Journal of Engineering Science and Technology
Vol. 17, No. 1 (2022) 0472 - 0490

NAÏVE BAYES BASED MULTIPLE PARALLEL FUZZY REASONING METHOD FOR MEDICAL DIAGNOSIS

THIRUMALAIMUTHU THIRUMALAIAPPAN RAMANATHAN[1,*], MD. JAKIR HOSSEN[2], MD. SHOHEL SAYEED[1]

[1]Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Bukit Beruang, Melaka, Malaysia
[2]Faculty of Engineering and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Bukit Beruang, Melaka, Malaysia
*Corresponding Author: 1181402216@student.mmu.edu.my

Abstract

There are millions of sample medical cases recorded in many digital medical datasets that can be used by the data mining techniques for predicting any particular disease. Improving the classification accuracy in medical diagnosis based on patterns extracted from the available medical datasets is a challenging research problem as the medical datasets contain many complex patterns. In artificial intelligence, hybrid intelligent systems can support the data mining process to improve the accuracy of classification for medical diagnosis. Hybrid intelligent system is an integrated design of different artificial intelligence techniques such as neuro-fuzzy, genetic-fuzzy, etc., that has been successful in many applications such as data mining, computer vision, speech synthesis, etc. This paper proposes a hybrid intelligent method of integrating Naïve Bayes classifier and parallel fuzzy systems for the classification of type 2 diabetes. The proposed method employs multiple hybrid fuzzy systems in a parallel structure for effective classification on the data. The proposed method showed better classification accuracy of 90.26% when tested using the Pima diabetes dataset.

Keywords: Diabetes mellitus, Fuzzy logic, Medical data mining, Naïve Bayes.

1. Introduction

The medical and healthcare data that are available in the digital world can be analysed by using the artificial intelligence techniques which can facilitate medical diagnosis. For example, the machine learning algorithms has a great potential to discover patterns from the medical datasets for detection of various diseases. Diabetes mellitus is one of the serious diseases in the world that requires careful attention. Diabetes can cause death if not treated correctly at the earlier stage [1]. Both male and female can have chance for diabetes [2]. Diabetes can be extremely difficult for pregnant women. There are different types of diabetes such as type 1, type 2, and gestational diabetes [3]. Among these types, type 2 diabetes can be controlled by taking appropriate medications and following proper lifestyle guidelines [4].

In recent years, many hybrid intelligent architectures were proposed for classifying type 2 diabetes, out of which the hybrid fuzzy system [5, 6] is the most popular one. A fuzzy expert system is an implementation of fuzzy logic technique which is used for complex domains that involves analysis of complicated data. The fuzzy systems have an advantage over machine learning algorithms. The machine learning algorithms that extract patterns from the diabetic datasets will show outcomes only based on the datasets. The risk level of diabetes is not identified by the machine learning algorithms. For example, in the diabetic dataset where the outcome classes are non-diabetic and diabetic, the machine learning algorithms trained with the diabetic dataset will only be able to classify whether the person is non-diabetic or diabetic. But the risk level of diabetes between non-diabetic and diabetic is not identified. Designing a hybrid fuzzy system by integrating the machine learning algorithms with the fuzzy logic technique can benefit in identifying the risk level of diabetes from the dataset. The accuracy of the fuzzy system is based on the design of fuzzy sets and fuzzy rule base. As diabetes is a complicated disease, the design of proper fuzzy rule base for accurate prediction of diabetes is a research problem. There are various hybrid fuzzy approaches proposed for optimizing the fuzzy rules in fuzzy system to support the type 2 diabetes diagnosis.

Mansourypoor and Asadi [7] proposed a reinforcement learning-based evolutionary fuzzy rule-based system (RLEFRBS) for type 2 diabetes diagnosis. According to their proposed model, the numerical data are used to form the initial rules, then the genetic algorithm (GA) [8] is used to choose proper subset of rules for creating an initial rule base, then the evolutionary approach is used to tune the membership functions (MFs) of the rules, and then the reinforcement learning [9] is used to adjust rule weights to enhance the consistency among the rules in fuzzy system. The performance of RLEFRBS showed a classification accuracy of 84% when tested using Pima diabetes dataset [10].

Vaishali et al. [11] studied the classification of type 2 diabetes by combining the GA and multiple objective evolutionary (MOE) fuzzy classifier [12] where GA is used for feature selection. Their investigations applied MOE fuzzy classifier for the type 2 diabetes diagnosis because it functions based on the principle of maximum classification rate and minimum rules. Their proposed algorithm showed a classification accuracy of 83.04% when tested using Pima diabetes dataset.

Chen et al. [13] proposed a decision tree based neuro-fuzzy approach for classification of type 2 diabetes where the decision tree learning [14] is used to optimize the fuzzy rule base in adaptive neuro-fuzzy inference system (ANFIS) [15]. According to their proposed approach, initially a crisp rule base is produced

using decision tree learning, then the crisp rule base is transformed into fuzzy rule base where the Gaussian MFs is used for replacing the crisp intervals. The transformed fuzzy rule base is finally given as input to ANFIS. The performance of their proposed model showed a classification accuracy of 75.67% when tested using Pima diabetes dataset.

Cheruku et al. [16] proposed a model for generating fuzzy rules by combining rough set theory (RST) [17] and bat optimization algorithm (BA) [18]. According to their proposed model, RST based QUICK-REDUCT [19] algorithm is used for feature selection from the dataset. Then the BA with Ada-Boosting [20] technique is applied to generate fuzzy rules. Their proposed model showed a classification accuracy of 85.33 % when tested on Pima diabetes dataset.

The curse of dimensionality is the major issue in these approaches where the number of fuzzy rules rises exponentially with the number of input variables. For a large number of input fuzzy sets, the fuzzy expert system will become difficult to implement because of the size of rule base. Different types of fuzzy system architectures such as hierarchical or parallel architectures [21-23] can be utilized to overcome this drawback where the number of rules rises only linearly with the number of input variables.

The objective of this paper is to present an effective hybrid parallel fuzzy method that can overcome the curse of dimensionality problem found in the hybrid fuzzy systems and improve the classification accuracy in type 2 diabetes diagnosis. A hybrid intelligent method called Naïve Bayes (NB) based multiple parallel fuzzy reasoning (NB-MPFR) method is proposed in this paper which combines the NB classifier and parallel fuzzy reasoning approach for type 2 diabetes diagnosis. The NB-MPFR method utilizes the benefits of machine learning technique and parallel fuzzy reasoning approach where the NB classifier is used for selecting rules in the fuzzy rule base and the parallel fuzzy reasoning approach is used to overcome the curse of dimensionality problem found in fuzzy systems. The NB-MPFR method showed a good classification accuracy for type 2 diabetes diagnosis when implemented in a three-layered parallel fuzzy system.

