Turkish Journal of Computer and Mathematics Education *Vol.12 No.3 (2021), 3957-3965*

Research Article

A Model-Based Approach for an Early Diabetes Prediction Using Machine Learning Algorithms

Abrar M. Alajlan

Self-Development Skills Department, Common First Year Deanship, King Saud University, KSA
Email: aalajlan1@ksu.edu.sa

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; published online: 5 April 2021

Abstract:

Diabetes is a chronic serious health condition that occurs when the pancreas is no longer produces insulin, or the human body cannot beneficially use the insulin it produces. Recognizing and predicting it at an early stage is the first step towards preventing its progression. With the advent of information technology and its emergence in the medical and healthcare sector, diabetes cases and symptoms are well documented. Knowledge can be discovered for predictive purposes through machine learning and data mining techniques. This work concentrates on evaluating the dataset through classification analysis by utilizing Decision tree, Adaptive boosting, and K-nearest neighbor's algorithms. Thus, a faster model of predicting diabetes is introduced, where the aim is to develop the best model that derives the conclusion on an early detection of undiagnosed diabetes.

Keywords: machine learning, diabetes prediction, decision tree, adaptive boosting, K-nearest neighbors, supervised classifications

1. Introduction

Diabetes is a chronic serious health condition that occurs when the pancreas is no longer produces insulin, or the human body cannot beneficially use the insulin it produces. According to WHO report, diabetes has affected more than 463 million people worldwide, mostly were women, and estimated to rise to 10.2% by 2030 and 10.9% by 2045. Indeed, about half a billion people are diagnosed with diabetes worldwide, and the number is expected to increase by 255 in 2045[1].

With the advent of information technology and its continuing emergence in the medical and healthcare sector, diabetes cases and symptoms are well documented. Knowledge discovery for predictive purposes can be made through machine learning and data mining techniques [2][3]. These techniques gained strength by the reason of their capability to handle and integrate a large amount of data from different sources, where in this case, is an early detection of undiagnosed diabetes.

Several studies are conducted for diagnosing diabetes using different classification algorithms of machine learning approaches [4]. This work focuses on reducing the complications of diabetes through early prediction. A person with diabetes has specific characteristics that cause the disease and vary from one type to another, such as age, glucose level, heredity, etc. The research evaluates the patterns founded in PIDD (Pima Indians Diabetes Database) through classification analysis by utilizing Decision tree, Adaptive boosting, and K- nearest neighbor's algorithms. Hence, a faster model of predicting diabetes is introduced to detect diabetes at an early stage and prevent complications.

The following is how the rest of the paper is organized: The second section summarizes the work of various machine learning models for predicting diabetics. The proposed model's methodology and background are described in Section 3. The proposed model's implementation and evaluation are presented in Section 4. The conclusion and future work are presented in Section 5.

2. Related work

Multiple prediction methods have been proposed and developed in the literature by researchers using different data mining techniques and machine learning algorithms. The improvement of prescient models for the conclusion of diabetes has been a functioning territory of research over the previous decade.

Gauri et al. in [5] predicted diabetes type, the future risk associated with it, and the treatment according to the risk level. Hadoop's system is being implemented to discover missing values in the dataset and its patterns.

Aiswarya et al. in [6] highlighted diabetes recorded in pregnant women where Decision Tree and Naïve Bayes have been implemented and compared to predict whether diabetes are recorded or not.

Hang lei et al. in [7] presented a predictive model based on demographic data and laboratory results that used Gradient Boosting Machine and Logistic Regression techniques to predict the likelihood of patients having DM (Diabetes Mellitus). Machine learning techniques such as Rpart and Random Forest were used to compare the results of these methods.

Karim et al. in [8] proposed a diabetes prediction system based on a decision tree technique, with the goal of predicting a candidate at a specific age. The results were promising; the system accurately predicts diabetes incidents based on age.

Nongyao and Rungruttikarn investigated four classification models: Decision Tree, Artificial Neural Networks, Logistic Regression, and Naive Bayes [9]. The model's robustness was improved using bagging and boosting techniques. The results of the experiments revealed that of all the algorithms used, the random forest algorithm produces the best results.

