Diabetes classification application with efficient missing and outliers data handling algorithms

Hanaa Torkey¹ · Elhossiny Ibrahim¹ · EZZ El-Din Hemdan¹ · Ayman El-Sayed¹ · Marwa A. Shouman¹

Received: 11 September 2020 / Accepted: 17 March 2021 / Published online: 17 April 2021
© The Author(s) 2021

Abstract

Communication between sensors spread everywhere in healthcare systems may cause some missing in the transferred features. Repairing the data problems of sensing devices by artificial intelligence technologies have facilitated the Medical Internet of Things (MIoT) and its emerging applications in Healthcare. MIoT has great potential to affect the patient's life. Data collected from smart wearable devices size dramatically increases with data collected from millions of patients who are suffering from diseases such as diabetes. However, sensors or human errors lead to missing some values of the data. The major challenge of this problem is how to predict this value to maintain the data analysis model performance within a good range. In this paper, a complete healthcare system for diabetics has been used, as well as two new algorithms are developed to handle the crucial problem of missed data from MIoT wearable sensors. The proposed work is based on the integration of Random Forest, mean, class' mean, interquartile range (IQR), and Deep Learning to produce a clean and complete dataset. Which can enhance any machine learning model performance. Moreover, the outliers repair technique is proposed based on dataset class detection, then repair it by Deep Learning (DL). The final model accuracy with the two steps of imputation and outliers repair is 97.41% and 99.71% Area Under Curve (AUC). The used healthcare system is a web-based diabetes classification application using flask to be used in hospitals and healthcare centers for the patient diagnosed with an effective fashion.

Keywords Diabetes · Imputation · Anomaly detection · MIoT and wearable devices · Flask

Introduction

Recently, there is a huge mutation in the different directions of science like data analytics, data gathering, data mining, healthcare systems, machine learning, deep learning, cloud computing, and storage. All of these technologies must help the human to take care of his/her health. Statistics improves

that, diabetes disease is one of the most harmful diseases that cause other dangerous diseases and may cause death in some cases. For example, diabetes affects the ability of the body in producing the insulin hormone that makes the metabolism of carbohydrates abnormal and raises the levels of glucose in the blood. Patients with diabetes generally suffer from high blood sugar. Intensify thirst, intensify hunger and frequent urination are some of the symptoms caused due to high blood sugar [1]. Parkinson's Disease diagnosis According to Gender [2] by using a hybrid greedy step-wise algorithm. Nowadays the medical data records have become very huge due to the increasing number of patients, social medical data, and sensing data. For example, when increasing the scale of the diabetes datasets, the analysis process requires distributing the dataset to be processed over a distributed system. Also, machine learning techniques are required to build models that make a good classification about the patients from some given features like Glucose, Insulin, BMI, or Diabetes Pedigree Function.

If this data is exploited perfectly, this can change it from an unvalued thing to help us in solving most of our problems

✉ Elhossiny Ibrahim
Elhossiny@el-eng.menofia.edu.eg

Hanaa Torkey
htorkey@el-eng.menofia.edu.eg

EZZ El-Din Hemdan
ezzeldinhemdan@eleng.menofia.edu.eg

Ayman El-Sayed
ayman.elsayed@el-eng.menofia.edu.eg

Marwa A. Shouman
marwa.shouman@el-eng.menofia.edu.eg

¹ Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Menouf, Egypt

in a cost-effective way. Now the science-based on analyzing the historical records in different fields.

From businesses and research institutions to governments, organizations now routinely generate data of unprecedented scope and complexity. Gleaning meaningful information and competitive advantages from massive amounts of data has become increasingly important to organizations globally. Trying to efficiently extract meaningful insights from such data sources quickly and easily is challenging. Thus, analytics has become inextricably vital to realize the full value of Data to improve their business performance and increase their market share. The tools available to handle the volume, velocity, and variety of data have improved greatly in recent years. In general, these technologies are not prohibitively expensive, and much of the software is open-source [3].

One of the most challenges that face data analytics is dealing with the missed data, which sometimes comes from sensors problem, human mistakes, or during the transmission between the system parts in different locations as in cloud servers. If the number of missing items raises in the large datasets, repairing the dataset becomes critical to preserve, or even raise the statistical power of the data [4]. In this paper, a framework of smart wearable devices based on big data analytics for diabetes detection in MIIoT is presented. Likewise, we applying data analytics in the health care system over the diabetes dataset. The model learned to predict the person's status from the given features. Comparative study of different algorithms such as Random Forest (RF), Decision Tree (DT), Naïve Bayes (NB), and Support Vector Machine (SVM) as classifiers, using the accuracy, Root Mean Square Error (RMSE), F1-measure (FM), and time as an evaluation metric. The size of the dataset is considered.

The main contribution of this work is presenting a framework for smart wearable devices. With developing an efficient machine learning system on apache-spark. Under the assumption that handling missing using our two novel missing data handling algorithms. Two anomaly repairing techniques, which yield higher risk stratification accuracy. As well as diabetes classification with minimal time and computational energy. The contributions could be summarized as:

- A comprehensive study of data from gathering using sensors, cleaning using missing data algorithms, learning to produce models, apply models in applications.
- Developing two efficient missing data imputation algorithms. The first algorithm is based on predicting the missed values by using the RF algorithm, but with initializing not predicted values with the mean value while the second algorithm working on replacing the mean value with the class' mean value.
- Demonstration of enhancement in classification which reached 92.02% in RF as the best case against other ML algorithms. The mean class value increases accuracy by

10%. The results prove that RF is preferred when the dataset size is increased. SVM is the most accurate algorithm when the dataset is small, and when an increase in the dataset size the model created is better.

- Apply interquartile range (IQR) anomaly detection technique to remove outliers, and introduce a new efficient way to detect outliers based on separate dataset classes when determining the outliers, to give more accurate results.
- Repairing the outliers instead of removing them, DL and RFMC are used to impute the outliers by replacing them with Null, then treat it as a missed value by using deep learning.
- The model accuracy reaches 97.41% when using DL to repair the outliers, by increasing rate reaches to 7.19%, 5% when compared to without outliers repair, and the traditional outliers repair techniques respectively.
- Utilizing the proposed machine learning models for diabetes classification for developing a web-based application using the Flask web framework to be used in hospitals and healthcare centers for patient diagnosis in an operational manner.
- By using the RF algorithm, RFCM imputation algorithm, IQR anomaly detection applied over each class of the dataset, the accuracy reaches 92.46%, then uses the learned model to introduce the web service flask interface.