The remaining part of this paper is organized as follows. A brief overview of fuzzy logic and NB algorithm is given in Section 2. The NB-MPFR method is described in Section 3. The application of NB-MPFR method in classification of type 2 diabetes is discussed in Section 4. The performance of NB-MPFR method in classification of type 2 diabetes is discussed in Section 5. Finally, the conclusion and future work are given in Section 6.

2. Background

2.1. Fuzzy logic

The fuzzy logic which is a kind of many valued logic was introduced by Zadeh [24] in the proposal of fuzzy set theory in the year 1965. Fuzzy logic has the ability to capture uncertainty in the data by supporting the concept of partial truth. In fuzzy logic, the truth values of variables may be of any value between 0 and 1 which represent completely false and completely true respectively. The fuzzy logic is implemented in the fuzzy expert system. There are different types of fuzzy inference systems and the most commonly used is the Mamdani fuzzy inference system [25]. In general, the fuzzy expert system involves the following steps.

2.1.1. Fuzzification

Fuzzification involves designing the fuzzy sets with appropriate degree of membership for all inputs and output values. The degree of membership is set within the interval [0,1]. There are different types of MFs [26] such as triangular, trapezoidal, gaussian, sigmoid, etc.

2.1.2. Rule execution

After creating the input and output fuzzy sets, a number of rules are defined in the rule base to perform fuzzy operation on the fuzzy sets using AND or OR operator. In each rule, the MFs of the input fuzzy sets are taken as antecedents and the MFs of the output fuzzy set are the consequent. When executing the rules, the outputs of all rules in the rule base are unified to form a single aggregate output fuzzy set [25].

2.1.3. Defuzzification

Defuzzification involves converting the aggregate output fuzzy set into a crisp output value. Out of different types of existing defuzzification methods, the most common method that is used in data mining applications is the centre of gravity (*COG*) method. The *COG* is computed over a continuum of points in the aggregate output fuzzy set by using Eq. (1) [27]

$$COG = \frac{\sum_{x=a}^{b} \mu_A(x)x}{\sum_{x=a}^{b} \mu_A(x)} \quad (1)$$

where A is the fuzzy set on the interval $[a, b]$, and $\mu_A(x)$ is the membership degree of the element x in the fuzzy set A.

2.2. NB

NB algorithm can be viewed as a probability classifier that employs Bayes theorem and is based on a strong independent assumption between all variables in the dataset, given the class variable [28]. The NB algorithm is easy to implement and has proved to be effective for many applications particularly in classification of large datasets by showing good classification accuracy despite of its simplified assumption. Suppose there is an instance x_i having n attributes, $A_1, A_2, \ldots, A_n$ with m classes, $c_1, c_2, \ldots, c_m$ in the dataset. The NB classifier classifies x_i in c_i if and only if $P(c_i|x_i) > P(c_j|x_i)$, for $1 \leq j \leq m$, $j \neq i$. $P(c_i|x_i)$ is computed using the Bayes theorem as shown in Eq. (2) and (3)

$$P(c_i|x_i) = \frac{P(x_i|c_i)P(c_i)}{P(x_i)} \quad (2)$$

$$\text{where } P(x_i|c_i) = \prod_{k=1}^{n} P(x_k|\, c_i) \quad (3)$$

Here, $P(x_i|c_i)$ is the probability of x_i for a given class c_i, $P(c_i)$ is the priori probability of c_i, and $P(x_i)$ is the priori probability of x_i.

When handling continuous data, the assumption is made that the continuous values associated with each class are distributed according to a gaussian distribution. For the gaussian distribution, the conditional probability $P(x_i|c_i)$ is computed using Eq. (4)

$$P(x_i|c_i) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x_i - m)}{2\sigma^2}\right) \quad (4)$$

where the parameter m is the mean, and σ^2 is the variance.

3. The proposed method

The NB-MPFR method presents a hybrid fuzzy reasoning approach in a parallel structure for the data classification problem by using several low-dimensional sub fuzzy systems (SFSs) which are integrated with the gaussian NB classifiers. The NB-MPFR method is strongly based on the training dataset. In NB-MPFR method, the training dataset is split into groups based on the attributes and accordingly the number of SFSs are created. The quartile values computed from each training dataset group are used to define the MFs of fuzzy sets in the corresponding SFSs. The gaussian NB classifier trained using each training dataset group is used to define the fuzzy rules in the corresponding SFSs. After the defuzzification process in all the SFSs, the average value of the crisp outputs obtained from all the SFSs is taken as the final output. The NB-MPFR method is described in detail through the following steps.

3.1. Splitting dataset into groups for SFSs

At first, the training dataset is split into groups based on the attributes such that each dataset group contains the data values of maximum of three input attributes and the data of target attribute. A number of SFSs are developed based on the dataset groups. For instance, if there are three dataset groups formed from the training dataset, then three SFSs are developed. Each SFS will correspond to each dataset group. The input attributes and the output classes of each dataset group will be the inputs and outputs of the corresponding SFS respectively.

3.2. Finding quartiles from dataset

For each input attributes in the SFSs, the minimum and maximum data values are identified from the corresponding dataset. Using the quartile concept, the number of data points in each attribute is divided around four equal parts. According to the quartile concept, the lower quartile or first quartile is the middle number between the minimum and median data value of the attribute. The second quartile is the median data value of the attribute. The upper quartile or third quartile is the middle number between the median and maximum data value of the attribute. Then finally, the middle number between the upper quartile and maximum data value of the attribute is considered as the uppermost quartile. For each input attribute, the lower quartile, second quartile, upper quartile and uppermost quartile values are computed from the corresponding dataset. Using these values, the input fuzzy set is designed for each input attribute. For each input attribute, the fuzzy set is designed in such a way that, the quartile values are used to define the parameters value of the MFs on x-axis of the fuzzy set. Each quartile value will represent the peak value of each MF in the fuzzy set. The endpoint of x-axis in the fuzzy set will be the minimum and maximum data value of the corresponding input attribute.

