98	

VI with the comparison before applying GA and after applying GA. The confusion matrices for both the training set and test set for both with GA and without GA are also shown in Table VII. The ROC graph for test set is also displayed in Fig 8.

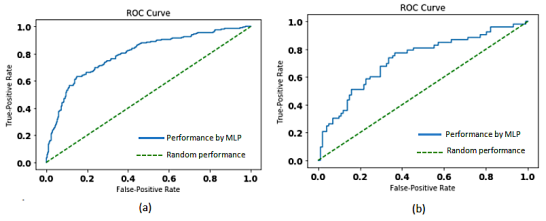

Fig. 8. ROC graph for MLP (a)with GA and (b)without GA.

TABLE VI
RESULTS OF MLP WITH AS WELL AS WITHOUT GA FOR PIDD DATASET.

	MLP without GA		MLP with GA	
	Test	Train	Test	Train
Accuracy	77.04%	71.44%	74.92%	72.15%
Precision	0.615	0.821	0.740	0.797
Recall	0.453	0.845	0.437	0.274
F-score	0.522	0.833	0.549	0.408
Kappa statistics	0.325	0.865	0.391	0.279
ROC score	0.652	0.929	0.677	0.618
MAE	0.286	0.067	0.251	0.278
RMSE	0.650	0.258	0.642	0.622
RAE	144.33	0.134	556.05	521.15
RRSE	16.99	0.518	33.35	32.28

TABLE VII
CONFUSION MATRIX FOR MLP WITHOUT GA AND WITH GA.

	Test		Train		Classified as
	a	b	a	b	
MLP without GA	86	15	350	49	a = tested_negative
	29	24	92	123	
MLP with GA	366	33	384	15	b = tested_positive
	121	94	156	59	

TABLE VIII
RESULTS OF NB WITH AS WELL AS WITHOUT GA FOR PIDD DATASET.

	NB without GA		NB with GA	
	Test	Train	Test	Train
Accuracy	72.72%	78.13%	74.89%	78.67%
Precision	0.709	0.679	0.689	0.678
Recall	0.583	0.598	0.519	0.656
F-score	0.640	0.634	0.592	0.667
Kappa statistics	0.424	0.486	0.415	0.504
ROC score	0.706	0.735	0.696	0.75
MAE	0.273	0.213	0.251	0.219
RMSE	0.522	0.462	0.501	0.468
RAE	0.561	0.495	0.551	0.492
RRSE	1.059	0.997	1.050	0.992

D. Result of Naive Bayes (NB)

The fourth model, we used in our system is NB, which is also applied with 20-fold cross-validation algorithm for splitting the dataset of the system. The results of our system using an NB model for both test set and training set are shown using Table VIII with the comparison before applying GA and after applying GA. The confusion matrices for both the training set and test set for both with GA and Without GA are also shown in Table IX. The ROC graph for the test set is also displayed in Fig 9.

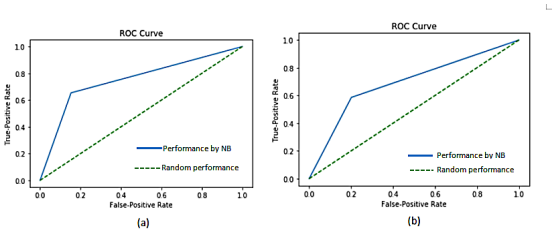

Fig. 9. ROC graph for NB (a)with GA and (b)without GA.

As already mentioned, we used GA in every model of our

TABLE IX
CONFUSION MATRIX FOR NB WITHOUT GA AND WITH GA.

	Test		Train		Classified as
	a	b	a	b	
NB without GA	112	23	225	33	a = tested_negative
	40	56	47	70	
NB with GA	131	19	211	39	b = tested_positive
	39	42	43	82	

TABLE X
COMPARISON AMONG MODELS WITH ACCURACY AND ROC SCORE.

	Accuracy	ROC score
KNN	93.33%	0.865
DNN	77.27%	0.748
MLP	74.92%	0.677
NB	74.89%	0.696

system. The reason for this is to select important features of the dataset for better performance of our system. The above tables also showed the comparison of the result of different models before applying GA and after applying GA. Most of the time, after applying GA, we got better results with less error which indicates better performance of each model. We used k-fold cross-validation algorithm for splitting dataset which is for increasing the performance measurements of each model.

