Advances in Computational Intelligence (2023) 3:5
https://doi.org/10.1007/s43674-022-00051-x

ORIGINAL ARTICLE

# A transparent machine learning algorithm to manage diabetes: TDMSML

Amrit Kumar Verma[1] · Saroj Kr. Biswas[1] · Manomita Chakraborty[2] · Arpita Nath Boruah[3]

Received: 19 June 2022 / Revised: 22 December 2022 / Accepted: 23 December 2022 / Published online: 10 February 2023


## Abstract

Diabetes is nowadays a very common medical problem among the people worldwide. The disease is becoming more prevalent with the modern and hectic lifestyle followed by people. As a result, designing an adequate medical expert system to assist physicians in treating the disease on time is critical. Expert systems are required to identify the major cause(s) of the disease, so that precautionary measures can be taken ahead of time. Several medical expert systems have already been proposed, but each has its own set of shortcomings, such as the use of trial and error methods, trivial decision-making procedures, and so on. As a result, this paper proposes a Transparent Diabetes Management System Using Machine Learning (TDMSML) expert system that uses decision tree rules to identify the major factor(s) of diabetes. The TDMSML model comprises of three phases: rule generation, transparent rule selection, and major factor identification. The rule generation phase generates rules using decision tree. Transparent rule selection stage selects the transparent rules followed by pruning the redundant rules to get the minimized rule-set. The major factor identification stage extracts the major factor(s) with range(s) from the minimized rule-set. These factor(s) with certain range(s) are characterized as major cause(s) of diabetes disease. The model is validated with the Pima Indian diabetes data set collected from Kaggle.

**Keywords** Data mining · Decision tree · Rule pruning · Expert system · Diabetes management

## 1 Introduction

Diabetes Mellitus (DM), also known as Diabetes, is a chronic disease characterized by elevated blood glucose levels (hyperglycemia). Diabetes Mellitus is classified into two types: Type 1 Diabetes Mellitus (T1DM) and Type 2 Diabetes Mellitus (T2DM) (T2DM). T1DM raises blood glucose levels due to insufficient insulin hormone secretion by the pancreas, but it is manageable with an ongoing supply of insulin hormone and blood glucose testing equipment. Type 2 Diabetes Mellitus (T2DM) occurs when blood glucose becomes resistant to insulin hormone, and T2DM increases the risk of Cardiovascular Disease (CVD). T2DM accounts for the vast majority of Diabetes cases, despite being largely preventable. The International Diabetes Federation (IDF) (International diabetes federation (IDF) diabetes atlas 2017) presents the most recent data on diabetes in 2017 and reports that approximately 425 million people aged 20–79 are estimated to have diabetes worldwide (T2DM).According to the IDF, if current trends continue, the number of people aged 20–79 will reach 629 million by 2045. In general, one out of every ten people will be affected (International diabetes federation (IDF) diabetes atlas 2017).To manage the severe impact of Diabetes on humanity, this paper proposes a Transparent Diabetes Management System (TDMSML), which can assist doctors and the general public in managing the severe impact of Diabetes on humanity.

With the advent of the internet and efficient communication, the medical science industry has amassed a massive

✉ Manomita Chakraborty
mou.look@gmail.com

Amrit Kumar Verma
amritkumar.verma8@gmail.com

Saroj Kr. Biswas
saroj@cse.nits.ac.in

Arpita Nath Boruah
arpita.boruah@hotmail.com

[1] Computer Science and Engineering Department, National Institute of Technology Silchar, Silchar, Assam, India

[2] School of Computer Science and Engineering, VIT-AP, Amravati, Andhra Pradesh, India

[3] Computer Science and Engineering, Alliance University, Bangalore, Karnataka, India

amount of relevant and valuable data that has not been properly mined and organized for optimal use. The discovery of hidden patterns and their relationships in these data sets is frequently underutilized and unknown. Fortunately, many data mining techniques, such as Artificial Neural Network (ANN), Support Vector Machine (SVM), Decision Tree (DT), and others, are available to extract useful knowledge from data. However, the majority of the techniques are black-box and cannot be used to make decisions with explanation. However, Decision Tree (DT) (Sankaranarayanan 2014) is a transparent data mining technique that identifies the major constituents of a decision and explicitly explains how a decision is made by generating user understandable decision rules. As a result, the proposed Transparent Diabetes Management System (TDMSML) employs Decision Tree (Sankaranarayanan 2014) to identify the key factors causing diabetes.

The decision tree algorithm is used to generate the rule-set at the very beginning. Following that, the transparent and significant rules are chosen from the extracted rule-set, and the redundant rules are pruned to yield a minimized rule-set using the proposed Sequential Transparent Floating Forward–Rule Pruning Algorithm (STFF–RPA).This STFF–RPA algorithm is designed based on the Sequential Floating Forward Search (SFFS) (Lv et al. 2015) technique. However, it has been observed that sometimes the SFFS technique is not able to detect some subsets of rules that could outperform the current subset of rules. To address this issue, the proposed SDTFFS algorithm incorporates a step known as 'Deep Transparency Search' (DTS) to allow these potential subsets to be found. During the rule selection step, the decision criteria for selecting a rule is based on the value of Significance of Rule (SOR), which is calculated using two formulas shown in Eqs. 6 and 7. Once the transparent rule-set is obtained, the algorithm may extract some redundant rules, increasing the difficulty in understanding the major attributes/causes of a decision. Removing these redundant rules will increase transparency even more. Therefore, in the subsequent stages, the proposed STFF–RPA prunes the redundant and irrelevant rules and merges the pruned rule-set into a single rule. The range of each attribute in the merged rule is then reversed, and the misclassification rate is calculated individually. The attributes with the highest misclassification rates are considered as the major factor(s)/attribute(s), because the misclassification rate is very high after reversing their calculated data ranges. This means the attributes and their calculated data ranges are very important for accurately diagnosing the disease. As a result, these factor(s)/attribute(s), along with their calculated ranges, are regarded as major cause(s) of diabetes. Diabetes can be managed/controlled to a large extent if the data ranges of the major factor(s)/attribute(s) are controlled, with proper medication as per experts' advice, food habits, or exercise.

The paper is organized as follows: Sect. 2 highlights the existing literature related to Diabetes diagnosing and prediction using DT, Sect. 3 describes the proposed TDMSML system in detail, Sect. 4 presents the result of Diabetes management using the Pima Indian Diabetes Data Set collected from Kaggle, and finally the Sect. 5 draws conclusion.

