


RESEARCH

Open Access



Diabetes emergency cases identification based on a statistical predictive model

Kebira Azbeq^{1*} , Mohcine Boudhane², Ouail Ouchetto¹ and Said Jai Andaloussi¹

*Correspondence:
kebira.azbeq-etu@etu.
univh2c.ma

¹ Department
of Mathematics
and Computer Sciences,
Computer Sciences
and Systems Laboratory,
Faculty of Sciences Ain
Chock, Hassan II University
of Casablanca, Casablanca,
Morocco
Full list of author information
is available at the end of the
article

Abstract

Diabetes is a chronic metabolic disease which is characterized by a permanently high blood sugar level. A distinction is made between two forms: Type 1 diabetes and Type 2 diabetes. It is believed that there are around 415 million people between the ages of 20 and 79 worldwide who have some form of diabetes illness today. In Europe, over 60 million people are diabetic, a diabetes incidence of 10.3% of men and 9.6% of women is estimated. The prevalence of diabetes is increasing among all ages in the European Region, mainly due to increases in overweight and obesity, unhealthy diet, and physical inactivity. A huge people in this population have type 2 diabetes, and the numbers will continue to rise over the next few years. So one can speak of a real widespread disease. The problem is not only the increased blood sugar, but also complications and accompanying diseases such as heart attack, stroke, or diabetic foot. However, as a type 2 diabetic, we can significantly influence the course of the disease and the success of therapy. To do this, it is important that we early detect the person that have (or likely have) a serious problem or an emergent case, and know about it as fast as possible. Early detection and treatment of this disease are very important to help diabetics live a healthy and near normal life. It can also help to avoid several serious complications. In addition, the evolution of wearable and Internet of Things medical devices can help to collect various health data for diagnosis using machine learning algorithms. In this paper, we present an IoT-based system architecture which ensures the collection of patient data in order to predict serious cases of diabetes. To secure data, Blockchain and IPFS are used, and to analyze data, we propose a statistical-based method for predictions. The process is as follows. First, data were collected from IoT devices, and a dataset was constructed and stored using IPFS. Then, the data will be scaled and filtered using noise-invariant data expansion. Next, an adaptive random forest algorithm is made in order to train data on the training dataset, and people with diabetes were classified using the proposed model. Three datasets were used, namely, the Pima Indian diabetes dataset, the Frankfurt Hospital diabetes dataset, and the last is the fusion of these two datasets. Finally, the performance of the method was evaluated and compared with other recent prediction methods. Based on the experiment result, an accuracy of 85.9%, 99.5%, and 99.8% has been achieved based on the three datasets, respectively. Thus, the model can be used to predict and alert physicians or hospitals serious cases that need urgent reactions.

Keywords: Diabetes disease, Internet of things (IoT), Machine learning, Medical treatment, Artificial intelligence

Introduction

Diabetes is one of the most common chronic diseases in the world and requires that patients living with it have a continuous self-management to control it. It is characterized by an excess of sugar in the blood and therefore a level of glucose (blood sugar) that is too high. This disease happens either when the pancreas does not produce enough insulin (type 1 diabetes) or when the produced insulin can not be used efficiently by the body (type 2 diabetes) [1, 2]. On the one hand, type 1 diabetes is also known as juvenile diabetes and usually begins before the age of 20 [3]. The cause is an autoimmune disease, i.e. a disease in which the immune system attacks its own body. In type 1 diabetes, insulin-producing cells in the pancreas are destroyed. On the other hand, the much more common form, type 2 diabetes, usually occurs only after the age of 30 and is therefore often referred to as old-age diabetes. However, young people can also be affected. Hereditary factors, obesity and lack of exercise play a role as the cause [4]. According to the global report, published by the World Health Organization (WHO), they estimated that 422 million people in the world were living with diabetes in 2014 compared to 108 million in 1980 [5]. To live healthy with this disease, diabetics should have regular check-ups to control blood glucose to reduce the development of complications and premature death. However, WHO reported also that between 24% and 62% of people with type 2 diabetes were undiagnosed and untreated.

With the advancement of Internet of Things (IoT), especially medical devices, wearable sensors and smartphones the quality of life for diabetic peoples will be improved [6]. It will improve diabetes care by collecting health records, such as blood pressure and glucose levels, regularly to ensure self-management of diabetes. IoT devices can help doctors make better decisions because the data collected by these devices are highly accurate. We can also take advantage of this advancement to collect health data in order to use it to predict future diabetes based on machine learning (ML) algorithms.

The purpose of this study is twofold. On one hand, it aims to help patients with diabetes live better lives by collecting their health data and keeping track of them. On the other hand, it allows healthcare professionals to obtain real-time data on their patients thanks to the use of IoT devices, making it easier to administer the proper medication, intervene and build the right treatment plan. In this paper, we propose a complete architecture which incorporates IoT for health information collection, Blockchain and Interplanetary File System (IPFS) to secure and store data, and ML for predicting possible future diabetes. Based on an adaptive random forest algorithm, we proposed a model and we trained it on two datasets in order to predict diabetes urgent cases.

The rest of the paper is organized as follows: The "[Related work](#)" Section presents the related work. The "[Research methodology](#)" Section explains our research methodology. In "[Theoretical principles](#)" Section, we present the theoretical background needed in our work. In "[The proposed algorithm](#)" Section, we discuss the proposed algorithm. The experimental results are discussed in "[Experimental results](#)" Section. The "[Performance analysis](#)" Section gives the performance analysis and in "[Conclusion and future work](#)" Section, we conclude the paper and introduce some future work.

Related work

Several researches have taken advantage of technological advances such as IoT and AI to improve the healthcare industry. The integration of such technologies in the medical field is promoting the healthcare services by helping both patients and doctors to monitor and manage patients' health situation. We review here some relevant work that used ML and IoT to predict diabetes using the well-known diabetes datasets, Pima Indians diabetes dataset [7] and Hospital Frankfurt Germany diabetes dataset [8] (which are available in the Kaggle data repository). Table 1 highlights the state-of-the-art solutions related to the prediction of diabetes.

Gandhi et al. [9] proposed a system based on feature selection and support vector machine classifier (SVM). Sowjanya et al. [10] proposed a mobile application based solution which used ML algorithm to predict diabetes. Four ML algorithms were tested in their work. After analysis, J48 algorithm demonstrated to give better results compared to others algorithms. Authors in [11] proposed a new methodology, based on novel pre-processing techniques, and K-nearest neighbor classifier (KNN). Komi et al. [12] treated early prediction of diabetes using five data mining algorithms. Based on their experiment result, they proved that artificial neural network (ANN) provided the highest accuracy compared to the others treated algorithms. Kaur et al. [13] proposed a cloud IoT based framework for diabetes prediction. Their solution incorporates wearable devices to collect blood glucose levels and cloud infrastructure for data storage. They used Decision Tree (DT) and ANN models to predict diabetes in patients. Lukmanto et al. [14] proposed a classification framework to detect and classify diabetes using F-Score feature selection and Fuzzy SVM. In [15], the authors had integrated principal component analysis (PCA) and K-means techniques to improve the logistic regression (LR) model for predicting diabetes using electronic health records. Most recently, Pradhan et al. [16] conducted a study to compare 8 different ML algorithms to predict diabetes and concluded that SVM provided the highest accuracy. The authors in [17] used four

Table 1 Comparison summary of the existent solutions

Authors	Year	Methods	Used dataset	IoT
Gandhi et al. [9]	2014	SVM	Pima Indian	No
Sowjanya et al. [10]	2015	DT (J48)	Pima Indian	No
Panwar et al. [11]	2016	KNN	Pima Indian	No
Komi et al. [12]	2017	ANN	Not specified	No
Kaur et al. [13]	2018	DT + ANN	Pima Indian	Yes
Sarwar et al. [20]	2018	KNN	Pima Indian	No
Lukmanto et al. [14]	2019	Fuzzy SVM	Pima Indian	No
Zhu et al. [15]	2019	PCA + K-means + LR	Pima Indian	No
Pradhan et al. [16]	2020	SVM	Pima Indian	No
Naz and Ahuja [17]	2020	DL	Pima Indian	No
Jashwanth Reddy et al. [18]	2020	RF	Pima Indian	No
Nath et al. [19]	2020	LR	Pima Indian	No
Yaganteeswarudu et al. [23]	2020	SVM	Pima Indian / Frankfurt hospital	No
Malik et al. [21]	2021	RF	Frankfurt hospital	No
Beghriche et al. [22]	2021	DNN	Frankfurt hospital	No
Ihnaini et al. [24]	2021	DL	Fusion of the two datasets	No

classification algorithms based on the Pima database to predict diabetes. They showed that among the four algorithms, deep learning (DL) provided the best results for diabetes onset. Another recent work done by Jashwanth Reddy et al. [18] designed models using 6 ML classifiers to predict the diabetes. After comparison, Random Forest (DF) algorithm ranked first. Nath et al. [19] used logistic regression (LR) to predict type 2 diabetes, their model showed an accuracy of 75.32%. The authors in [20], used six different ML algorithms (LR, SVM, Naive Bayes (NB), DT, RF, and KNN) to predict diabetes. After comparison, SVM and KNN achieved the highest accuracy which is 77%.

All the works mentioned above are based on the Pima Indian diabetes dataset which has only 768 instances. Other researches were carried out on a large diabetes dataset provided by Frankfurt Hospital. The authors in [21] provided a comparison of ten different ML algorithms and deduced that KNN, RF, and DT outperformed the other algorithms in terms of all metrics. Beghriche et al. [22] performed a comparison between six well-known ML algorithms and showed that the DNN provided better accuracy. Yaganteeswarudu et al. [23] evaluated several ML algorithms using the two datasets. The results showed that SVM achieved the highest accuracy. Most recently, Ihnaini et al. [24] proposed a healthcare system for diabetes prediction and recommendation based on DL and the fusion of the two datasets.

Deshkar et al. [25] presented a set of applications based on IoT for diabetes management such as web-based services, robot assistant and systems based on mobile health (m-health). They also discussed the major challenges facing the integration of IoT in healthcare applications, namely, security and privacy issues, interoperability and legal regularities. This research study fills these security gaps by using Blockchain and IPFS to secure data and enhance integrity.