The structure of this paper as follows: Sect. “[Background and related work](#)” provides related work of several classification techniques for diabetes classification, while Sect. “[Smart wearable device based data analytics for diabetes detection in MIIoT](#)” introduces a smart wearable device based on big data analytics for diabetes detection MIIoT framework. Section “[The proposed data imputation algorithms](#)” provides two new missing data imputation algorithms while the proposed diabetes prediction model is presented in Sect. “[Diabetes classification model using machine learning](#)”. Section “[A web-based- classification application using flask in MIIoT](#)” offers the new web-based application for diabetes predications using Flask, the results of using the new approaches over different classification algorithms are presented in Sect. “[Experimental evaluations and testing](#)”. Finally, Sect. “[Conclusion](#)” shows the conclusion of the paper.

Background and related work

There is little research work in data imputation that can be generated from wearable sensors and most of them address it with customary methods without examining the performance of the model. When building models in the healthcare

system and classify the state of the patients or the probability of being affected by some diseases or creating a recommendation system. The dataset correctness and truthfulness are very weighty, but in most cases, some values are lost due to devices' problems or human errors. So the way of adjusting these missed data is very vital.

Diabetes patient monitoring

Daily controlling of diabetes by patients depend on know-how properly measuring the levels of glucose concentration in the blood by using the right sensors (Like CGM sensors) [5]. As shown in Fig. 1 the patient wears the sensor and can monitor the change in Glucose level.

This point is very important and saves a large number of patients from death due to high glucose levels, or other diseases like blindness. Specific authors' study complete systems, from the data sources (sensors), until the classification model learning process the summary of related work is tabulated in Table 1. Georga et al. [6] build Continuous Glucose Monitoring (CGM), with an automated insulin pump, and a mobile application integrated system for self-monitoring and adjusting the diabetes patient's state. M. Chen et al. [7] make integration of smart clothes designs with cloud computing (storage), and big data (decision making) for making a complete system for monitoring and take care of patients. Murali et al. [8] use a wearable smart device for monitoring and classifying the physical human emotion by capturing the physiological signals: ECG, respiration, Impedance Cardiogram (ICG).

There are many papers, working on building classification models like Saravana Kumar et al. [9] by using analyzing algorithms over the Hadoop framework for the type of diabetes, entanglement linked with it, and the preferred treatment for the patient. This analysis provides a perfect way to care for diabetics. Kumar et al. [10] a comparative study of different machine learning algorithms to determine the most effective one for making the classification.

Missing data handling

One of the biggest problems facing machine learning is how to deal with missed values on the datasets. As if the rows contain missing values are deleted, some important data are lost. Treating the missing values in a good manner can affect the entire model behavior. So data imputation for diabetes data is still at the primitive stage. As proposed by Sanadi et al. [11] judge the different imputation methods (a Hot deck, a Naïve–Bayes, mean method) on the different classification algorithms. The results showed that there is no general best method, but each approach is preferred with a specific classifier. They conclude that using the Naïve–Bayes method to substitute the missing value gives an improved accuracy with the RIPPER classifier.

Also in this work, Tang et al. [12], which develop a random forest imputation algorithm to be used for different size datasets, and their results show that performance is enhanced with increasing correlation. Noor et al. [13] three diverse varieties of mean value are used to replace the missed value (pure Mean, Mean Above, and Mean Above–Below). And the results indicate that using the Mean above–below method gives the best results based on their study case dataset. Biessmann et al. [14] uses the deep learning model for truthful imputation of non-numeric values. Dzulkalnine et al. [15] implement the feature selection hybrid model to impute the missing data by the integration of the Fuzzy Principle component analysis (FPCA), support vector machine, and the Fuzzy c-means (FCM) to select the relevant features only in the missed data treatment process. Sherif et al. [16] offered a new approach using clustering, the local least square imputation method, then select the smallest Euclidian distance to catch the missed data value from a similar cluster to the missed value.

Bartlett et al. [17] present how imputation by fully conditional requirement, a popular method for acting multiple imputations. Cantor et al. [18] use the KNN algorithm for missed data estimation for DNA microarrays. Choi et al. [19] target to deliver guidance in selecting the optimal approach of missed mechanism and display the effect on the graph

Fig. 1 How to use sensors for continuous diabetes monitoring

Table 1 Summary of current big data analytics algorithms on Diabetes classification

Author	Title	Dataset size	Description	Algorithms	Year
Saravana Kumar et al. [9]	Predictive methodology for diabetic data analysis in big data	768	Determining the diabetes type, and the suitable treatment	Hadoop/Map Reduce	2015
Perveen et al. [28]	Performance analysis of data mining classification techniques to predict diabetes	650	Study the effect of Blood Pressure on Dietetics	DT, SVM and NB	2016
Orabi et al. [29]	Early predictive system for diabetes mellitus disease	100	Develop a new prediction system, by adding a regression technique with a randomization code to predict the age	DT	2016
Rashid et al. [30]	An Intelligent Approach for Diabetes Classification	500	Determine the impact of FBS (Fasting Blood Sugar) rate and the symptoms on the patient's health by predicting the rate of FBS, and its relation to symptoms	Artificial Neural Network(ANN), DT	2016
Farid et al. [31]	An adaptive rule-based classifier for mining big biological data	148	An optimal ensemble classifier that will help in improving the prediction accuracy of DNA variant identification and classification task	DT and k-nearest-neighbor (KNN)algorithms	2016
Sisodia et al. [1]	Prediction of diabetes using classification algorithms	768	Design a model which can prognosticate the likelihood of diabetes in patients with maximum accuracy	DT, SVM and NB	2018
Ali Zia et al. [32]	Predicting diabetes in medical datasets using machine learning techniques	768	Analyzing medical Bioinformatics data to predict if the patients affected by diabetes or not	NB, DT and KNN	2017

in illustrating the missing structure. Zhang [20] Focus on single imputation, but a complete case analysis by using R code. Qiu et al. [21] define a new technique called denoising autoencoder with partial loss (DAPL) based on deep learning for data imputation. Results on pan-cancer gene expression data and DNA methylation data from more than 11,000 samples ascertain key improvement over regular denoising autoencoder. Beaulieu-Jones et al. [22] compare the effectiveness of popular multiple imputations approaches with a deeply learned auto-encoder using the Pooled Resource Open-Access ALS Clinical Trials Database (PRO-ACT).

Some studies taking the outcome of missing data in the model power [23] validates that on substituting the missing values and outliers by group median and median values, respectively, also using the random forest as both feature selection and classification method incomes a more truthful model. Yilmaz et al. [24] Propose a new data preparing method based on the modified K-means clustering algorithm by eliminating the noise or inconsistent data and classifying the data by using SVM, the results confirm that the proposed approach improves the model effectiveness. Most of the missed data imputation approaches are restricted to one type of variables categorical or continuous, but the missforest(MF) algorithm [25] repair all types by using random forest, which certainly creates a multiple imputation scheme. Phung et al. [26] employ deep learning(DL) for predicting the values of the missing data on the healthcare dataset. "Multivariate imputation by chained equations" (MICE) has evidence as a righteous technique of fixing missing data [27].

