Analysis of Prediction Accuracy of Diabetes Using Classifier and Hybrid Machine Learning Techniques

Shekharesh Barik, Sambit Mohanty, Surajit Mohanty, and Debabrata Singh

Abstract In the past few years, the growth of diabetes among people became exponential. A health report tells that about 347 million of world populations are affected by diabetes. Diabetes not only affects the older person but the younger generation too. To detect diabetes at an early stage is also a big challenge. This detection will be helpful for decision-making process of medical system. Early prediction of diabetes helps us to save the human life from diabetes. A prolong diabetes leads to the risk of damage in vital organs of human body. So, early prediction of diabetes is very crucial in order to save human being from diabetes. Data analysis is concerned with finding a pattern from a large dataset. This helps us to build certain conclusion out of the available datasets. The analytical process can be done by different machine learning algorithms. This paper presents two sets of machine learning approach for prediction of diabetes. One of them is a classification-based algorithm, and the other one is a hybrid algorithm. In classification, we have taken the random forest algorithm. For hybrid approach, we have chosen XGBoost algorithm. These two algorithms were implemented and compared in order to explore the prediction accuracy in diabetes for two different machine learning approaches and got the mean score 74.10% which is better than the Random Forest algorithm.

S. Barik · S. Mohanty
Department of Computer Science and Engineering, DRIEMS (Autonomous), Cuttack, Odisha, India
e-mail: shekharesh@gmail.com

S. Mohanty
e-mail: mohanty.surajit@gmail.com

S. Mohanty
SLFS Lab, Bhubaneswar, Odisha, India
e-mail: sambitmohanty778@gmail.com

D. Singh (✉)
Department of Computer Science and Information Technology, Institute of Technical Education and Research (ITER), Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha 751030, India
e-mail: debabratasingh@soa.ac.in

D. Mishra et al. (eds.), *Intelligent and Cloud Computing*,
Smart Innovation, Systems and Technologies 153,
https://doi.org/10.1007/978-981-15-6202-0_41

Keywords Decision-making system · Data analysis · Hybrid algorithm · Random forest · XGBoost

1 Introduction

Diabetes arises when sugar level in blood increases. Nowadays, diabetes is a fast-growing disease. Most physician suggests that the main factors for diabetes are bad life style, bad diet, and lack of exercise. The other factors can be obesity, viral infection, chemical in food, environment pollution, immune reaction, bad food habit, i.e., more oil and salt intake in food. Diabetes can be classified into three categories: Type 1, Type 2, and Type 3 [1]. In Type 1 diabetes, the immune system of body destroys the important cells which produce insulin to absorb the sugar. Type 1 diabetes can attack child or adult. Type 2 diabetes is found in adults, mainly in middle aged or old aged people. Obesity is one of the causes for Type 2 diabetes; obesity is the imbalance of body mass index (BMI) [2]. In Type 2 diabetes, the human body is unable to produce insulin [3]. Type 3 diabetes is also known as gestational diabetes. This type of diabetes is mainly found in case of pregnant women. They develop high glucose level in blood, which needs urgent medical attention. Otherwise, it will lead to various complications during pregnancy. If we neglect toward diabetes, then it can affect and damage the vital organs of body such as heart, liver, kidney, and eyes. Apart from that, the progression of diabetes occurs through five stages except Stage 0 which is considered as normal [4, 5]. Stage 1: Sometimes called compensation where secretion of insulin increases to maintain normoglycemia and β cell mass decreases. Stage 2: Glucose levels start increasing and β cell mass also decreases. Stage 3: It is an unstable period in which glucose levels relatively increase and approach to Stage 4. Stage 4: It is characterized as stable decompensation, and β cell is more dedifferentiation. Stage 5: Severe decompensation occurs, and β cell mass severely reduces and progression to ketosis as depicted in Fig. 1a, b.