In [10], Sajida et al. used the AdaBoost and bagging ensemble techniques with the J48 decision tree to classifies patients with diabetes mellitus based on diabetes attributes and risk factors. The AdaBoost algorithm outperforms both bagging and a standalone decision tree, according to the results.

Madhavi and Bamnote proposed a classifier for diabetes detection using GP (Genetic Programming) in [11], that uses a reduced function pool of only arithmetic operators, allowing for less validation and leniency during crossover and mutation.

All previous methods focused on identifying outliers or improving diabetic prediction. Our research focuses on predicting diabetes incidents with greater accuracy at an early stage.

3. Methodology and Background

A. Model diagram

This work utilizes different machine learning algorithms to solve classification problems. Figure-1 demonstrates the proposed model diagram. The proposed model starts by transmitting the data through pre- processing techniques to clean irrelevant data and convert the raw data into a suitable form that can be used byML algorithms. When the review texts are all cleaned up, the rows sequence is changed randomly, which might affect the learning process. The dataset is divided into training and testing, 70% to 30%. The training set is partitioned into smaller sections, and the testing set is used to validate the trained model and measure overall performance.

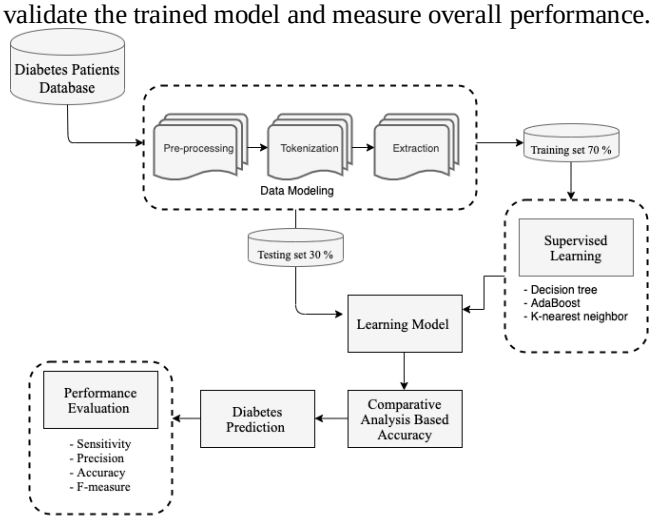

Fig.1: System architecture

B. Supervised machine learning methods

All the supervised machine learning algorithms mentioned below were installed in Python using the scikit- learn module. This work used three different machine-learning algorithms to solve the classification problem.

Decision Tree

As shown in algorithm 1, the decision tree algorithm represents rules in a hierarchical, sequential structure, where each object corresponds to a single node that provides a solution. The processes can be divided into three categories, as follows:

• Information about the data. Decision trees are useful for storing data information in a compact format.

• Categorization. Discrete values are required for the target variable.

• There will be a regress. Decision trees can be used to set the dependent (target) variable's dependence on independent (input) variables if the dependent (target) variable has continuous values. Numerical prediction tasks, for example (prediction of target variable values), fall into this category.

Algorithm 1. Decision Tree

Step 1: load the data.

Step 2: Find the best predictor in the dataset.

Step 3: Divide the training dataset into a subset that contains potential values for the best predictor.

Step 4: Make a decision tree node that contains the best predictor.

Step 5: Recursively generate new decision trees using a subset of the data created in step 3 until you reach a stage where the data cannot be classified further. Represent the class as a leaf node

Adaptive Boosting

As shown in algorithm 2, AdaBoost is a machine learning algorithm proposed by Yoav Freund and Robert Schapier. To improve the efficiency of several classification algorithms, this algorithm can be used in conjunction with them. The algorithm combines the classifiers into a "committee" to strengthen them. AdaBoost is adaptive in the sense that each classifier committee uses objects that previous committees incorrectly classified. AdaBoost is also susceptible to data noise and outliers. In comparison to other machine learning algorithms, it is less likely to be retrained.