It's noticed that some authors introduced a lot of research on the classification of diabetes, most of the techniques,

applying machine learning models, but not concerning the missing data and outlier. But it is very weighty and the model performance is affected. Regarding these limitations in the state of art diabetes classification methods, the main achievement in this work is two enhanced algorithms to clean the data from any missing or outlier values. So the results show that the final model performance is enhanced through altered metrics like accuracy, recall, and MSE.

Smart wearable device based data analytics for diabetes detection in MIoT

With our day's digital age, a massive number of smart devices have been connected to the Medical Internet of things (MIoT). Connected MIoT devices generate an extensive volume of data. However, the enormous amount of data produced by the connected Medical IoT devices is requiring fast parallel processing platforms. Big data platforms (i.e. Apache Spark) are used to handle and manage large volumes of gathering data from wearable sensors. Thus, big data processing challenges need to be resolved by adopting a big data mining approaches in the classification and detection processes of various diseases.

The main idea of the proposed MIoT system is to send the data to the cloud and big data server for analysis. Then the cloud and big data server send back the results of predicting and detecting diabetes (i.e. yes or no) as shown in Fig. 2. The proposed system consists of the following main parts:

1. Smart wearable devices.

Fig. 2 The proposed MIoT framework for diabetes detection

2. The cloud and big data server will process the medical data and then replay the decision to the medical staff server (i.e. user mobile device and medical dashboard in the hospital system). The diabetes classification can be through using the classification model as shown in Fig. 2.
3. The communication system between the cloud server and smart wearable devices. These can be through the use of the MQTT protocol as the most common messaging protocol.
4. User mobile device and medical dashboard in the hospital system that will be used for decision making by diabetes patients.

In the previous system, the phases that the data passes through cause some errors, due to sensor problems, or human mistakes. These errors may lead to wrong decisions or case handling which can cause disease to enter dangerous stages or even death. So the way of interacting with missing data is a very important and urgent research area, especially when used in building healthcare models. In this paper, we are motivated to work at this point to solve the problem of missed data to improve the performance of the healthcare model and solve one of the most crucial challenges that face it.

The proposed data imputation algorithms

In this section, we provide the idea of the two new approaches for handling missing values in diabetes data. The two approaches are discussed in the following subsections:

• Algorithm 1: Random Forest with Mean (RFM)

The main idea of this algorithm is to apply the RF algorithm to predict the missed values, firstly the model needs to be learned with a complete dataset. When predicting the missed values in one column other columns may contain null values. Replace the null values with the mean value of not null for each column, then divide the dataset to train (that doesn't have missed values), and predict (that contains missing values). Finally, replace the missed values with the predicted values, each one on the corresponding index.

Repeat the same steps for the other columns, but do not forget to return the null places to predict it again by using the random forest as a regression algorithm. Ordering the dataset descending according to the correlation factor, then start to predict the most correlated columns, until the accuracy of the model starts to decrease, return to the previous state. The decrease in the accuracy is due to predict a value of a less correlated column gives that values not related to the missed. Each iteration measure the overall model accuracy. To be more dedicated, Suppose we have n features ($X_1, X_2, \dots, X_1, \dots, X_n$), when predicting X_i missed values, the other features ($X_1, X_2, \dots, X_n$) are replaced initially with the mean

value, while X_i values are predicted by using the RF algorithm. After handling X_i by return X_{i+1} to the original state (that has null values). Predict the null values of X_{i+1} with the new predicted X_i , and so on. In each step measures the model accuracy AC_i (i is the iteration index) by using Eq. 4.

Compare AC_i with AC_{i-1} , to stop or continue the prediction. Keep the time and the accuracy of the model. When building the model to predict the missed values, split the dataset into Train Subset (TS), and Predict subset (PS).

The predicted value for the missed data by RF is calculated by the following equation:

$$X_i = \frac{1}{T} \sum_{t=1}^T Ft_i(PS_i) \quad (1)$$

X_i is the missed value prediction, Ft_i is the model learned by TS_i for each tree t in the RF ensemble model, and PS_i is the matrix of features used to predict the missing values.

$PS_i = X_1, X_2, \dots, X_n$, of the predicted subset. T is the tree number in the RF, i is the index of the column to be predicted (its missing values).

• Algorithm 2: Random Forest with Class' Mean (RFCM)

After working on the previous approach, instead of using the mean value in initialization, use the class' mean value. In our case the dataset rows represent patients' records, some of them are diabetics and the others are not, so when replacing the null value with the average value of all it is not correct. The idea here is when initializing the missed values of non-predicted features, replace the missed value with the mean of its class (diabetics or not). To apply this, we must divide the dataset into two parts: diabetic patients Yes Subset (YS), and not diabetic patients No Subset (NS), then calculate the mean value of each part: Mean of Yes (MY) and the Mean of No (MN) by using Eqs. (5) and (6). Finally, searching for the missed values, when finding it, replace it with MY or MN according to its label. Also here using RF to predict the missed columns, after ordering the columns according to their correlation with the outcome of the dataset. Iterate throws the sorted list of the features to predict the missed values with RF, initially the non-predicted features with the class' mean values.

$$MN = \frac{1}{N1} \sum_{i=1}^{i=N1} XV_i \text{ if } XV_i \neq 0, \text{ outcome class}(XV_i = 0) \quad (2)$$

$$MN = \frac{1}{N2} \sum_{i=1}^{i=N2} XV_i \text{ if } XV_i \neq 0, \text{ outcome class}(XV_i = 0) \quad (3)$$

$N1$ and $N2$ are the numbers of patients who have diabetes, and no diabetics respectively. XV_i is the value of the feature of index i . The missed value calculation as mentioned calculated by using Eq. (4).

Algorithm 1: *Random Forest with Mean (RFM)***Input:** *dataset with missed data.***Output:** *imputed dataset.***Steps:**

1. Ordering the dataset n features ($X_1, X_2, \dots, X_i, \dots, X_n$) descending according to the correlation factor.
2. For X_i in the sorted list
 - a. Initialize the missed values of non-predicted features with their mean values.
 - b. Return X_i to its original state before initialization in previous iterations.
 - c. Split the dataset into two parts TS, and PS.
 - d. Use TS to learn the RF model.
 - e. Use the learned model to predict the missed values of PS by using equation (4).
 - f. Replace the missed values with the predicted ones.
 - g. Measure the model accuracy AC_i after the missed column values are handled.
 - h. If $AC_i > AC_{i-1}$:
Return to the previous state of the column before the prediction.
3. End of the For Loop
4. Finally, use this complete dataset to build the complete model.