Data analysis deals with identifying a pattern from a large set of data. From these analysis, we can derive certain predictive conclusions. This analytical work can be accomplished by the machine learning algorithms [6]. Machine learning is a part of artificial intelligence. Like human, brain learns from the past experience or analyzing the past history. Machine also learns in the similar way. Hence, machine can take its decision depending upon the knowledge fed into it. Knowledge is gathered by analyzing huge amount of data and by providing training to machine. We are using the machine learning algorithms: random forest classifier and XGBoost. We have implemented these two algorithms using a medical dataset to explore their techniques. These algorithms belong to two different machine learning approaches. So, we can get a thoughtful insight and conclusive remark regarding the approaches by comparing the prediction results [7, 8].

The remaining part of this research paper is organized as follows: Section 2 briefly represents the literature survey of various techniques for prediction of the diabetes. Section 3 presents the methodology used, i.e., Random forest classifier and XGBoost

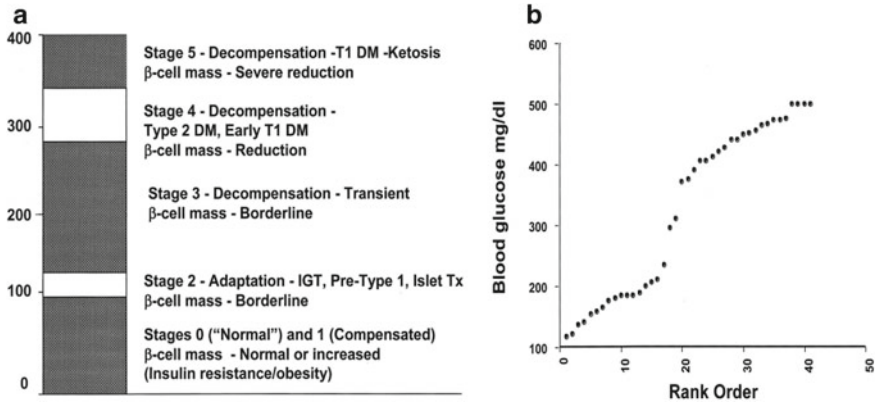

Fig. 1 **a** Five stages of progression of diabetes. **b** Normal and severe diabetic sugar levels

(hybrid machine learning technique) with the considered dataset. In Sect. 4, we compared the two methodologies and discussed the result analysis elaborately, and finally in Sect. 5 we conclude our paper with future work and some references.

2 Related Works

Sisodia et al. in [9] predict the diabetes using machine learning approaches. Using classification algorithm, i.e., decision tree, SVM, and Naïve Bayes, the authors detect diabetes at an early stage. From the result analysis, authors show that Naïve Bayes gives 76.3% highest accuracy comparatively with other approaches. Aljumahi et al. in [10] predict the diabetes treatment using the deterministic regression-based techniques. They adopted Oracle Data Miner (ODM) for predicting modes of treating diabetes into two groups, i.e., young and old age groups. From their analysis, young group can be delayed to avoid side effect, whereas in old age group, prescribed drug should be immediately taken. Lyer in [11] find out one of the finest solutions to diagnose the diabetic disease through the patterns found by the help of data classification analysis on J48 decision tree and Naïve Bayes algorithm. From their analysis, Naïve Bayes techniques give least error rate as compared to the other approaches.

Velu in [12] employed the most emerged three techniques for classification of the diabetic patients, i.e., EM algorithms, H Means + clustering, and Genetic Algorithm (GA) [6]. From their result analysis, H Means + clustering techniques give a better result as compared to other two techniques in case of diabetes disease. Ganji in [13] adopted fuzzy ant colony optimization techniques to find the set of rules for the adiabatic patient and their diagnosis. Now it is also used for the prima Indian diabetes datasets. Jayalakshmi T. in [14] diagnoses the adiabatic patient through their new approach—ANN techniques. The authors preprocessed and replaced the missing values in the datasets used for detecting the diabetic patient. Their modification

on dataset gives a better accuracy as compared to other training datasets, due to getting result in lesser time. Aishwarya et al. in [15] used classification techniques by using machine learning approaches for diabetes. To detect diabetes disease at an early stage, a greatest support of machine learning is needed. Authors trying a promising technique support vector machine (SVM) in machine learning approaches for classification.