E. Robustness performance of the models

Our prediction of having diabetes of a person doesn't have higher accuracy for each of the model, but it is much better in our working environment. The same dataset, as well as same models, are used for prediction and so the performance of KNN, DNN, MLP, and NB was compared based on accuracy and ROC score of the test set after training the dataset with GA. Table X shows the comparative study of the models with test data results after applying GA to train dataset. Here, among the four algorithms, KNN gives a higher level of accuracy with fewer errors. Comparing with other models, 93.33% (KNN) is achieved after applying genetic algorithm with k-fold cross validation where, DNN, MLP, and NB gave 77.27%, 74.92%, and 74.89%. The performance is also evaluated using the ROC score which is highest for KNN (0.865) where, the other algorithms have 0.748, 0.677, and 0.696. As we know, with increasing accuracy and ROC score, the performance of an algorithm also increases. So, in this paper, we can see, KNN works best for predicting diabetes with the highest accuracy and ROC score. However, we can compare our work with other paper [6], which used NB as a classifier and we followed the same procedure for our 4 models like their work in the paper. The accuracy of our KNN model (93.33%) is much higher than NB, proposed by them though the accuracy using NB (78.6%) is pretty higher than ours (74.89%). Again, the ROC value of their proposed model (0.844) is less than our KNN model's ROC score value (0.865).

F. Discussion

In this paper, KNN, DNN, MLP, and NB algorithms are applied to PIDD dataset with search-based optimization technique called genetic algorithm (GA). GA is applied to get the best-suited features for prediction. However, it can be said that KNN gives higher performance among the four algorithms.

V. CONCLUSIONS

Early prediction of diabetes of a person is very important. In this paper, different machine learning classification models are applied for the prediction of diabetes. KNN, DNN, MLP, and NB are applied with search-based optimization techniques for better accuracy of prediction. Performances of each model with another is analyzed with the prediction accuracy and ROC score. The analysis and comparison between the models give us a clear understanding of how good or bad the model works for prediction using the dataset.

In the future, more models will be used for prediction of diabetes. Prediction of other different diseases can also be analyzed with the present system. Other search-based optimization techniques can also be used for selecting the best feature and comparison.

VI. ACKNOWLEDGEMENT

The authors wholeheartedly like to thank the Department of Computer Science and Engineering (CSE), Khulna University of Engineering & Technology (KUET) to facilitate the work of our research.

REFERENCES

[1] "Diabetes prediction using machine learning algorithms," *Procedia Computer Science*, vol. 165, pp. 292 – 299, 2019, 2nd International Conference on Recent Trends in Advanced Computing ICRTAC - DISRUP - TIV INNOVATION , 2019 November 11-12, 2019.

[2] A. Yahyaoui, A. Jamil, J. Rasheed, and M. Yesiltepe, "A decision support system for diabetes prediction using machine learning and deep learning techniques," in *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, 2019, pp. 1–4.

[3] P. Sonar and K. JayaMalini, "Diabetes prediction using different machine learning approaches," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 2019, pp. 367–371.

[4] "Diabetes," https://www.mayoclinic.org/diseases-conditions/diabetes/diagnosis-treatment/drc-20371451, 2018, [Online; accessed 14-January-2020].

[5] "Pima indians diabetes database," https://www.kaggle.com/uciml/pima-indians-diabetes-database, 2016, [Online; accessed 07-October-2016].

[6] D. K. Choubey, S. Paul, S. Kumar, and S. Kumar, "Classification of pima indian diabetes dataset using naive bayes with genetic algorithm as an attribute selection," in *Communication and Computing Systems: Proceedings of the International Conference on Communication and Computing System (ICCCS 2016)*, 2017, pp. 451–455.

[7] R. Manimaran and V. Muthuraman, "Prediction of diabetes disease using classification data mining techniques," *International Journal of Engineering and Technology*, vol. 9, 11 2017.

[8] N. Sravani, V. Ravuri, J. Vinita, k. Nikath, and s. Ramsubbareddy, "Diabetic detection using data mining techniques," vol. 8, pp. 704–711, 11 2018.

[9] A. Iyer, J. S, and R. Sumbaly, "Diagnosis of diabetes using classification mining techniques," *International Journal of Data Mining Knowledge Management Process*, vol. 5, no. 1, p. 01–14, Jan 2015. [Online]. Available: http://dx.doi.org/10.5121/ijdkp.2015.5101

[10] G. Magudeeswaran and D. Suganyadevi, "Forecast of diabetes using modified radial basis functional neural networks," *the Proceedings on Research Trends in Computer Technologies*, pp. 35–9, 2013.

[11] M. Nilashi, O. Bin Ibrahim, A. Mardani, A. Ahani, and A. Jusoh, "A soft computing approach for diabetes disease classification," *Health Informatics Journal*, vol. 24, no. 4, pp. 379–393, 2018.

[12] A. G. Karegowda, M. Jayaram, and A. Manjunath, "Cascading k-means clustering and k-nearest neighbor classifier for categorization of diabetic patients," *International Journal of Engineering and Advanced Technology*, vol. 1, no. 3, pp. 147–151, 2012.

[13] N. Tabassum, D. Das, and A. Das, "A cost-effective multisensor based framework to assist visually disable person," in *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)*, 2020, pp. 47–52.