Algorithm 2: *Random Forest with Class' Mean (RFCM)***Input:** *dataset with missed data.***Output:** *imputed dataset.***Steps:**

1. Divide the dataset into YS and NS according to the outcome.
2. Calculate the average value of not null for each part MY and MN by using equations (5) and (6).
3. Ordering the dataset n features ($X_1, X_2, \dots, X_i, \dots, X_n$) descending according to the correlation factor.
4. For X_i in the sorted list
 - a. Initialize the missed values of non-predicted features with their mean values (MY or MN), according to its label class.
 - b. Return X_i to its original state before initialization in previous iterations.
 - c. Split the dataset into two parts TS, and PS by using equation (4).
 - d. Use TS to learn the RF model.
 - e. Use the learned model to predict the missed values of PS.
 - f. Replace the missed values with the predicted ones.
 - g. Measure the model accuracy AC_i after the missed column values are handled.
 - h. If $AC_i > AC_{i-1}$:
Return to the previous state of the column before the prediction.
5. End of the For Loop
6. Finally, use this complete dataset to build the healthcare high-performance model.

Fig. 3 Proposed diabetes classification model

Diabetes classification model using machine learning

In this section, we will discuss the steps for learning our model for predicting the diabetes patients accord some features, as shown in Fig. 3, using machine learning classification algorithms as mentioned in the following paragraph.

The first in Fig. 3 is the preprocessing of the dataset by converting it to a suitable format to be compatible with the spark environment. As mentioned above Spark has ML API which is working on data expressed in the data frame format,

after that handling the missed data values by different approaches like mean, KNN, and the two new approaches (RFM and RFCM). At the end of this step, the dataset is complete, also in a suitable format (Data frames) to be processed over the apache spark environment. The second step is randomly to divide the dataset into a training set, and a test set by the ratio (70–30) respectively, the training set used to learn the model by using tenfold cross-validation to produce the right fitted model after the model is learned, the test set is used to measure the performance of the model. According to the real values of the output, and generate the

Fig. 4 Proposed MIoT system

predicted values from the model, the final step of the model can be evaluated according to some metrics like accuracy, recall, and precision. The classification algorithms used in the model are DT, RF, NB, and SVM.

A web-based- classification application using flask in MIoT

The Healthcare system presents a service for the patients; it needs to be more comfortable and easier to use it. This section provides a portable diabetes prediction application using the flask as shown in Fig. 4. This web application is to predict the real-time level of diabetes. It takes inputs such as pregnancies, blood pressure, age, bmi, insulin level and suggests diabetic users take appropriate diet to maintain their sugar level. It predicts the glucose level using efficient machine learning algorithms of the multivariate regression. The classifiers have been trained using the diabetes dataset. They are developed using Python, sci-kitlearn, Flask, HTML, matplotlib. In the proposed application, the user only needs to fill the form on the web application, then press the predict button to know the patient status, without any previous knowledge about machine learning, or any programming languages. To implement this system, the flask server is used. The Flask is a lightweight Web Service Gateway Interface (WSGI) web application framework [33].

In real-world applications, the proposed application will start by collecting data from sensors, handling the missed values due to any mistakes by using two new algorithms. After preprocessing the data and sure that it is complete data, divide it to train and test parts, from this step model is created and evaluated. The trained model was used as part of the proposed flask application to predict patient status.

In summary, the proposed application work as follows:

Table 2 Dataset description

Attribute	Description	Type
Pregnancies	Number of times pregnant	N
Glucose	Plasma glucose concentration 2 h in an oral glucose tolerance test	N
Blood pressure	Diastolic blood pressure (mm Hg)	N
Skin thickness	Triceps skinfold thickness (mm)	N
Insulin	2-h serum insulin (mu U/ml)	N
BMI	Body mass index (weight in kg/ (height in m) ²)	N
Diabetes Pedigree Function	Diabetes pedigree function	N
Age	Age (years)	N
Outcome	Class variable (0 or 1)	C

Table 3 Dataset missed values statistics for each feature

Attribute	Size_1	Size_2	Size_3	Percentage %
Pregnancies	111	301	412	15%
Glucose	5	13	18	0.65%
Blood pressure	35	90	125	4.5%
Skin thickness	227	573	800	29%
Insulin	374	956	1330	48%
BMI	11	28	39	1.4%
Diabetes pedigree function	0	0	0	0%
Age	0	0	0	0%

- The patients, doctors, or any other users can access the website,
- Fill the form with the patient's diabetes information, then this form communicates with the pre-trained model (proposed efficient machine learning algorithms in this paper) using the flask sever.

- The report or the result appears to the user without any knowledge about machine learning, or programming.

Finally, any user who doesn't have any data science experience or any medical knowledge can use the website to determine the patient's status. This section shows how technology facilitates the patient's life and minimize their pain. The full code can be accessed from <https://github.com/elhossiny/diabetes-flask>.

Experimental evaluations and testing

All experiments is done on Apache Spark version 2.1.2, using Python 3.6.6, and the device is Intel (R) Core I7(4700MQ) CPU at 2.40 GHz of processor and 16.00 GB of RAM. All of these are working on Ubuntu 20.04 Operating System.

Evaluation dataset

The dataset contains 2768 patients and eight features (PIDDD) which is sourced from the UCI machine learning repository [34]. Apply the work on three different sizes (768, 2000, and 2768). Table 2 shows the attributes of the dataset. Table 3 shows the statistics of our dataset; the number of missing values of each column in the dataset and the percentage of the missed values of the total number of items.

Classification system evaluation metrics

We need to measure the effect of treating the missed data, but in this work, a different and effective way is used. The most important sign of any new approach is model efficiency. When dealing with the healthcare system, the model performance is very critical. The following metrics are calculated from the confusion matrix elements (True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN)).

- Accuracy(Ac): the degree to which the result of a measurement, calculation, or specification conforms to the correct value

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (4)$$

- RMSE: it compares a predicted value and an observed or known value.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (P_i - O_i)^2}{n}} \quad (5)$$

- F1-Measure (FM): F1-Measure means that a measure of a test's accuracy.

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

- Area Under Curve (AUC): calculate the whole two-dimensional area under the entire Receiver Operating Characteristic (ROC) curve. ROC is a graph showing the classification quality.

Data imputation analyses

Firstly, extracting a pure dataset that doesn't contain any missed data. Randomly put a null value with different ratios from 10 to 50%. Apply the two new algorithms to treat the missing data, evaluating their performance compared to the traditional algorithms, shows the effect of missed data and different treatment on the model accuracy.