Algorithm 2. Adaptive Boosting

Step 1: given training sample set S**,** $S = ((x_1 , y_1),\ldots\ldots, (x_n , y_n))$and a predetermined number of iterations, $x_n \in X$ is the feature vector of sample, $y_n \in \{+1, -1\}$ sample category labels.

Step 2: initializing the weights of training samples ford$_n$ = 1/m, m normalized weight for the total sample.

Step 3: by iterating to obtain the weak classifier, the number of iterations t= 1,2,…., T (T to solve a number of weak classifiers)

1-Use with weight distribution d_n the training dataset to learn, get the basic weak classifier, $h_t : x \rightarrow \{+1,-1\}$

2-Compute the h_t the classification error rate on the training dataset $\varepsilon = \sum_{n=1}^{N} d_n \, I(h_t(x_n) \neq y_n)$

3-The calculation of the weak classifier h_t weighting coefficients

$\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$ Where ε_t weighted classifier error h_t.

4-Update the sample weights for the next iteration:

$D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$ where Z_t is the normalizing parameter

Step 4: Combined weak classifier to get final strong classifier

$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

K-Nearest Neighbors

The KNN algorithm is a supervised machine learning algorithm that can be used to solve problems such as classification and regression. It classifies objects based on the feature space's nearest training data, as shown in algorithm 3. KNN is a non-paramedic learning algorithm as well as a lazy learning algorithm. There is no specialized training phase in a lazy learning algorithm, and the model uses all of the data for training during classification. Non-paramedic learning algorithm means it does not make any assumption, and the model is made up entirely of the data given to it. KNN has the following characteristics:

- All data Every data point in n-dimensional Euclidean space corresponds to a data instance.
- The target function might be discrete or exaggerated.
- Classification is done by comparing feature vectors from various points in a space region. The following procedures must be followed in order to categorize each of the test sample objects:

- Determine the distance between each object in the training sample.
- Pick k objects from the training sample with the shortest distance between them.
- The class of the object to be classified is the most common class among the k-nearest neighbors.

Algorithm 3. K-Nearest Neighbors

Step 1: load the data.

Step 2: initialize the value of K to the chosen number of neighbors.

Step 3: For each example in the data Do:

Step 3.1: Calculate the distance between test data and each row of training data using (Euclidean, Manhattan or Hamming distance)

Step 4: Sort the ordered collection of distances and indices in ascending order by the distances.

Step 5: Pick the first K entries from the sorted collection.

Step 6: Get the labels of the selected K entries.

Step 7: If regression, return the mean of the K labels

Step 8: If classification, return the mode of the K labels

C. Dataset and Data Description

The quality and quantity of data affect how good the model is and how accurate the desired result will be. Non-relevant data must be cleaned as it may reduce the accuracy measures, including missing values, extraneous data, and other data out of the common set. Here are two options: either this data is removed from the sample or replaced with dummy values. For example, missing values can be replaced with zeros. For the algorithm, obtaining approximate values is more proper than the missing ones. So, the prioritized factors should be taken into consideration for analyzing the dataset: the accuracy of the forecast or the time spent on preparing the data.

The dataset used in this study was taken from the National Institute of Diabetes and Digestive and Kidney Diseases, where all patients are females (https://www.kaggle.com/uciml/pima-indians-diabetes-database). The proposed model includes data collection and understanding the data to study the patterns and trends that predict and evaluate the results. Dataset description is given below in table 1. The diabetes dataset contains 768 recordsand 9 attributes. Medical independent predictors used to form the proposed model, which include:

Table 1. Dataset description

Predictor	Description
NP	Number of past pregnancies
Glucose	Plasma glucose concentration, 2 hours in an oral glucose tolerance test (how the bodyresponse to sugar)
Blood Pressure	Diastolic blood pressure measured in units of millimeters of mercury (mmHg)
Skin sickness	Triceps skinfold thickness gives information about the fat reserves of the body. Measured in(mm)
Insulin	Indicates 2-Hour serum insulin (mu U/ml)
BMI	Body mass index (weight in kg/(height in m^2))
Pedi	Diabetes Pedigree Function: a function that scores the likelihood of diabetes based on familyhistory
Age	Age (years)
Outcome	Class variable (0 if non-diabetic, 1 if diabetic)

Implementation and Evaluation

Decision tree, Adaptive boosting, and KNN algorithms were implemented to reduce the complications of diabetes through early prediction. All models were written in Python and its libraries such as NumPy, pandas, Matplotlib, and XGBoost while the coding in Jupyter Notebook. The implementation starts with indicating features and importing the dataset. The dataset has 8 representing features and one target value that indicates '0' for no diabetes and '1' for diagnosing diabetes.

