

# Effective Diagnosis of Diabetes with a Decision Tree-initialised Neuro-fuzzy Approach

Tianhua Chen<sup>1</sup>, Changjing Shang<sup>2</sup>, Pan Su<sup>3</sup>, Grigoris Antoniou<sup>1</sup>, and Qiang Shen<sup>2</sup>

<sup>1</sup> Department of Computer Science, School of Computing and Engineering,  
University of Huddersfield, Huddersfield, UK

<sup>2</sup> Department of Computer Science, Institute of Mathematics, Physics and Computer  
Science, Aberystwyth University, Aberystwyth, UK

<sup>3</sup> School of Control and Computer Engineering, North China Electric Power  
University, Baoding, China

**Abstract.** Diabetes mellitus is a serious hazard to human health that can result in a number of severe complications. Early diagnosis and treatment is of significant importance to patients for the acquisition of a better quality life and precaution against subsequent complications. This paper proposes an approach by learning a fuzzy rule base for the effective diagnosis of diabetes mellitus. In particular, the proposed approach starts with the generation of a crisp rule base through a decision tree learning mechanism, which is data-driven and able to learn simple rule structures. The crisp rule base is then transformed into a fuzzy rule base, which forms the input to the powerful neuro-fuzzy framework of ANFIS, further optimising the parameters of both rule antecedents and consequents. Experimental study on the well-known Pima Indian diabetes data set is provided to demonstrate the promising potential of the proposed approach.

## 1 Introduction

Diabetes mellitus is a complex metabolic disorder characterised by persistent hyperglycemia, resulting from defects in insulin secretion, insulin action or both [1]. Diabetes is associated with a number of complications and it can increase the risk of developing blindness, blood pressure, heart disease, kidney disease and nerve damage [2]. Type 1 and type 2 are the most commonly seen forms of diabetes. In particular, type 2/adult-onset diabetes is associated with obesity and can be delayed or controlled with proper medication, healthy diet and exercise. It is therefore of significant importance to have an early detection of such disease to help patients suffering from it obtain a better quality life and subsequently prevail over the complications diabetes may bring.

The early diagnosis of type 2 diabetes is not an easy work, as diabetes patients show many symptoms in common with those that also appear in other types of disease. Despite recent medical progress, it has been reported that about half of the patients with type 2 diabetes are unaware of their disease and may

take more than ten years as the delay from disease onset to diagnosis while early diagnosis and treatment is vital [3]. With the staggering development of computer technology and the rapid computerisation of business nowadays, huge volume of medical data taken from patients and evaluations of medical experts is being accumulated at a dramatic pace. However, raw data is barely of direct interest unless potentially useful information is extracted, which can then be utilised for future diagnosis.

Knowledge discovery in databases refers to the overall process of extracting useful high-level knowledge from low-level data with data mining methods. Such knowledge is typically required to be human interpretable as well as computationally useful. In particular, techniques developed on the basis of fuzzy set theory (FST) generally facilitate the tolerance of imprecision, uncertainty and approximation, where many problems in real-life cannot be handled with binary encoding. Amongst those, fuzzy rule-based systems (FRBSs) are one of the most important applications of FST in data mining. A fuzzy rule-based model consists of a set of fuzzy rules in the form of if-then statement, specifying what actions or behaviour should be taken under given circumstances, which allows such a system to reason about how it reaches a conclusion and provide an explanation of its reasoning to human users. Many approaches [4,5,6,7] have been proposed for generating and learning FRBSs to represent the input-output behaviour with applications to a number of domain areas [8,9], including applications to medical diagnosis problems [10,11].