Research methodology

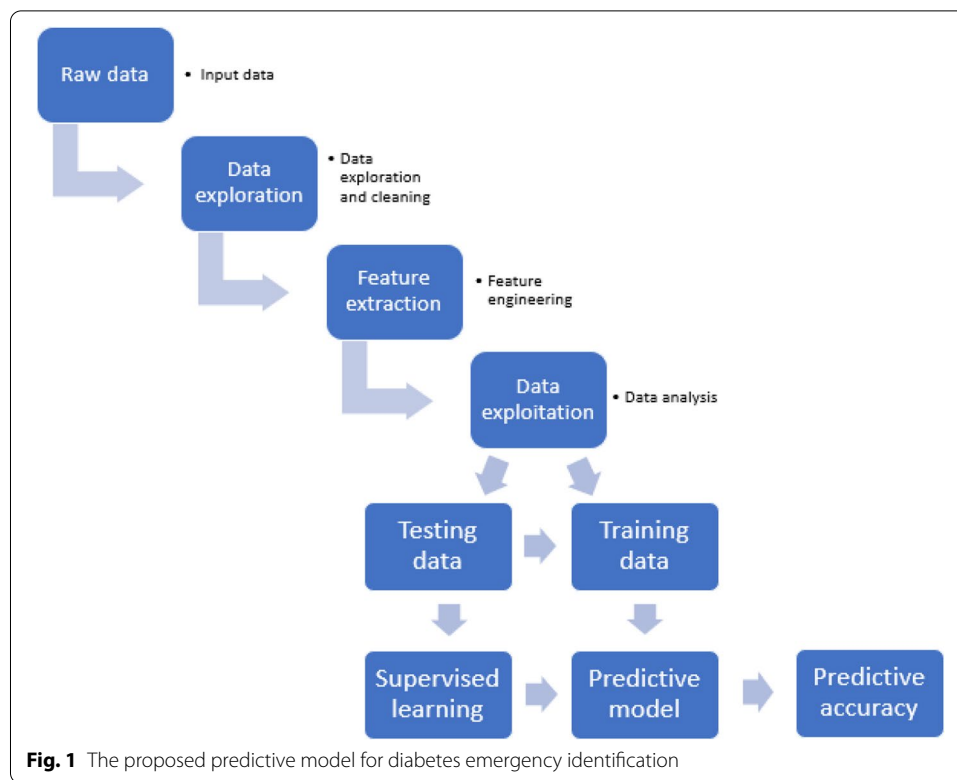
In this section, we first explain the main steps that make up a prediction system. Then we expose the composition of our architecture. In the end, we express the implemented ML algorithms to predict diabetes.

Prediction system

Every prediction system consists of three main steps: Extraction, Risk and Alert. Data extraction is the first step and it refers to the process of collecting or retrieving different types of data from different sources for further processing, storage and analysis. Once the data is collected and stored, it can be analyzed and explored to get all the relevant information. In this step, different algorithms can be created in order to predict and classify risks, suggest solutions and propose decisions. The second step is about risk prediction, the system predicts the state from the created algorithms, then a reaction will be done according to the gravity of the situation and an alert will be sent in case of a risk (serious situation). In the last step, when an alert is triggered, the appropriate decision will be taken based on the created classification decisions.

Prediction model overview

Figure 1 illustrates the model used for prediction. The system is divided into seven layers: input data, data exploration, feature extraction, data exploitation, training and



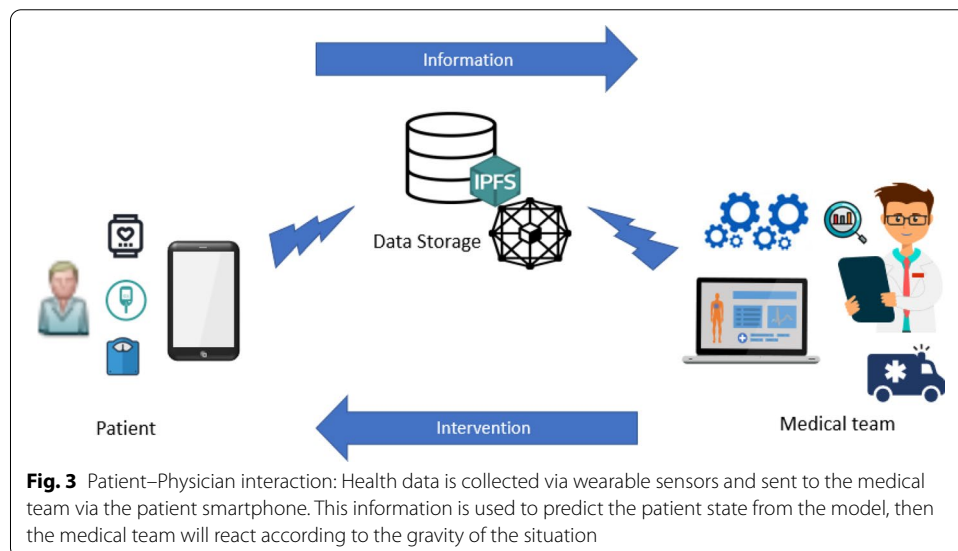
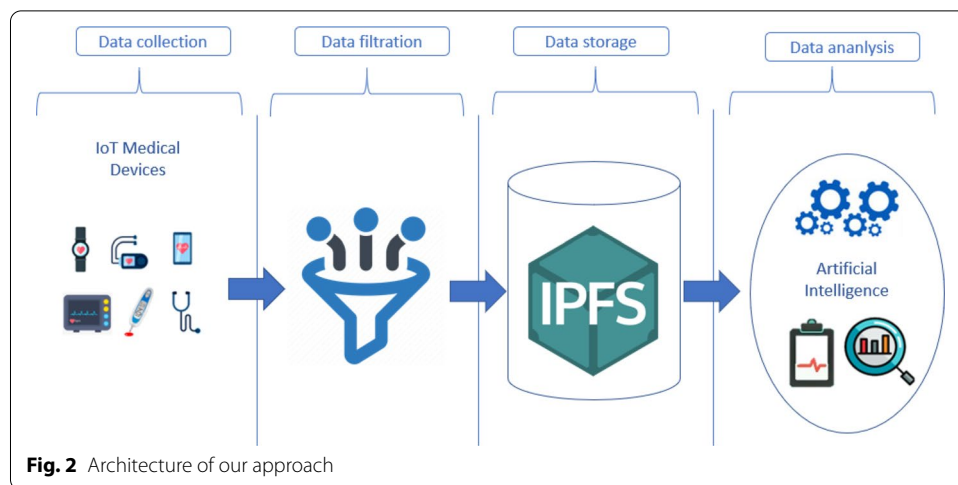
testing data, predictive model, and predictive accuracy. The system has two phases—training phase and application phase. Expert knowledge is used to develop class labels in the training phase. The relationships between features are defined according to historical data. Then ML algorithms process the data. Within this phase, we used 80% of the dataset. In the application phase, the remaining 20% of data is used for checking the algorithm accuracy. The predictive model is applied to test if these people would develop the diabetes disease. By using this model high-impact factors can be recognized to help organizations to focus their strategies and decisions on most relevant issues. Each phase is important because it has a huge impact on the final accuracy.

Proposed approach

Our proposed architecture summarizes the prediction process in several steps (Fig. 2).

Data collection

Data collection is carried out by a set of medical devices that collect health data such as blood pressure, glucose level, sleep patterns, heart rate, and patient weight. These features are necessary to track patient's health. In the proposed solution, we developed a mobile application that collects health data from patient's devices and allows people to enter their medical data (such as medical test results) to store them as a csv file and send them to the medical team using their smartphones. Figure 3 explains the interaction between the patient and the medical team.



Data filtration

- (a) *Dealing with dataset errors* In this phase, we try to keep track and take a look at the patterns where most errors originate, as this will make it much easier to recognize the correction of erroneous or degenerate information. This is especially important if we need to integrate different arrangements in the model, that is, the framework programs, so that mistakes do not end with a mistake made by different divisions.
- (b) *Standardize your processes* It is important to normalize the waypoint and check its meaning. By institutionalizing all information procedure, we ensure a valid transit declaration and reduce the risk of duplication.
- (c) *Approve the accuracy* We approve the accuracy of all the information after we cleaned up the current database. Precisely, we find out and put resources into information devices that allow us to gradually clean up the information. Currently, many devices even use AI to test accuracy even more.

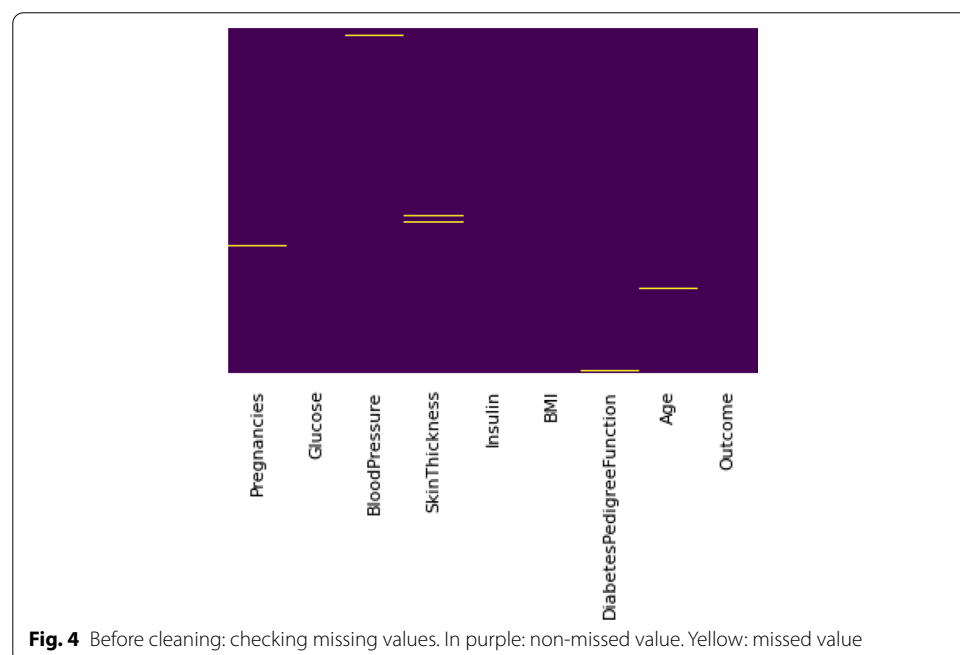
- (d) *Check for duplicate data* Recognize the copies, as this will save you time when reviewing the information. Therefore, you can keep a strategic distance from the exploration and use of resources in various information cleaning instruments, such as those mentioned above, which can mass decompose raw information and automate the procedure.