As shown in Table 4 a comparison between the used algorithms for missing data imputation.

As presented in Table 5 four different classification algorithms, with eight ways of dealing with missing data. The model accuracy decrease as the ratio of missed data increase, RF is the most effective classifier as it uses an ensemble learning. The two new algorithms give enhancement in the model performance in most cases, as they merge to algorithms, by using one in initial imputation (Mean) then use RF regression iteratively to predict the missed values.

Table 4 The used algorithms to evaluate the proposed work

Algorithm	Advantages	Disadvantages
Mean	Easy to implement	Average performance
KNN [18]	Easy to implement	Average performance
Deep learning [14]	High-performance models	Imputation takes a long time
MICE [27]	Multiple imputations	Average performance
Miss-forest [25]	Can repair mixed kinds of missing data	Average performance
The proposed algorithms	Good performance models, and Multiple imputations	Use additional memory space

Table 5 Ac, FM, AUC of different algorithms on different missed ratios using imputation algorithms

Missed data ratio		10%			20%			30%			40%			50%		
Method		Ac	FM	AUC	Ac	FM	AUC	Ac	FM	AUC	Ac	FM	AUC	Ac	FM	AUC
DT	Without	87.00	0.79	0.87	85.26	0.78	0.86	85.15	0.79	0.82	85.28	0.79	0.82	83.57	0.78	0.80
	KNN	86.01	0.78	0.88	85.76	0.79	0.88	85.87	0.78	0.86	84.63	0.78	0.85	84.10	0.78	0.84
	Mean	87.04	0.78	0.87	85.85	0.77	0.87	86.62	0.79	0.87	85.46	0.78	0.87	84.41	0.78	0.86
	DL	86.29	0.78	0.89	86.24	0.78	0.88	87.87	0.79	0.88	87.58	0.78	0.89	86.91	0.78	0.88
	Mice	87.33	0.78	0.90	86.99	0.79	0.89	86.29	0.79	0.85	86.27	0.78	0.89	85.04	0.78	0.87
	MF	86.80	0.79	0.85	88.50	0.79	0.90	87.37	0.79	0.89	87.70	0.79	0.90	86.50	0.78	0.88
	RFM	88.00	0.78	0.89	89.64	0.80	0.91	90.12	0.81	0.91	90.29	0.80	0.89	87.76	0.78	0.90
	RFCM	88.20	0.78	0.92	89.21	0.81	0.91	89.06	0.77	0.89	88.15	0.76	0.88	88.07	0.81	0.90
RF	Without	89.73	0.78	0.96	88.80	0.78	0.95	87.39	0.80	0.94	87.87	0.79	0.94	85.86	0.79	0.93
	KNN	89.58	0.79	0.96	88.31	0.79	0.95	88.14	0.79	0.95	87.92	0.80	0.95	86.91	0.79	0.94
	Mean	89.89	0.80	0.96	89.71	0.79	0.96	86.91	0.78	0.94	86.97	0.80	0.94	87.72	0.79	0.95
	DL	89.13	0.80	0.95	89.00	0.79	0.96	89.25	0.79	0.96	89.26	0.79	0.96	89.96	0.79	0.96
	Mice	89.00	0.79	0.96	89.82	0.79	0.96	88.03	0.79	0.95	87.85	0.79	0.95	88.73	0.80	0.95
	MF	90.15	0.79	0.96	88.82	0.79	0.95	89.60	0.80	0.96	88.44	0.79	0.95	88.78	0.79	0.96
	RFM	88.56	0.79	0.95	89.19	0.80	0.96	90.42	0.81	0.96	88.57	0.79	0.95	91.21	0.81	0.96
	RFCM	90.30	0.80	0.96	90.53	0.80	0.97	91.81	0.78	0.97	90.43	0.82	0.95	90.13	0.77	0.96
NB	Without	65.03	0.75	0.15	64.51	0.75	0.17	64.72	0.75	0.20	62.37	0.74	0.21	62.59	0.73	0.23
	KNN	64.30	0.74	0.17	66.54	0.75	0.18	64.24	0.75	0.19	64.77	0.76	0.18	66.06	0.76	0.18
	Mean	66.20	0.76	0.16	63.36	0.75	0.17	64.67	0.75	0.17	64.48	0.75	0.18	64.10	0.75	0.17
	DL	66.30	0.76	0.17	64.34	0.76	0.16	64.94	0.75	0.17	65.64	0.75	0.16	64.24	0.74	0.18
	Mice	64.03	0.75	0.17	64.31	0.74	0.17	64.02	0.74	0.17	64.63	0.75	0.17	63.62	0.74	0.17
	MF	65.38	0.75	0.17	62.99	0.74	0.19	64.30	0.74	0.17	65.66	0.75	0.16	63.60	0.73	0.17
	RFM	65.84	0.77	0.20	65.38	0.75	0.18	66.87	0.76	0.18	65.38	0.75	0.20	65.57	0.74	0.17
	RFCM	67.33	0.73	0.14	64.74	0.76	0.16	66.42	0.74	0.17	65.67	0.75	0.18	65.53	0.74	0.17
SVM	Without	78.85	0.79	0.85	74.44	0.76	0.80	76.35	0.79	0.82	76.02	0.78	0.81	76.35	0.79	0.80
	KNN	78.66	0.78	0.85	76.44	0.79	0.82	77.01	0.78	0.81	76.25	0.78	0.79	76.26	0.78	0.80
	Mean	76.40	0.77	0.83	77.92	0.78	0.84	77.01	0.79	0.82	75.36	0.77	0.81	78.29	0.79	0.82
	DL	78.34	0.79	0.84	77.83	0.79	0.85	77.97	0.78	0.83	78.48	0.79	0.85	75.72	0.76	0.83
	Mice	77.10	0.77	0.83	75.08	0.79	0.81	75.97	0.78	0.82	77.23	0.79	0.81	77.44	0.77	0.84
	MF	78.38	0.79	0.85	78.93	0.79	0.84	78.63	0.79	0.84	78.69	0.80	0.84	76.82	0.78	0.81
	RFM	80.06	0.81	0.84	81.25	0.81	0.83	82.30	0.81	0.89	80.21	0.78	0.88	79.19	0.81	0.83
	RFCM	79.82	0.78	0.85	79.49	0.79	0.86	78.80	0.80	0.83	77.78	0.79	0.85	78.65	0.80	0.84

As stated in the above comparison between different algorithms according to the accuracy is not adjuster enough in some cases. The AUC parameter is important to differentiate between algorithms honestly. Firstly, as shown in Table 4 the most accurate result (96.15%) is RF for the reasons mentioned in the accuracy section.