This work used the confusion matrix to obtain evaluation metrics: accuracy, sensitivity, precision, and f_ measure. The confusion matrix is used to form the outcome of testing calculation and deliver the number of True Positive (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) cases as depicted in table 2.

Table 2. Confusion matrix

	Predicted Positive	***Predicted Negative***
Actual Positive	T_p	F_n
Actual Negative	F_p	T_n

A. Accuracy

Accuracy measures the prediction correctness for the target dataset, where the highest value is 1and the lowest value is 0.

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \quad (1)$$

B. Sensitivity

Sensitivity measures the prediction completeness of the model, where the highest value is 1, and the lowest value is 0.

$$Sensitivity = \frac{T_p}{T_p + F_n} \quad (2)$$

C. Precision

Precision measures the prediction exactness of model, where the highest value is 1, and the lowest value is 0.

$$Precision = \frac{T_p}{T_p + F_p} \quad (3)$$

D. F_measure

F_ measure is used to measure the performance in case of imbalanced data sets, in other words, it conveys the balance between sensitivity and precision, where the highest value is 1, and the lowest value is 0.

$$F_{measure} = 2\,\frac{Sensitivity * Precision}{Sensitivity + Precision} \qquad (4)$$

The experimental result shows that there is different accuracy related to different machine learning used. Corresponding classifiers performance of Decision tree, Adaboost and KNN algorithms over Accuracy arepresented in figure 2**,** 3, and 4, respectively.

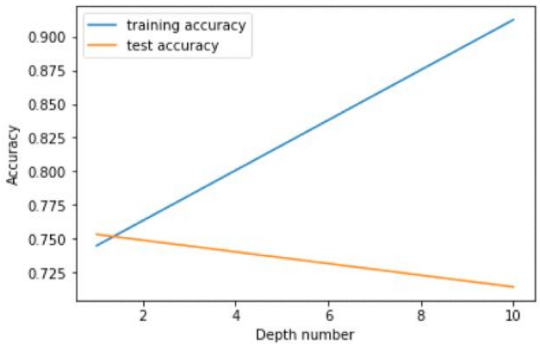

Fig. 2: The performance of DecisionTree over Accuracy

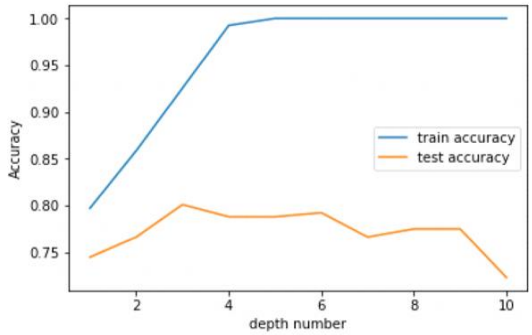

Fig. 3: The performance of Gradient boosting over Accuracy

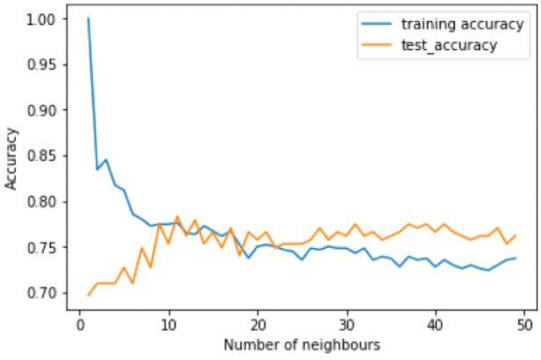

Fig. 4:

The performance of KNN over Accuracy

The classifiers' performance based on classified instances are defined in table 3.