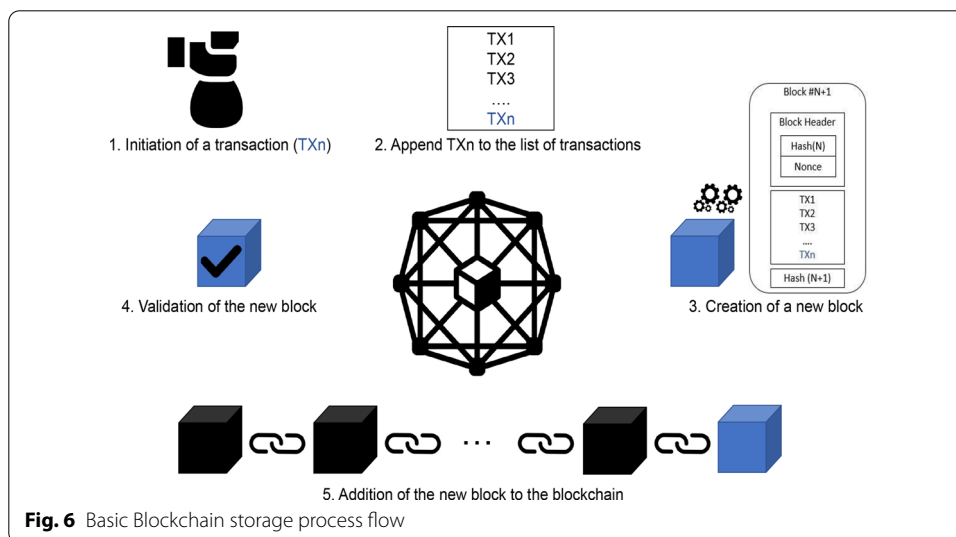
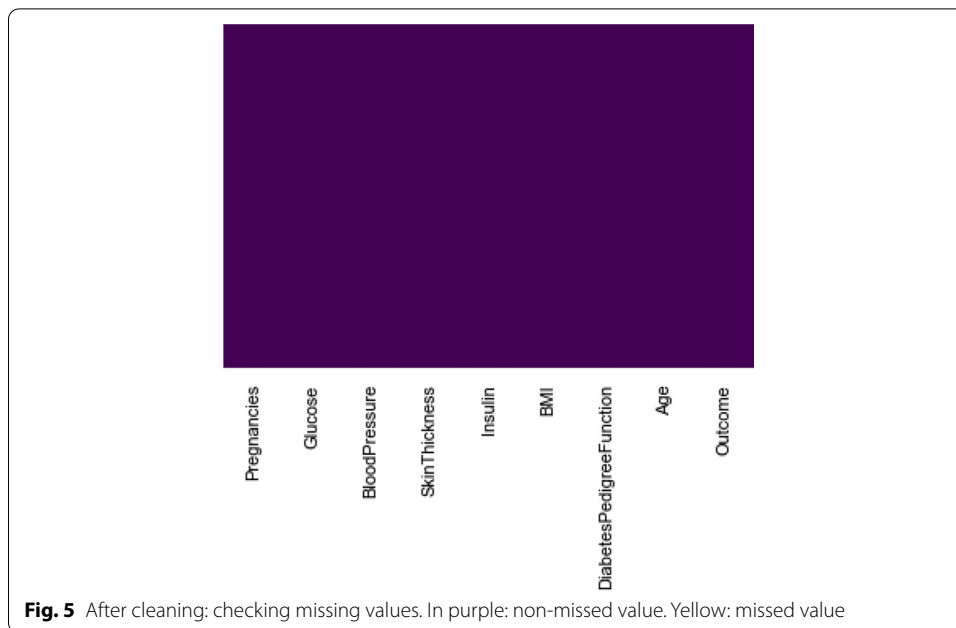
Figure 5 illustrates data representation after the filtration process. In this phase, the data is explored and cleaned. Categorical (non-numeric) values were converted to binary fields. Also, non-desirable features were deleted. After exploring the data, no missing values were found. Figures 4 and 5 use an indicator of missing (or NAN) values. Overall, it can be seen that there are no missing values on the dataset after the cleaning process.

Data Storage

IPFS is a protocol built on top of BitTorrent [26] and Kademlia DHT [27]. It is based on a distributed system for storing and sharing data in a peer-to-peer network. It uses content-addressing to give a unique identifier for each file. In our system, once data is collected and filtered, we store it in the IPFS in order to ensure distributed storage and high storage throughput and to avoid single point of failure. The data will then be the subject of further analysis using ML algorithms. Another interest of using the IPFS for storage is to be able to combine this work with another previous work that uses Blockchain for security concerns [28]. The idea, is to use Blockchain for storing the hash of data stored in the IPFS to ensure the integrity.

Figure 6 illustrates an overview of the storage process adopted in the Blockchain technology. Five basic steps were presented :





- Step 1: First of all, the transaction (TX_n) is initiated and broadcasted to the Blockchain network for treatment. In our case, this transaction will contain the hash of the data.
- Step 2: Once the transaction is received, the nodes participating in the Blockchain network authenticate the digital signature and append the transaction to the previous ones.
- Step 3: Each node tries to create a new block that contains the set of transactions. In the case of Bitcoin, for example, a block is generated each 10 minutes. The process used to create a block in Bitcoin is called proof of work (PoW) and it consists

in using the computing power to solve a mathematical problem. When a node succeeds in creating a block, it sends it to the entire network for verification.

- Step 4: The network verifies the generated PoW to check if the solution is correct or not.
- Step 5: If the given solution is correct, then the block is added to the Blockchain.

Data analysis

The Data analysis consists of analyzing data of each patient in order to diagnostically predict whether the patient will develop diabetes in the future. In this work, our ML models are trained and tested using a real diabetes database. Furthermore, the data used in this investigation was acquired from two open databases 'Pima Indians diabetes database' (in the next, we will call it 'Dataset1') and 'Hospital Frankfurt Germany diabetes dataset' (which will be called 'Dataset2'). The sample is 768 and 2000, respectively, with a total of 9 attributes (768, 9) and (2000, 9). Since the two datasets have the same features, we merged them to create a third dataset. The merged dataset consists of 2768 cases with eight features and we will denote it 'Dataset3'.

Attributes include several descriptive measures such as glucose, blood pressure, insulin, age, etc. The features are presented in the table below (Table 2). Patients in the two databases are women over 21 years of age. The implemented models are explained above, and the accuracy of results is compared in the "Discussion" Section.

The data is spitted in two big groups: diabetic and non diabetic cases. Figures 7 and 8 illustrate the distribution of the population in each group of study. The figure 7 shows that the Indian dataset contains 500 diabetic and 268 non-diabetic persons. Figure 8 shows that the Frankfurt dataset contains 1316 diabetic and 684 non-diabetic persons.

Once we get the data, we need to analyze the distribution in terms of each feature. In other words, we need to study the distribution of the population of each characteristic used as parameter of prediction. Based on the Dataset1, the results are represented in Figure 9. In this last, the features are shown respectively from the top left as Age, BMI, Blood pressure, Diabetes Pedigree Function, Glucose, Insulin, Outcome, Pregnancies, and Skin Thickness.

Table 2 Features used in this method to predict diabetes patient in strong situation

Features	Description	Index
Pregnancies	Number of times pregnant	1
Glucose	Plasma glucose concentration 2 hours in oral glucose tolerance test	2
Blood Pressure	Diastolic blood pressure (mm Hg)	3
Skin Thickness	Triceps skin fold thickness (mm)	4
Insulin	2-Hour serum insulin (mu U/ml)	5
BMI	Body mass index (weight in kg/(height ² in m)	6
Diabetes Pedigree Function	Diabetes pedigree function	7
Age	Age (years)	8

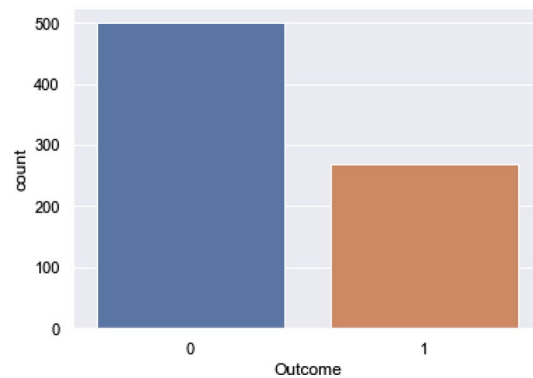


Fig. 7 Population distribution of the Dataset1 0: diabetic cases, 1: non-diabetic cases

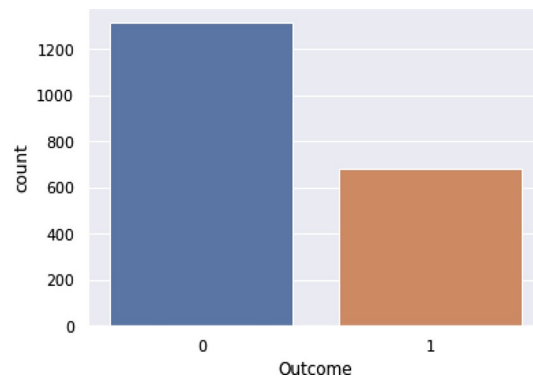


Fig. 8 Population distribution of the Dataset2 0: diabetic cases, 1: non-diabetic cases

Data exploitation and analysis

In this phase, we need to compare the features with each other. This step is very important because it shows how react conjoint features. It can obtain the statistic link inter-features. Figure 10 resumes the distribution of each with all rest of features using the Dataset1.