3.3. Fuzzification of dataset attributes in SFSs

The gaussian MF is used to set the membership degree in each input fuzzy set of the SFS. The gaussian MF for any two points on x-axis of the fuzzy set '*A*' can be represented by its mean *m* and standard deviation σ as shown in Eq. (5)

$$\mu_A(x) = exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) \quad (5)$$

Suppose there are A_i attributes where i = 1, 2, ..., n is the number of attributes. Let a^i be the data value of the A_i and let $a^i_{min}, a^i_{LQ}, a^i_{SQ}, a^i_{UQ}, a^i_{UMQ}, a^i_{max}$ be the minimum, lower quartile, second quartile, upper quartile, uppermost quartile and maximum data value of the input attribute respectively. Initially, a^i_{min} and a^i_{max} are used as the limits of x-axis in the input fuzzy sets. Then the values of $a^i_{LQ}, a^i_{SQ}, a^i_{UQ}, a^i_{UMQ}$ are used to design four gaussian MFs on the fuzzy sets such that each quartile value on x-axis of the fuzzy set will be the mean value for each gaussian MF. For each gaussian MF, the mean value is the corresponding quartile value, and the standard deviation is computed for the continuous data between the preceding quartile and succeeding quartile value. That is, for the first gaussian MF, the mean value is a^i_{LQ} and the standard deviation is computed between a^i_{min} and a^i_{SQ}. For the second gaussian MF, the mean value is a^i_{SQ} and the standard deviation is computed between a^i_{LQ} and a^i_{UQ}. For the third gaussian MF, the mean value is a^i_{UQ} and the standard deviation is computed between a^i_{SQ} and a^i_{UMQ}. Finally, for the fourth gaussian MF, the mean value is a^i_{UMQ} and the standard deviation is computed between a^i_{UQ} and a^i_{max}. Figure 1 shows the sample of how the MFs are designed on input fuzzy sets based on the quartile values.

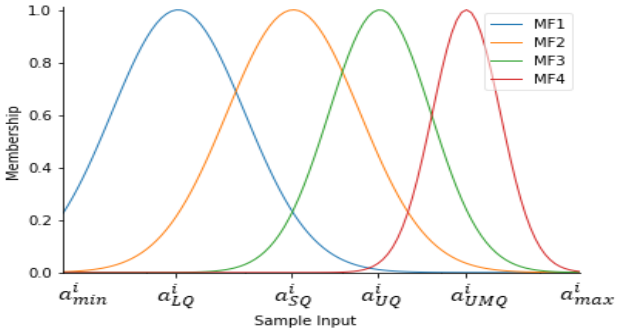

Fig. 1. Sample input fuzzy set based on quartile values.

The output fuzzy set of each SFS is based on the number of output classes in target attribute of the training dataset. Trapezoidal MF is used to plot the output class in the output fuzzy set. The trapezoidal MF can be represented by four parameters {*a*, *b*, *c*, *d*} which represent the lower limit, lower support limit, upper support limit, and upper limit respectively on x-axis of the fuzzy set *A* as shown in Eq. (6)

$$\mu_A(x) = \begin{cases} 0, & if\ x \leq a \\ \frac{x-a}{b-a}, & if\ a \leq x \leq b \\ 1, & if\ b \leq x \leq c \\ \frac{d-x}{d-c}, & if\ c \leq x \leq d \\ 0, & if\ d \leq x \end{cases} \quad (6)$$

For the output fuzzy set, the limits of x-axis in the fuzzy set are set randomly and the x-axis is split into equal parts to plot all trapezoidal MFs in the output fuzzy set. For instance, if the target attribute has two output classes, then two trapezoidal MFs are created for the output fuzzy set. Figure 2 shows the sample of output fuzzy set which has two trapezoidal MFs representing two output classes of the target attribute.

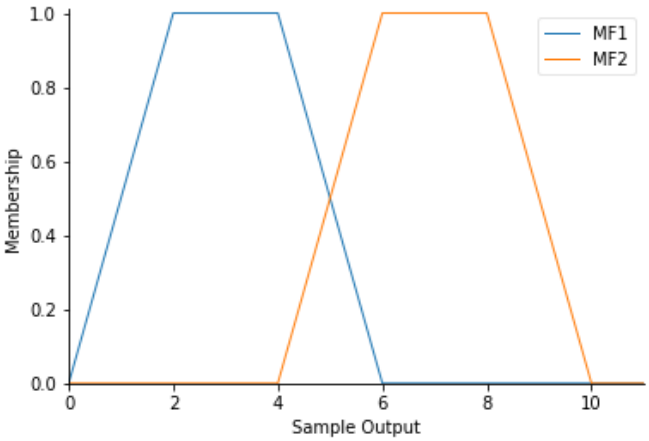

Fig. 2. Sample output fuzzy set.

3.4. Gaussian NB trained with dataset for fuzzy rule base

Gaussian NB classifiers are trained using the dataset groups. The gaussian NB classifier trained by a dataset group is used to define the fuzzy rules in the corresponding SFS. In the input fuzzy sets of each SFS, the quartile values marked on the x-axis represents the peak value of each MF. To define the fuzzy rules based on gaussian NB classifier, it is assumed that each quartile value on the x-axis represents the corresponding MF. For all possible combinations of the quartile values between different input fuzzy sets, their outcomes are predicted using the corresponding trained gaussian NB classifier which is in turn used to set the consequent MFs for the corresponding combinations of MFs between different input fuzzy sets in the fuzzy rule base.

Suppose $a^i_{LQ}, a^i_{SQ}, a^i_{UQ}, a^i_{UMQ}$ represent the MFs: μ_{A_i} [$MF1$], μ_{A_i} [$MF2$], μ_{A_i}[$MF3$], μ_{A_i}[$MF4$] respectively in the input fuzzy set. Let IC_k be the combination of quartile values between different input attributes where k = 1, 2, ..., n represent the number of possible combinations of quartile values between different input attributes and let R_k be the rule in which the antecedents are the combination of MFs between different input fuzzy sets where k = 0, 2, ..., n represent the number of possible combinations of MFs between different input fuzzy sets.

For instance, let IC_k represents the combination of inputs: ($a^0_{LQ}, a^2_{SQ}, a^3_{UQ}$, ..., a^n_{UMQ}) and R_k represents the rule having the antecedents as the corresponding combination of MFs as antecedent: (μ_{A_0} [$MF0$], μ_{A_2} [$MF2$], μ_{A_3} [$MF3$], ..., μ_{A_n} [$MF4$]). Then, if gaussian NB classifier prediction of IC_k is 1 then the consequent of R_k is set as μ_{OP}[$MF2$] where μ_{OP}[$MF2$] is the MF of output fuzzy set representing the output class 1 of target attribute in the training dataset. That is,

If prediction of gaussian NB (IC_k: $a^0_{LQ}, a^2_{SQ}, a^3_{UQ}$, ..., a^n_{UMQ}) = 1, then

R_k : IF (μ_{A_0}[$MF0$], μ_{A_2}[$MF2$], μ_{A_3}[$MF3$], ..., μ_{A_n}[$MF4$]) THEN μ_{OP}[$MF2$]

In this way, the gaussian NB classifiers trained by the dataset groups are used to set the consequent MFs of all rules for the corresponding combinations of MFs between different input fuzzy sets in the fuzzy rule base of all SFSs.

3.5. Defuzzification and final output

The defuzzification process of each SFS is done using the *COG* method to convert the aggregate output fuzzy set into crisp output. The final output of NB-MPFR method is the average value of the crisp outputs obtained from all the SFSs. Figure 3 shows the pseudocode of complete NP-MPFR method.