## 2 Literature survey

In recent years, DT are broadly used by the researchers to predict as well as diagnosing health-related problems. There are various DT algorithms are available which performs exceptionally well to diagnose the diseases, such as diabetes, breast cancer, heart disease and many more (Podgorelec et al. 2002; Stiglic et al. 2012). DT is the most prevailing algorithm as compared to other classification techniques viz. Artificial Neural Network, Naïve Base Classifier, SVM etc., due to its transparency in nature which helps in identifying the major responsible attributes for a disease and also to validating the outcomes (Podgorelec et al. 2005; Rajkumar and Reena 2010; Dangare and Apte 2012; Kokol et al. 1994; Azar and Bitar 2015; Zorman et al. 1997). Apart from transparency in nature DT extracts the range of values for the major responsible attributes in diagnosing a disease. Till now many interesting models have been proposed using decision trees for medical diagnosis, some of the relevant implementations are mentioned below:

Biswas et al. (2018) proposed an algorithm, called Rule-Based Major Feature Identification (RBMFI) which extracts the most important responsible factors of a diseases by pruning production-rules generated by DT. Azar et al. (2018) made a comparative study of nine machine learning techniques for classification of eight different diseases using two modeling techniques: cross-validation and boot-strapping. Purushottam et al. (2015) designed a system that can efficiently discover the rules to predict the risk level of patients, based on the given parameter about their health. Followed by the comparison of results of the proposed system using C4.5 rules and partial tree, in terms of different parameters. Panigrahi et al. (2016) used different classification approach named DT algorithm, Bayes algorithm and rule-based algorithm. These algorithms are evaluated on the basis of error rates. (Shen et al. 2010) proposed an attribute reduction algorithm, where the irrelevant condition attributes are pruned. After pruning the irrelevant attributes, this model evaluates the superiority of the different subsets of the candidate attribute based on a fitness function derived by formulation. Liu et al. (1970) designed a new rule pruning technique by removing unwanted terms from the rule. For rules which hold multiple test conditions in each attribute, the proposed technique examined both the upper and lower bound individually as well as together and retains the attribute with

higher fitness value. Vijayan et al. (2015) provided a review to highlight the benefits of different preprocessing techniques for decision support systems for predicting diabetes which are based on Support Vector Machine (SVM), Naive Bayes classifier and Decision Tree. Shettty et al. (2017) proposed a model to assemble Intelligent Diabetes Disease Prediction System (IDDPS) that gives analysis of diabetes malady utilizing diabetes patient's database. In this system, they used Bayesian and K-NN (K-Nearest Neighbor) algorithm, analyzed them by taking various attributes of diabetes for prediction of diabetes disease. Morteza et al. (2015) proposes a RF + HC method in which for rule extraction the Random Forest is used. Once the random forest is built, hill climbing algorithms are used to search for an efficient set of rules which reduces the number of rules significantly. Morteza et al. (2017) this paper proposes three algorithms to extract important rules from decision tree ensembles. All the three algorithms viz. RF + DHC, RF + SGL and RF + MSGL uses random forest to generate the decision rules. However, the selection of the significant rule-set is different. RF + DHC uses downhill climbing algorithm for rule selection, whereas RF + SGL uses sparse group lasso and RF + MSGL uses multi class sparse group lasso to extract the significant rules. Pradeep et al. (2016) proposed a methodology comprises of two steps including feature extraction and classification by J48 decision tree algorithm for diabetes detection along with online web-based remote patient monitoring application. International Diabetes Federation (IDF) (International diabetes federation (IDF) diabetes atlas 2017) diabetes atlas—8th edition provided some very useful information regarding statistics of diabetes disease, the rate at which diabetes is increasing rapidly and how the humanity is going to be affected. Guo et al. (2012) aimed at the discovery of a decision tree model for diagnosis of type-2 diabetes. This model uses some pre-processing techniques and followed by the Naïve Bayes model construction. Bashir et al. (2014) used decision trees as base classifier, which differ on splitting criterion named ID3, C4.5 and CART. These decision trees are then combined using different ensemble techniques. Huang et al. (2016) used Support Vector Machine (SVM) to classify the given data step by step. Incorrectly classified patterns were fed to the succeeding stage to find a better split point in SVM. Split point was used to calculate information gain that can identify principal features from candidate attributes. Tsipouras et al. (2006) presented a decision support system using fuzzy model and optimization of the parameters obtained from decision tree using fuzzy model. Chen et al. (2017) proposed a hybrid prediction model, where K-means was used for the reduction of data with J48 decision tree as a classifier to help the diagnosis of Type 2 diabetes. In the proposed model. Tanner et al. (2008) reported that decision algorithms can predict and diagnose dengue disease using simple clinical and hematological parameters. The proposed model used C4.5 decision tree algorithm along with a parameter called minimal cases which acts as the stopping criterion for further partitioning of the data at a particular decision node. Various decision trees were generated using the parameter and on the basis of performance value, the corresponding tree is chosen for further analysis of sensitivity and specificity. (Liu et al. 2016) designed a modified version of C4.5 DT algorithm based on RELIEFF technique which is used for attribute weighting in disease diagnosing. Here, the RELIEFF method is used to prune the redundant attributes. (Albu 2017) used DT algorithm for hepatitis prediction which presents the simple automated system that diagnoses hepatitis. (Pashaei et al. 2015) proposed a model which uses C4.5 decision tree algorithm with boosted version of C5.0 DT algorithm as the fitness function to improvise the outcome of Particle Swarm Optimization (PSO) for medical analysis. Amiri et al. (2013) developed a medical diagnostic system which is implemented on heart disease detection and identification using Classification and Regression Trees (CART). They have demonstrated the potential of CART and suitable encoding scheme for the heart sound data as innocent or pathological murmurs in newborns. The output of their system can help the physicians to decide whether to send a new born for an echocardiogram or not. (Saranya et al. 2017) proposed a system which uses cluster and decision tree analytics-based supervised learning and forms a hub between patients, doctors and dieticians by providing mobile and web application-based solution to pregnant women's on various health-related issues, diet tips etc. Corpus of responses collected from physicians and dietitians is used for creating the system. Sankaranarayanan et al. (2014) used the concepts of classification methods that had been applied to have a watchful study of Diabetes. The data set contained 8 continuous attributes and 768 instances and two classes along with a decision attribute that determines either a person is or not having diabetes mellitus. Ronald et al. (2018) in this paper the basic concepts of SFFS algorithm is used for sub-group feature identification. Nakariyakul et al. (2009) proposes an algorithm which is basically the improvised version of sequential forward floating search named Improved Forward Floating Search (IFFS). Jia et al. (2015) proposes sequential deep floating forward search algorithm which uses the concept of Sequential Floating Forward Search (SFFS) with a minor modification which includes an extra step of deep search. Fazil et al. (2017) focused on finding the linkage between blood glucose and cholesterol levels in the pre-diabetic subjects as to explain the cause of diabetes and cardiovascular. Wei et al. (2018) made a comprehensive exploration to the most popular classification techniques (Deep Neural Network (DNN), Support Vector Machine (SVM), Logistic Regression, Decision Tree, and Naïve Bayes) used to identify diabetes. Sumangali et al. (2016) focused to classify the

data in diabetic or non-Diabetic class labels and improved the classification accuracy, using the advantage of combination of the CART and RF which increases accuracy and resolves the problem of over fitting. Shivakumar et al. (2014) provided a survey of data mining methods that have been commonly applied viz. association rule, clustering, classification on Pima Indian diabetes data sets to Diabetes data analysis and prediction of the disease. A method for computer-assisted diagnosis of skin cancers in dermatology was put forwarded by (Handels et al. 1999). Malignant melanoma and nevocytic nevi (moles) were automatically recognized through the analysis of high resolution skin surface profiles. (Salem et al. 2018) proposed a two-phase technique to classify images of lesions into benign or malignant. The first phase consisting of an image processing-based method that extracts the Asymmetry, Border Irregularity, Color Variation and Diameter of a given mole. The second phase classifies lesions using a Genetic Algorithm. Podgorelec et al. (2001) introduced the integrated computerized environment DIAPRO enabling the diagnostic process optimization which is based on a single approach–evolutionary algorithms.