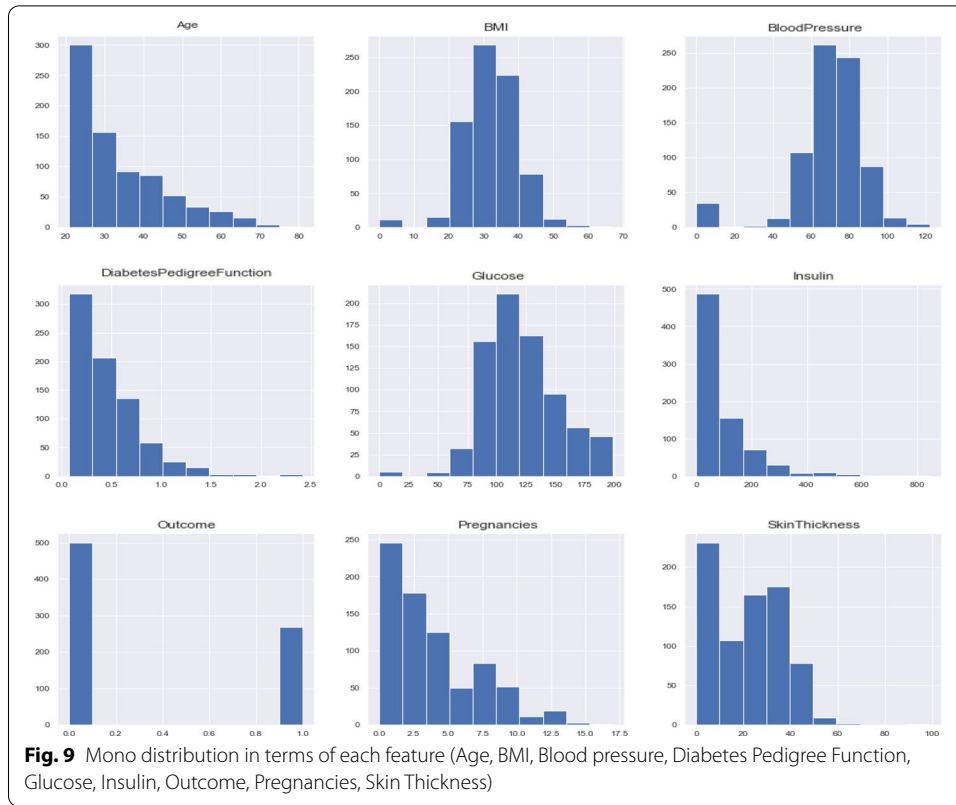
Theoretical principles

Theoretical approach

The objective of this work is to build a predictive model in emergency cases caused by diabetes. The predictive model used in this study is the RF algorithm.

DT and RF are now standard methods in supervised learning. They offer many advantages (wide applicability, ease of use, good performance, etc.) and are now commonly used in many fields, particularly in healthcare purposes [29, 30] or to make predictions/estimates [31, 32]. Before presenting these methods in more detail, we define the mathematical framework within which this research fits.

Let $D_n = (X_1, Y_1), \dots, (X_n, Y_n)$ n independent copies of the pair of random variables (X, Y) . The pair (X, Y) is independent of D_n and its law is unknown. D_n represents the training set.



Denote by X and Y the measurable spaces in which the random variables X and Y respectively live. In this manuscript, we consider the case $X = R_d$. The variable $X = (X_1, \dots, X_d)$ denotes the vector of explanatory variables and Y is the response variable.

In this article, we consider the case of supervised classification where Y denotes the class with $Y = 1, \dots, K, K \geq 2$, and f^* is the Bayes classifier (unknown), defined on X by :

$$f^*(x) = \operatorname{argmax}_{(k \in 1, \dots, K)} \mathbb{P}[Y = k | X = x] \quad (1)$$

In each context, the problem is to estimate the link between the vector X and the response variable Y , that is, to estimate the function f^* from the data of the training sample D_n . An estimator of f^* is a measurable function

$$\hat{f} : (X \times (X \times Y))^n \Rightarrow Y \quad (2)$$

which, for any new observation x , predicts the value of the response Y by $\hat{f}(x, D_n)$. In the following, we will note for convenience $\hat{f}(x)$. The function \hat{f} is called a prediction rule or a decision rule. A set of reference books deals with the issue of supervised learning; see, for example [33].

In many problems in supervised learning, the explanatory variables can have a group structure. The grouping of variables can be natural or well defined to capture / model the relationships between the different variables. The explanatory variables can

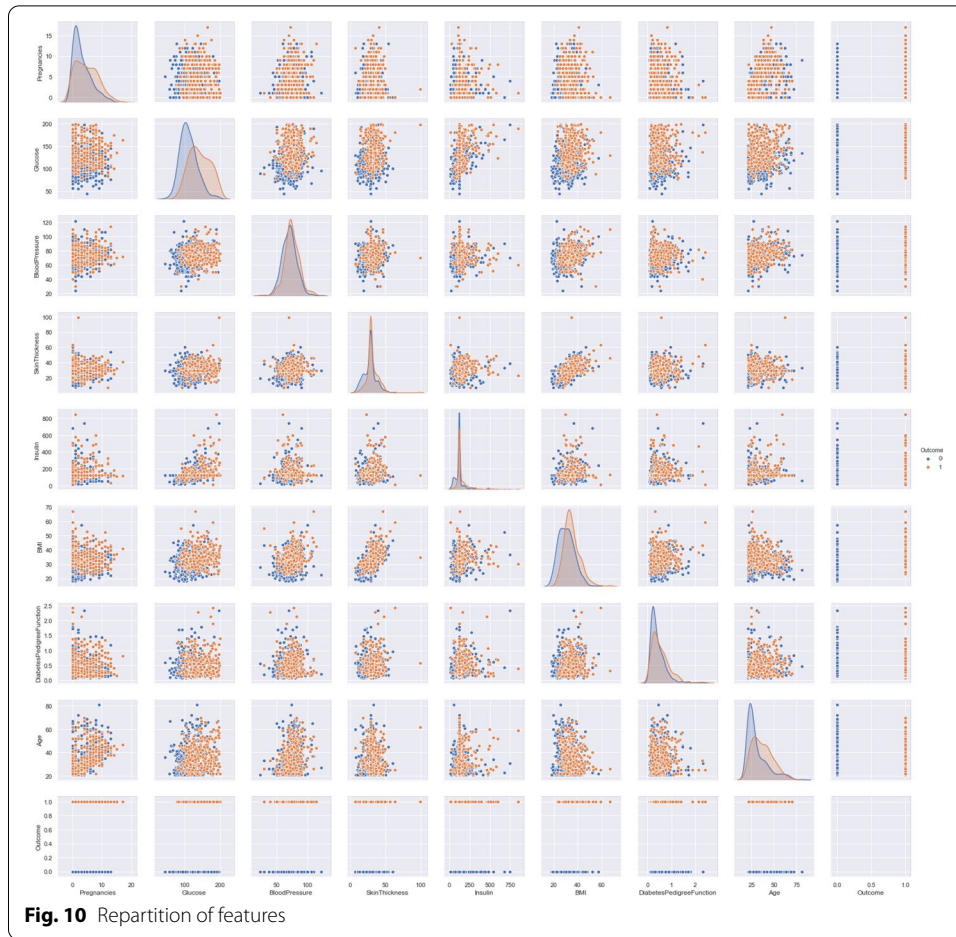


Fig. 10 Repartition of features

act in groups on the response variable. Thus, the exploitation of such a structure can be very useful for building a prediction rule.

In this work, we are interested in the case where the vector X is structured in J known groups. Each one represents a group of patients. We define the j – th group $X_j, j = 1, \dots, J$, by:

$$X_j = (X_{j1}, X_{j2}, \dots, X_{jd_j}), \quad (3)$$

where the set $j_1, j_2, \dots, j_{d_j} \subseteq 1, \dots, d$ denotes the d_j index of the explanatory variables belonging to the group $j, d_j \leq d$. Note that the groups are not necessarily disjoint. The objective is to use this structure to build a prediction rule \hat{f} .

Data pre-processing

In the previous phases, we have improved the modeling and made it more relevant. Modeling emphasizes the development of predictive or descriptive models according to the previously defined analytical approach. Usually, the modeling process is extremely iterative, as you start with the first version of the prepared dataset, get interim analyzes, and use them to refine the preparation of model data and specifications.

This phase takes advantage of mathematical tools such as statistics, correlations, and visualizations, which can be used to communicate major trends within the data exploitation. A statistical description describes a large and complex dataset using a few key figures. But it is dangerous to rely only on statistical descriptions and ignore the overall distribution.

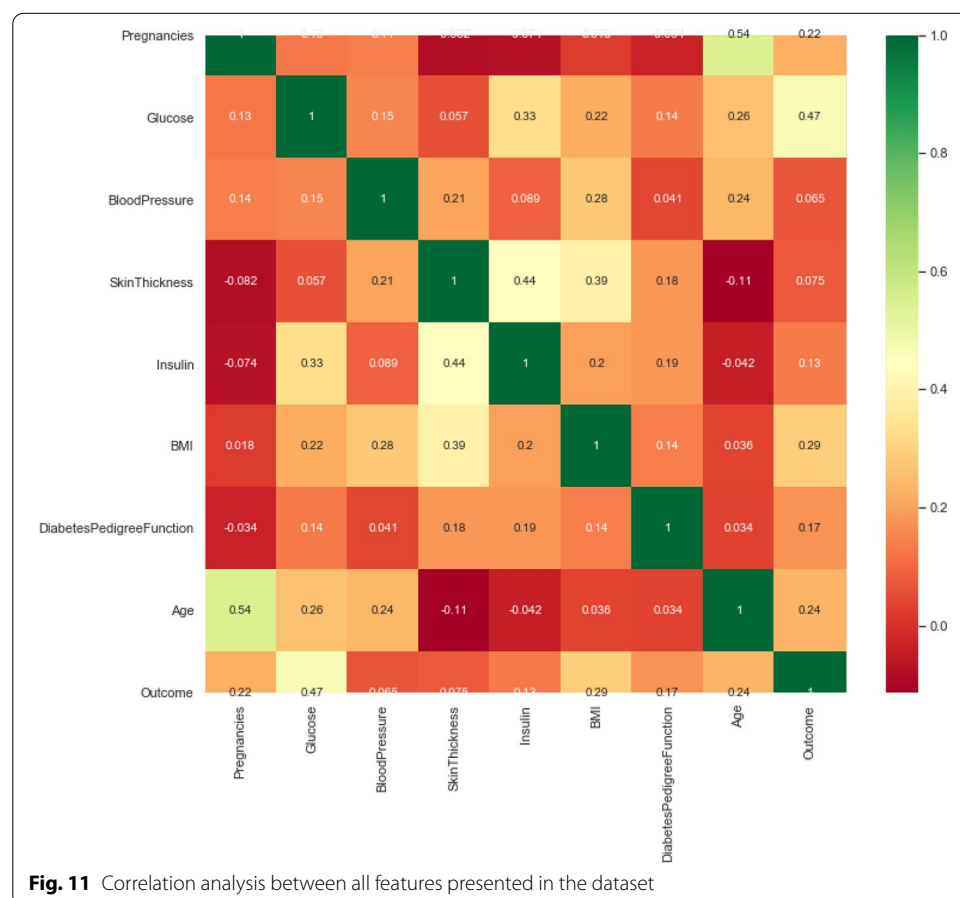
In Fig. 11, variables were compared and the correlation hypothesis was tested using Dataset1. From the analysis of correlations several hypotheses were proposed about the underlying generation of the dataset. This analysis gave the initial idea of the dataset. As the main focus of this study is emergency diabetes cases identification, the highest correlations with emergency cases measures within the dataset were extracted.

The initial analysis proposes that the most important factor affecting diabetes is glucose level. After data training and the application of the best performing ML algorithm, these results will be compared.

The proposed algorithm

Decision tree

The DT CART is a classification and prediction tool. Its popularity is largely based on its simplicity. A decision tree is made up of a root node through which data is entered, leaf nodes which correspond to a classification of questions, and answers which condition

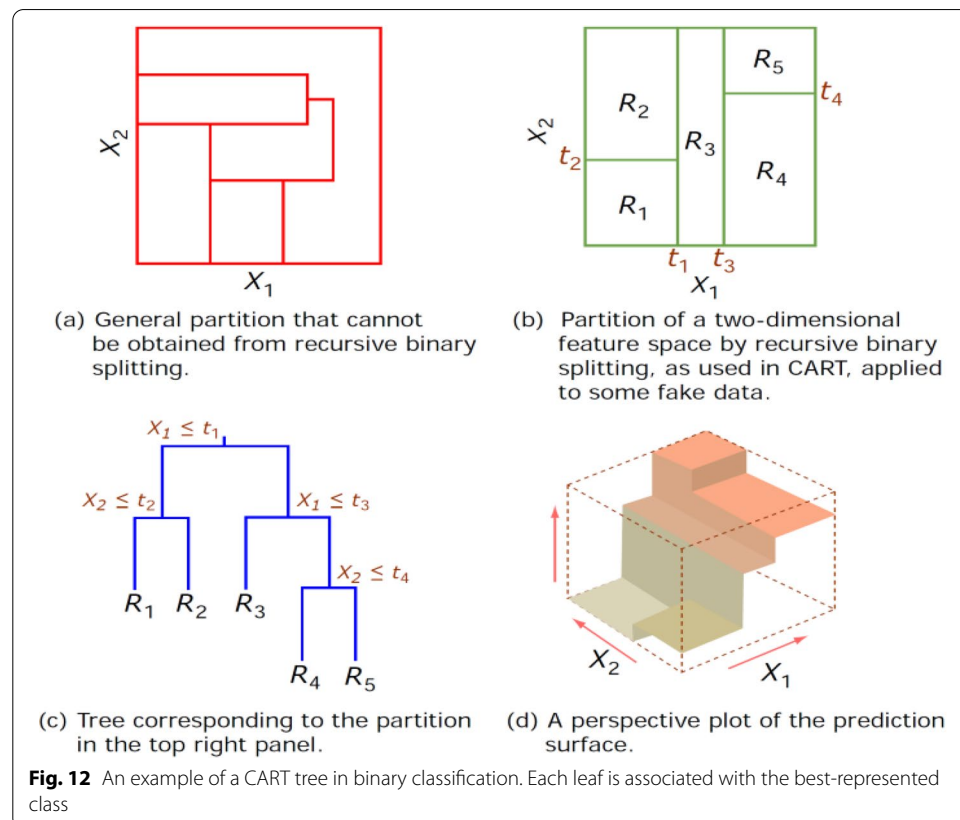


the following question. It is an interactive rule-inducing process that leads to a well-justified assignment. The connection of the nodes involves the calculation of various criteria according to the chosen algorithm. There are different algorithms for building decision trees such as ID3, C4.5, CHAID, and CART. This last is represented in the Fig. 12. To construct a CART tree from the data of the training sample D_n , the algorithm proceeds in two steps.

- Step 1: Development of a maximal tree. This step consists of a recursive and dyadic partitioning of the X data space.
- Step 2: Pruning and selection of the final tree. The often too complex maximum tree T_{max} is generally not optimal within the meaning of a chosen performance criterion (for example in classification, classification error). An excessive number of cuts results in a tree that tends to over-adjust.

Random forest

This is an approach based on RF algorithm that consists of aggregating a collection of estimators constructed from bootstrap samples. A RF is an aggregation of random trees. The principle of building a forest is first of all to independently generate a large number (denoted $ntree$) of bootstrap samples $D_n^1, \dots, D_n^{ntree}$ by randomly drawing, for each of them, observations (with or without replacement) in the training



sample D_n . Then, $ntree$ decision trees T^1, \dots, T^{ntree} are built from the bootstrap samples $D_n^1, \dots, D_n^{ntree}$ and using a variant of CART. In fact, each tree is here constructed as follows. To split a node, the algorithm chooses randomly and without replacement a number $mtry$ of explanatory variables, then it determines the best cut only according to the selected $mtry$ variables. In addition, the constructed tree is fully developed and is not pruned. The RF, denoted by T_n^1 , is finally obtained by aggregating the $ntree$ trees thus constructed. It defines a prediction rule that corresponds to the empirical mean of the predictions in regression and the majority vote in classification. The construction of Breiman's RF is described in Fig. 13 (and later in Algorithm Fig. 13).

Experimental results

In this context, we take into consideration several parameters in the proposed algorithm.

- The number of trees in the forest. Its default value is 500. Note that this parameter is not really a parameter to calibrate in the sense that a larger value of this parameter will always lead to more stable predictions than a smaller value of this parameter.

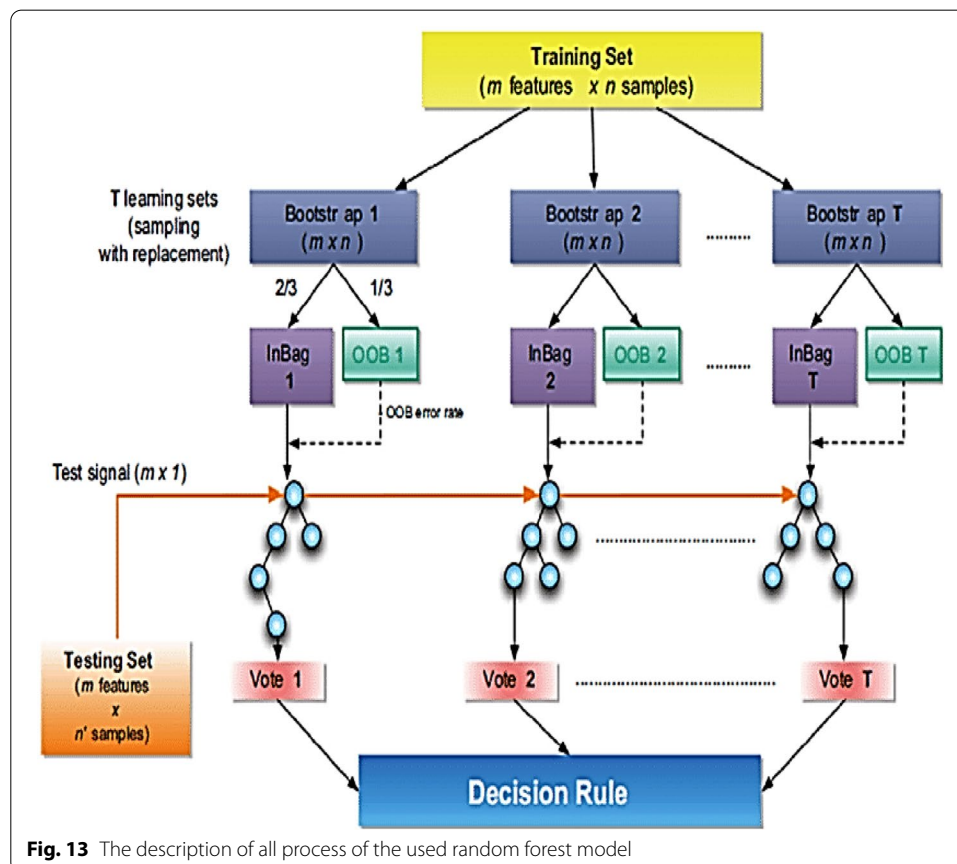


Fig. 13 The description of all process of the used random forest model

- *mtry* number of variables chosen for the division of each node. This is arguably the most important parameter to calibrate as it can greatly influence the performance of the forest.
- *nodesize* minimum number of observations below which a node is no longer split. In this context, the default value for this parameter is *nodesize* = 1.
- *an* The number of observations per time unit in each bootstrap sample. By default, each bootstrap sample contains *an* = *n* observations drawn with replacement in the initial sample D_n .

Several authors have been interested in the choice and influence of these parameters ([33] in 2018 and [34] in 2016). In general, the default values of the parameters work well. Indeed, there are few theoretical results available for the Breiman RF. We can nevertheless cite a major result recently established in [35] and focusing on the convergence of RF in the additive model. Theoretical guarantees have also been obtained for simplified versions of the method [36]. A summary of the main theoretical results is illustrated in the algorithm in Fig. 14.

Model configuration

The model uses two phases: the training phase and the classification phase (testing phase). Algorithm in Fig. 15 illustrates the update of algorithm in Fig. 14 using the proposed approach.

The model validation technique used for this particular dataset is decomposed as follows: 80% of data is used for training and 20% is held out for testing. After validating the data and training the models optimal configuration these models were tested on 20% holdout sample. The algorithm is tested in (RAM: 8Gb, CPU: Intel core i5-8250U CPU 1.80 GHz). The implementation code is published publicly [37].

```

Procedure: BuildForest( $\mathcal{D}$ ,  $E$ ,  $p$ )
Data: Training set  $\mathcal{D}$ , ensemble size  $E$ , number of queries per sub-sample  $p$ 
Result: Tree ensemble  $Trees$ 
begin
   $Trees \leftarrow \emptyset$ ;
  for  $i \in \{1, \dots, E\}$  do
     $\mathcal{D}_i \leftarrow \emptyset$ ;
    while  $|\mathcal{Q}_{\mathcal{D}_i}| < p$  do
       $q \leftarrow \text{chooseRandom}(\mathcal{Q}_{\mathcal{D}} \setminus \mathcal{Q}_{\mathcal{D}_i})$ ;
       $\mathcal{D}_i.add(\langle x_{q,j}, l_{q,j} \rangle_{j=1}^{n_q})$ ;
    end
     $Trees.add(\text{BuildTree}(\mathcal{D}_i))$ ;
  end
  return  $Trees$ ;
end

Where the function  $\text{chooseRandom}(A)$  selects an item uniformly at random from the set  $A$ .

```

Fig. 14 The proposed model based on random forest algorithm

Training Phase

Given

- X : the objects in the training data set (an $N \times n$ matrix)
- Y : the labels of the training set (an $N \times 1$ matrix)
- L : the number of classifiers in the ensemble
- K : the number of subsets
- $\{\omega_1, \dots, \omega_c\}$: the set of class labels

For $i = 1 \dots L$

- Prepare the rotation matrix R_i^a :
 - Split F (the feature set) into K subsets: $F_{i,j}$ (for $j = 1 \dots K$)
 - For $j = 1 \dots K$
 - * Let $X_{i,j}$ be the data set X for the features in $F_{i,j}$
 - * Eliminate from $X_{i,j}$ a random subset of classes
 - * Select a bootstrap sample from $X_{i,j}$ of size 75% of the number of objects in $X_{i,j}$. Denote the new set by $X'_{i,j}$
 - * Apply PCA on $X'_{i,j}$ to obtain the coefficients in a matrix $C_{i,j}$
 - Arrange the $C_{i,j}$, for $j = 1 \dots K$ in a rotation matrix R_i as in equation (1)
 - Construct R_i^a by rearranging the columns of R_i so as to match the order of features in F .
- Build classifier D_i using $(X R_i^a, Y)$ as the training set

Classification Phase

- For a given x , let $d_{i,j}(x R_i^a)$ be the probability assigned by the classifier D_i to the hypothesis that x comes from class ω_j . Calculate the confidence for each class, ω_j , by the average combination method:

$$\mu_j(x) = \frac{1}{L} \sum_{i=1}^L d_{i,j}(x R_i^a), \quad j = 1, \dots, c.$$

- Assign x to the class with the largest confidence.

Fig. 15 The proposed algorithm

Table 3 Description of the confusion matrix

TP	FP
FN	TN

Performance analysis

It is very important to test, measure, and monitor the performance of a predictive model before and after deploying it to production. We must then define the measures to be used for the evaluation of this performance.

Confusion matrix

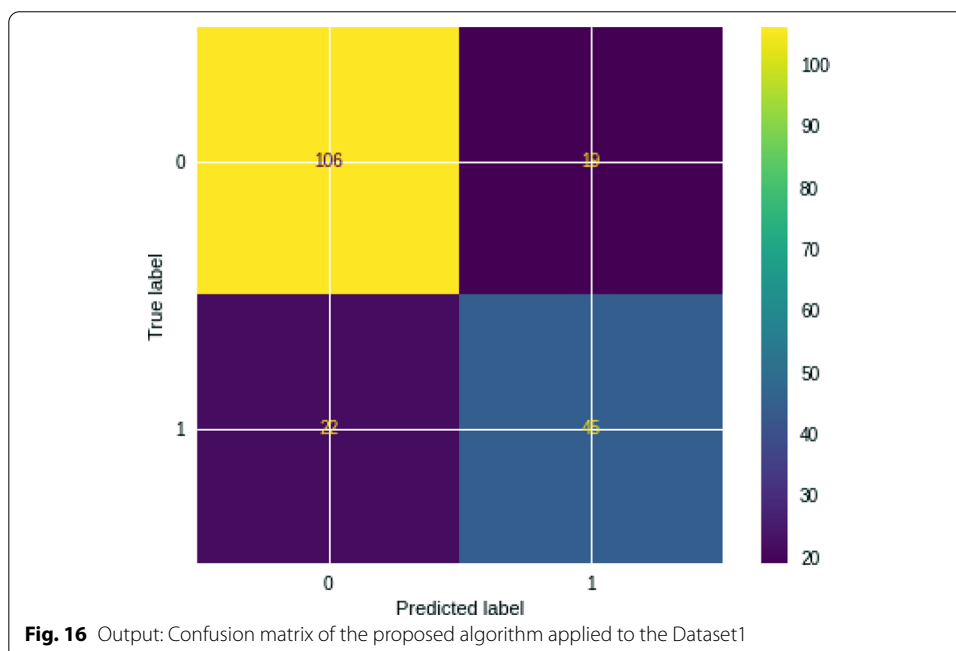
This is an array of size $n \times n$ to visualize the results of predictive models for classification problems. Where n is the number of classes in our datasets. In this matrix, the real target classes are crossed with the predicted classes obtained (see Table 3). This gives

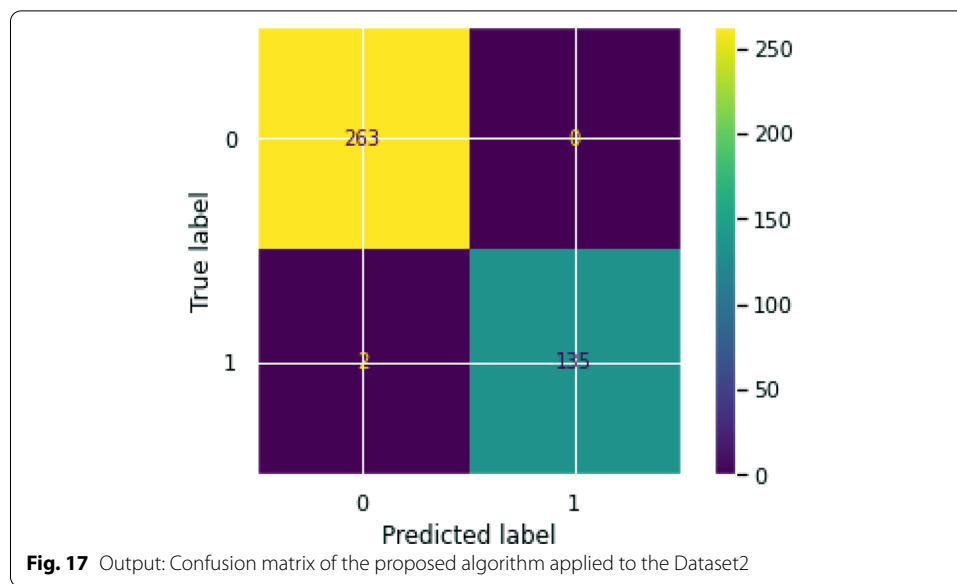
us the number of instances that are correctly classified and the number of misclassified instances.

- TP: this is the number of True Positives, the number of correctly classified positive instances,
- FP: this is the number of False Positives, the number of instances which are not positive and which are predicted to be positive,
- FN: this is the number of False Negatives, the number of non-negative instances classified as negative,
- TN: this is the number of True Negatives, the number of correctly classified negative instances.

In Fig. 16 the confusion matrix of the proposed algorithm applied to the Dataset1 and in Fig. 17 the confusion matrix of the proposed algorithm applied to the Dataset2. They show results for predicted and actual labels. True Positive (TP) presents patients who define the most urgent cases and were correctly determined by the algorithm. True Negative (TN) of patients who present dangerous cases and were correctly determined by the algorithm. False Negative (FN) are the cases that the algorithm predicted they would need to be treated as emergency cases without presenting such dangerous case. And finally, the False Positive (FP) presents patients who need to be alerted as emergency cases, and the algorithm does not. In this case, we must rescale our data so that it fulfills these requirements. The accuracy given by the proposed algorithm was around 85.9% for the Dataset1, 96.4% for the Dataset2, and 99.8% for the Dataset3.

The algorithm places great importance on the 'Glucose' feature, but it also chooses 'BMI' to be the second most informative feature overall. The randomness in building the algorithm forces the algorithm to consider many possible explanations, the result being



**Table 4** Summary of the different algorithms on Dataset1

Authors	Year	Algorithm-based	Accuracy
Yu et al. [41]	2010	SVM	73%
Panwar et al. [11]	2016	KNN	78%
Ramezankhani et al. [39]	2016	DT	74%
Mingqi et al. [40]	2020	ADABOOST	79.2%
Pradhan et al. [42]	2020	ANN	80.4%
Tigga et al. [38]	2021	LR	75.32%
Ihnaini et al. [24]	2021	DL	72.7%
Our proposition	2022	RF	85.9%

that the proposed algorithm captures a lot of details compared to the conventional RF algorithm. That is due to the ability of the proposed approach to deal with the perturbations that affect generate errors in predictions. It helps more to select and reject unimportant features in the model. To demonstrate more the performance of the proposed algorithm, we compare the results with different algorithms. Next section proceed the comparison of the proposed approach against recent prediction methods for diabetes identification.

Comparison of different ML algorithms

In this part, we give a selection of the most relevant works for predicting diabetes. We also give a comparison of the different ML algorithms used in these works. Tables 4, 5, and 6 give a summary of the studied works and Fig. 18 shows a comparison of the proposed method with recent prediction methods.

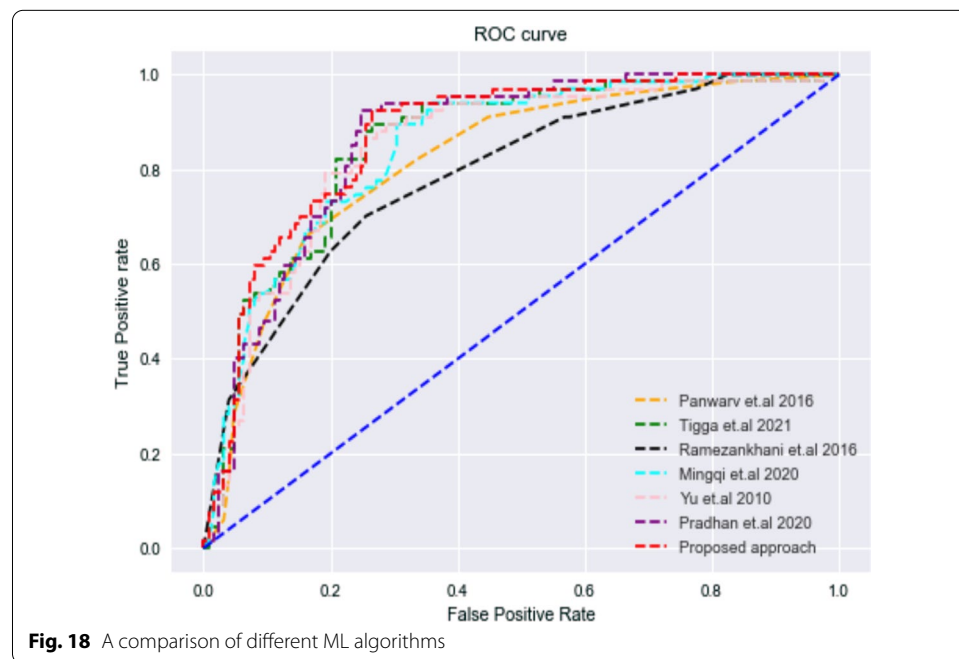
In Panwar et al. [11], the authors utilize KNN, The algorithm is arguably a ML algorithm which is considered to be the simplest. Building the model consists only of storing the training dataset. To make a prediction for a new data point, the algorithm looks for the closest data points in the training dataset, its “nearest neighbors”.