A_i: Input attribute of dataset, D where i = 1, 2, ..., n
T: Target attribute of D
Split D into groups ()
$D[A_1, A_2, \ldots, A_n \vee T] = D[A_1, A_2, A_3 \vee T], D[A_4, A_5, A_6 \vee T], \ldots, D[A_l, A_m, A_n \vee T]$
For all D groups ()
  Create SFS ()
    $SFS_1() = D_1 [A_1, A_2, A_3 \vee T]$
    $SFS_2() = D_2 [A_4, A_5, A_6 \vee T]$
    .....
    $SFS_n() = D_n [A_l, A_m, A_n \vee T]$
$SFS_i()$ //where i indicates the number of SFS//
  For each SFS_i ()
  For each A_i ()
    a^i_{min} = minimum data value of A_i
    a^i_{max} = maximum data value of A_i
    $a^i_{SQ} = (a^i_{min} + a^i_{max}) / 2$
    $a^i_{LQ} = (a^i_{min} + a^i_{SQ}) / 2$
    $a^i_{UQ} = (a^i_{SQ} + a^i_{max}) / 2$
    $a^i_{UMQ} = (a^i_{UQ} + a^i_{max}) / 2$
    Create Input Fuzzy Sets ()
      μ_{A_i}[MF1] = gaussian MF (m= a^i_{LQ} ; σ (start: a^i_{min}, stop: a^i_{SQ}))
      μ_{A_i}[MF2] = gaussian MF (m= a^i_{SQ} ; σ (start: a^i_{LQ}, stop: a^i_{UQ}))
      μ_{A_i}[MF3] = gaussian MF (m= a^i_{UQ} ; σ (start: a^i_{SQ}, stop: a^i_{UMQ}))
      μ_{A_i}[MF4] = gaussian MF (m= a^i_{UMQ} ; σ (start: a^i_{UQ}, stop: a^i_{max}))
    Create Output Fuzzy Set () //In case of binary output class//
    //where the parameters a, b, c, d, e, f represent the limits of trapezoidal MF//
      μ_{OP}[MF1] = trapezoidal MF (a, b, c, d)
      μ_{OP}[MF2] = trapezoidal MF (c, d, e, f)
    Fuzzy rule base ()
      For all IC_k in D ()
      For all R_k in SFS ()
        If gaussian NB-Prediction (IC_k: $a^1_{LQ}, a^2_{SQ}, a^3_{UQ}, \ldots, a^n_{UMQ}$) = 0 then,
        R_k :IF (μ_{A_1}[MF1], μ_{A_2}[MF2], μ_{A_3}[MF3], …, μ_{A_n}[MF4]) THEN μ_{OP}[MF1])
        Else
        If gaussian NB-Prediction (IC_k: $a^1_{LQ}, a^2_{SQ}, a^3_{UQ}, \ldots, a^n_{UMQ}$) = 1 then,
        R_k :IF (μ_{A_1}[MF1], μ_{A_2}[MF2], μ_{A_3}[MF3], …, μ_{A_n}[MF4]) THEN μ_{OP}[MF2])
    Defuzzification ()
      COG method ()
Computing final output ()
    Average value = $\dfrac{Output\ of\ SFS_1 + Output\ of\ SFS_2 + \ldots + Output\ of\ SFS_n}{n}$
    Final output = Average value

Fig. 3. Pseudocode of NP-MPFR method.

4. Application of NB-MPFR in Diabetes Diagnosis

The NB-MPFR method is implemented as a three-layered parallel fuzzy system using python programming language. Figure 4 shows the architecture of NB-MPFR system. The Pima diabetes dataset is used for testing the NB-MPFR method. The attributes of Pima diabetes dataset are number of times pregnant (Preg.), plasma glucose concentration 2 hours in an oral glucose tolerance test (Glu.), diastolic blood pressure (BP), triceps skin fold thickness (Tri.), 2-Hour serum insulin (Ser.), body mass index (BMI), diabetes pedigree function (Pedi), age, and the output classes: 0 (non-diabetic) and 1 (diabetic). The Pima diabetes dataset contains 768 samples where 500 samples belong to the output class of non-diabetic and 268 samples belong to the output class of diabetic [10]. As per the NB-MPFR method, the below steps are used for the classification of type 2 diabetes.

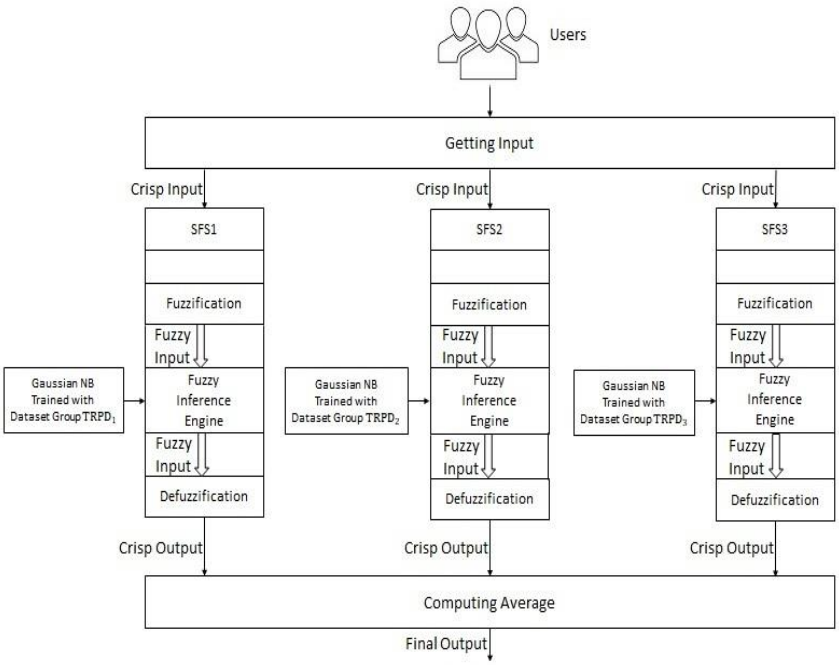

Fig. 4. NB-MPFR system architecture.

4.1. Splitting diabetes dataset into groups for SFSs

The Pima diabetes dataset is initially divided into three dataset groups based on the attributes where the first dataset group PD_1 has three input attributes: Preg., Glu., BP, and the target attribute, the second dataset group PD_2 has three input attributes: Tri., Ser., BMI, and the target attribute, the third dataset group PD_3 has two input attributes: Pedi, age, and the target attribute. The divided dataset groups (PD_1, PD_2, PD_3) are further divided into two groups as training dataset groups ($TRPD_1$, $TRPD_2$, $TRPD_3$) and testing dataset groups ($TEPD_1$, $TEPD_2$, $TEPD_3$) in 80:20 ratio respectively. The training dataset groups: $TRPD_1$, $TRPD_2$, $TRPD_3$ are used for designing the fuzzy sets and for training the gaussian NB classifier. The testing dataset groups: $TEPD_1$, $TEPD_2$, $TEPD_3$ are used for testing the NB-MPFR system.