## 3 Proposed TDMSML model

This section describes the proposed Transparent Diabetes Management System with Machine Learning (TDMSML) model, which identifies preventive measures for Diabetes disease by identifying the most significant factors with ranges using a Decision tree. The proposed TDMSML is divided into three phases: rule generation, transparent rule selection, and identification of major factors. The decision tree is used to generate a rule-set, and the Transparent Rule Selection stage selects the transparent rules from the rules generated by the decision tree, followed by Major Factor Identification, which identifies the significant factors responsible for diabetes disease. Figure 1 depicts a schematic representation of the proposed TDMSML model.

### 3.1 Rule generation

The TDMSML model generates production rules using the C4.5 decision tree algorithm. To build the decision tree during the training stage, the C4.5 employs a top-down strategy based on the divide-and-conquer approach. It maps the training set and uses the information gain ratio as a metric to select splitting attributes before generating nodes from the root to the leaves. Every illustrating path from the root node to the leaf node constitutes a decision rule for determining which class a new instance belongs to. To account for unknown attribute values, the root node contains the entire training set, with all training case weights set to 1.0. If all of the current node's training cases belong to the same class, the algorithm terminates. Otherwise, if all training cases belong to more than one class, the algorithm computes the information gain ratio for each attribute. The attribute with the highest information gain ratio is selected as the best attribute for splitting. Equation (1) gives the expected information needed to split a node based on an attribute **A**, where $\boldsymbol{D}$ is a set of ($\boldsymbol{D_1 \ldots D_j}$) samples with '$\boldsymbol{m}$' distinct classes.

$$\text{Gain Ratio}\ (A) = \frac{\text{Gain}\ (A)}{\text{Split} - \text{Info}_A(D)} \tag{1}$$

$$\text{Gain}\ (A) = \text{Info}(A) - \text{Info}_A(D) \tag{2}$$

$$\text{Split} - \text{Info}_A(D) = -\sum \frac{|D_\text{i}|}{|D|} * \log_2\left(\frac{|D_\text{i}|}{|D|}\right) \tag{3}$$

$$\text{Info}\ (D) = -\sum \text{Prob}_\text{i} * \log_2(\text{Prob}_\text{i}) \tag{4}$$

$$\text{Info}_A(D) = -\sum \frac{|D_\text{i}|}{|D|} * \text{Info}\ (D_\text{i}) \tag{5}$$

Gain ($A$) in (2) represents the expected reduction in entropy caused by knowing the value of attribute $A$, and Split-Info$_\text{A}$ ($D$) in (3) represents the potential information obtained by partitioning the training data set $D$ into $n$ partitions while considering attribute A. Info ($D$) in (4) measures the class impurity before splitting the data set $D$, whereas Info$_A$($D$) in (5) specifies the average entropy after the split of data set $D$ based on the attribute values of $A$ (Prob$_\text{i}$ = probability of distinct class $C_\text{i}$, ($|D_\text{i}|/|D|$) = act as the weight of $i^{\text{th}}$ partition) (Navada et al. 2011).

### 3.2 Transparent rule selection

To obtain a transparent and minimized rule-set, the TDMSML model proposes the Sequential Transparent Floating Forward–Rule Pruning Algorithm (STFF–RPA). The proposed algorithm takes decision rules as input and outputs significant ranges identifying the major factor(s)/attribute(s) responsible for diabetes disease. STFF–RPA is divided into two stages: transparent rule extraction and rule pruning. It extracts the most transparent rule-set from a set of decision rules in the transparent rule extraction phase, whereas the rule pruning phase determines the best rules from each training set in terms of highest SOR value, followed by pruning the unnecessary rules and merging the refined rules into a single rule. The following are the stages of the proposed STFF–RPA:

#### 3.2.1 Transparent rule extraction

The Sequential Deep Transparent Floating Forward Search (SDTFFS) algorithm is proposed for extracting transparent

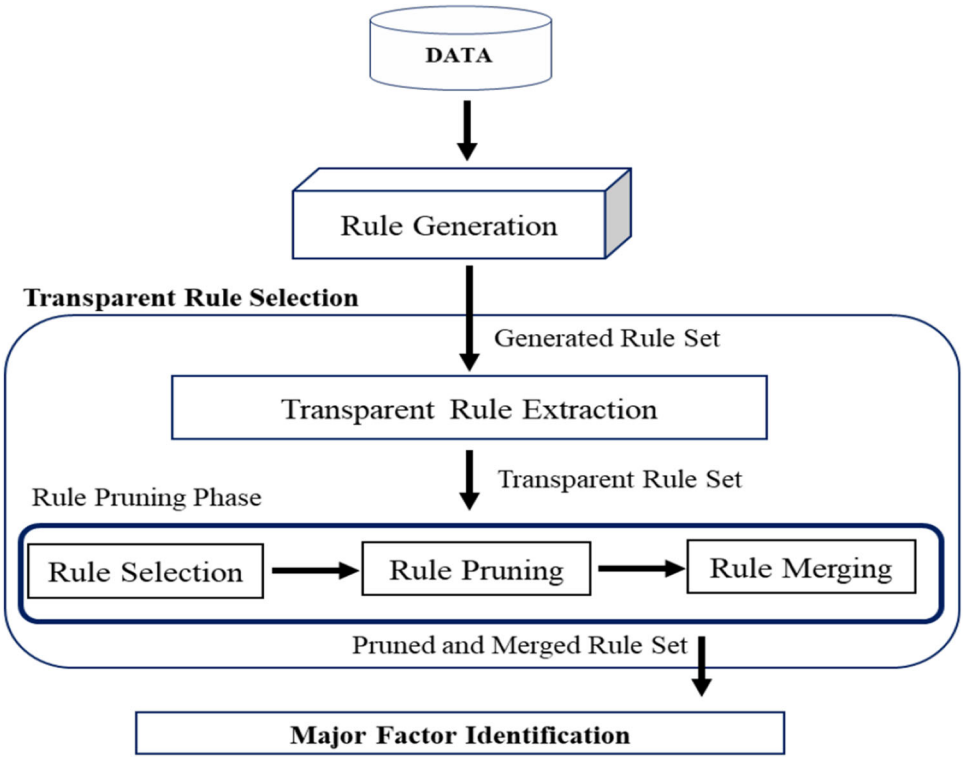


**Fig. 1** Transparent disease management system using machine learning

rules. SDTFFS has three steps: Sequential Forward Search (SFS), Deep Transparency Search (DTS), and Sequential Backward Search (SBS).In all three steps, the deciding criteria for whether or not to select any rule is based on the calculation of Significance of Rule (SOR). Equations 6 and 7 provide the formula for calculating SOR value:

$$AOR = \left[\left(\frac{CC - IC}{CC + IC}\right) + \left(\frac{CC}{IC + 1}\right) - \frac{IC}{CC}\right] \tag{6}$$

$$SOR = AOR + \frac{CC}{RL} \tag{7}$$

where $AOR$ = Accuracy of the Rule, $SOR$ = Significance of Rule, $CC$ = Total number of correctly classified patterns by a rule-set, $IC$ = Total number of incorrectly classified patterns by a rule-set and $RL$ = Rule Length. The process flow of the SDTFFS is shown in Fig. 2.