Case study with real dataset contains missed values

As pronounced in Table 5 three different sizes of the dataset, four classification algorithms, with six ways of addressing the missing data. From the following cases: the best algorithm is RF, due to it is one of the ensemble learning techniques that consists of several decision trees. Each tree predicts the output. Then makes voting to choose the final

decision. When the dataset size increases, the accuracy of the model increases. As it gives the model more knowledge to learn from it.

Finally, the accuracy of our proposed approaches RFM and RFCM in all machine-learning algorithms are high compared to without any data imputation and the best KNN. As an example, in the case of the RF algorithm, the accuracy of the RF without imputation, KNN, and RFCM are 78.24%, 82.62%, and 92.02% respectively. In all cases the average accuracy of RFCM increases than the best KNN, and without by 10% and 14% respectively. This means that the two proposed approaches in the case of missed data improve the performance. Through this enhancement, human life becomes safe when using our model.

Table 6 Ac, FM, AUC of different algorithms on different dataset sizes using imputation algorithms

Imputation Method		DataSet_size_1			DataSet_size_2			DataSet_size_3		
		Ac %	FM %	AUC %	Ac %	FM %	AUC %	Ac %	FM %	AUC %
DT	Without	74.59	77.27	64.77	75.73	77.25	73.65	75.55	77.21	76.85
	Mean	74.64	76.62	69.75	75.2	77.29	74.04	76.75	77.79	75.54
	KNN	72.56	75.52	58.17	75.38	76.43	69.85	76.18	76.55	65.65
	DL	75.98	78.24	70.27	78.51	77.98	78.19	79.40	78.33	75.96
	Mice	74.50	77.41	69.52	76.17	77.08	76.38	77.18	77.60	77.02
	MF	74.09	76.84	70.58	76.26	76.92	75.18	76.47	76.89	76.68
	RFM	73.36	77.03	63.58	76.16	77.41	68.57	75.74	76.85	64.77
	RFCM	84.07	77.74	81.44	83.5	78.34	83.77	84.01	78.14	85.37
RF	Without	72.53	76.78	81.52	78.03	77.78	89.42	78.24	77.57	90.94
	Mean	74.53	75.65	84.01	81.67	77.04	90.38	82.38	77.47	90.52
	KNN	74.6	75.53	81.9	80.97	76.64	89.5	82.62	77.54	89.79
	DL	78.29	76.69	84.87	87.73	78.96	94.76	90.13	78.93	95.76
	Mice	76.98	76.60	83.18	82.98	77.42	91.24	82.86	76.79	90.93
	MF	76.30	75.91	83.70	82.50	76.87	90.51	83.83	77.26	91.30
	RFM	74.79	75.43	80.91	81	77.2	89.39	82.48	77.62	90.31
	RFCM	87.53	77.35	94.01	91.5	78.49	97.01	92.01	78.39	97.19
NB	Without	60.9	76.82	32.22	61.08	71.08	31.71	59.83	71.05	31.73
	Mean	63.74	77.13	20.83	66.13	75.76	19.35	67	75.8	19.9
	KNN	61.51	76.88	20.76	62.11	73.55	21.29	61.64	73.36	21.01
	DL	66.98	75.33	15.80	64.79	75.63	16.74	65.05	75.23	16.00
	Mice	67.92	72.74	19.41	68.06	73.16	18.59	65.05	75.23	16.00
	MF	67.59	74.54	19.83	68.75	75.18	19.25	76.47	76.89	76.68
	RFM	64.64	73.74	20	67.07	74.52	21.15	64.23	73.99	20.92
	RFCM	81.62	75.75	14.28	79.76	76.71	15.24	80.57	76.27	14.96
SVM	Without	76	70.47	78.68	74.32	77.36	79.64	76	77.62	80.97
	Mean	72.57	74.01	79.14	71.02	77.69	80.58	72.94	77.89	80.43
	KNN	73.68	72.33	80.83	73.49	77.6	80.33	72.88	76.97	79.34
	DL	76.09	78.13	81.65	77.25	78.86	82.52	78.52	78.54	84.49
	Mice	72.82	76.03	80.38	75.38	78.04	81.25	75.22	77.22	82.21
	MF	73.46	76.23	81.46	74.97	77.13	81.24	74.19	76.84	80.67
	RFM	72.18	77.67	78.24	72.51	77.67	77.55	72.25	77.23	78.37
	RFCM	77.46	76.59	83.71	76.72	76.96	82.61	77.29	76.87	83.75

When working on three different sizes of the dataset the first one (DataSet_size_1) is 768 rows, the second one (dataset_size_2) is 2000 rows, and the third one (dataset_size_3) is 2768 rows. The result proves that when the dataset is much larger as possible the accuracy of the classification becomes superior.

F1-measure (FM) as mentioned in Eq. (9), merges between recall and precision metrics. As shown in Table 6, in most cases the RFCM is the best one, as a way of replacing the missed value.

As mentioned the above comparison between different algorithms according to the accuracy is not fair enough in some cases. The AUC parameter is important to distinguish between algorithms reasonably. Firstly, as shown in Table 6 the most accurate result (97.19%) is RF for the reasons mentioned in the accuracy section. Also, when the dataset

size increases the AUC value increase. SVM approximately doesn't change with the size of the dataset as it is based on the edge points (Vectors), so no matter about many other points.

As shown in Fig. 5, when the dataset size increased, the RMSE will be reduced. Increasing the size of the dataset as much as possible helps in reducing the RMSE as produces a good learned model with many more cases. The proposed approach (RFCM) gives the minimum RMSE of all algorithms with all dataset sizes.

Anomaly detection

The outlier (anomaly) is the object which is very diverse from the remaining dataset. Anomaly detection is the process of detecting the odd values which differ from the normal

(a) Dataset size_1

(b) Dataset size_2

(c) Dataset size_3

Fig. 5 RMSE of different algorithms on different dataset sizes by using different imputation algorithms

distribution. There are many approaches to determine and remove the outliers for example hidden Markov Model, which learning the model of the normal patterns, to avoid the others. Sau et al. [35] proposed an outliers detection

technique, based on the minimum distance for the unknown parameters of a parametric density on the unit sphere. The IQR is a measure of unevenness, based on cutting up a data set into quartiles [36]. Quartiles divide a rank-ordered data

Fig. 6 Boxplot and a probability density function of a Normal Population

set into four equal parts. The values that separate parts are called the first, second, and third quartiles; and they are denoted by Q_1 , Q_2 , and Q_3 , respectively. As shown in Fig. 6 by using Eq. 7, determine the outliers of the dataset.