As three dataset groups are obtained from the Pima diabetes dataset, a three-layered parallel fuzzy system is used to implement the NB-MPFR method. That is, three SFSs (SFS_1, SFS_2, SFS_3) are used in a parallel architecture to implement the NB-MPFR method where SFS_1 is based on the training dataset group, $TRPD_1$, SFS_2 is based on the training dataset group, $TRPD_2$ and SFS_3 is based on the training dataset group, $TRPD_3$. The inputs and outputs of the three SFSs are shown in Table 1 based their corresponding training dataset groups.

Table 1. Inputs and outputs of SFSs.

SFS	Inputs	Output
SFS_1	Preg., Glu., BP	non-diabetic or diabetic
SFS_2	Tri., Ser., BMI	non-diabetic or diabetic
SFS_3	Pedi, age	non-diabetic or diabetic

4.2. Finding quartiles from diabetes dataset

Initially the values of $a^i_{min}, a^i_{LQ}, a^i_{SQ}, a^i_{UQ}, a^i_{UMQ}, a^i_{max}$ are computed from each Pima training dataset group to plot four gaussian MFs in the input fuzzy sets of each SFS. As per the NB-MPFR method, the values of $a^i_{min}, a^i_{LQ}, a^i_{SQ}, a^i_{UQ}, a^i_{UMQ}, a^i_{max}$ are shown in the Table 2.

Table 2. Quartile values of input attributes.

Input Attributes	a^i_{min}	a^i_{LQ}	a^i_{SQ}	a^i_{UQ}	a^i_{UMQ}	a^i_{max}
Preg.	0	4.25	8.5	12.75	14.88	17
Glu.	0	49.5	99	148.5	173.25	198
BP	0	30.5	61	91.5	106.75	122
Tri.	0	24.75	49.5	74.25	86.63	99
Ser.	0	211.5	423	634.5	740.25	846
BMI	0	16.78	33.55	50.32	58.71	67.1
Pedi	0.08	0.66	1.25	1.83	2.13	2.42
age	21	36	51	66	73.5	81

4.3. Fuzzification of diabetes dataset attributes in SFSs

Based on the values shown in Table 2, the input fuzzy sets are created in the SFSs (SFS_1, SFS_2, SFS_3). For creating the input fuzzy sets, at first, the limits of x-axis in the fuzzy set are set from a^i_{min} to a^i_{max}. Then using the values: $a^i_{LQ}, a^i_{SQ}, a^i_{UQ}, a^i_{UMQ}$, four gaussian MFs are plotted in the input fuzzy sets of all SFSs.

As per the NB-MPFR method, four gaussian MFs (m, σ) are plotted in the input fuzzy sets of all SFSs by considering the quartile values of their corresponding attributes as the mean values m and the standard deviation values σ are computed over a continuous data between the respective preceding and succeeding quartiles of the mean quartiles as shown below.

$\mu_{A_i}[MF0]$ = (A_i, *gaussian MF* ($m = a^i_{LQ}$; σ (*start:* a^i_{min}, *stop:* a^i_{SQ})))

$\mu_{A_i}[MF2]$ = (A_i, *gaussian MF* ($m = a^i_{SQ}$; σ (*start:* a^i_{LQ}, *stop:* a^i_{UQ})))

$\mu_{A_i}[MF3]$ = (A_i, *gaussian MF* ($m = a^i_{UQ}$; σ (*start:* a^i_{SQ}, *stop:* a^i_{UMQ})))

$\mu_{A_i}[MF4]$ = (A_i, *gaussian MF* ($m = a^i_{UMQ}$; σ (*start:* a^i_{UQ}, *stop:* a^i_{max})))

Following this order, four gaussian MFs are created for each input fuzzy set in all the SFSs. Figure 5 shows the fuzzy sets of all input attributes. For the output fuzzy set, the limits of x-axis are set randomly from 0 to 10. As there are two output classes in the Pima diabetes dataset, the x-axis is split into two equal parts to plot two trapezoidal MFs that represent the outcome classes: non-diabetic and diabetic in the output fuzzy set. The output fuzzy set is same for all the three SFSs. Figure 6 shows the output fuzzy set. Table 3 shows the parameters of MFs for input and output fuzzy sets.

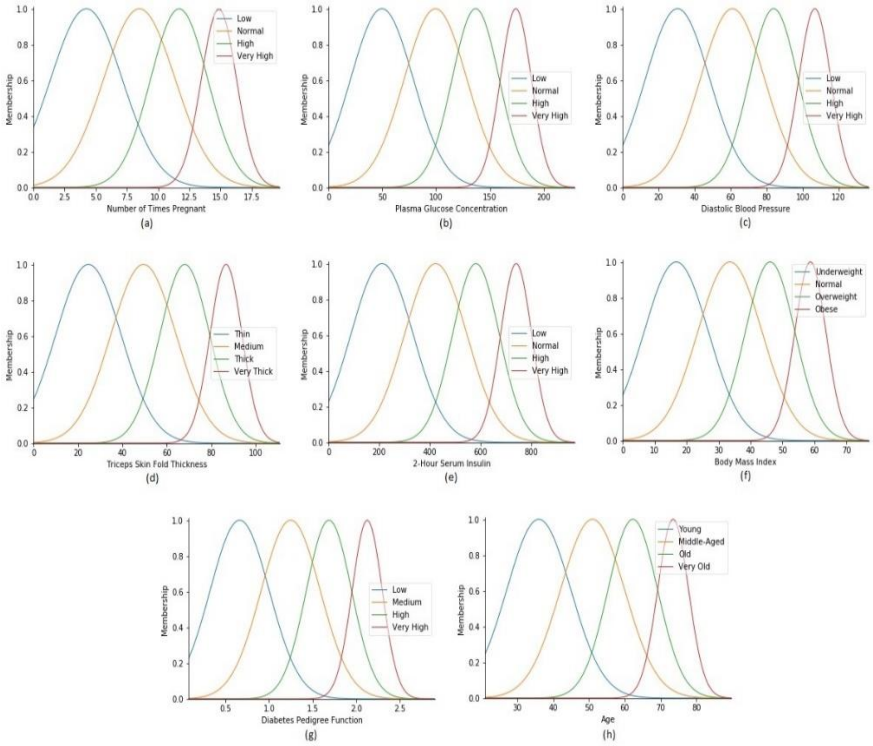

Fig. 5. Fuzzy sets of input attributes: (a) Preg., (b) Glu., (c) BP, (d) Tri., (e) Ser., (f) BMI, (g) Pedi, (h) age.