In working towards providing assistance for medical doctors to conduct effective diagnosis of diabetes, this paper proposes a neuro-fuzzy approach with a dedicated application to the well-known Pima Indian diabetes benchmark data. The proposed approach works by first discretising each of the continuous attributes into a certain number of categorical ones, such that the original continuous data set can be mapped onto a new data set with only nominal values, facilitating the rapid generation of a set of crisp rules through the exploitation of advanced decision tree learning. The generated crisp rules are able to reveal the basic relationship between attribute-value pairs, whereas the attribute-value pairs which do not appear in the rules can be removed. The generated crisp rules are then transformed to corresponding fuzzy rules with categorical values replaced by Gaussian membership functions. Finally, the set of fuzzy rules are adapted in the neuro-fuzzy framework of ANFIS with gradient descent and least square estimation, for the acquisition of an optimal set of accurate fuzzy rules. Experiments are provided to illustrate the working mechanism of the proposed approach and its performance.

The remainder of this paper is organised as follows. Section 2 introduces the background of fuzzy rules and ANFIS. Section 3 describes the proposed methodology. Section 4 presents and discusses a comparative experimental study. Section 5 concludes the paper and outlines ideas for further development.

## 2 Background

The task of learning an FRBS for the detection of diabetes is to find a finite set of fuzzy production or if-then rules capable of classifying a given input. Without losing generality, the classification system to be modelled is herein assumed to be multiple-input-single-output, receiving  $n$ -dimensional input patterns and producing one output which is determined to be of one of the  $M$  classes. The fuzzy rule set to be induced is required to perform the mapping  $\varphi : X^n \rightarrow Y$ , where  $X^n = X_1 \times X_2 \times \dots \times X_n$ ,  $X_1, X_2, \dots, X_n$  are the domains of discourse of the input variables and  $Y$  represents the set of possible output classes of a cardinality of  $M$ . The information about the behaviour of the system is described by a set of input-output example pairs  $E$ , where for each instantiation of the input variables  $\bar{x}^p = (x_1^p, x_2^p, \dots, x_n^p)^T$ ,  $x_i^p \in X_i, i = 1, 2, \dots, n$ , an associated class  $y^p \in Y$  is indicated.

Owing to its capability to approximate nonlinear functions to any degree of accuracy in any convex compact region and being computationally efficient [12], a TSK fuzzy if-then rule is adopted in this paper. In general, a TSK fuzzy if-then rule  $F_j$  can be represented as follows:

$$\text{If } x_1 \text{ is } A_{j1} \text{ and } \dots \text{ and } x_n \text{ is } A_{jn}, \text{ Then } z = f(x_1, \dots, x_n) \quad (1)$$

where  $j = 1, 2, \dots, N$ , with  $N$  denoting the number of all such fuzzy rules within the system;  $x_i, i = 1, \dots, n$  are the underlying domain variables, jointly defining the  $n$ -dimensional pattern space and respectively taking values from  $X_i$ ;  $A_{ji} \in X_i$  denotes a fuzzy set that the variable  $x_i$  may take; and  $z_j$  is the consequent of the fuzzy rule describing the output of the model within the fuzzy region specified by the antecedent part of the rule, which is defined as the first order polynomial in this paper, i.e.,  $z_j = \sum_{i=1}^n x_i p_i + q$ .

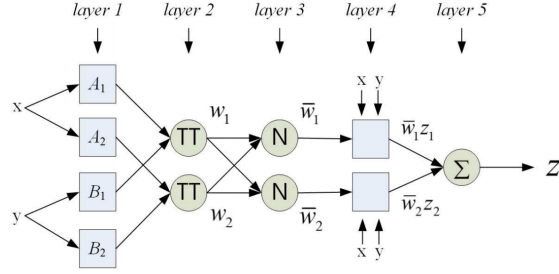
ANFIS [13] is a popular TSK fuzzy inference system built under the framework of artificial neural network, capturing the benefits of both neural networks and fuzzy logic principles. The general ANFIS architecture containing five layers can be simply illustrated with a two-input and one-output system. Suppose that there are only two fuzzy if-then rules in the rule base as follows:

$$\begin{aligned} \text{Rule 1: If } x \text{ is } A_1 \text{ and } y \text{ is } B_1, \text{ Then } z_1 &= p_1 x + q_1 y + r_1 \\ \text{Rule 2: If } x \text{ is } A_2 \text{ and } y \text{ is } B_2, \text{ Then } z_2 &= p_2 x + q_2 y + r_2 \end{aligned} \quad (2)$$

The structure of the neuro-fuzzy ANFIS that is equivalent to the flat fuzzy TSK rule base is shown in Figure 1, where square nodes stand for the network nodes with parameters to be adapted, and circle nodes represent fixed ones without modifiable parameters. For completeness, details of individual layers within the ANFIS are briefly summarised below.