Table 3. performance evaluation of all models

	ExecT	True D	True Non-D	Accu	Prec	Sens	F
DT	1.38	573.7	194.3	0.744	0.747	0.751	0.781
AB	2.97	**694.27**	**73.73**	**0.926**	**0.904**	**0.928**	**0.900**
KNN	12.5	592.12	175.87	0.776	0.771	0.768	0.853

Table 3 represents different performance values of all classification algorithms calculated on various measures. It is obvious that AdaBoost can predict the chances of diabetes with more accuracy as compared to other classifiers. Performances of all classifiers based on various measures are plotted via a graph in figure 5.

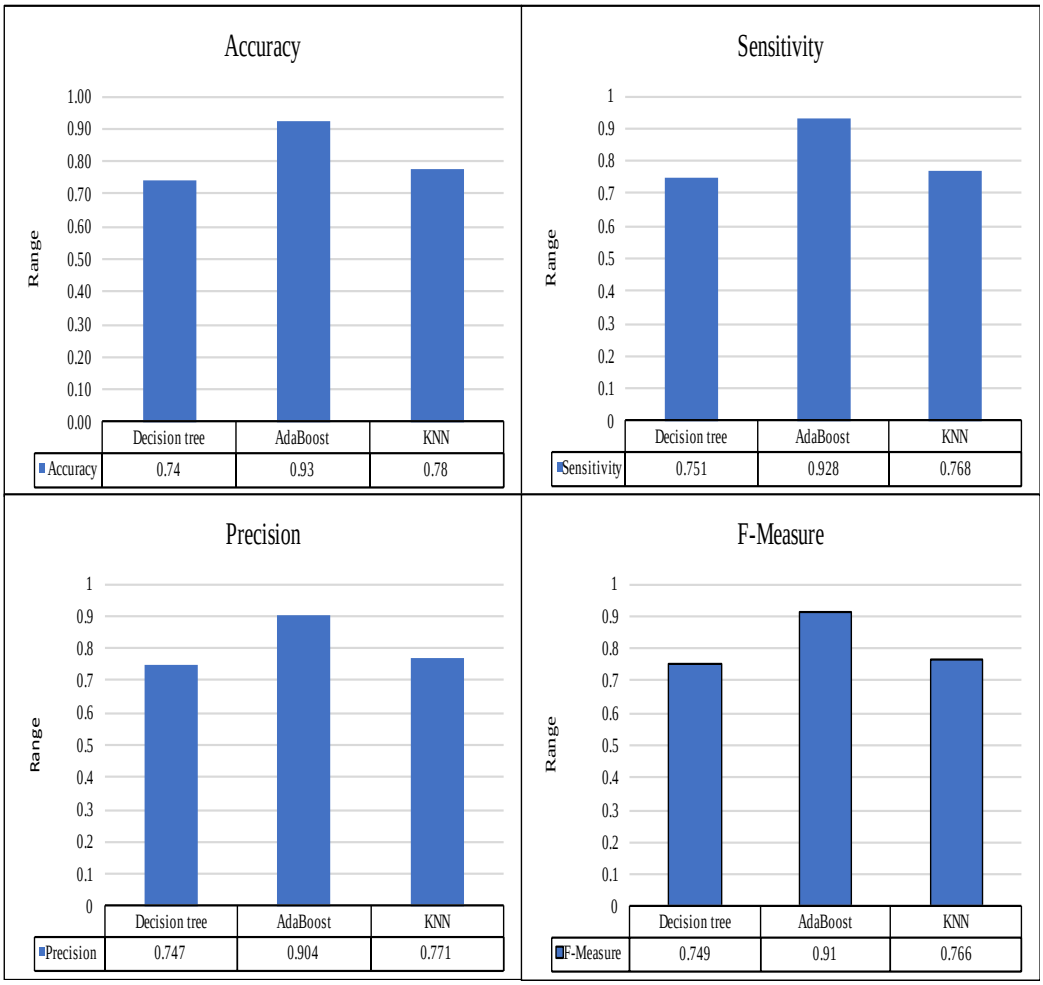

Fig. 5: Classifiers Performance on Various Measures

Area Under the ROC Curve (AUC) is also calculated in figure 6 as relevant for representing the model. The AUC takes the value between 0 and 1, where a well-performing classifier should have a high AUC value.

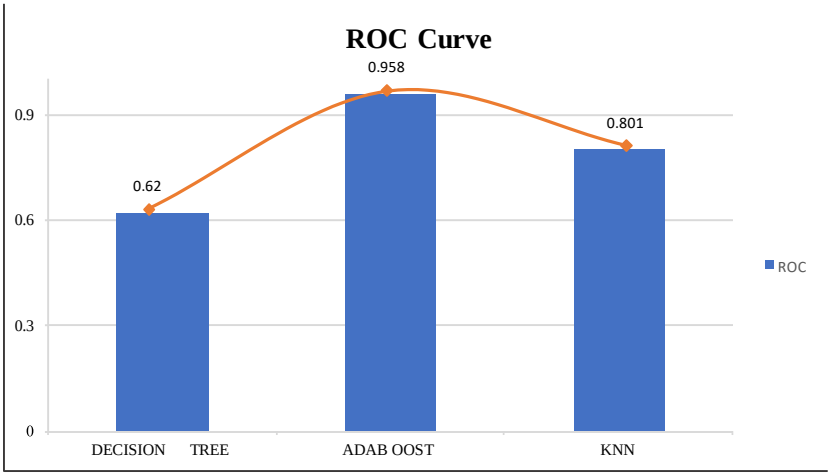

Fig. 6: ROC Area of classification Algorithms

Performance of individual algorithm is evaluated based on the number of true diabetes, and non-true diabetes records out of a total number of records as presented in figure 7.

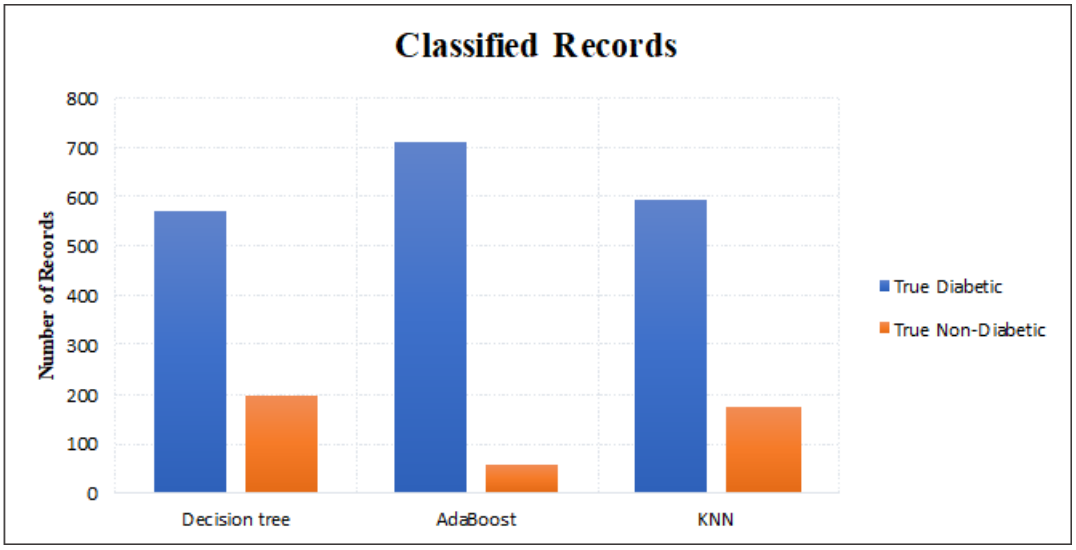

Fig. 7: True diabetes and non-true diabetes of tested algorithms

All models have a relatively small difference in error rate from the results obtained, though the percentage split of 70:30 for Adaptive boosting gives the least error rate compared to decision tree and KNN. On the other hand, decision tree has the fastest execution time but performs the worst and only classifies 74%of our dataset.