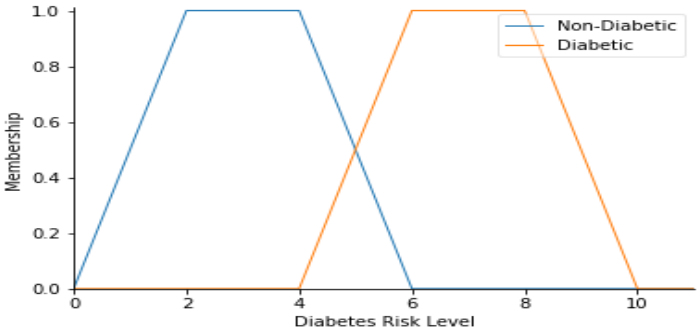

Fig. 6. Output fuzzy set.

Table 3. Parameters of MFs for input and output fuzzy sets.

Attributes	Linguistic Terms	Parameters of MFs
Preg.	"Low" "Normal" "High" "Very High"	gaussian MF (4.25, 2.87) gaussian MF (8.5, 2.87) gaussian MF (12.75, 2.29) gaussian MF (14.88, 1.41)
Glu.	"Low" "Normal" "High" "Very High"	gaussian MF (49.5, 28.87) gaussian MF (99, 28.87) gaussian MF (148.5, 21.94) gaussian MF (173.25, 14.43)
BP	"Low" "Normal" "High" "Very High"	gaussian MF (30.5, 17.9) gaussian MF (61, 17.9) gaussian MF (91.5, 13.56) gaussian MF (106.75, 8.94)
Tri.	"Thin" "Medium" "Thick" "Very Thick"	gaussian MF (24.75, 14.72) gaussian MF (49.5, 14.72) gaussian MF (74.25, 11.25) gaussian MF (86.63, 7.21)
Ser.	"Low" "Normal" "High" "Very High"	gaussian MF (211.5, 122.4) gaussian MF (423, 122.4) gaussian MF (634.5, 92.09) gaussian MF (740.25, 61.2)
BMI	"Underweight" "Normal" "Overweight" "Obese"	gaussian MF (16.78, 10.1) gaussian MF (33.55, 10.1) gaussian MF (50.32, 7.79) gaussian MF (58.71, 4.9)
Pedi	"Low" "Medium" "High" "Very High"	gaussian MF (0.66, 0.34) gaussian MF (1.25, 0.34) gaussian MF (1.83, 0.26) gaussian MF (2.13, 0.17)
Age	"Young" "Middle-Aged" "Old" "Very Old"	gaussian MF (36, 8.94) gaussian MF (51, 8.94) gaussian MF (66, 6.92) gaussian MF (73.5, 4.32)
Output	"Non-Diabetic" "Diabetic"	trapezoidal MF (0, 2, 4, 6) trapezoidal MF (4, 6, 8, 10)

4.4. Gaussian NB trained with diabetes dataset for fuzzy rule base

Gaussian NB classifiers are trained using the three training dataset groups: $TRPD_1$, $TRPD_2$, $TRPD_3$. The gaussian NB classifier trained with $TRPD_1$is used to set the

fuzzy rules in SFS_1 as per the NB-MPFR method. Similarly, the gaussian NB classifier trained with $TRPD_2$ is used to set the fuzzy rules in SFS_2 and the gaussian NB classifier trained with $TRPD_3$ is used to set the fuzzy rules in SFS_3. The receiver operating characteristics (ROC) graphs [29] were used to examine the performance of the gaussian NB classifiers. Figure 7 shows the ROC curves of gaussian NB for the three Pima diabetes dataset groups (PD_1, PD_2, PD_3). Gaussian NB classifiers trained using the three training dataset groups: $TRPD_1$, $TRPD_2$, $TRPD_3$ are used to predict the outcomes for the testing dataset groups: $TEPD_1$, $TEPD_2$, $TEPD_3$ respectively and the results are plotted using the ROC curves. According to the shape of ROC curves, the gaussian NB classifier is reasonably good for all the three Pima diabetes dataset groups (PD_1, PD_2, PD_3). When comparing the area under the ROC curves (AUC) of gaussian NB classifiers for the three dataset groups, it can be seen that the AUC of gaussian NB classifier for PD_1 is better than the AUC of gaussian NB classifier for PD_2 and PD_3.

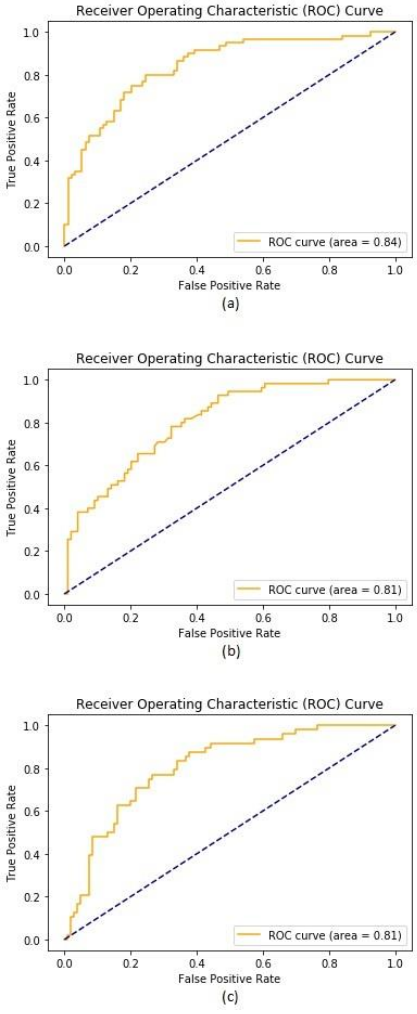

Fig. 7. ROC of gaussian NB for dataset groups: (a) PD_1, (b) PD_2, (c) PD_3.

4.5. Final output for diabetes diagnosis

The crisp output is obtained by using the *COG* defuzzification method in all the three SFSs. The final output of the three-layered parallel fuzzy system is the average value of the crisp outputs obtained from all the three SFSs.

5. Performance evaluation

The performance of NB-MPFR method implemented in a three-layered parallel fuzzy system is evaluated by using ROC graph, sensitivity, specificity, and classification accuracy. The sensitivity, specificity, and classification accuracy measures are based on the confusion matrix. In confusion matrix, the output class predicted by the model is compared with the actual class specified in the dataset using the test outcomes: true positive (*TP*), true negative (*TN*), false positive (*FP*), false negative (*FN*). Where *TP* is the number of testing samples belonging to class 1 (diabetic) that has been predicted correctly, *TN* is the number of testing samples belonging to class 0 (non-diabetic) that has been predicted correctly, *FP* is the number of testing samples belonging to class 1 that has been predicted incorrectly, and *FN* is the number of testing samples belonging to class 0 that has been predicted incorrectly. Using the confusion matrix, the sensitivity, specificity, and accuracy are calculated using Eqs. (7), (8), and (9) [30].

$$Sensitivity = \frac{TP}{TP+FN} \tag{7}$$

$$Specificity = \frac{TN}{TN+FP} \tag{8}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

The testing dataset groups ($TEPD_1$, $TEPD_2$, $TEPD_3$) are used to test the three-layered parallel fuzzy system. Figure 8 shows the ROC curves of NB-MPFR method. Table 4 shows the sensitivity, specificity, and accuracy of NB-MPFR method.