Table 5 Summary of the different algorithms on Dataset2

Authors	Year	Algorithm-based	Accuracy
Malik et al. [21]	2021	RF	98.8%
Ihnaini et al. [24]	2021	DL	91%
Our proposition	2022	RF	99.5%

Table 6 Summary of the different algorithms on Dataset3

Authors	Year	Algorithm-based	Accuracy
Yaganteeswarudu et al. [23]	2020	SVM	92%
Ihnaini et al. [24]	2021	DL	99.6%
Our proposition	2022	RF	99.8%

**Fig. 18** A comparison of different ML algorithms

The precision was around 78%. Tigga et al. [38] use LR, a popular machine learning classification algorithm to predict the risk of type 2 diabetes among individuals. The aim of this study is to improve prediction so that the LR algorithm can be used on any dataset to give a result with good accuracy 75.32%. Ramezankhani et al. [39] used a large population-based sample in their study. Direct measurements of glucose value and anthropometric indices were used rather than self-reported information for predictor variables and outcome. The study proposes an approach for detecting interactions between predictors. There were no data available on the dietary intake among the participants. The average rate is 74%. Another work by Mingqi et al. [40] utilizes the ADABOOST algorithm to perform diabetes dataset. To reduce overfitting, they apply a prior-adjustment of data by limiting the maximum depth or lower the

learning rate. Through the comparative analysis with the integrated algorithm, they propose an improved feature combination algorithm based on XGBoost. The prediction rate was about 79.2%. The authors in Ref. [41] used SVM algorithms in order to predict diabetes. The SVM models were used to select sets of variables that would yield the best classification of individuals into these diabetes categories. The model overfits quite substantially, with a perfect score in the training set and only a rate of 65% in the testing phase. SVM requires all the features to vary on a similar scale. They need to rescale our data that all the features are approximately on the same scale. The obtained accuracy using the same dataset was 73%. In [42], the accuracy of the MLP is not as good as the other models at all, this is likely due to scaling of the data. ANN also expects all input features to vary in a similar way and ideally to have a mean of 0, and a variance of 1. The obtained overall accuracy by Pradhan et al. [42] is 80.4%. Tables 4, 5, and 6 resume this comparative study based on Dataset1, Dataset2, and Dataset3, respectively. These tables clearly show that the proposed method outperforms the other methods.

Discussion

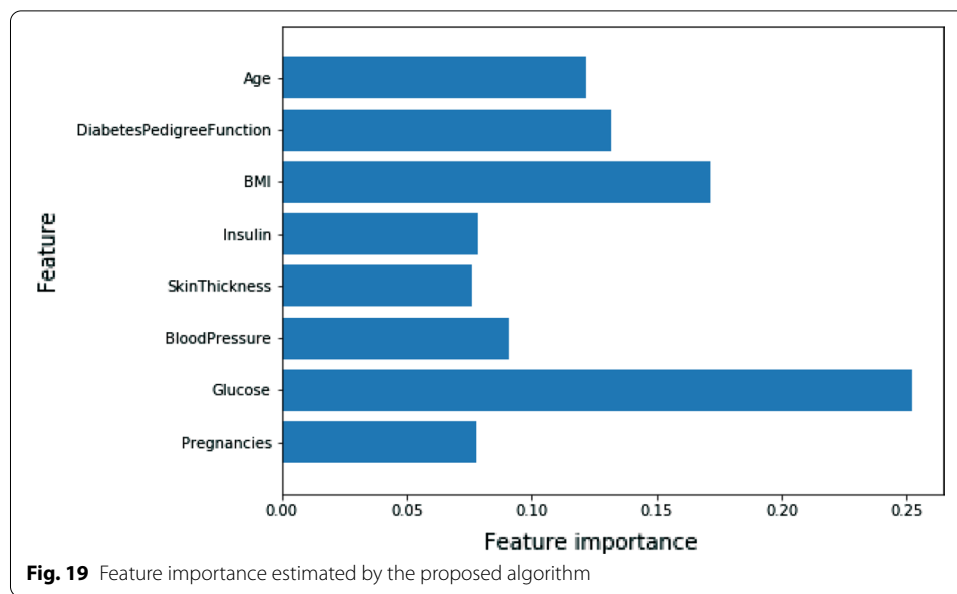
The proposed algorithm was found to be the best prediction algorithm for the dataset used (Dataset1) compared to the other algorithms cited previously. But it has the disadvantage of being more difficult to interpret. In order to overcome this, several indexes of the importance of the variables are defined. These scores make it possible to establish a hierarchy of explanatory variables based on the importance in relation to answer Y . The proposed approach mainly offers two criteria: the importance of Gini and the importance by permutation.

The importance of a variable is first assessed in each tree in the forest. Thus, for a given tree, it corresponds to the overall reduction in impurity, that is, the weighted sum of the reductions in impurity induced when the variable is used to cut a node from the said tree. The Gini importance of a variable is then defined by the average (over all trees in the forest) of the overall impurity reductions. In other words, the permutation importance index is based on the idea that an explanatory variable can be considered important in predicting the Y response if breaking the link between this variable and the Y response deteriorates the quality of the prediction. In this sense, random permutations of the values of the variable are used to mimic the breaking of this link. Formally, the calculation of the measure of importance by permutation for a variable X_j (with $j = 1, \dots, d$) consists first of all in defining the out-of-bag (OOB) sample associated with each sample bootstrap [43].

These steps are repeated on all the trees in the forest. The importance index then corresponds to the average over all trees of the increase in error:

$$\mathcal{I}_{perm}(X_j, \{T^b\}_{ntree}^1) = \frac{1}{ntree} \sum_{b=1}^{ntree} \mathcal{R}(T^b, \bar{D}_n^b) - \mathcal{R}(T^b, \bar{D}_n^{bj}) \quad (4)$$

If the random permutation of the j -th variable induces a large increase in the error then $\mathcal{I}_{perm}(X_j, \{T^b\}_{ntree}^1)$ is large and the variable is considered important. Conversely, if the perturbations do not affect the error, then the permutation importance index of X_j is close to zero and the variable is considered unimportant in predicting the response Y . feature

**Table 7** Relevant feature predicted by random forest

Features	Index
Glucose	1
BMI	2
Diabetes Pedigree Function	3
Age	4
Blood pressure	5
Pregnancies	6
Insulin	7
Skin thickness	8

importance rates how important each feature is for the decision a tree makes. It is a number between 0 and 1 for each feature, where 0 means “not used at all” and 1 means “perfectly predicts the target”. The feature importance always sum to 1. In this context, Fig. 19 illustrates the importance of each feature according to the dataset used.

According to the proposed algorithm, the most important features are Glucose, BMI, DPF (Diabetes Pedigree Function) and the Age. Only glucose was recognized as the top features. high that represent high performance in prediction. The proposed algorithm shows the importance and efficiency of ML Algorithm in predicting diabetes and having results that are more precise. The value calculated by the proposed algorithm are listed in Table 7.

Conclusion and future work

Diabetes is a chronic disease which causes a lot of death per year. The number of people living with diabetes is increasing each year. It is clear now that the early detection and management of diabetes is the only solution to have a nearly normal life with this illness. The early detection of diabetes is then a very important step for the management

and treatment of diabetes. In this work, we proposed a novel method to efficiently predict diabetes based on a statistical predictive model using the two well-known databases, namely, Pima Indians diabetes dataset and Hospital Frankfurt Germany diabetes dataset. In addition, we created a third dataset by merging these two datasets. The experimental results show that the proposed algorithm outperforms the state of the art methods in terms of accuracy. Based on an adaptive random forest algorithm, the proposed model achieved an accuracy of 85.9% based on the Pima Indian dataset, 99.5% based on the Frankfurt dataset, and 99.8% on the merged dataset. That is due to the process made for data management, storage, and analysis. The system integrates the IoT medical devices with Blockchain and IPFS to securely collect and store data. That makes the mechanism more robust and consistent. At the end, an alarm is transferred to the persons responsible to deal with the urgent cases.

In future work, we plan to address this problem by designing training algorithms that are more suitable for specific city by gathering data from a big area and alert the nearest hospital/family doctor by means of real time monitoring.

Abbreviations

WHO: World Health Organization; IoT: Internet of things; AI: Artificial Intelligence; ML: Machine learning; IPFS: Interplanetary File System; SVM: Support vector machine; KNN: K-nearest neighbor; ANN: Artificial Neural Network; DT: Decision tree; PCA: Principal component analysis; LR: Logistic regression; DL: Deep learning; DF: Random forest; NB: Naive Bayes; PoW: Proof of work; TP: True Positives; FP: False Positives; TN: True Negatives; FN: False Negatives; OOB: Out of Bag; DPF: Diabetes Pedigree Function.

Acknowledgements

Not applicable.

Authors' contributions

All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

Not applicable. For any collaboration, please contact the authors.

Declarations

Ethics approval and consent to participate

The author confirms the sole responsibility for this manuscript. The author read and approved the final manuscript.

Consent for publication

The authors hereby consent to the publication of the work in the Journal of Big Data.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Department of Mathematics and Computer Sciences, Computer Sciences and Systems Laboratory, Faculty of Sciences Ain Chock, Hassan II University of Casablanca, Casablanca, Morocco. ²Scio-Technical Systems Engineering Institute, Vidzeme University Applied Sciences, Valmiera, Latvia.

Received: 26 October 2021 Accepted: 18 February 2022

Published online: 09 March 2022

References

1. Diabetes. <https://www.who.int/news-room/fact-sheets/detail/diabetes> Accessed 27 Jan 2022.
2. CDC: What is Diabetes? (2021). <https://www.cdc.gov/diabetes/basics/diabetes.html> Accessed 2022-01-27.
3. Type 1 diabetes - symptoms and causes. <https://www.mayoclinic.org/diseases-conditions/type-1-diabetes/symptoms-causes/syc-20353011> Accessed 27 Jan 2022.

4. Type 2 diabetes - symptoms and causes. <https://www.mayoclinic.org/diseases-conditions/type-2-diabetes/symptoms-causes/syc-20351193> Accessed 27 Jan 2022.
5. Organization WH. Global Report on Diabetes. Geneva: WHO Press, World Health Organization. 2016. OCLC: 948336981.
6. Longva AM, Haddara M. How can IoT improve the life-quality of diabetes patients? MATEC Web Conf. 2019;292:03016. <https://doi.org/10.1051/mateconf/201929203016>.
7. Data-society: Pima Indians Diabetes Database - dataset by data-society. 2015. <https://data.world/data-society/pima-indians-diabetes-database> Accessed 9 Mar 2021.
8. Diabetes. <https://kaggle.com/johndasilva/diabetes> Accessed 23 Jan 2022.
9. Gandhi Khyati K. (2014) Diabetes prediction using feature selection and classification. International Journal of Advance Engineering and Research Development. 1(5) 2. <https://doi.org/10.21090/IJAERD.0105110>
10. Sowjanya K, Singhal A, Choudhary C. MobDBTest: A machine learning based system for predicting diabetes risk using mobile devices. In: 2015 IEEE international advance computing conference (IACC). New york: IEEE. pp. 397–402. 2015. <https://doi.org/10.1109/IADCC.2015.7154738>. <http://ieeexplore.ieee.org/document/7154738/> Accessed 12 Dec 2020.
11. Panwar M, Acharyya A, Shafik RA, Biswas D. K-nearest neighbor based methodology for accurate diagnosis of diabetes mellitus. In: 2016 Sixth international symposium on embedded computing and system design (ISED). New york: IEEE. 2016. pp. 132–136. <https://doi.org/10.1109/ISED.2016.7977069>. <http://ieeexplore.ieee.org/document/7977069/> Accessed 12 Dec 2020.
12. Komi M, Li J, Zhai Y, Zhang X. Application of data mining methods in diabetes prediction. In: 2017 2nd International conference on image, vision and computing (ICIVC). New york: IEEE. pp. 1006–1010. 2017. <https://doi.org/10.1109/ICIVC.2017.7984706>. <http://ieeexplore.ieee.org/document/7984706/> Accessed 12 Dec 2020.
13. Kaur P, Sharma N, Singh A, Gill B. CI-DPF. A cloud IoT based framework for diabetes prediction. In: 2018 IEEE 9th annual information technology, electronics and mobile communication conference (IEMCON). Newyork: IEEE. pp. 654–660. 2018. <https://doi.org/10.1109/IEMCON.2018.8614775>. <https://ieeexplore.ieee.org/document/8614775/> Accessed 11 Dec 2020.
14. Lukmanto RB, Suhajito Nugroho A, Akbar H. Early detection of diabetes mellitus using feature selection and fuzzy support vector machine. Procedia Comput Sci. 2019;157:46–54. <https://doi.org/10.1016/j.procs.2019.08.140>.
15. Zhu C, Idemudia CU, Feng W. Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques. Inform Med Unlocked. 2019;17:100179. <https://doi.org/10.1016/j.imu.2019.100179>.
16. Pradhan R, Aggarwal M, Maheshwari D, Chaturvedi A, Sharma D.K. Diabetes mellitus prediction and classifier comparative study. In: 2020 International conference on power electronics and IoT applications in renewable energy and its control (PARC). IEEE: Newyork. pp. 133–139. 2020. <https://doi.org/10.1109/PARC49193.2020.236572>. <https://ieeexplore.ieee.org/document/9087108/> Accessed 11 Dec 2020.
17. Naz H, Ahuja S. Deep learning approach for diabetes prediction using PIMA indian dataset. J Diabetes Metab Disord. 2020;19(1):391–403. <https://doi.org/10.1007/s40200-020-00520-5>.
18. Reddy JD, Mounika B, Sindhu S, Reddy TP, Reddy NS, Sri GJ, Swaraja K, Meenakshi K, Kora P. Predictive machine learning model for early detection and analysis of diabetes. Mater Today. 2020. <https://doi.org/10.1016/j.matpr.2020.09.522>.
19. Nath V, Mandal, JK. (eds.): Proceedings of the Fourth International conference on microelectronics, computing and communication systems: MCCS 2019. Lecture Notes in Electrical Engineering, vol. 673. Singapore: Springer. 2021. <https://doi.org/10.1007/978-981-15-5546-6>.
20. Sarwar MA, Kamal N, Hamid W, Shah MA. Prediction of Diabetes Using Machine Learning Algorithms in Healthcare. In: 2018 24th International conference on automation and computing (ICAC), pp. 1–6. IEEE, Newcastle upon Tyne, United Kingdom 2018. <https://doi.org/10.23919/ICAC.2018.8748992>. <https://ieeexplore.ieee.org/document/8748992/> Accessed 23 Jan 2022.
21. Malik S, Harous S, El-Sayed H. Comparative analysis of machine learning algorithms for early prediction of diabetes mellitus in women. In: International symposium on modelling and implementation of complex systems. Cham: Springer. 2020; p. 95–106.
22. Beghriche T, Djerioui M, Brik Y, Attallah B, Belhaouari SB. An efficient prediction system for diabetes disease based on deep neural network. Complexity. 2021;2021:1–14. <https://doi.org/10.1155/2021/6053824>.
23. Yaganteeswarudu A, Dasari P. Diabetes analysis and risk calculation-auto rebuild model by using flask api. In: Yaganteeswarudu A, editor. International conference on image processing and capsule networks. Cham: Springer; 2020. p. 299–308.
24. Ihnaini B, Khan MA, Khan TA, Abbas S, Daoud MS, Ahmad M, Khan MA. A smart healthcare recommendation system for multidisciplinary diabetes patients with data fusion based on deep ensemble learning. Comput Intell Neurosci. 2021;2021:1–11. <https://doi.org/10.1155/2021/4243700>.
25. Deshkar S. A review on IoT based m-Health systems for diabetes. Int J Comput Sci Telecommun. 2017;8(1):6.
26. Legout A. Understanding bittorrent: an experimental perspective. Vol. 17. 2005.
27. Maymounkov P, Mazières D. Kademlia: A peer-to-peer information system based on the XOR metric. In: Druschel P, Kaashoek F, Rowstron A (eds.) Peer-to-Peer Systems. Series title: lecture notes in computer science. Cham: Springer vol. 2429, pp. 53–65. 2002. https://doi.org/10.1007/3-540-45748-8_5. http://link.springer.com/10.1007/3-540-45748-8_5 Accessed 13 Dec 2020.
28. Azbeg K, Ouchetto O, Andaloussi SJ, Fetjah L, Sekkaki A. Blockchain and IoT for security and privacy: a platform for diabetes self-management. In: 2018 4th International conference on cloud computing technologies and applications (Cloudtech). New york: IEEE pp. 1–5. 2018. <https://doi.org/10.1109/CloudTech.2018.8713343>. <https://ieeexplore.ieee.org/document/8713343/> Accessed 20 Dec 2019.
29. Panda S, Panda G. Intelligent classification of iot traffic in healthcare using machine learning techniques. In: 2020 6th International Conference on control, automation and robotics (ICCAR). New york: IEEE. 2020. pp. 581–585.
30. Pratt M, Boudhane M, Cakula S. Predictive data analysis model for employee satisfaction using ml algorithms. In: Pratt M, editor. Advances on smart and soft computing. Singapore: Springer; 2021. p. 143–52.

31. Lan H, Pan Y. A crowdsourcing quality prediction model based on random forests. In: 2019 IEEE/ACIS 18th International conference on computer and information science (ICIS). New York: IEEE. 2019. pp. 315–319.
32. Sinha NK, Khulal M, Gurung M, Lal A. Developing a web based system for breast cancer prediction using xgboost classifier. *Int J Eng Res*. 2020.
33. Genuer R, Poggi J-M, Tuleau-Malot C, Villa-Vialaneix N. Random forests for big data. *Big Data Res*. 2017;9:28–46. <https://doi.org/10.1016/j.bdr.2017.07.003>.
34. Biau Gérard SE. A random forest guided tour. *TEST*. 2016. <https://doi.org/10.1007/s11749-016-0481-7>.
35. Gao W, Zhou Z-H. Towards convergence rate analysis of random forests for classification. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, editors. *Advances in neural information processing systems*, vol. 33. Red Hook: Curran Associates Inc.; 2020. pp. 9300–11. <https://proceedings.neurips.cc/paper/2020/file/6925f2a16026e36e4fc112f82dd79406-Paper.pdf>. Accessed 2 Feb 2022.
36. Fromont LA, Royle P, Steinhauer K. Growing random forests reveals that exposure and proficiency best account for individual variability in I2 (and I1) brain potentials for syntax and semantics. *Brain Lang*. 2020;204:104770. <https://doi.org/10.1016/j.bandl.2020.104770>.
37. KebAz: KebAz/DiabetesPrediction. original-date: 2022-02-07T14:19:51Z. 2022. <https://github.com/KebAz/DiabetesPrediction>. Accessed 20 February 2022.
38. Tigga NP, Garg S. Predicting type 2 diabetes using logistic regression. In: *Proceedings of the fourth international conference on microelectronics, computing and communication systems*, 2021. Cham: Springer. pp. 491–500.
39. Ramezankhani A, Hadavandi E, Pournik O, Shahrabi J, Azizi F, Hadaegh F. Decision tree-based modelling for identification of potential interactions between type 2 diabetes risk factors: a decade follow-up in a middle east prospective cohort study. *BMJ Open*. 2016;6(12):e013336.
40. Li M, Fu X, Li D. Diabetes prediction based on xgboost algorithm. In: *IOP conference series: materials science and engineering*. Bristol: IOP Publishing. 2020. vol. 768, p. 072093.
41. Yu W, Liu T, Valdez R, Gwinn M, Khoury MJ. Application of support vector machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes. *BMC Med Inform Decis Mak*. 2010;10(1):1–7.
42. Pradhan N, Rani G, Dhaka VS, Poonia RC. Diabetes prediction using artificial neural network. In: Pradhan N, editor. *Deep learning techniques for biomedical and health informatics*. Amsterdam: Elsevier; 2020. p. 327–39.
43. Janitza S, Hornung R. On the overestimation of random forest's out-of-bag error. *PloS ONE*. 2018;13(8):0201904.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)