**Layer 1:** Every node  $i$  in this layer is a square node with the following function:

$$O_i^1 = \mu_{A_i}(x) \quad (3)$$



**Fig. 1.** Illustrative ANFIS structure

where  $x$  denotes the input variable to node  $i$ , and  $A_i$  is the fuzzy set associated with this node. Usually the membership function  $\mu_{A_i}(x)$  is chosen to be bell-shaped, typically defined by the Gaussian membership function:

$$\mu_{A_i}(x) = e^{-\left(\frac{x-c_i}{\sigma_i}\right)^2} \quad (4)$$

where  $\{c_i, \sigma_i\}$  are the parameters associated with the corresponding variable, representing the mean value and standard deviation of the Gaussian membership function. These parameters are named premise parameters hereafter. Note that other continuous and piecewise differentiable functions, such as trapezoid or triangular functions may also be utilised if desired.

**Layer 2:** Every node in this layer is a circle node which accumulates the incoming values through multiplication and outputs the product. The output  $w_i$  in this layer acts as the firing strength of a certain rule (Rule  $i, i = 1, 2$  for the present example). That is,

$$w_i = \mu_{A_i}(x) \times \mu_{B_i}(y) \quad (5)$$

**Layer 3:** Each node in this layer is a circle node, computing the ratio of the  $i$ th rule's firing strength to the sum of all rules' firing strengths:

$$\bar{w}_i = \frac{w_i}{\sum_{j=1}^N w_j} \quad (6)$$

where again,  $i = 1, 2$  and the number of rules  $N = 2$ , for this particular example. The outputs of this layer are normalised firing strengths from previous layer.

**Layer 4:** Each node  $i$  in this layer is a square node with the following function:

$$O_i^4 = \bar{w}_i z_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (7)$$

where  $\bar{w}_i$  is the output of layer 3, and  $p_i, q_i, r_i$  are the parameters which appear in the rules and are referred to as consequent parameters hereafter.

**Layer 5:** The single node in this layer, the output layer, is a circle node that computes the overall output in response to all current inputs, defined as the summation of all incoming values, i.e.,

$$O_1^5 = \sum_i \bar{w}_i z_i = z \quad (8)$$

The parameters, including both premise and consequent ones, are trained using a hybrid learning method combining gradient descent and least square estimation.

### 3 Proposed Methodology

**Data Discretisation** The Pima Indian Diabetes dataset is obtained from the National Institute of Diabetes and Digestive and Kidney Diseases, which made studies on Pima Indian women of at least 21 years old and living at Phoneix, Arizona, USA [14]. The objective of the dataset is to predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. Among 768 cases, there are 268 (34.9%) positive tests for diabetes and 500 (65.1%) cases with tests being negative. There are eight continuous clinical measurements: 1) the number of times pregnant; 2) Plasma glucose concentration over 2 hours in an oral glucose tolerance test (mg/dl); 3) Diastolic blood pressure (mm Hg); 4) Triceps skin fold thickness (mm); 5) 2-hour serum insulin ( $\mu$  U/ml); 6) Body mass index (kg/m<sup>2</sup>); 7) Diabetes pedigree function; and 8) Age. A brief statistics of these eight features is given in Table 1.