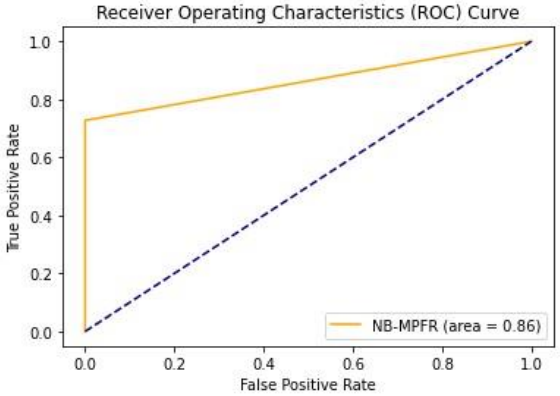

Fig. 8. ROC of NB-MPFR method.

Table 4. Performance of NB-MPFR method.

Proposed Method	Sensitivity (%)	Specificity (%)	Accuracy (%)
NB-MPFR	100	86.84	90.26

The performance of traditional machine learning algorithms which include decision tree learning, k-nearest neighbor (KNN) [31], NB, random forest algorithm [32], and support vector machine (SVM) [33] were also tested on Pima diabetes dataset with same 80:20 ratio. The performance of machine learning algorithms was evaluated using ROC curve and classification accuracy. Figure 9 shows the ROC curves of machine learning algorithm for Pima diabetes dataset. From Table 5, it can be seen that the NB-MPFR method shows a higher classification accuracy and AUC than decision tree, KNN, NB, random forest, and SVM classifiers. The NB method shows a classification accuracy of 74.68%, but the NB-MPFR method that integrates NB with the fuzzy system shows a higher classification accuracy of 90.26%. The reason is that the NB classifier is able to better classify the quarantine values of the input attributes. As the NB-MPFR method uses an effective method for designing the fuzzy sets and rule base which is based on the quarantine values of the dataset, the NB-MPFR method showed a higher AUC and classification accuracy. Table 6 compares the accuracy of NB-MPFR method with the other hybrid fuzzy methods that were tested using the Pima diabetes dataset. It is evident based on the comparison that the NB-MPFR method shows better classification accuracy in type 2 diabetes diagnosis.

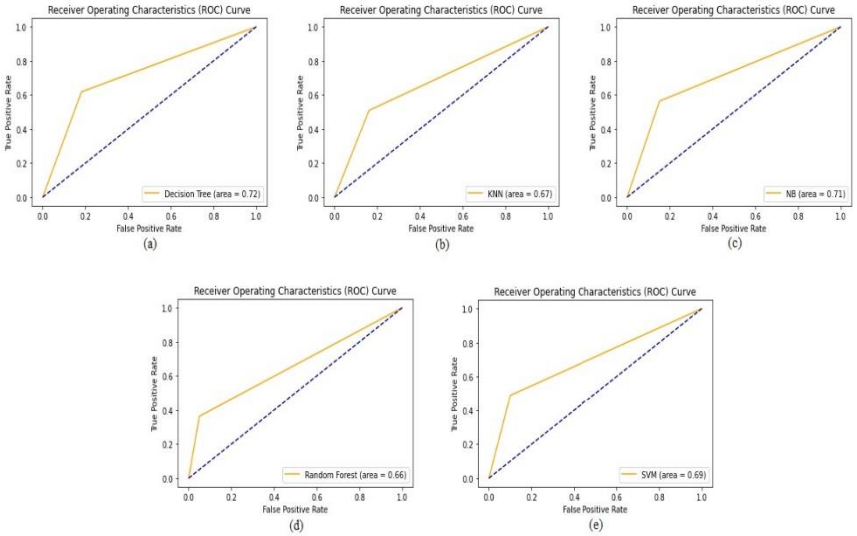

Fig. 9. ROC of (a) decision tree, (b) KNN, (c) NB, (d) random forest, (e) SVM.

Table 5. Performance comparison of NB-MPFR with machine learning algorithms.

Classification Methods	Classification Accuracy (%)	AUC
NB-MPFR	90.26	0.86
Decision Tree	74.68	0.72
KNN	72.08	0.67
NB	74.68	0.71
Random Forest	74.03	0.66
SVM	75.32	0.69

Table 6. Performance comparison of NB-MPFR with other hybrid fuzzy methods.

Classification Methods	Classification Accuracy (%)
NB-MPFR	90.26
RLEFRBS [7]	84
GA-MOE Fuzzy [11]	83.04
Decision Tree-ANFIS [13]	75.67
RST-BatMiner [16]	85.33

6. Conclusion

The major problem with the fuzzy system is that the number rules in fuzzy rule base rises exponentially with the number of input fuzzy sets. Designing appropriate fuzzy sets and fuzzy rule base based on the application is another complicated task. To deal with this issue, NB-MPFR method is proposed in this paper which integrates the gaussian NB classifiers with multiple parallel SFSs.

The NB-MPFR method implemented in a three-layered parallel hybrid fuzzy system has the advantage of minimizing the size of rule base when compared with the monolithic fuzzy system. The other advantage of the NB-MPFR method is that it is able to clear the uncertainty in the risk level of diabetes.

Based on the diabetes dataset, the input data of the patient is classified either as non-diabetic or diabetic. But the risk level of diabetes is uncertain. As the output fuzzy set of the fuzzy system gives the value between a defined range, the hybrid fuzzy system based on the NB-MPFR method is able to predict the risk level from a given range (0-10) for each patient's input data.

The NB-MPFR method showed a better classification accuracy in type 2 diabetes diagnosis when compared to the existing classification methods. The NB-MPFR method can be applied to similar types of applications that involve classification of complicated data.

The future work will be extending the NB-MPFR method to deal with more complicated applications by integrating it with different machine learning algorithms and feature selection methods.