3 Methodology Used

In this paper, we have classified the patient data to predict whether a patient has diabetes or not. For this classification purpose, we have used PIMA diabetes dataset which is provided by National Institute of Diabetes (NID) for Digestive and Kidney Diseases. This dataset consists of 768 rows and 9 columns. Each row has nine attributes like (i) Pregnancies (Number of times pregnant), (ii) Glucose (Plasma glucose concentration within 2 h duration with an oral glucose tolerance test), (iii) Blood Pressure [Diastolic/Systolic blood pressure (mm Hg)], (iv) The Skin Thickness [Triceps skin fold thickness (mm)], (v) Insulin [2-h duration serum insulin (mu U/ml)], (vi) BMI [body mass index (weight in kg/(height in m)^2)], (vii) Diabetes_Pedigree_Function (Diabetes pedigree function), (viii) Age[(years)], and (ix) Outcomes [Class variable (0 or 1)] [16]. These columns indicate some specific medical conditions, and the snapshot of the datasets is given below:

Out[4]:

	num_preg	glucose_conc	diastolic_bp	thickness	insulin	bmi	diab_pred	age	skin	diabetes
0	6	148	72	35	0	33.6	0.627	50	1.3790	True
1	1	85	66	29	0	26.6	0.351	31	1.1426	False
2	8	183	64	0	0	23.3	0.672	32	0.0000	True
3	1	89	66	23	94	28.1	0.167	21	0.9062	False
4	0	137	40	35	168	43.1	2.288	33	1.3790	True

Here the outcome column has two values, 0 and 1. 0 means patient has no diabetes, and 1 means patient has diabetes. Within this paper, we have used two algorithms like RandomForest classifier and XGBoost algorithm to predict whether a patient has diabetes or not [17].

We have used Jupyter Notebook which is freely available software for performing machine learning operations. For machine learning purpose, we need to import the Sklearn module which contains all the essential algorithm and functions. We need to import Python NumPy module and Pandas module for data analysis purpose. To plot different graphs, we need to import Matplotlib module which contains all the methods related to plotting graphs. Our data is stored in a CSV file which needs to be imported to the notebook by using Python Pandas module as a data frame. After importing the data, we can apply various data analysis and machine learning algorithms for classification and prediction. Before applying the machine learning algorithms on the dataset, we can see it through graphs by using Matplotlib.

If we want to see the correlation among the different parameters of the dataset, then we need to use the corr() function, which will find the correlation among the variables. Then, we can plot this correlation values in the form of graph by using seaborn module. To see this correlation properly, we have to use the heat map as shown below that also shows the correlation among the attributes of the datasets depicted in Fig. 2. The correlation values for the dataset are as follows:

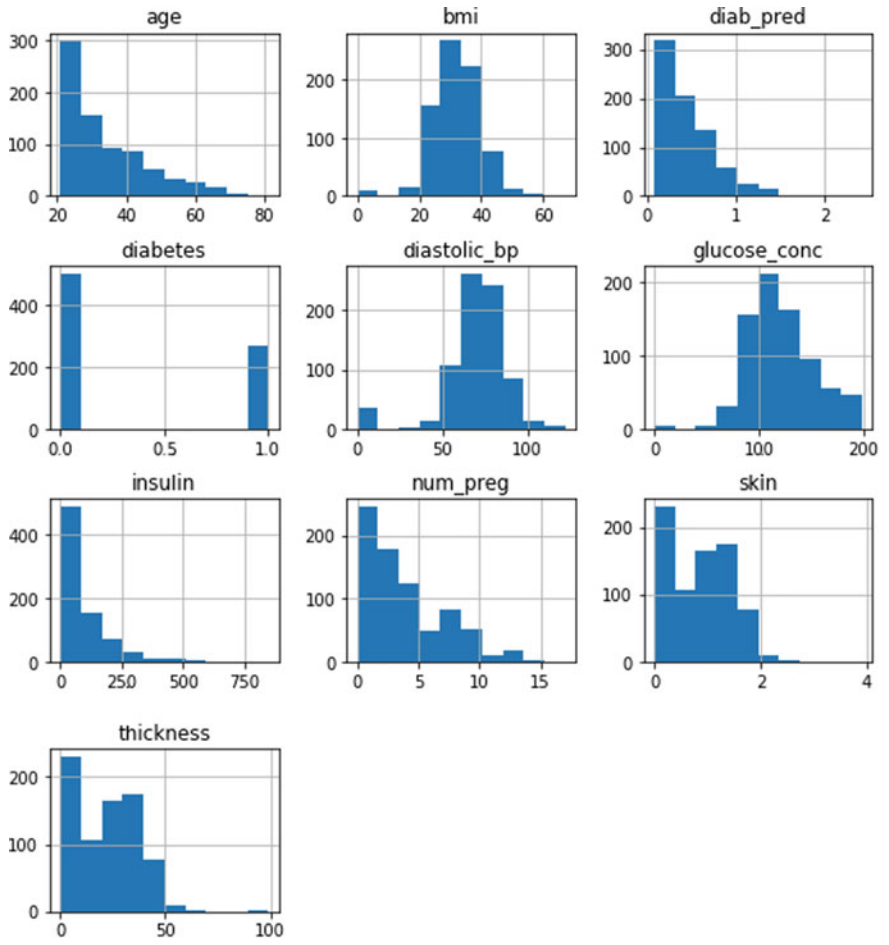

Fig. 2 Features/attributes of the dataset used in our classification

```
In [8]: data.corr()
```

Out[8]:

	num_preg	glucose_conc	diastolic_bp	thickness	insulin	bmi	diab_pred	age	skin	diabetes
num_preg	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	-0.033523	0.544341	-0.081672	0.221898
glucose_conc	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	0.137337	0.263514	0.057328	0.466581
diastolic_bp	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	0.041265	0.239528	0.207371	0.065068
thickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	1.000000	0.074752
insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	0.185071	-0.042163	0.436783	0.130548
bmi	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	0.140647	0.036242	0.392573	0.292695
diab_pred	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	1.000000	0.033561	0.183928	0.173844
age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	0.033561	1.000000	-0.113970	0.238356
skin	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	1.000000	0.074752
diabetes	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	0.173844	0.238356	0.074752	1.000000

3.1 Random Forest Classifier

Random forest belongs to the category of supervised machine learning. Forest refers to a set of trees, and tree refers to the decision tree. The prediction value of random forest depends on the decision trees present in the forest. Each decision tree represents some feature or label of dataset on random basis. Decision tree gives the prediction value after performing the operation on given data. Voting is done in case of classification problems. Majority in voting decides the class to be chosen [18]. Random forest can also be applied for regression-based problems. Here, the output is found by calculating the mean or median of all the predicted values given by the decision trees [19]. One of the main problems in decision tree is the high variance and low bias. Random forest overcomes this problem with many number of decision trees. This also gives us high degree of prediction accuracy [20, 21]. That is why random forest is a very popular machine learning algorithm. It can be applied for disease prediction, credit card fraud detection, design of recommendation system, classification of loan applicants, and so on. One of the challenges in random forest may be slower prediction time. This is because random forest uses many decision trees [17, 22].

In order to use Random forest classifier on our dataset, we have to import the RandomForestClassifier class from Sklearn module. We need to apply some exploratory data analysis using pandas and NumPy before applying RandomForest algorithm on the dataset. We have to find how many zero values are present in all the columns of the dataset except the Pregnancy column as its medical value is taken 0 or any other integer values basing on number of times patient is pregnant. But for all other columns, some values must be recorded that need to be checked. After applying exploratory data analysis, we got the results shown below:

```
total number of rows : 768
number of rows missing glucose_conc: 5
number of rows missing glucose_conc: 5
number of rows missing diastolic_bp: 35
number of rows missing insulin: 374
number of rows missing bmi: 11
number of rows missing diab_pred: 0
number of rows missing age: 0
number of rows missing skin: 227
```

We need to change these 0 values or missing values to some numerical value. For this purpose, we can either take mean or median value as a replacement. In this case, we have taken mean value to replace these missing values. After replacing all missing values with mean value, we need to divide the dataset into train and test parts. Here, 30% of total dataset is taken for testing purpose, and 70% is taken for training.

```
In [15]: ## Train Test Split

         from sklearn.model_selection import train_test_split
         feature_columns = ['num_preg', 'glucose_conc', 'diastolic_bp', 'insulin', 'bmi', 'diab_pred', 'age', 'skin']
         predicted_class = ['diabetes']

In [16]: X = data[feature_columns].values
         y = data[predicted_class].values


         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state=10)
```

After dividing dataset into train and test, now we can apply Random Forest algorithm on the train dataset for training the model.

```
In [19]: ## Apply Algorithm

         from sklearn.ensemble import RandomForestClassifier
         random_forest_model = RandomForestClassifier(random_state=10)

         random_forest_model.fit(X_train, y_train.ravel())
```

Now the model is trained by learning from the train dataset, i.e., X_train. Now we have to test our model by using the test dataset which we have separated from the original dataset.

```
In [23]: predict_train_data = random_forest_model.predict(X_test)

         from sklearn import metrics

         print("Accuracy = {0:.3f}".format(metrics.accuracy_score(y_test, predict_train_data)))

         Accuracy = 0.719
```

When we used our test dataset, i.e., X_test, it gives accuracy of 71.9%. The accuracy can be further improved by hyper-parameterization.

3.2 XGBoost

XGBoost is a type of hybrid machine learning algorithm which always provides better solution than any other machine learning algorithm. XGBoost or Extreme Gradient Boosting is a type of boosting algorithm based on ensemble. It combines several weak learners and provides an improve prediction accuracy. It builds weak models and understands various important features, parameters; using those conclusions, it builds a new stronger model and tries to reduce the rate of misclassification. It is also called gradient boosting because when we add the new models, it uses a gradient descent algorithm to minimize the loss. Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. XGBoost uses a tree ensemble [16, 18]. Tree ensemble is a set of classification and regression trees. These trees try to reduce the misclassification rates on each iteration. XGBoost can be used to solve any kind of machine learning problems like regression, classification, ranking, and user-defined prediction problems. It supports various cloud platforms like AWS, Azure, and so on. It is a type of algorithm which uses less computing resources to provide the better accuracy in less time. It provides better results than other algorithms because it has some inbuilt features like parallel tree building, tree pruning using depth-first approach, and cache awareness by using internal buffers to store data, and uses regularization to avoid over-fitting, efficient handling of missing data, and inbuilt cross validation capability. XGBoost provides some kind of tuning parameters which must be optimized for better performance as discussed below:

i. Learning_rate: step size shrinkage used to prevent over-fitting. Range is [0, 1].
ii. Max_depth: determines how deeply each tree is allowed to grow during any boosting round.
iii. Subsample: percentage of samples used per tree. Low value can lead to under-fitting.
iv. Gamma: controls whether a given node will split based on the expected reduction in loss after the split. A higher value leads to fewer splits. It supports only the tree-based learners.
v. Colsample_bytree: percentage of features used per tree. High value can lead to over-fitting.
vi. N_estimators: number of trees you want to build.
vii. Alpha: L1 regularization on leaf weights. A large value leads to more regularization.
viii. Lambda: L2 regularization on leaf weights is smoother than L1 regularization.

```
In [26]: ## Hyper Parameter Optimization

params={
 "learning_rate"    : [0.05, 0.10, 0.15, 0.20, 0.25, 0.30 ] ,
 "max_depth"        : [ 3, 4, 5, 6, 8, 10, 12, 15],
 "min_child_weight" : [ 1, 3, 5, 7 ],
 "gamma"            : [ 0.0, 0.1, 0.2 , 0.3, 0.4 ],
 "colsample_bytree" : [ 0.3, 0.4, 0.5 , 0.7 ]

}
```

We have used RandomizedSearchCv to optimize the hyperparameter for XGBoost.

```
In [27]: ## Hyperparameter optimization using RandomizedSearchCV
         from sklearn.model_selection import RandomizedSearchCV
         import xgboost

In [28]: classifier=xgboost.XGBClassifier()

In [29]: random_search=RandomizedSearchCV(classifier,param_distributions=params,n_iter=5,scoring='roc_auc',n_jobs=-1,cv=5,verbose=3)
```

We need to predict the best parameter for XGBoost algorithm using Randomized-SearchCv.

```
In [34]: random_search.best_estimator_

Out[34]: XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
                colsample_bytree=0.3, gamma=0.0, learning_rate=0.25,
                max_delta_step=0, max_depth=3, min_child_weight=7, missing=None,
                n_estimators=100, n_jobs=1, nthread=None,
                objective='binary:logistic', random_state=0, reg_alpha=0,
                reg_lambda=1, scale_pos_weight=1, seed=None, silent=True,
                subsample=1)

In [36]: classifier=xgboost.XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
                colsample_bytree=0.3, gamma=0.0, learning_rate=0.25,
                max_delta_step=0, max_depth=3, min_child_weight=7, missing=None,
                n_estimators=100, n_jobs=1, nthread=None,
                objective='binary:logistic', random_state=0, reg_alpha=0,
                reg_lambda=1, scale_pos_weight=1, seed=None, silent=True,
                subsample=1)
```

Now we have got our best classifier as per our problem, we can now apply the model on the dataset.

```
In [39]: from sklearn.model_selection import cross_val_score
         score=cross_val_score(classifier,X,y.ravel(),cv=10)

In [40]: score

Out[40]: array([0.72727273, 0.77922078, 0.67532468, 0.67532468, 0.7012987 ,
                0.74025974, 0.76623377, 0.76623377, 0.77631579, 0.80263158])

In [41]: score.mean()

Out[41]: 0.7410116199589883
```

We got the mean score, i.e., 74.10%, which is better than the RandomForest algorithm. This model can be improved by applying some other optimization method. We can deploy our model to cloud hosting services as cloud services are providing

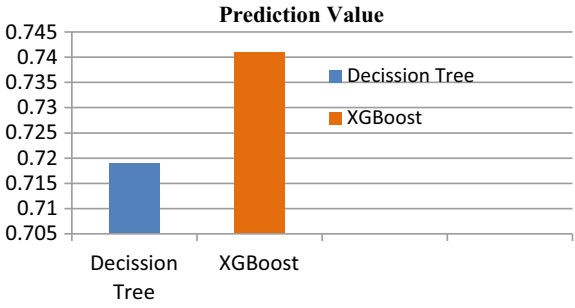

Fig. 3 Comparison of prediction value for machine learning algorithms

support for machine learning models which may not be supported by physical server. We can choose best cloud service provider as per our requirement. Some top most cloud service providers are Amazon, Google, Microsoft, and so on.

4 Result Analysis

We have used two machine learning algorithms on our dataset to get the prediction for diabetes. In case of Random forest with a fixed value of n, we get prediction value 0.719. But for XGBoost method, we get the prediction value 0.7410116. Hence, it is clear that XGBoost model gives more accuracy in our case (Fig. 3).

5 Conclusion and Future Scope

In this paper, we have applied two machine learning algorithms on the dataset for prediction of diabetes. RandomForest uses sequential decision trees, whereas XGBoost uses parallel trees for prediction. XGBoost provides better results and also faster than RandomForest as it optimally uses both hardware and software. These algorithms belong to two different machine learning approaches. More analytical report and concluding remark can be drawn by considering more machine learning algorithms which fall under these two approaches. We can also improve the performance of these algorithms by using hyper-parameterization and optimization methods which are beyond the scope of the paper. Furthermore, this work can be extended and improved by including other sets of machine learning approaches.

References

1. American Diabetes Association: Diagnosis and classification of diabetes mellitus. Diabet. Care **27**, S5 (2004)
2. Rasouli, B., Ahlbom, A., Andersson, T., Grill, V., Midthjell, K., Olsson, L., Carlsson, S.: Alcohol consumption is associated with reduced risk of type 2 diabetes and autoimmune diabetes in adults: results from the Nord Trøndelag health study. Diabet. Med. **30**(1), 56–64 (2013)
3. Singla, R., Singla, A., Gupta, Y.: Artificial intelligence/machine learning in diabetes care. Indian J. Endocrinol. Metabol. **23**(4), 495–497 (2019)
4. Jayalakshmi, T., Santhakumaran, A.: A novel classification method for diagnosis of diabetes mellitus using artificial neural networks. In: International Conference on Data Storage and Data Engineering, pp. 159–163 (2010)
5. Weir, G.C., Bonner-Weir, S.: Five stages of evolving beta-cell dysfunction during progression to diabetes. Diabetes **53**(Suppl 3), S16–S21 (2004)
6. Dash, R., Misra, B.B.: A multi-objective feature selection and classifier ensemble technique for microarray data analysis. Int. J. Data Min. Bioinform. **20**(2), 123–160 (2018)
7. Zheng, T., Xie, W., Xu, L.: A machine learning-based framework to identify type 2 diabetes through electronic health records. Int. J. Med. Inf. **97**, 120–127 (2017)
8. Nabi, M., Kumar, P., Wahid, A.: Performance analysis of classification algorithms in predicting diabetes. Int. J. Adv. Res. Comput. Sci. **8**(3), 456–461 (2017)
9. Sisodia, D., Sisodia, D.S.: Prediction of diabetes using classification algorithms. Proc. Comput. Sci. **132**, 1578–1585 (2018)
10. Aljumah, A.A., Ahamad, M.G., Siddiqui, M.K.: Application of data mining: diabetes health care in young and old patients. J. King Saud Univ. Comput. Inf. Sci. **25**(2), 127–136 (2013)
11. Iyer, A., Jeyalatha, S., Sumbaly, R.: Diagnosis of diabetes using classification mining techniques. arXiv preprint arXiv:1502.03774
12. Velu, C.M., Kashwan, K.R.: Visual data mining techniques for classification of diabetic patients. In: 2013 3rd IEEE International Advance Computing Conference (IACC), pp. 1070–1075. IEEE (2013)
13. Ganji, M.F., Abadeh, M.S.: Using fuzzy ant colony optimization for diagnosis of diabetes disease. In: 2010 18th Iranian Conference on Electrical Engineering, pp. 501–505. IEEE (2010)
14. Jayalakshmi, T., Santhakumaran, A.: A novel classification method for diagnosis of diabetes mellitus using artificial neural networks. In: 2010 International Conference on Data Storage and Data Engineering, pp. 159–163. IEEE (2010)
15. Aishwarya, R., Gayathri, P.: A method for classification using machine learning technique for diabetes (2013)
16. Mohanty, S., Sahoo J., Pramanik J., Jayanhu S.: A review of techniques in practice for sensing ground vibration due to blasting in open cast mining. In: International Conference on Remote Engineering and Virtual Instrumentation, pp. 306–314 (2019)
17. Aishwarya, R., Gayathri, P., Jaisankar, N.: A method for classification using machine learning technique for diabetes. Int. J. Eng. Technol. **5**, 2903–2908 (2013)
18. Rashid, T.A., Abdulla, S.M., Abdulla, R.M.: Decision support system for diabetes mellitus through machine learning techniques. Int. J. Adv. Comput. Sci. Appl. **7**, 170–178 (2016)
19. Wang N, Kang G (2012) Monitoring system for type 2 diabetes mellitus. In: IEEE Conference on E-health Networking, pp. 62–67
20. Kononenko, I.: Machine learning for medical diagnosis: history, state of the art and perspective. Artifi. Intell. Med. **23**(1), 89–109 (2001). (Elsevier)
21. Schulze, M.B., Hu, F.B.: Dietary patterns and risk of hypertension, type 2 diabetes mellitus, and coronary heart disease. Curr. Atherosclerosis Rep. **4**(6), 462–467 (2002)
22. Verikas, A., Gelzinis, A., Bacauskiene, M.: Mining data with random forests: a survey and results of new tests. J. Pattern Recogn. **44**, 330–349 (2011)