**Table 1.** Statistics on Pima Indian diabetes data set

Attribute	Abbreviation	Mean	Standard deviation	Min/max
1	Pregnant	3.8	3.4	1/17
2	Glucose	120.9	32	56/197
3	DBP	69.1	19.4	24/110
4	TSFT	20.5	16	7/52
5	INS	79.8	115.2	15/846
6	BMI	32	7.9	18.2/57.3
7	DPF	0.5	0.3	0.0850/2.3290
8	Age	33.2	11.8	21/81

Each of the eight continuous variables is then discretised into a corresponding categorical one by simply partitioning the universe of discourse into a certain number of equal intervals. Each interval length is set to:  $intl = \frac{\max(X_i) - \min(X_i)}{L}$  where  $X_i$  is the domain of attribute  $i$  with  $\max(X_i)$  and  $\min(X_i)$  being its maximum and minimal value, respectively;  $L$  is the user-defined number of partitions. Any original value  $x_i$  is mapped onto the integer  $k$ , if  $intl_i^k \leq x_i < intl_i^{k+1}$ ,  $k \in [1, \dots, L + 1]$ , where  $intl_i^k$  is the  $k$ -th interval point for attribute  $i$ . In so doing, the original continuous attributes are transformed into ordinal integers.

Although such heuristic partitioning may not be optimal at this stage, which is to be optimised later with an adaptive method, such a simple empirical data discretisation comes with two advantages: First, each interval can now be attached with a linguistic label with interpretable meanings over pure numerical numbers that hardly make any sense especially to novices in the domain. Second, from computational perspective, the small number of categorical values help expedite the construction of a decision tree for the acquisition of an initial crisp rule base as to be introduced in the subsequent section.

**Generating Crisp Rules with Decision Tree Learning** Once the discretisation of the original data set is carried out, a set of crisp rules can be generated using a decision tree learning mechanism such as the Classification and Regression Tree (CART) [15]. The basic working of this learning method starts with the full data set at the root node and iteratively applies the Gini index as defined by

$$Gini(S) = \sum_{i=1}^M p_i \sum_{k \neq i} p_k = \sum_{i=1}^M p_i(1 - p_i) = \sum_{i=1}^M p_i - \sum_{i=1}^M p_i^2 = 1 - \sum_{i=1}^M p_i^2 \quad (9)$$

where  $S$  denotes the current data set for which this index is calculated;  $M$  is the number of class labels;  $p_i, i \in \{1, \dots, M\}$  is the probability of an object with the label  $i$  being randomly chosen; and  $\sum_{k \neq i} p_k = 1 - p_i$  represents the probability of a mistake in categorising an object. It can be seen that the Gini index reaches its minimum when all cases fall into a single category, and maximum when all items are equally distributed among all classes. As such, this index can be used to capture the amount of uncertainty in a dataset, measuring how often a randomly chosen object from the dataset may be incorrectly labelled, if it is randomly labelled according to the distribution of all the labels in the data.

At each split, a decision tree node is generated with the attribute for which the resulting Gini index is minimum. The same procedure is then recursed on each of its subsets using the remaining attributes. When there are no more attributes to be selected for further split or every element in the subset belongs to the same class, a complete decision tree is generated, which can be easily transformed into a set of crisp rules by retrieving paths from each leaf node backwards through its parent to the root node. Without losing generality, suppose that a crisp rule  $C_j, j = 1, 2, \dots, N$  (with  $N$  denoting the number of all crisp rules available) is given as follows:

$$\text{If } x_1 \text{ is } I_{j1} \text{ and } \dots \text{ and } x_n \text{ is } I_{jn}, \text{ Then class is } y^{C_j} \quad (10)$$

where  $x_1, x_2, \dots, x_n$  represent the underlying domain attributes;  $I_{ji}, i \in \{1, 2, \dots, n\}$ , is the crisp interval of the antecedent attribute  $x_i$  that may be attached with a meaningful label for linguistic interpretation; and  $y^{C_j}$  is a class label, acting as the rule consequent.

**Converting Crisp Rules into Fuzzy Rules** The above data-driven set of crisp rules can then be converted into a set of corresponding fuzzy rules for further optimisation. From the viewpoint of rule structures, a rule is made up of an antecedent and a consequent part, be it fuzzy or crisp. Both the fuzzy and crisp rule antecedent are of a conditional statement form, describing the values that the antecedent attributes should take in order to derive the corresponding consequent, which are connected by logical operators. The only difference is that the attributes in a crisp rule are described with a crisp intervals, whereas an attribute in a fuzzy rule is depicted by a fuzzy set. Therefore, a straightforward approach is to replace each crisp interval (represented by an integer) with a fuzzy

membership function. In this work, a crisp interval  $I_i$ ,  $i \in \{1, 2, \dots, n\}$  as shown in Eqn. 10 is replaced with a Gaussian membership function  $\mu_{A_i}(x_i) = e^{-\left(\frac{x-c_i}{\sigma_i}\right)^2}$ , where  $c_i$  and  $\sigma_i$  are the mean value and standard deviation of the Gaussian membership function. Here, the mean value is set to the average of those values belonging to the corresponding crisp interval  $I_i$ , such that

$$c_i = \frac{\sum_{x_i \in I_i} x_i}{|\{x_i \in I_i\}|} \quad (11)$$

Similarly, the standard deviation is set to:

$$\sigma_i = \sqrt{\frac{\sum_{x_i \in I_i} (x_i - c_i)^2}{|\{x_i \in I_i\}|}} \quad (12)$$

Once the process of replacing crisp intervals with the above Gaussian membership functions is complete, the transformation of the entire crisp rule antecedent finishes with the logical ‘AND’ connector in the original crisp rules replaced with a T-norm operator that performs fuzzy ‘AND’ operation such as product in ANFIS (or minimum in Mamdani models). The consequent of a crisp rule with a decision class is then directly mapped onto that of the corresponding fuzzy rule. As the first order polynomial TSK rule shown in Eqn. 2 is utilised in this paper, the integer that represents the decision class in the crisp rule is taken as the bias term in the fuzzy rule. The resulting mapped fuzzy rule from an original crisp rule can thus, be generally represented as

$$\text{If } x_1 \text{ is } e^{-\left(\frac{x-c_1}{\sigma_1}\right)^2} \text{ and } \dots \text{ and } x_n \text{ is } e^{-\left(\frac{x-c_n}{\sigma_n}\right)^2}, \text{ Then } z = 0x_1 + \dots + 0x_n + r \quad (13)$$

where  $e^{-\left(\frac{x-c_i}{\sigma_i}\right)^2}$  is the fuzzy membership function for attribute  $x_i$ ,  $i = 1, \dots, n$  with  $c_i$  and  $\sigma_i$  calculated as above, and  $r$  is the integer that represents the decision class of the corresponding crisp rule.

Note that running the conventional method of grid partitioning [16] of each and every input space may suffer from the curse of dimensionality as the number of inputs increases. Therefore, instead of considering all of the possible combinations of the input and class attributes, it is herein proposed to utilise the existing crisp rules, which have been generated by decision tree learning and which are able to efficiently and sufficiently generalise the given data to guide the transformation, without resorting to pure and brute force search. Being fundamentally data-driven, such a rule generation method will omit the empty parts of the input space, substantially expediting the subsequent optimisation process.

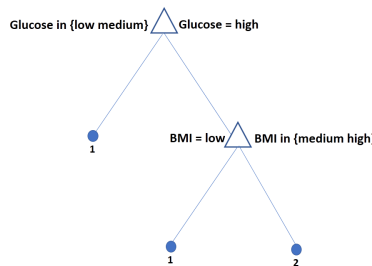
**Optimising Transformed Fuzzy Rules with ANFIS** As introduced in Section 2, ANFIS uses the framework of neural networks to represent an existing set of fuzzy rules, which are then optimised by tuning premise and consequent parameters. The set of fuzzy rules converted from the generated decision tree rules are herein employed as the initial fuzzy rule set to specify the ANFIS architecture. The ANFIS parameters are optimised with a hybrid learning algorithm

[13] that combines the gradient descent method and least squares estimator. In particular, each epoch of the hybrid learning procedure is composed of a forward pass and a backward pass. In running the forward pass, the antecedent parameters are fixed and a vector of input values is presented, and then the error between the actual output and the target output is calculated. In the backward pass, the error computed at the last forward pass is propagated backwards, from the output end towards the input end while fixing the consequent parameters, by the gradient method. The details of such an iteration of forward and backward computation is beyond the scope of this paper, but can be found in [13].

## 4 Experimental Study

To demonstrate the proposed approach at work for effectively aiding in diagnosis of diabetes, the experimental study is performed on the popular Pima Indian diabetes data set obtained from UCI machine learning repository [17], whose statistical information has already been listed in Table 1. Due to space limit only investigations into this single dataset are reported here.

As the range of different attributes vary significantly, a preprocessing step is to normalise each attribute so that their normalised values fall within the range of  $[0, 1]$ . This helps reduce the potential adverse effect caused by certain attributes being dominating, especially those undesired ones just because of their bigger value ranges. In the absence of testing data for the performance evaluation of the proposed approach, tenfold cross-validation (10-CV) is employed for result validation. In 10-CV, the given data set is partitioned into ten subsets. Of the ten, nine subsets are used to perform training, where the proposed approach is used to generate the desired fuzzy rule base, and the remaining single subset is retained as the testing data for assessing the learned classifier's performance. This cross-validation process is then repeated ten times in order to lessen the impact of any random factors.



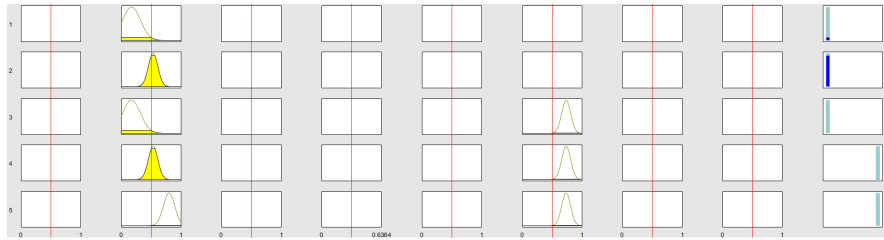
**Fig. 2.** Generated decision tree

The decision tree algorithm utilised in this paper is the popular Classification and Regression Trees (CART), which is characterised by its construction of a binary tree with each internal node having exactly two outgoing branches.



In particular, each original continuous attribute is empirically discretised into categorical one with 3 equally spaced bins. For instance, the resulting decision tree structure, which is taken from a single fold out of the complete 10-CV, is shown in Figure 2, where '1' and '2' in the leaf nodes stand for negative and positive test respectively; and 'low', 'medium' and 'high' are the labels used to denote the corresponding discretised crisp intervals. Note that certain attributes can take more than one interval as its value such as that Glucose can take either low or medium in its left branch, which is attributed to the mechanism that CART grows the trees. However, this can easily be transformed into a rule base with each attribute only taking a single value as follows:

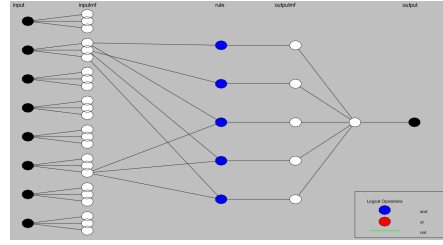
- Rule 1: If Glucose is *low*, Then test *negative*;
- Rule 2: If Glucose is *medium*, Then test *negative*;
- Rule 3: If Glucose is *high* and BMI is *low*, Then test *negative*;
- Rule 4: If Glucose is *high* and BMI is *medium*, Then test *positive*;
- Rule 5: If Glucose is *high* and BMI is *high*, Then test *positive*.



**Fig. 3.** Transformed fuzzy rule base

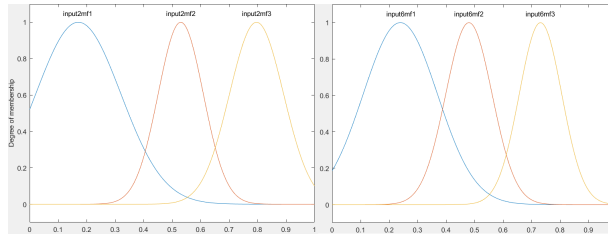
In this example, only two attributes out of the original eight are utilised to initiate the transformation process, with five rules generated, which could have been as many as  $3^8 \times 2$  if grid partitioning were used for rule generation. The above crisp rule base is then transformed into a fuzzy one with crisp intervals replaced by Gaussian membership functions as specified in Section 3. The resultant fuzzy rule base is shown in Figure 3. Each numbered row represents a converted fuzzy rule, where the last rectangle represents the output and the first eight rectangles are the input fuzzy membership functions. Obviously, only the second and the sixth rectangle come with Gaussian membership functions given that just the two corresponding attributes are utilised in the original crisp rules.

The transformed rule base serves as the input to the neuro-fuzzy ANFIS structure as shown in Figure 4. ANFIS then fine-tunes both the antecedent and consequent parameters based on the existing rule base structure by running the hybrid learning method. With regard to the two attributes used, Figure 5 shows the definitions of the initial membership functions, whose optimised definitions are shown in Figure 6. In order to better visualise the significant performance improvement after optimisation with ANFIS, Table 2 lists the accuracy of each

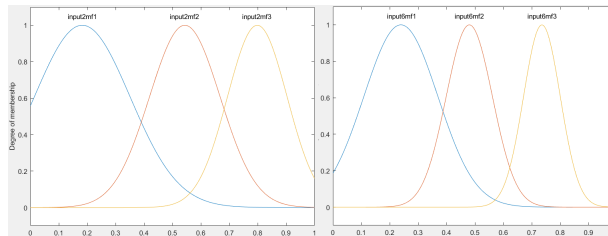


**Fig. 4.** ANFIS structure

10-CV fold on training data with initial rule base (Trn\_Initial\_ANFIS), training data with ANFIS optimised rule base (Trn\_Optimised\_ANFIS) and testing data with optimised rule base (Tst\_Optimised\_ANFIS). Clearly, the performance of each individual fold on training data is significantly improved along with ANFIS optimisation.



**Fig. 5.** Initial membership functions of 2nd attribute (Glucose) and 6th attribute (BMI)



**Fig. 6.** Optimised membership functions of 2nd attribute (Glucose) and 6th attribute (BMI)

To further compare the performance on testing data with alternative approaches, Table 3 lists the accuracy with another five popular learning classifiers. These include PTTD [5], a state-of-the-art fuzzy pattern-tree learning classifier that is composed of an ensemble of pattern trees; GP-COACH [7], a genetic programming-based learning approach that learns rules of a disjunctive normal form; SMO, a sequential optimisation algorithm for building support

**Table 2.** Accuracy (%) over data classified with different rule sets

Fold	Trn_Initial_ANFIS	Trn_Optimised_ANFIS	Tst_Optimised_ANFIS
1	75.11	79.45	77.92
2	73.52	76.85	80.52
3	63.24	81.19	74.03
4	74.13	78.32	81.58
5	67.73	81.62	75.32
6	76.12	79.16	76.62
7	72.50	83.50	61.04
8	73.95	81.62	75.32
9	75.58	79.34	82.89
10	70.04	83.65	71.43
Average	72.19	80.47	75.67

vector machines with the polynomial kernel adopted as kernel function; IBk, the classical  $k$ -nearest neighbour approach, and RIPPER which is a crisp rule induction algorithm following a separate-and-conquer strategy. Apart from the powerful support vector machines, the proposed approach beats all the other popular learning classifiers, demonstrating its promising performance.

**Table 3.** Accuracy (%) comparison with alternative methods

Proposed Approach	PTTD	GP-COACH	SMO	IBk	RIPPER
75.67	74.13	75.13	76.80	70.62	66.88

## 5 Conclusion

This paper has proposed an effective approach to learn a TSK fuzzy rule base with application to the diagnosis of diabetes mellitus. The proposed approach starts with the generation of a crisp rule base with a simple decision tree learning mechanism, which is data-driven and able to learn basic rule structures that reflect the characteristics between domain input and output attributes, with minor computational overheads. The crisp rule base is then transformed into a fuzzy rule base with crisp intervals replaced by Gaussian membership functions. This forms the input to the powerful neuro-fuzzy framework of ANFIS, which further optimises the rules' antecedent and consequent parameters. The resultant optimised fuzzy rules that are transformed from simple crisp rules show promising results with better or comparable performance to those derived from popular machine learning algorithms.

Whilst promising, interesting work remains. This includes: examining the influence of crisp rule bases of a different complexity upon the performance of the final fuzzy rules and the effect of any subsequent fine-tuning with ANFIS; investigating the use of more powerful data discretisation techniques (e.g., [18]); and comparing the proposed approach with alternative methods (e.g., [19]) that employ fuzzy systems for diabetes detection.

## References

1. R. I. Holt and N. A. Hanley, *Essential endocrinology and diabetes*. John Wiley & Sons, 2012, vol. 41.

2. H. Temurtas, N. Yumusak, and F. Temurtas, "A comparative study on diabetes disease diagnosis using neural networks," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8610–8615, 2009.
3. K. Polat, S. Güneş, and A. Arslan, "A cascade learning system for classification of diabetes disease: Generalized discriminant analysis and least square support vector machine," *Expert Systems with Applications*, vol. 34, no. 1, pp. 482–487, 2008.
4. T. Chen, C. Shang, P. Su, and Q. Shen, "Induction of accurate and interpretable fuzzy rules from preliminary crisp representation," *Knowledge-Based Systems*, vol. 146, pp. 152–166, 2018.
5. R. Senge and E. Hüllermeier, "Fast fuzzy pattern tree learning for classification," *Fuzzy Systems, IEEE Transactions on*, vol. 23, no. 6, pp. 2024–2033, 2015.
6. T. Chen, Q. Shen, P. Su, and C. Shang, "Fuzzy rule weight modification with particle swarm optimisation," *Soft Computing*, vol. 20, no. 8, pp. 2923–2937, 2016.
7. F. J. Berlanga, A. Rivera, M. J. del Jesús, and F. Herrera, "Gp-coach: Genetic programming-based learning of compact and accurate fuzzy rule-based classification systems for high-dimensional problems," *Information Sciences*, vol. 180, no. 8, pp. 1183–1200, 2010.
8. P. Su, Q. Shen, T. Chen, and C. Shang, "Ordered weighted aggregation of fuzzy similarity relations and its application to detecting water treatment plant malfunction," *Engineering Applications of Artificial Intelligence*, vol. 66, pp. 17–29, 2017.
9. P. Su, C. Shang, T. Chen, and Q. Shen, "Exploiting data reliability and fuzzy clustering for journal ranking," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1306–1319, 2017.
10. C. Zou and H. Deng, "Using fuzzy concept lattice for intelligent disease diagnosis," *IEEE Access*, vol. 5, pp. 236–242, 2017.
11. J. Wang, Y. Hu, F. Xiao, X. Deng, and Y. Deng, "A novel method to use fuzzy soft sets in decision making based on ambiguity measure and dempster–shafer theory of evidence: an application in medical diagnosis," *Artificial Intelligence in Medicine*, vol. 69, pp. 1–11, 2016.
12. G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 5, pp. 676–697, 2006.
13. J.-S. Jang, "Anfis: adaptive-network-based fuzzy inference system," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 23, no. 3, pp. 665–685, 1993.
14. W. C. Knowler, P. H. Bennett, R. F. Hamman, and M. Miller, "Diabetes incidence and prevalence in pima indians: a 19-fold greater incidence than in rochester, minnesota," *American Journal of Epidemiology*, vol. 108, no. 6, pp. 497–505, 1978.
15. L. Breiman, *Classification and regression trees*. Routledge, 2017.
16. L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 6, pp. 1414–1427, 1992.
17. K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>
18. T. Boongoen, C. Shang, N. Iam-On, and Q. Shen, "Extending data reliability measure to a filter approach for soft subspace clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 6, pp. 1705–1714, 2011.
19. R. Lukmanto and E. Irwansyah, "The early detection of diabetes mellitus (dm) using fuzzy hierarchical model," *Procedia Computer Science*, no. 59, pp. 312–319, 2015.