Nomenclatures

$P(c_i)$	Prior probability of a given class, c_i
$P(c_i\|x_i)$	Posterior probability of class, c_i given an instance, x_i
$P(x_i)$	Prior probability of an instance, x_i
$P(x_i\|c_i)$	Probability of an instance, x_i for a given class, c_i

Greek Symbols

$\mu_A(x)$	Membership degree of the element, x in the fuzzy set, A
$\mu_A[MF]$	Membership function, MF for the fuzzy set, A
σ	Standard deviation
σ^2	Variance

Abbreviations

ANFIS	Adaptive Neuro-Fuzzy Inference System
BA	Bat Optimization Algorithm
BMI	Body Mass Index
COG	Centre of Gravity
FN	False Negative
FP	False Positive
GA	Genetic Algorithm
KNN	K-Nearest Neighbor
MF	Membership Function
MOE	Multiple Objective Evolutionary
NB-MPFR	Naïve Bayes based Multiple Parallel Fuzzy Reasoning
RLEFRBS	Reinforcement Learning-based Evolutionary Fuzzy Rule-based System
ROC	Receiver Operating Characteristics
RST	Rough Set Theory
SFS	Sub Fuzzy System
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

References

1. Seshasai, S.R.K.; Kaptoge, S.; Thompson, A.; Di Angelantonio, E.; Gao, P.; Sarwar, N.; Whincup, P.H.; Mukamal, K.h.; Gillum, R.F.; Holme, I.; Njølstad, I.; Fletcher, A.; Nilsson, P.; Lewington, S.; Collins, R.; Gudnason, V.; Thompson, S.G.; Sattar, N.; Selvin, E.; Hu, F.B.; and Danesh, J. (2011). Diabetes mellitus, fasting glucose, and risk of cause-specific death. *New England Journal of Medicine*, 364(9), 829-841.
2. Rao, P.V.; Ushabala, P.; Seshiah, V.; Ahuja, M.M.S.; and Mather, H.M. (1989). The Eluru survey: prevalence of known diabetes in a rural Indian population. *Diabetes Research and Clinical Practice*, 7(1), 29-31.
3. Kaul, K.; Tarr, J.M.; Ahmad, S.I.; Kohner, E.M.; and Chibber, R. (2013). Introduction to diabetes mellitus. *Diabetes*, Springer, New York, 1-11.
4. Stenlöf, K.; Cefalu, W.T.; Kim, K.A.; Alba, M.; Usiskin, K.; Tong, C.; Canovatchel, W.; and Meininger, G. (2013). Efficacy and safety of canagliflozin monotherapy in subjects with type 2 diabetes mellitus inadequately controlled with diet and exercise. *Diabetes, Obesity and Metabolism*, 15(4), 372-382.
5. Kirisci, M.; Yılmaz, H.; and Saka, M.U. (2019). An ANFIS perspective for the diagnosis of type II diabetes. *Annals of Fuzzy Mathematics and Informatics*, 17(2), 101-113.
6. Patil, R.; Tamane, S.; and Rawandale, N. (2020). Hybrid ANFIS-GA and ANFIS-PSO based models for prediction of type 2 diabetes mellitus. *In Proceedings of Computational Methods and Data Engineering*, Springer, Singapore, 11-23.
7. Mansourypoor, F.; and Asadi, S. (2017). Development of a reinforcement learning-based evolutionary fuzzy rule-based system for diabetes diagnosis. *Computers in Biology and Medicine*, 91, 337-352.

8. Forrest, S. (1996). Genetic algorithms. *ACM Computing Surveys* (*CSUR*), 28(1), 77-80.
9. Kaelbling, L.P.; Littman, M.L.; and Moore, A.W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4, 237-285.
10. Rossi, R.A.; and Ahmed, N.K. (2015). The network data repository with interactive graph analytics and visualization. *In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. Retrieved May 5, 2020, from http://networkrepository.com.
11. Vaishali, R.; Sasikala, R.; Ramasubbareddy, S.; Remya, S.; and Nalluri, S. (2017). Genetic algorithm based feature selection and MOE fuzzy classification algorithm on Pima Indians diabetes dataset. *Proceedings of the* 2007 *International Conference on Computing Networking and Informatics*, Lagos, 1-5.
12. Jiménez, F.; Sánchez, G.; and Juárez, J.M. (2014). Multi-objective evolutionary algorithms for fuzzy classification in survival prediction. *Artificial Intelligence in Medicine*, 60(3), 197-219.
13. Chen, T.; Shang, C.; Su, P.; Antoniou, G.; and Shen, Q. (2018). Effective diagnosis of diabetes with a decision tree-initialised neuro-fuzzy approach. *In UK Workshop on Computational Intelligence*, Springer, Cham, 227-239.
14. Breiman, L.; Friedman, J.H.; Olshen, R.A.; and Stone, C.J. (1984). CART: classification and regression trees, *Wadsworth and Brooks*/*Cole*, Monterey, CA.
15. Jang, J.S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), 665-685.
16. Cheruku, R.; Edla, D.R.; Kuppili, V.; and Dharavath, R. (2018). RST-Batminer: A fuzzy rule miner integrating rough set feature selection and bat optimization for detection of diabetes disease. *Applied Soft Computing*, 67, 764-780.
17. Pawlak, Z. (2012). Rough sets: Theoretical aspects of reasoning about data. *Springer Science & Business Media*, 9.
18. Yang, X.S. (2010). A new metaheuristic bat-inspired algorithm. *In Nature Inspired Cooperative Strategies for Optimization*, Springer, Berlin, Heidelberg, 65-74.
19. Shen, Q.; and Chouchoulas, A. (2000). A modular approach to generating fuzzy rules with reduced attributes for the monitoring of complex systems. *Engineering Applications of Artificial Intelligence*, 13(3), 263-278.
20. Schapire, R.E. (2013). Explaining adaboost. *In Empirical Inference*, Springer, Berlin, Heidelberg, 37-52.
21. Wang, L.X. (1999). Analysis and design of hierarchical fuzzy systems. *IEEE Transactions on Fuzzy systems*, 7(5), 617-624.
22. Razak, T.R.; Abd Halim, I.H.; Jamaluddin, M.N.F.; Ismail, M.H.; Fauzi, S.S.M.; and Gining, R.A.J. (2019). Towards designing a hierarchical fuzzy system for early diagnosis of heart disease. *Journal of Computing Research and Innovation*, 4(2), 31-41.
23. Jatobá, A.; Bellas, H.C.; Koster, I.; Burns, C.M.; Vidal, M.C.R.; Grecco, C.H.S.; and de Carvalho, P.V.R. (2018). Supporting decision-making in

patient risk assessment using a hierarchical fuzzy model. *Cognition, Technology & Work*, 20(3), 477-488.

24. Zadeh, L.A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338-353.
25. Mamdani, E.H.; and Assilian, S. (1999). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Human-Computer Studies*, 51(2), 135-147.
26. Negnevitsky, M. (2005). *Artificial intelligence: a guide to intelligent systems*. Pearson Education.
27. Van Leekwijck, W.; and Kerre, E.E. (1999). Defuzzification: criteria and classification. *Fuzzy Sets and Systems*, 108(2), 159-178.
28. Webb, G.I. (2010). Naïve Bayes. *Encyclopaedia of Machine Learning*, 15, 713-714.
29. Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861-874.
30. Zhu, W.; Zeng, N.; and Wang, N. (2010). Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations. *NESUG Proceedings: Health Care and Life Sciences*, Baltimore, Maryland, 67.
31. Altman, N.S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), 175-185.
32. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.
33. Cortes, C.; and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297.