The detailed explanation of the stages of proposed SDTFFS algorithm is given below:

**Sequential forward search (SFS)** Starting with an empty rule-set, in each iteration the algorithm adds a new rule to the current rule-set. While adding a new rule to the current rule-set, it checks whether addition of the rule increases the overall SOR value of the current rule-set or not. If it increase the SOR value of the current rule-set then the corresponding rule is added otherwise discarded. If there is no such rule available that could increase the SOR value of current rule-set then the proposed SDTFFS algorithm terminates and the current rule-set will be the resultant rule-set of this stage.

**Deep transparency search (DTS)** It is assumed that when the proposed SDTFFS algorithm enters into DTS stage, the rule that was newly added by SFS algorithm, marked as seed and the initial size of the rule-set is assumed to be $k$. For each iteration, DTS algorithm will create a set of rule-sets of size ($k$–1) removing each rule at a time from current rule-set ($S$) keeping the seed intact, which will produce $N = {}^{k-1}C_{k-2}$ new rule-sets of size ($k$–1). To every $N$ new rule-sets, it adds a new rule having highest SOR value from the remaining set of rules those were not present in current rule-set ($S$). Therefore, in total $N$ new $k$-rule-set (rule-set of size $k$) is obtained. Now, among these $N$ new $k$–rule-sets, if there exists any rule-set having highest SOR value as well as greater than previous SOR value of current rule-set ($S$), is considered as new current rule-set ($S$). If no such rule-set is found then the current rule-set ($S$) is kept as it is.

**Sequential backward search (SBS)** The $k$-rule-set (rule-set of size $k$) obtained from DTS step, fed as input to SBS step. In this step the algorithm starts removing each rule at a time from current $k$-rule-set, and finds all ($k$–1)-rule-set (rule-set of size $k$–1). Now it calculates the SOR values of all ($k$–1)-rule-sets. The rule-set having highest SOR value as well as greater than the SOR value of ($k$–1)-rule-set of previous iteration, is considered as current best ($k$–1)-rule-set and forwards it to next iteration.

The summarized SDTFFS algorithm is explained below:

*//Sequential Deep Transparent Floating Forward Search (SDTFFS)//*

***Input*** *: Initial rule-set R*
***Output :*** *Transparent rule-set S*

***Notations :***
$S$ → *Subset of rules under consideration*
$K$ → *Number of rules in subset S*
$R$ → *Initial rule-set*
$R_i$ → $i^{th}$ *rule in R*
$C_i$ → *SOR on inclusion of* $R_i$ *into S*
$P_c$ → *Previous SOR value.*
$P_{cp}$ → *Previous SOR value on S of size (k-1)*
$A_c$ → *Current SOR vlaue.*
$P$ → *Seed (newly added rule to subset S)*
$M$ → *Set of rules consisting (S+P)*
$M_j$ → $j^{th}$ *rule of set M*
$N$ → *Set of rules containing (R-S)*
$N_k$ → $k^{th}$ *rule of N*
$Q$ → *Set of subsets of M*
$Q_a$ → $a^{th}$ *subset of Q*
$A_a$ → *SOR value of each* $Q_a$
$X$ → *Significance of Rule*
$L$ → *A rule of N*
$Y_i$ → $i^{th}$ *rule of S*
$V$ → *Subset of rules on removal of* $Y_i$
$U_i$ → *Array of SOR value on V*

***Step 1 :*** *Initialize S and k.*
***Step 2 :*** *For all* $R_i$ *of R*
***2(a) :*** *Insert* $R_i$ *into S and calculate* $C_i$
***2(b) :*** *Remove* $R_i$ *from S*
***Step 3 :*** *For all* $C_i$
***3(a) :*** *Take highest* $C_i$
***3(b) :*** *Check if* $C_i > P_c$
***3(b)(i) :*** *If yes, select the corresponding rule*
***3(b)(ii) :*** *Insert the selected rule in S and mark as P*
***3(b)(iii) :*** *Update the value of* $P_c$ *by* $C_i$ *and go to* ***step 5***
***3(c) :*** *If not then go to* ***step 4***
***Step 4 :*** *Exit.*
***Step 5 :*** *Initialize S and k.*
***Step 6 :*** *For all* $M_j$ *of M except P*
***6(a) :*** *Remove* $M_j$ *from M*
***6(b) :*** *Save resultant M into Q*
***6(c) :*** *Add* $M_j$ *back to M*
***Step 7 :*** *For all* $N_k$ *of N*
***7(a):*** *Calculate X*
***7(b):*** *Take highest significant rule L*
***7(c):*** *For all* $Q_a$ *of Q*

***7(c)(i) :*** *Add L into* $Q_a$

***7(c)(ii):*** *Calculate* $A_a$ *on* $Q_a$

***7(d):*** *For all* $Q_a$ *of* $Q$

***7(d)(i) :*** *Take highest* $A_a$

***7(d)(ii) :*** *Check if* $A_a > P_c$

***7(d)(ii)(A) :*** *If yes update* $P_c$ *by* $A_a$

***7(d)(ii)(B) :*** *Update* $S$ *by* $Q_a$

***7(d)(ii)(C) :*** *Go to* ***step 5***

***7(d)(iii) :*** *Else go to* ***step 8***

***Step 8 :*** *Initialize* $S$ *and* $k$.

***Step 9 :*** *For all* $Y_i$ *of* $S$

***9(a) :*** *Find* $V$

***9(b) :*** *Calculate* $U_i$

***Step 10 :*** *For all* $U_i$

***10(a) :*** *Take highest* $U_i$

***10(b) :*** *Check if* $U_i > P_{cp}$

***10(b)(i) :*** *If yes then update* $S$ *by* $V$

***10(b)(i) :*** *Update* $P_{cp}$ *by* $U_i$ *and go to* ***step 11***

***10(c) :*** *If not then go to* ***step 11***

***Step 11 :*** *Go to* ***step – 1***

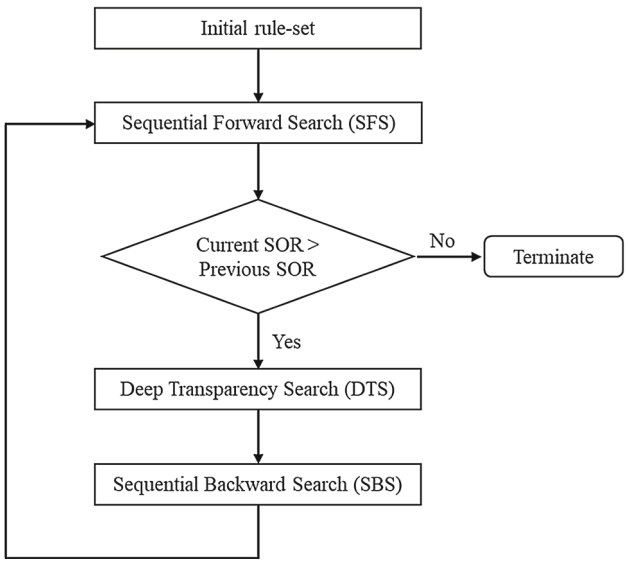


**Fig. 2** Sequential deep transparent floating forward search (SDTFFS)

### 3.2.2 Rule pruning phase

The proposed STFF–RPA prunes the transparent rule-set to obtain a single optimal rule. STFF–RPA algorithm is divided into three stages: rule selection, rule pruning, and rule merging. In rule selection stage, the most promising transparent rule is selected by calculating the SOR from each training set (tenfold cross validation). Then, by Rule pruning, redundant rules are pruned to make the proposed TDMSML more transparent. The obtained pruned rules are merged in to a single rule. The detailed explanations on each step is given below:

**Rule selection** From the obtained transparent rule-set for each training set, the rule which produces highest SOR value is identified and this rule is considered as the final and most promising rule for that training set. This selection procedure is continued for all the training sets. The algorithmic representation is given below:

| *// Rule Selection //* |
|---|
| ***Input*** *: Rule-set S*<br>***Output :*** *Single Rule R* |
| ***Notations :***<br>*S* → *Subset of rules under consideration*<br>*k* → *Number of rules in subset S*<br>$R_i$ → $i^{th}$ *rule in S.*<br>$C_i$ → *SOR of* $R_i$<br>$P_c$ → $R_i$ *with highest SOR in S* |
| ***Step 1 :*** *Initialize S and k.*<br>***Step 2 :*** *For all* $R_i$ *of S*<br>***2(a) :*** *Calculate* $C_i$<br>***Step 3 :*** *Find* $P_c$<br>***Step 4 :*** *Exit.* |

**Rule pruning** In the rule pruning process, pruning is done on the basis of sensitivity analysis of each rule obtained from the rule selection phase. While pruning, first if there are different ranges of an attribute in a single rule, then sensitivity of each range is calculated for that rule only. The range which on pruning lowers the sensitivity of that rule considered as important range. After pruning on the ranges, the sensitivity is calculated for each rule. The rules which on pruning lower the sensitivity of the rule-set are considered as more significant and kept unchanged. Rest redundant rules are removed from the current rule-set. The rule pruning algorithm is summarized below:

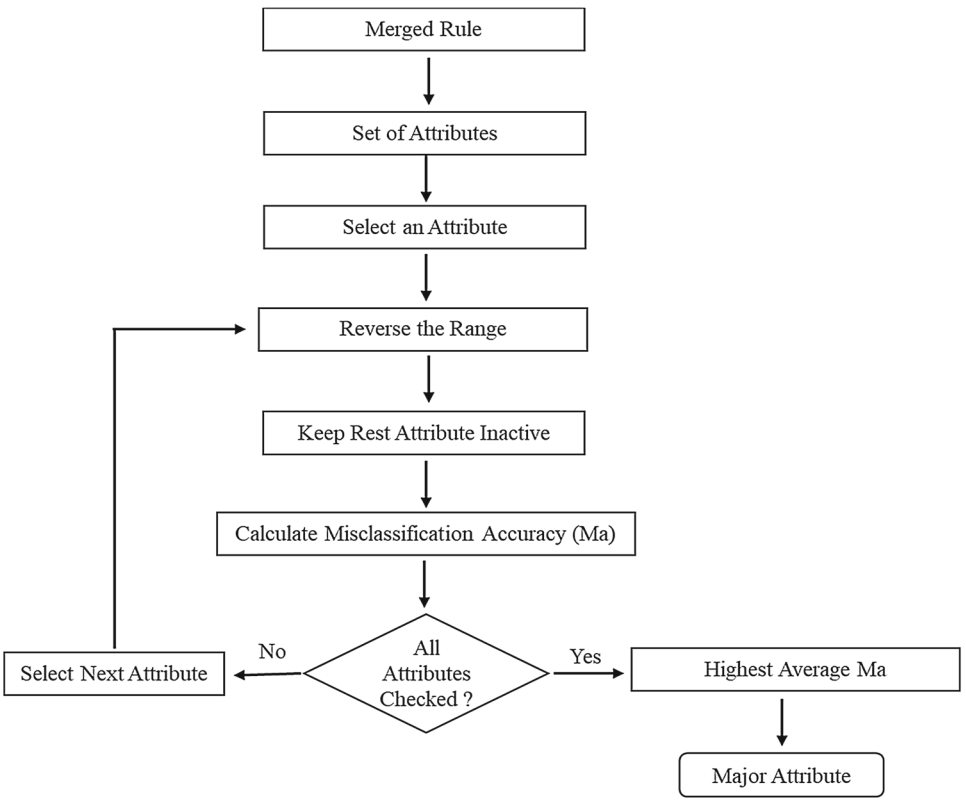


**Fig. 3** Major factor identification

*// Rule Pruning //*

***Input***: - *A set of transparent rules.*
***Output***: - *Final transparent pruned rule-set.*

***Notations:-***

***S*** → *new rule-set obtained after rule selection phase.*
$A_{ac}$ → *classification accuracy of* ***S***.
$r_i$ → $i^{th}$ *rule in* ***S***.
$A_i$ → *classification accuracy of* ***S*** *after pruning* $r_i$.
***k*** → *no. of rules in* ***S***.
***R*** → *the final pruned rule obtained.*
$a_i$ → $i^{th}$ *attribute of* $r_i$.
***P*** → *Set of all attributes in* $r_i$
***Q*** → *Set of all ranges of* $n^{th}$ *attribute in* $P_i$
$Q_n$ → *The* $n^{th}$ *element of* ***Q***
$A_{acn}$ → *Classification accuracy on pruning* $Q_n$

***Step 1:*** *Compute the classification accuracy* $A_{ac}$ *of* ***S***.
***Step 2:*** *Set* $i=1$
***Step 3:*** *Select* $i^{th}$ *Rule*
***Step 4 :*** *Find P*
***Step 5 :*** *Find Q*
***Step 6 :*** *If (Q != Empty)*
***Step 6.1 :*** *Set* $n=1$
***Step 6.2 :*** *Prune* $Q_n$ *and calculate* $A_{acn}$ **.**
***Step 6.3 : If*** $(A_{acn} > A_{ac})$
***Step 6.3.1 :*** *Prune* $Q_n$ *permanently and set* $A_{ac} = A_{acn}$ .
***Else***
***Step 6.3.2 :*** *Restore* $Q_n$
***End***
***Step 6.4:*** $n = n + 1$
***Step 7:*** *Set* $i=i+1$
***Step 8:*** *If* $(i \leq n)$
***Step 8.1 :*** *Go to* ***Step 3***
***Else***
***Step 8.2 :*** *Go to* ***Step 9***
***End***
***Step 9 :*** *Set* $i=1$
***Step 10 :*** *Remove* $r_i$ *and calculate* $A_i$ .
***Step 11 : If*** $(A_i<A_{ac})$
*Keep the selected rule.*
***Else***
*Prune rule* $r_i$ *from set* ***P*** *and Set* $A_{ac}=A_i$.
***End***
***Step 12:*** *Set* $i=i+1$.
***Step 13: If*** $(i \leq n)$
*Go to* ***Step 10***
***Else***
*Stop the process*
***End***
***Step 14:*** *Exit*