4. Conclusion and future work

This work examined the execution of three machine learning algorithms, namely, Decision tree, AdaBoost, and KNN, for early diabetic prediction. The methods evaluation measures are determined in terms of accuracy, sensitivity, precision, f-measure, and ROC curve are achieved to select the optimal features based on the correlation values. Adaptive Boosting gives the highest performance and outperformed many other supervised machine learning classification algorithms in various cases. It gives the best fit to the data concerning the diabetic and non-diabetic patients.

The future work involves presenting intelligent machine learning algorithms useful to a huge collection of a real dataset and higher accuracy percentage. More attributes can enhance performance and give a more precise predictions.

5. Author Details

Dr. Abrar Mohammed Alajlan is an Assistant Professor of Computer Science at King Saud University, where currently, she is the vice chair of Self Development Skills Department at Common First Year Deanship, KSU. Dr. Alajlan completed her Ph.D. in Computer Science and Engineering at The University of Bridgeport in 2016.

Dr. Alajlan has been recognized by the International Honor Society for the Computing and Information Disciplines Upsilon Pi Epsilon, and the Honor Society of Phi Kappa Phi. She has published several papers in influential international journals and conferences. Her fields of interest are Wireless Sensor network, artificial intelligent and real-time programming.

References

[1] Saeedi P, Petersohn I, Salpea P, et al. Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the International Diabetes Federation Diabetes Atlas, 9thedition. Diabetes Research and Clinical Practice. 2019 Nov;157:107843. DOI: 10.1016/j.diabres.2019.107843.

[2] Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., Chouvarda, I., 2017. Machine Learning and Data Mining Methods in Diabetes Research. Computational and Structural Biotechnology Journal 15, 104–116. doi:10.1016/j.csbj.2016.12.005.

[3] Kumar, P.S., Umatejaswi, V., 2017. Diagnosing Diabetes using Data Mining Techniques. International Journal of Scientific and Research Publications 7, 705–709.

[4] Dhomse Kanchan B., M.K.M., 2016. Study of Machine Learning Algorithms for Special Disease Prediction using Principal of Component Analysis, in: 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication, IEEE. pp. 5– 10.

[5] Kalyankar, G., Poojara, S., & Dharwadkar, N. (2017). Predictive analysis of diabetic patient data using machine learning and Hadoop. International Conference on ISMAC (IoT in Social, Mobile, Analytics and

Cloud) (ISMAC). doi:10.1109/I-SMAC.2017.8058253

[6] Iyer, A., S., & Sumbaly, R. (2015). Diagnosis of diabetes using classification mining techniques. International Journal of Data Mining & Knowledge Management Process, 5(1), 1- 14. doi:10.5121/ijdkp.2015.5101

[7] Lai, H., Huang, H., Keshavjee, K. et al. Predictive models for diabetes mellitus using machine learning techniques. BMC Endocr Disord 19, 101 (2019).

[8] Orabi K.M., Kamal Y.M., Rabah T.M. (2016) Early Predictive System for Diabetes Mellitus Disease. In: Perner P. (eds) Advances in Data Mining. Applications and Theoretical Aspects. ICDM 2016. Lecture Notes in Computer Science, vol 9728. Springer, Cham.

[9] Nongyao Nai-arun, Rungruttikarn Moungmai, Comparison of Classifiers for the Risk of Diabetes Prediction, Procedia Computer Science, Vol 69, 2015, pp. 132-142.

[10] Sajida Perveen, Muhammad Shahbaz, Aziz Guergachi, Karim Keshavjee, Performance Analysis of Data Mining Classification Techniques to Predict Diabetes, Procedia Computer Science,Vol 82, 2016, pp. 115- 121.

[11] Pradhan M., Bamnote G.R. (2015) Design of Classifier for Detection of Diabetes Mellitus Using Genetic Programming. In: Satapathy S., Biswal B., Udgata S., Mandal J. (eds) Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014. Advances in Intelligent Systems and Computing, vol 327. Springer, Cham