In this work, we apply this concept to interact with the odd values of the dataset, which may lead to low accuracy models. In the healthcare system, anything that may affect the model performance becomes very crucial.

$$IQR = Q_3 - Q_1 \quad (7)$$

To show the effect of outliers and the different ways to repair it, some ideas about that are covered in this work. Some of these are very good in results, and some are not. Finally, we know how to correctly deal with the outliers.

Using IQR to determine the range of accepted data (25%–75% range is the commonest), three different proposed ways to repair outliers; remove the rows which contain values out of the accepted range. Instead of determining the outliers for the entire dataset at one time, which may lead to removing important rows, instead determine the outliers for each class separately. Another approach is replacing the outliers with null values and treating them as missing data.

New techniques to deal with outliers

Two proposed techniques to deal with the outliers to improve the quality of the dataset, and produce a healthcare model with very good accuracy, and save human life.

(A) Detecting the outliers at the level of dataset different classes.

As shown in Fig. 6 the detection of the anomaly values is by calculating the distribution of the entire dataset, then remove the outliers' values according to that. But this may cause a problem due to the values of different classes of the dataset have different outliers calculation.

As clear in Fig. 7 the distribution of the three parts is different, so the anomaly detection on each one will differ from the others. As mentioned above, when determining odd values for each class, gives better performance of the model.

(B) Using the missing data imputation to repair the outliers

In this approach, first of all, repairing the missed data by using the proposed techniques RFCM, then instead of removing the entire records, which contains an anomaly value, replace this value by null, and repair it by using RFCM, DL missing data imputation techniques as shown in Fig. 8. These two techniques are used as giving the best results in the previous work.

As shown in Table 7 there are 4 classification algorithms, 5 methods for treating with the outliers, and 3 metrics to measure the difference in each case.

Finally, we try to interact with outliers in completely different ways, which is based on replacing outliers with a null value and use imputation techniques like RFCM and DL. As shown in the results when repairing outliers instead of removing them, gives model accuracy reaches 97.41%, and AUC is 99.71%. Also, restraint the complete dataset without removing any part of it.

From the previous results, treating with the outliers of each class of the dataset is the best scenario. Determining

Fig. 7 The distribution of the total dataset, the diabetic, and non-diabetic parts

Fig. 8 Outliers repairing by missing data imputation (RFCM, DL)

the outliers at the start is very important. Divide the dataset according to its label class. Choose a suitable algorithm to

Table 7 the accuracy, RMSE, and AUC of different algorithms with different outliers repair

Method		Ac %	RMSE %	AUC %
DT	With outliers	80.81	43.78	81.34
	Entire outliers	81.55	42.88	78.21
	Class outliers removal	83.8	40.2	69.18
	Outliers imputation with RFCM	79.05	45.71	80.08
	Outliers imputation with DL	94.29	23.85	93.71
RF	With outliers	90.22	31.08	96.15
	Entire_outliers	92.11	27.91	97.14
	Class outliers removal	92.46	27.25	97.79
	Outliers imputation with RFCM	90.88	30.03	96.74
	Outliers imputation with DL	97.41	16	99.71
NB	With outliers	64.37	59.64	17.45
	Entire_outliers	68.71	55.91	15.54
	Class outliers removal	69.95	54.75	10.34
	Outliers imputation with RFCM	68.48	56.1	15.64
	Outliers imputation with DL	85.9	37.52	5.21
SVM	With outliers	77.98	46.84	83.91
	Entire_outliers	78.68	46.11	83.48
	Class outliers removal	80.14	44.46	87.52
	Outliers imputation with RFCM	76.55	48.34	82.48
	Outliers imputation with DL	91.1	29.63	96.31

Fig. 9 the RMSE of different imputation algorithms

repair the outliers instead of removing them and lose an important part of the dataset.

As shown in Fig. 9 the mean algorithm gives the minimum RMSE value. If only these values determine the best imputation algorithm, the miss-forest (MF) algorithm is the best. From the previous results of the complete model evaluation of different imputation algorithms, the best imputation algorithm is the one that gives a good model performance. RMSE is calculated by Eq. 5, but the difference here is that $\hat{p}_i$ is the predicted value for the missed item, p_i is the real value of the item before it is removed to generate missed places in a pure dataset from null values.

From the previous experimental results: the proposed imputation algorithms increase the accuracy of the model over the case study datasets by an average 10%. The proposed outliers' approach which implements anomaly detection over each class of the dataset, then repairs them by DL, increases the accuracy by an average + 10%.

Conclusion

The objective of this paper is to present a new MIoT framework which integrates big data analytics, cloud system, and smart wearable device. This framework can be used for diabetes detection to save patients' lives around the world. Within the proposed framework, two new algorithms are proposed to handle missing data from wearable sensors in MIoT. The proposed new algorithms derived and made a big performance enhancement by an average 10% increase in the model accuracy, which the most crucial part of the proposed framework system. Moreover, a web-based diabetes

classification application using the flask-framework is efficiently developed for patient diagnosis. RMSE is not sufficient to evaluate the imputation algorithms. Outliers can be treated as missing data and can be repaired. DL is a very good repairing technique for outliers' imputation (97.41% accuracy and 99.71% AUC).

Acknowledgements This work is funded and supported by the computer science and engineering department, faculty of electronic engineering, Menoufia University.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Sisodia D, Sisodia DS (2018) Prediction of diabetes usingdoi: classification algorithms. *Procedia Comput Sci* 132: 1578–1585, no. Iccids. <https://doi.org/10.1016/j.procs.2018.05.122>.
2. Yücelbaş Ş (2020) Simple Logistic Hybrid system based on greedy stepwise algorithm for feature analysis to diagnose Parkinson's disease according to gender. *Arab J Sci Eng* 0123456789, <https://doi.org/10.1007/s13369-020-04357-1>.
3. Zakir J (2015) Issues in information systems 16, no. Ii, pp 81–90
4. Petrozziello A, Jordanov I, Sommeregger C, Distributed neural networks for missing big data imputation. *Proc Int Jt Conf Neural*