**Rule merging** The significant rules in the rule-set are further refined by focusing on the attributes of the rule-set. Similar attributes with different ranges in more than one rule are identified and they are pruned one by one if their absence from the rule set increases the accuracy of the set. The final rule is generated by merging all the refined significant rules in the set into a single rule in a reasonable way. The abstract is represented in algorithmic way below:

*// Rule Merging //*

***Input**: - A set of transparent rules.*
***Output: -** Final transparent single merged rule.*

***Notations:-***
***S*** → *final pruned transparent rule-set.*
$A_{ac}$ → *classification accuracy of* ***S****.*
$r_i$ → $i^{th}$ *rule in* ***S****.*
$A_i$ → *classification accuracy of* ***S*** *after pruning* $r_i$*.*
***n*** → *no. of rules in* ***S****.*
***R*** → *the final pruned rule obtained.*
$a_i$ → $i^{th}$ *attribute of* $r_i$**.**
$k_i$ → *range of* $i^{th}$ *attribute.*

***Step 1:*** *Calculate* $A_{ac}$

***Step 2:*** *For all* $r_i$

***Step 2.1: If*** *(*$(a_i == a_j)$ *&&* $(k_i != k_j)$*)*

***Step 2.1.1:*** *Prune* $a_i$ *and calculate* $A_{aci}$**.**

***Step 2.1.2:*** *Restore* $a_i$ *.*

***Step 2.1.3:*** *Prune* $a_j$ *and calculate* $A_{acj}$ *.*

***Step 2.1.4:*** *Restore* $a_j$ *.*

***Step 2.2: If*** *(*$(A_{acj} > A_{aci})$ *&&* $(A_{acj} > A_{ac})$*)*

*Prune* $a_j$ *permanently.*

***Else if*** *(*$(A_{aci} > A_{acj})$ *&&* $(A_{aci} > A_{ac})$*)*

*Prune* $a_i$ *permanently.*

***Else***

*Retain both the ranges.*

***End***

***Step 3:*** *For each rule* $r_i$ *of S.*

***Step 3.1:*** $R=R \cup \{r_i\}$

***Step 4: Exit***

## 3.3 Major factor identification

To identify the major factor(s) from the pruned and merged (single) rule, the Sequential Search algorithm is used. The basic concept of this algorithm is based on the evaluation of each attribute independently. Therefore, in this stage, each attribute present in merged rule, is taken under consideration, while keeping other attributes inactive and prevention/misclassification rate is calculated. The attribute which produces highest average misclassification rate for positive class, when its range is reversed, is considered as major factor for diabetes disease. If more than one major factor needs to be identified then the incremental Sequential Search process is iteratively executed to identify intended number of major factor(s) with range(s). The flow chart of the algorithm is shown in Fig. 3. The abstract of the algorithm is written below:

| *// Major Factor Identification//* |
|---|
| ***Input**: - Merged Rule.*<br>***Output**: - Attribute(s) with highest average misclassification rate when ranges reversed.* |
| ***Notations:-***<br>***C*** → *Positive class.*<br>***P*** → *Rule obtained after rule pruning and merging.*<br>***M*** → *Set of attributes present in **P**.*<br>$A_i$ → $i^{th}$ *attribute of the rule in **M** .*<br>$MR_i$ → *Number of misclassified patterns after reversing the range of* $A_i$.<br>***B*** → *Set of major factors* |
| ***Step 1:** For all* $a_i$ *of M*<br>*Step 1.1: Make* $A_i$ *active and rest inactive in P*<br>*Step 1.2: Reverse the range of* $A_i$.<br>*Step 1.3: Calculate average* $MR_i$ *of C.*<br>***Step 2:** Select attribute* $a_i$ *with highest average* $MR_i$, *and include the attribute in set **B**.* |

## 4 Results and discussion

The Pima Indian Diabetes data set is used here. In this section the entire TDMSML model work flow is explained clearly with results. The diabetes data set contains total 768 patterns, 8 attributes and 2 classes indicating the occurrences and non-occurrences of diabetes disease. tenfold cross-validation is performed to validate the model. The data set comprises of the following:

- x1 → Pregnancies—Number of times pregnant
- x2 → Glucose—Plasma glucose concentration a 2 h in an oral glucose tolerance test
- x3 → Blood Pressure—Diastolic blood pressure (mm Hg)
- x4 → Skin Thickness—Triceps skin fold thickness (mm)
- x5 → Insulin—2-h serum insulin (mu U/ml)
- x6 → BMI—Body mass index (weight in kg/(height in m)^2)
- x7 → Diabetes pedigree function
- x8 → Age (years)

### 4.1 Rule generation

Set of rules generated for the positive class (patients diagnosed with diabetes) of training set 1 are shown below. The rules are generated by C4.5 decision tree algorithm.

1) *if ( x2 < 0.640704 && x8 >= 0.125 && x6 >= 0.392697 && x7 < 0.233561 && x2 >= 0.469849 && x4 < 0.328283 && x3 < 0.631148 ) then Class = 1*
2) *if ( x2 < 0.640704 && x8 >= 0.125 && x6 >= 0.392697 && x7 >= 0.233561 ) then Class =1*
3) *if ( x2 >= 0.640704 && x6 < 0.444858 && x2 >= 0.728643 ) then Class = 1*
4) *if ( x2 >= 0.640704 && x6 >= 0.444858 && x2 < 0.791457 && x7 < 0.153074 && x6 >= 0.571535 ) then Class = 1*
5) *if ( x2 >= 0.640704 && x6 >= 0.444858 && x2 < 0.791457 && x7 >= 0.153074 ) then Class = 1*
6) *if ( x2 >= 0.640704 && x6 >= 0.444858 && x2 >= 0.791457 ) then Class = 1*

## 4.2 Transparent rule selection

As discussed earlier, this stage finds out the minimized transparent rule-set by extracting transparent rule-set followed by rule pruning (to remove the redundant rules) and rule merging (to merge the rules into one single rule). The detailed explanation along with results of each and every step is given below:

### 4.2.1 Transparent rule extraction

This section extracts the transparent rule-set from the set of rules generated by decision tree. In this stage from the training set 1, rule number-3 and rule number-6 (shown in the above box), are selected as the transparent rule-set using the proposed SDTFFS algorithm. The resultant transparent rule-set is shown in the box given below:

*1) if ( x2 >= 0.640704 && x6 < 0.444858 && x2 >= 0.728643 ) then Class = 1*
*2) if ( x2 >= 0.640704 && x6 >= 0.444858 && x2 >= 0.791457 ) then Class = 1*

### 4.2.2 Rule pruning phase

Once the transparent rule-set is obtained, the proposed STF-F–RPA selects the most promising rule followed by the removal of redundant rules which might increase the transparency as well as the classification accuracy. Rule pruning is done in the following manner.

*1) if ( x2 >= 0.640704 && x6 >= 0.444858 && x2 >= 0.791457 ) then Class = 1*
*2) if ( x2 >= 0.640704 && x6 >= 0.444858 && x2 >= 0.791457 ) then Class = 1*
*3) if ( x2 >= 0.721106 && x2 >= 0.836683 ) then Class = 1*
*4) if ( x2 >= 0.640704 && x6 >= 0.447094 && x2 >= 0.776382 ) then Class = 1*
*5) if ( x2 >= 0.640704 && x6 >= 0.446349 && x2 >= 0.791457 ) then Class = 1*
*6) if ( x2 >= 0.640704 && x6 >= 0.446349 && x2 >= 0.791457 ) then Class = 1*
*7) if ( x2 >= 0.640704 && x6 >= 0.444858 && x2 >= 0.791457 ) then Class = 1*
*8) if ( x2 >= 0.640704 && x6 >= 0.446349 && x2 >= 0.791457 ) then Class = 1*
*9) if ( x2 >= 0.640704 && x6 >= 0.444858 && x2 >= 0.781407 ) then Class = 1*
*10) if ( x2 >= 0.721106 && x2 >= 0.776382 ) then Class = 1*

**Rule selection** It is clearly visible that the transparent rule-set extracted in Sect. 4.2.1 contains two rules. Among those two rules one rule is selected which has higher SOR value than other(s). In this case the second ($2^{nd}$) rule has highest SOR value. Therefore, the rule given in the box below is considered as the most promising and transparent rule among all the rules in training set 1.

*1) if ( x2 >= 0.640704 && x6 >= 0.444858 && x2 >= 0.791457 ) then Class = 1*

Similarly, most promising rules are selected from all the remaining training sets. As a result, finally total 10 rules are selected from 10 training sets. The final transparent rule-set is given in the box below:

**Rule pruning** From the set of 10 rules obtained in Sect. 4.2.2.1, it is clearly visible that there are some repetitive rules and in some of the rules the same attribute has different ranges. These redundant rules and attributes needs to be removed to increase the transparency of the set. The sensitivity of attribute(s) with different ranges in a rule is calculated individually keeping all other rules inactive and that attribute range is dropped from the rule which is less sensitive in making the decision. For example, the first rule of the set has attribute x2 with two different ranges. With these two ranges in the rule, 81 positive patterns out of a total of 268 positive patterns were correctly classified, and without the range (x2 > = 0.791457), the number of correctly classified positive patterns increased to 151. As a result, the range (x2 > = 0.791457) is removed from rule 1 of the set. This procedure is repeated for all of the rules in the rule-set. After removing all such ambiguous ranges of single attribute in a rule, the sensitivity of each rule is calculated. If the number of properly classified positive patterns increases when a rule is removed, that rule is permanently dropped. The pruned rule-set is shown in the box below:

1) *if ( x2 >= 0.640704 && x6 >= 0.444858 ) then Class = 1*
2) *if ( x2 >= 0.721106 ) then Class = 1*
3) *if ( x2 >= 0.640704 && x6 >= 0.447094 ) then Class = 1*
4) *if ( x2 >= 0.640704 && x6 >= 0.446349 ) then Class = 1*

**Rule merging** The minimized rule-set obtained during the rule pruning phase reveals two attributes with different data ranges, x2 and x6. As a result, all such ranges are identified and pruned one by one during the rule merging procedure. If removing a data range from the rule-set increases the number of correctly classified positive patterns, that range is permanently removed. Following this, all of the different rules are combined into a single rule, as shown below:

1) *if ( x2 >= 0.640704 || x6 >= 0.444858 ) then Class = 1*

## 4.3 Major factor identification

The previous step resulted in a merged rule with two attributes, x2 and x6. Therefore, during the major factor identification phase, it is determined which of the attribute(s) is/are most important for diagnosing the disease. To test the misclassification rates, the ranges of each attribute are reversed individually. The greater the misclassification rate, the greater the importance of the respective attribute. Table 1 shows the misclassification rates for single attributes.

**Table 1** Misclassification rates when condition reversed (one attribute)

| Testing set | X2 < 0.640704 | X6 < 0.444858 |
|---|---|---|
| Testing set-1 | 66.95% | 82.20% |
| Testing set-2 | 65.45% | 82.11% |
| Testing set-3 | 63.68% | 81.20% |
| Testing set-4 | 65.13% | 83.61% |
| Testing set-5 | 65.15% | 82.57% |
| Testing set-6 | 63.87% | 82.35% |
| Testing set-7 | 64.96% | 82.68% |
| Testing set-8 | 64.20% | 81.48% |
| Testing set-9 | 65.57% | 83.61% |
| Testing set-10 | 64.14% | 82.70% |
| **Average** | **64.91%** | **82.45%** |

The bold values represent the average results

Table 1 clearly shows that attribute x6 is the most promising cause, with a high misclassification rate when its range is reversed. Therefore, if the scenario is to select single major factor responsible for diabetes then x6 is selected as the one. The same procedure is now followed incrementally when selecting more than one major factor. The misclassification rates with two attributes are shown in Table 2.

All of the results presented above show that reversing the data ranges of more attributes increases the misclassification rate. Therefore, it can said that prevention rate will also increase with the increase in the number of factors to be controlled. Though the prevention rate increases with the

**Table 2** Misclassification rates when condition reversed (two attribute)

| Testing set | X2 < 0.640704, X6 < 0.444858 |
|---|---|
| Testing set-1 | 91.10% |
| Testing set-2 | 90.65% |
| Testing set-3 | 91.02% |
| Testing set-4 | 91.60% |
| Testing set-5 | 90.87% |
| Testing set-6 | 90.76% |
| Testing set-7 | 91.34% |
| Testing set-8 | 90.54% |
| Testing set-9 | 91.39% |
| Testing set-10 | 91.14% |
| **Average** | **91.04%** |

The bold values represent the average results

increase in factor(s) but it may be difficult to control more factor(s) within its range that are responsible for diabetes disease. However, from the analysis it is clear that, even if the range of one significant attribute is controlled, the disease can be managed effectively.

## 5 Conclusion

The proposed TDMSML model is a method for determining the significant factor(s) and their range(s) for managing diabetes. If the range(s) of only the significant factor(s)/attribute(s) can be controlled, the occurrence of diabetes disease can be managed to a great extent as well as individuals can be saved from different health issues. The proposed model effectively generates transparent production rules from diabetes data sets using decision tree algorithm. To identify the major factor(s) from these rules, the transparent rules are selected as well as pruned to remove the redundancies and the significant factor(s)/attribute(s) with range(s) are finally identified using the proposed transparent rule selection and rule pruning algorithms. According to the experimental results, it can be said that diabetes can be significantly managed by controlling one or two attributes. Consequently, it can also be concluded that the TDMSML can be used to assist the physicians in analyzing the medical records and in effectively preventing diabetes. This proposed model is also expected to bring about a revolutionary change in the medical field for diabetes prevention. The proposed model generates production rules using a decision tree; however, other machine learning algorithms such as neural networks, etc. can be used. Rule and attribute pruning algorithms can also be improved for greater efficiency.

**Data availability** The data set used for the experiments are collected from kaggle. No third party data has been used in this work.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

Albu A (2017) From logical inference to decision trees in medical diagnosis. E-Health Bioeng Conf (EHB), 65–68

Amiri AM, Armano G (2013) Early diagnosis of heart disease using classification and regression trees. The 2013 International Joint Conference on Neural Networks (IJCNN), 1–4

Anderson RC, Baker MC (2018) Preliminary assessment of an SFFS method for sub-group feature identification in heterogeneous data sets. 2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS), Karlstad, pp. 182–187

Azar D, Bitar M (2015) AI-based methods for predicting required insulin doses for diabetic patients. Int J Artif Intell 13(1):8–24

Azar D, Moussa RO, Jreij G (2018) A comparative study of nine machine learning techniques used for the prediction of diseases. Int J Artif Intell 16(2):25–40

Bashir S, Qamar U, Khan FH, Javed MY (2014) An efficient rule-based classification of diabetes using ID3, C4.5, & CART Ensembles. 2014 12th International Conference on Frontiers of Information Technology, Islamabad, pp. 226–231

Bing L, Wynne H, Yiming M (1970) Pruning and Summarizing the discovered associations. Int Conf Knowl Discov Data Mining (KDD-99)

Biswas SK, Chowdhury SR, Chakraborty M, Purkayastha B (2018) A medical expert system to identify major factor of diseases using P-rules, 2018 International Conference on Intelligent Autonomous Systems (ICoIAS), Singapore, pp. 82–87

Chen W, Chen S, Zhang H, Wu T (2017) A hybrid prediction model for type 2 diabetes using K-means and decision tree. 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, pp. 386–390

Dangare CS, Apte SS (2012) Improved study of heart disease prediction system using data mining classification techniques. Int J Computer Appl 47(10):44–48

Fazil QA, Jamaludin UK (2017) Investigation on the relationship between cholesterol and blood glucose levels using decision tree method in healthy subjects. 2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS), Kota Kinabalu, pp. 161–166

Guo Y, Bai G, Hu Y (2012) Using bayes network for prediction of Type-2 diabetes. 2012 International Conference for Internet Technology and Secured Transactions, London, pp. 471–472

Handels H, Rob T, Kreusch J, Wolff H, Poeppl SJ (1999) Feature selection for optimized skin tumor recognition using genetic algorithms. Artif Intell Med 16(3):283–297

Huang Y, Nashrullah M (2016) SVM-based decision tree for medical knowledge representation. 2016 International Conference on Fuzzy Theory and Its Applications (iFuzzy), Taichung, pp. 1–6

International diabetes federation (2017) (IDF) diabetes atlas. 8th edn

Kokol P, Mernik M, Zavrsnik J, Kancler K, Malcic I (1994) Decision trees based on automatic learning and their use in cardiology. J Med Syst 18(4):201–206

Liu Q, Xu X, Tao Y, Wang X (2016) An improved decision tree method base on RELIEFF for medical diagnosis. 6th International Conference on Digital Home (ICDH), 133–138

Lv J, Peng Q, Sun Z (2015) A modified sequential deep floating search algorithm for feature selection. 2015 IEEE International Conference on Information and Automation, Lijiang, pp. 2988–2993

Mashayekhi M, Gras R (2015) Rule extraction from random forest: the RF+HC methods. In: Barbosa D, Milios E (eds) Advances in artificial intelligence. Canadian AI 2015. Lecture notes in computer science, vol 9091. Springer, Cham

Mashayekhi M, Gras R (2017) Rule extraction from decision trees ensembles: new algorithms based on heuristic search and sparse group lasso methods. Int J Information Technol Decision Making (IJITDM) 16(6):1707–1727 (**World Scientific Publishing Co. Pte. Ltd.**)

Nakariyakul S, Casasent DP (2009) An improvement on floating search algorithms for feature subset selection. Pattern Recogn 42(9):1932–1940

Navada A, Ansari AN, Patil S, Sonkamble BA (2011) Overview of use of decision tree algorithms in machine learning. 2011 IEEE Control and System Graduate Research Colloquium, Shah Alam, pp. 37–42

Pashaei E, Ozen M, Aydin N (2015) Improving medical diagnosis reliability using boosted C5.0 decision tree empowered by particle swarm optimization. 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 7230–7233

Podgorelec V, Kokol P (2001) Towards more optimal medical diagnosing with evolutionary algorithms. J Med Syst 25(3):195–219

Podgorelec V, Kokol P, Stiglic B, Rozman I (2002) Decision trees: an overview and their use in medicine. J Med Syst 26(5):445–463. https://doi.org/10.1023/a:1016409317640. (**PMID: 12182209**)

Podgorelec V, Kokol P, Stiglic MM, Heričko M, Rozman I (2005) Knowledge discovery with classification rules in a cardiovascular dataset. Comput Methods Programs Biomed 80:S39–S49

Pradeep KR, Naveen NC (2016) Predictive analysis of diabetes using J48 algorithm of classification techniques. 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), Noida, pp. 347–352

Rajkumar A, Reena GS (2010) Diagnosis of heart disease using datamining algorithm. Global J Comp Sci Technol 10(10):38–43

Salem C, Azar D, Tokajian S (2018) An image processing and genetic algorithm-based approach for the detection of melanoma in patients. Methods Inf Med 57(01/02):74–80

Sankaranarayanan S, Perumal TP (2014) A predictive approach for diabetes mellitus disease through data mining technologies. 2014 World Congress on Computing and Communication Technologies, Trichirappalli, pp 231–233

Saranya G, Geetha G, Safa M (2017) E-Antenatal assistance care using decision tree analytics and cluster analytics based supervised machine learning. Int Conf IoT Appl (ICIOT), 1–3

Saxena PK, Sharma R (2015) Diabetes mellitus prediction system evaluation using C4.5 rules and partial tree. 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trendsand Future Directions), Noida, pp.16

Shen H, Yang S, Liu J (2010) On attribute reduction of rough set based on pruning rules. In: Yu J, Greco S, Lingras P, Wang G, Skowron A (eds) Rough set and knowledge technology. RSKT 2010. Lecture notes in computer science, vol 6401. Springer, Berlin, Heidelberg

Shetty D, Rit K, Shaikh S, Patil N (2017) Diabetes disease prediction using data mining. 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, pp. 1–5

Shivakumar BL, Alby S (2014) A survey on data-mining technologies for prediction and diagnosis of diabetes. 2014 International Conference on Intelligent Computing Applications, Coimbatore, pp. 167–173

Srikanth P, Deverapalli D (2016) A critical study of classification algorithms using diabetes diagnosis. 2016 IEEE 6th International Conference on Advanced Computing (IACC), Bhimavaram, pp. 245–249

Stiglic G, Kocbek S, Pernek I, Kokol P (2012) Comprehensive decision tree models in bioinformatics. PLoS ONE 7(3):e33812

Sumangali K, Geetika BSR, Ambarkar H (2016) A classifier based approach for early detection of diabetes mellitus. 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, pp.389–392

Tanner T et al (2008) Decision tree algorithms predict the diagnosis and outcome of dengue fever in the early phase of illness. PLOS Neglected Tropical Dis 2(3):e196

Tsipouras MG, Exarchos TP, Fotiadis DI, Kotsia A, Naka A, Michalis LK (2006) A decision support system for the diagnosis of coronary artery disease. 19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06), 279–284

Vijayan VV, Anjali C (2015) Decision support systems for predicting diabetes mellitus—a review. 2015 Global Conference on Communication Technologies (GCCT), Thuckalay, pp. 98–103

Wei S, Zhao X, Miao C (2018) A comprehensive exploration to the machine learning techniques for diabetes identification. 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), Singapore, pp. 291–295

Zorman M, Stiglic MM, Kokol P, Malcic I (1997) The limitations of decision trees and automatic learning in real world medical decision making. J Med Syst 21(6):403–415