- Networks 2018–July, no. July, 2018, <https://doi.org/10.1109/IJCNN.2018.8489488>.
5. Cappon G, Acciaroli G, Vettoretti M, Facchinetti A, Sparacino G Wearable continuous glucose monitoring sensors: A revolution in diabetes treatment. *Electron* 6(3), 2017, <https://doi.org/10.3390/electronics6030065>.
 6. Georga EI, Protopappas VC, Bellos CV, Fotiadis DI (2014) Wearable systems and mobile applications for diabetes disease management. *Health Technol (Berl)* 4(2):101–112. <https://doi.org/10.1007/s12553-014-0082-y>
 7. Chen M, Ma Y, Song J, Lai CF, Hu B (2016) Smart clothing: connecting human with clouds and big data for sustainable health monitoring. *Mob Networks Appl* 21(5):825–845. <https://doi.org/10.1007/s11036-016-0745-1>
 8. Murali S, Rincon F, Atienza D A wearable device for physical and emotional health monitoring *Comput. Cardiol.* (2010) 42 May 2016, pp. 121–124, 2015, <https://doi.org/10.1109/CIC.2015.7408601>.
 9. Saravana Kumar NM, Eswari T, Sampath P, Lavanya S Predictive methodology for diabetic data analysis in big data. *Procedia Comput Sci* 50:203–208 <https://doi.org/10.1016/j.procs.2015.04.069>.
 10. Kumar PS, Pranavi S Performance analysis of machine learning algorithms on diabetes dataset using big data analytics. 2017 Int. Conf. Infocom Technol. Unmanned Syst. Trends Futur. Dir. ICTUS 2017, vol. 2018-Janua, no. Iddm, pp. 508–513, 2018, <https://doi.org/10.1109/ICTUS.2017.8286062>.
 11. Sanadi RA, Prakash B, Dhainje B (2016) A Survey paper on use of data mining in various fields 7301(12): 11–13
 12. Tang F, Ishwaran H (2017) Random forest missing data algorithms. *Stat Anal Data Min* 10(6):363–377. <https://doi.org/10.1002/sam.11348>
 13. Noor MN, Yahaya AS, Ramli NA., Al Bakri AMM Mean imputation techniques for filling the missing observations in air pollution dataset. *Key Eng. Mater* 594–595, no. February 2016, pp. 902–908, 2013, <https://doi.org/10.4028/www.scientific.net/kem.594-595.902>.
 14. Biessmann F, Salinas D, Schelter S, Schmidt P, Lange D (2018) ‘Deep’ learning for missing value imputation in tables with non-numerical data pp 2017–2025. <https://doi.org/10.1145/3269206.3272005>.
 15. Dzulkalnine MF, Sallehuddin R (2019) Missing data imputation with fuzzy feature selection for diabetes dataset. *SN Appl Sci* 1(4). <https://doi.org/10.1007/s42452-019-0383-x>.
 16. Sherif AS, Harb H, Zaky S (2011) A new data imputing algorithm. *Int J Comput Sci Issues* 8(3):133–139
 17. Bartlett JW, Seaman SR, White IR, Carpenter JR (2015) Multiple imputation of covariates by fully conditional specification: accommodating the substantive model. *Stat Methods Med Res* 24(4):462–487. <https://doi.org/10.1177/0962280214521348>
 18. Cantor M, Sherlock G, Hastie T Missing value estimation methods for DNA microarrays. no. July, 2001, <https://doi.org/10.1093/bioinformatics/17.6.520>.
 19. Choi J, Dekkers OM, le Cessie S (2019) A comparison of different methods to handle missing data in the context of propensity score analysis. *Eur J Epidemiol* 34(1):23–36. <https://doi.org/10.1007/s10654-018-0447-z>
 20. Zhang Z (2016) Big-data Clinical Trial Column Missing data imputation: focusing on single imputation. *Ann Transl Med* 4(1):8. <https://doi.org/10.3978/j.issn.2305-5839.2015.12.38>
 21. Qiu YL, Zheng H, Gavaert O A deep learning framework for imputing missing values in genomic data. *bioRxiv*, p. 406066, 2018, <https://doi.org/10.1101/406066>.
 22. Beaulieu-Jones BK, Moore JH (2017) Missing data imputation in the electronic health record using deeply learned autoencoders * the Pooled Resource Open-Access Als Clinical Trials Consortium †, p. 208, 2017.
 23. Rahman J, Suri HS, Abedin M Accurate Diabetes Risk stratification using machine learning : role of missing value and outliers pp 1–17, 2018.
 24. Yilmaz N, Inan O, Uzer MS (2014) transactional processing systems a new data preparation method based on clustering algorithms for diagnosis systems of heart and diabetes diseases, <https://doi.org/10.1007/s10916-014-0048-7>.
 25. Stekhoven DJ, Bühlmann P (2012) MissForest—non-parametric missing value imputation for mixed-type data 28(1): 112–118 , <https://doi.org/10.1093/bioinformatics/btr597>.
 26. Phung S, Kumar A, Kim J A deep learning technique for imputing missing healthcare data, 2019 41st annual international conference of the IEEE engineering in medicine and biology society (EMBC), Berlin, Germany, 2019, pp 6513–6516.”
 27. Azur PJ, Stuart MJ, Frangakis EA, C., & Leaf, “2011, Multiple imputation by chained equations: what is it and how does it work? *International Journal of Methods in Psychiatric Research*, 20(1): 40–49. doi:<https://doi.org/10.1002/mpr.329>.”
 28. Perveen S, Shahbaz M, Guergachi A, Keshavjee K (2016) Performance analysis of data mining classification techniques to predict diabetes. *Procedia Comput Sci* 82:115–121. <https://doi.org/10.1016/j.procs.2016.04.016>
 29. Orabi KM, Kamal YM, Rabah TM (2016) Early predictive system for diabetes mellitus disease, 9728
 30. Rashid TA, Abdullah S (2015) An intelligent approach for diabetes classification , prediction and description an intelligent approach for diabetes classification , Prediction and Description, no. January 2016, <https://doi.org/10.1007/978-3-319-28031-8>.
 31. Farid DM, Al-Mamun MA, Manderick B, Nowe A (2016) An adaptive rule-based classifier for mining big biological data. *Expert Syst Appl* 64:305–316. <https://doi.org/10.1016/j.eswa.2016.08.008>
 32. Ali Zia U, Khan N (2017) Predicting diabetes in medical datasets using machine learning techniques. *Int J. Sci Eng Res* 8(5):1538–1551
 33. “[https://en.wikipedia.org/wiki/Flask_\(web_framework\)](https://en.wikipedia.org/wiki/Flask_(web_framework)).”.
 34. Sigillito V <https://www.kaggle.com/uciml/pima-indians-diabetes-database>, Research center, RMI Group Leader Applied Physics Laboratory he Johns Hopkins University, "last accessed 18/8/2019" .”.
 35. Sau MF, Rodriguez D (2018) Minimum distance method for directional data and outlier detection. *Adv Data Anal Classif* 12(3):587–603. <https://doi.org/10.1007/s11634-017-0287-9>
 36. Sunitha L, Balraju M, Sasikiran J, Ramana EV (2014) Automatic outlier identification in data mining using IQR in real-time data, 3(6):7255–7257.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations