SEP **TNM**

INSTITUTO TECNOLÓGICO DE TIJUANA

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

Nuevo enfoque Modular para Modelos de Clasificación basados en Lógica Difusa Tipo-2 y Redes Neuronales

TRABAJO DE TESIS

PRESENTADO POR

EDUARDO RAMIREZ FLORES

PARA OBTENER EL GRADO DE

DOCTOR EN CIENCIAS

EN COMPUTACION

DIRECTOR DE TESIS

DRA. ELBA PATRICIA MELIN OLMEDA

TIJUANA, BC., JUNIO 2021

INSTITUTO TECNOLOGICO DE TIJUANA
POSGRADO EN COMPUTACION

Asunto: Se autoriza impresión de Trabajo de Tesis

Tijuana, B.C., 16 de Junio del 2021

C. DRA. YAZMIN MALDONADO ROBLES
Jefe de la Div. de Estudios de Posgrado e Investigación
Presente.

En lo referente al trabajo de tesis escrito, con título " **Nuevo enfoque Modular para Modelos de Clasificación basados en Lógica Difusa Tipo-2 y Redes Neuronales"**, presentado por el ***C. Eduardo Ramírez Flores,*** alumno del Doctorado en Ciencias en Computación con número de control **D99210205**, informamos a usted que se autoriza el escrito de tesis y se aprueba en todas sus partes, en virtud de reunir los requisitos de un trabajo de grado de Doctorado y a la vez se autoriza al interesado para que proceda de inmediato a la impresión del mismo y a presentar su examen de grado, ya que cumple con todos los requisitos.

ATENTAMENTE

DRA. ELBA PATRICIA MELIN OLMEDA
PRESIDENTE

DR. OSCAR CASTILLO LOPEZ
SECRETARIO

DRA. DANIELA ADRIANA SANCHEZ VIZCARRA
VOCAL

DR. FEVRIER ADOLFO VALDEZ ACOSTA
VOCAL

DR. JOSE MARIO GARCIA VALDEZ
VOCAL

c.c.p. Oficina de Titulación
c.c.p. División de Estudios de Posgrado e Investigación
c.c.p. Expediente
c.c.p. Interesado

EPMO/*inf

CARTA DECLARACIÓN DE PROPIEDAD INTELECTUAL

Tijuana, BC a 17 de junio del 2021

Yo **EDUARDO RAMIREZ FLORES** reconozco que el Trabajo de Tesis de Doctorado que realice durante mis estudios en el Doctorado en Ciencias en Computación del Instituto Tecnológico de Tijuana fue parte del Proyecto de Investigación Titulado: **Nuevo enfoque Modular para Modelos de Clasificación basados en Lógica Difusa Tipo-2 y Redes Neuronales No. 257008** que desarrolla mi directora de tesis el **Dra. Elba Patricia Melin Olmeda**-y del cual es responsable del proyecto de investigación. Por esta razón, los métodos, modelos, algoritmos, y software realizados, así como datos y resultados obtenidos durante el desarrollo de mi tesis de doctorado son propiedad intelectual de mi Director de Tesis, del Tecnológico Nacional de México, Instituto Tecnológico de Tijuana y del Conacyt, y No podré utilizarlos por mi cuenta durante, Ni después de terminar mi beca o estudios, excepto a solicitud escrita para poder utilizarlos bajo una colaboración directa con mi directora de tesis la cual es responsable del proyecto de investigación. Por tanto, estoy de acuerdo en que No podre utilizar ni tomar modelos, ni datos utilizados en este proyecto de investigación y en el desarrollo de tesis para: presentaciones, publicaciones ni desarrollo de mi propia investigación que pudiera desarrollar una vez concluidos mis estudios.

Atentamente

EDUARDO RAMIREZ FLORES

Estudiante del Doctorado en Ciencias en Computación
Instituto Tecnológico de Tijuana

DECLARACIÓN DE ORIGINALIDAD

Tijuana, BC., 17 de Junio de 2021,

Yo, **EDUARDO RAMIREZ FLORES**, estudiante del Doctorado en Ciencias en Computación, en mi calidad de autor manifiesto que este documento de tesis es producto de mi trabajo original y que no infringe los derechos de terceros, tales como derechos de publicación, derechos de autor, patente y similaridad. Por lo tanto, la obra realizada es de mi exclusiva autoría y no infringí en copiar el texto o imágenes, de fuentes de información por lo cual soy responsable del escrito que aquí se presenta.

Así mismo, declaro que en las citas textuales que he incluido (las cuales aparecen entre comillas) y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y las publicaciones.

En caso de presentarse cualquier reclamación o acción por parte de terceros en cuanto a los derechos de autor sobre la obra en cuestión, acepto toda la responsabilidad de tal infracción y relevo de esta a mi director de tesis, así como al Tecnológico Nacional de México, al Instituto Tecnológico de Tijuana y a sus respectivas autoridades.

EDUARDO RAMIREZ FLORES
Nombre completo del estudiante y firma
Estudiante de Doctorado en Ciencias en Computación

Resumen

En la actualidad, diferentes métodos han sido utilizados para resolver problemas de clasificación, sin embargo, existen problemas complejos en los que un método no puede resolver satisfactoriamente el problema. En el área médica, en específico cardiología, diferentes métodos han sido utilizados para ayudar en la interpretación de diversas morfologías o patrones electrocardiográficos que representan diferentes enfermedades cardiacas que son diagnosticadas por la interpretación de los mismos como son LVQ, MLP, modelos de Markov, mapas auto-organizados, sistemas lineales discriminantes, enfoques Bayesianos, SVM, sistemas expertos, sistemas de inferencia difusos, así como también sistemas híbridos, entre otros.

En este trabajo de tesis la principal aportación es la creación de un nuevo clasificador con enfoque modular utilizando lógica difusa tipo 2 y redes neuronales artificiales.

La metodología propuesta se aplicó como casos de estudio en el ámbito médico utilizando las bases de datos de electrocardiogramas, primeramente la base datos de arritmias del MIT-BIH y posteriormente la base de datos diagnóstica de electrocardiogramas PTB. Finalmente se muestra un estudio estadístico para demostrar las ventajas de la metodología propuesta con respecto a otros métodos.

Abstract

Currently, different methods have been used to solve classification problems, however, there are complex problems in which a method cannot satisfactorily solve the problem. In the medical area, in specific, different methods have been used to help in the interpretation of various morphologies or electrocardiographic patterns that represent different heart diseases that are diagnosed by their interpretation, such as LVQ, MLP, Markov models, self-organized maps, Discriminant linear systems, Bayesian approaches, SVM, expert systems, fuzzy inference systems, as well as hybrid systems, among others.

In this thesis work, the main contribution is the creation of a new classifier with a modular approach using type 2 fuzzy logic and artificial neural networks.

The proposed methodology was applied as case studies in the medical field using the electrocardiogram databases, firstly the MIT-BIH arrhythmia database and later the PTB diagnostic ECG database. Finally, a statistical study is shown to demonstrate the advantages of the proposed methodology with respect to other methods.

Dedication

First of all, to God, for the blessing of allowing me to reach this stage in my professional life and my family, especially my wife and children and my parents for the support, trust and encouragement they have given me during this time in my academic preparation.

Acknowledgment

To my family, for supporting me during the continuation of my post graduate studies.

To my thesis director Dra. Patricia Melin and Dr. Oscar Castillo for their advice, guidance, knowledge and support me during my training in the continuation of my post graduate studies.

To Cardiologist Dr. Germán Prado Aréchiga for share his knowledge and supporting as specialist in the medical area.

To all my teachers, partners and classmates who contributed my training during my course of this thesis.

To the National Council of Science and Technology (CONACYT), for the scholarship awarded under number 257008 for the completion of this thesis.

To all my Friends and colleagues, thank you for your friendship.

Contents

Chapter 1. Introduction.......... 13
Chapter 2. Background Theory.......... 15
2.1 Hybrid Intelligent Systems 13
2.2 Artificial Neural Networks.......... 15
2.3 Electrocardiograms.......... 14
2.4 Cardiac Arrhythmia.......... 15
2.5 The QRS wave 15
2.6 Waves, Intervals and Segments 16
2.7 Autoregressive Models, Shannon Entropy and Multifractal Analysis Wavelets 16
2.8 Fuzzy K-Nearest Neighbor algorithm.......... 17
2.9 Euclidean distance 19
2.10 Hamming distance.......... 19
2.11 Cosine Similarity distance.......... 20
Chapter 3. Proposed method 23
3.1 A new Modular approach for forming Classifier Models based on Type-2 Fuzzy Logic and Neural Networks 21
3.2 MIT-BIH arrhythmias database. 23
3.3 PTB Diagnostic ECG database.......... 23
3.4 Preprocessing stage.......... 23
3.5 Classification methods.......... 24
3.6 Combined by Fuzzy Inference Systems.......... 24
3.7 List of production research 24
Chapter 4. Study Cases.......... 26
4.1 Hybrid model based on neural networks, type-1 and type-2 fuzzy systems for 2-lead cardiac arrhythmia classification.......... 26
4.2 Hybrid model based on neural networks, type-1 and type-2 fuzzy systems for 12-lead cardiac arrhythmia classification 51
4.3 A new variant of Fuzzy K-Nearest Neighbor using Interval Type-2 Fuzzy Logic 77
Chapter 5. Conclusions and Future works.......... 82
References 84
Appendix 89
A.1 Function MLPPTB. 88

A.2 Function MLPPTB2. 90
A.3 Function FuzzyKNNPTB. 92
A.4 Function HelperRandomSplit. 92
A.5 Function Main. 93

List of Figures

Figure 2.1 13
Figure 3.1 24
Figure 4.1 28
Figure 4.2 28
Figure 4.3 30
Figure 4.4 30
Figure 4.5 36
Figure 4.6 43
Figure 4.7 46
Figure 4.8 48
Figure 4.9 49
Figure 4.10 53
Figure 4.11 54
Figure 4.12 55
Figure 4.13 57
Figure 4.14 57
Figure 4.15 58
Figure 4.16 59
Figure 4.17 59
Figure 4.18 64
Figure 4.19 64
Figure 4.20 67
Figure 4.21 68
Figure 4.22 71
Figure 4.23 79

List of Tables

Table 2.1 33
Table 4.1 34
Table 4.2. 34
Table 4.3. 34
Table 4.4. 35
Table 4.5 35
Table 4.6 36
Table 4.7 37
Table 4.8. 37
Table 4.9 38
Table 4.10 38
Table 4.11 39
Table 4.12. 40
Table 4.13. 40
Table 4.14. 41
Table 4.15. 42
Table 4.16. 56
Table 4.17. 58
Table 4.18. 60
Table 4.19. 61
Table 4.20. 62
Table 4.21. 63
Table 4.22. 64
Table 4.23. 65
Table 4.24. 83
Table 4.25. 83
Table 4.26. 84
Table 4.27. 85
Table 4.28. 85

List of Equations

Equation 2.1 16
Equation 2.2 17
Equation 2.3 18
Equation 2.4 18
Equation 2.5 19
Equation 2.6 19
Equation 4.1 23
Equation 4.2 24
Equation 4.3.......... 24
Equation 4.4.......... 29
Equation 4.5.......... 30
Equation 4.6 30
Equation 4.7 33
Equation 4.8 55
Equation 4.9.......... 55
Equation 4.10 55
Equation 4.11 56
Equation 4.12 56
Equation 4.13.......... 56

Chapter 1

Introduction

An electrocardiogram (ECG) illustrates the electrical activity of the heart. The ECG contains relevant information for the physicians to perform a complete medical diagnosis of the patient. The ECG is the common standard tool used for obtaining heart disease diagnosis. The physicians obtain the signals easily and non-invasively by adding electrodes to patient's body. The Holter device is a frequently used device for ECG recording. Physicians use the Holter device on the patient when it is required to monitor their ECG to find for the existence of some abnormal heartbeats in the ECG of a complete day. A person can register around 100,000 cardiac cycles in a day and represents a challenge wanting to analyze all this information to make a diagnosis.

Different methods have been currently proposed to help in the interpretation of the different morphologies or electrocardiographic patterns that represent different diseases that are diagnosed by interpreting the electrocardiograms [24, 28, 29, 42]. Such as learning vector quantization (LVQ) [1, 39], multilayer perceptron (MLP) [12, 56, 21, 19, 52, 35, 4, 5, 43, 48, 64, 65], Markov models, self-organizing maps (SOM), linear discriminant systems, Bayesian approach, support vector machines (SVM) [22, 15, 43], higher order statistics [31, 32, 33, 34], expert systems, type-1 fuzzy systems [16, 56, 61, 4, 49], type-2 fuzzy systems [37, 6, 38, 36, 60], statistical and syntactic pattern recognition, fuzzy KNN algorithm [56, 4, 23, 36], as well as hybrid systems [11, 62, 41, 40, 46, 47, 18, 44, 57, 66], Other solutions implement optimization algorithms such as Ant Colony and Particle Swarm Optimization [9, 25, 63]. Some solutions are focus on the preprocessing process such as PCA, LDA, ICA, and Discrete Wavelet Transform [32, 45], linear and nonlinear features [7, 8, 10, 17, 26, 27, 59]. Finally, with respect to the ECG signal analysis approach [50, 51, 58].

The hybridization of methods can increase the performance in a system and take advantage of the benefits offered by such techniques in solving complex problems. The interpretation of electrocardiograms is a useful task for physicians, but when it comes to reviewing more of 24 hours of information it becomes a laborious task for them. For this reason, design a computational model that helps such a task, is very useful for the timely medical diagnosis [9, 11, 14, 20].

The rest of this thesis is structured as follows: Important basic concepts as background theory are presented in Chapter 2 to understand the context in which we have applied this research. In Chapter 3, the new modular approach for forming classifier models of the proposed method and the problem description are discussed. In Chapter 4, details for the performed experiments of study cases are presented. The conclusions and future works of this research are described in Chapter 5. Finally, the references and appendix are presented.

Chapter 2.
Background Theory

In this chapter, we present a brief description of basic concepts about the methods used for the development of this thesis such as Hybrid Intelligent Systems, Artificial Neural Networks: Multilayer Perceptron and Radial Basis Function Neural Networks, Fuzzy Logic, Fuzzy KNN algorithm and general review of adapting in classification systems. All these concepts are important in order to described the proposed method of this research.

2.1 Hybrid Intelligent Systems

The hybridization of methods can increase the performance in a system and take advantage of the benefits offered by such techniques in solving complex problems [9, 11, 14, 20].

2.2 Artificial Neural Networks

An artificial neuron is a mathematical model of a biological neuron. The body of the neuron is a node that performs two functions: it computes the sum of the weighted input signals, and it applies an output function to the sum. The input signals are multiplied by weights before the sum and output functions are applied; this models the synapse. The output function is usually nonlinear; examples are: (1) converting the neuron's output to a set of discrete values (turn a light on or off); (2) limiting the range of the output values (the motor power can be between −100 and 100; (3) normalizing the range of output values (the volume of a sound is between 0 (mute) and 1 (maximum). In Figure 2.1, the ANN with one neuron and one and two inputs are presented [2].

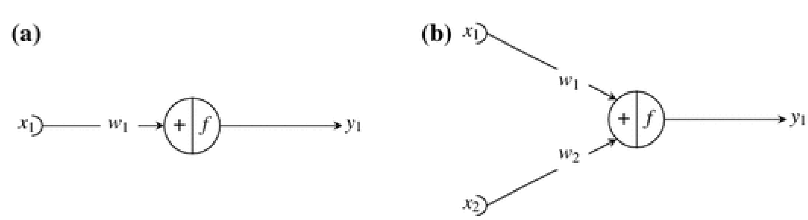


Figure 2.1 a) ANN with one neuron and one input and

b) ANN with one neuron and two input

Where *f* represent neuron output function, symbol “+” the sum of the inputs, x_i the inputs, y_i the outputs, w_i the weights for the inputs and “1” constant input of value 1 [2].

Artificial neurons are analog models, that is, the inputs, outputs, weights and functions can be floating point numbers. Here we start with an unrealistic activity that demonstrates how artificial neurons work within the familiar context of digital logic gates [2].

2.3 Electrocardiograms

An electrocardiogram (ECG) illustrates the electrical activity of the heart. The ECG contains relevant information for the physicians to perform a complete medical diagnosis of the patient. The ECG is the common standard tool used for obtaining heart disease diagnosis. The physicians obtain the signals easily and non-invasively by adding electrodes to patient’s body. The Holter device is a frequently used device for ECG recording. Physicians use the Holter device on the patient when it is required to monitor their ECG to find for the existence of some abnormal heartbeats in the ECG of a complete day. A person can register around 100,000 cardiac cycles in a day and represents a challenge wanting to analyze all this information to make a diagnosis.

2.4 Cardiac Arrhythmia

By cardiac arrhythmia, we mean any alteration in the activity of the heart rhythm in duration, amplitude or form of the rhythm and present changes from normal electrical impulses. In other words, the arrhythmias are abnormal heartbeats.

2.5 The QRS Wave

The morphology of the QRS wave is represented by the sequence of ventricular activation and the dominant vector force associated with each step. Normal ventricular depolarization can be simplified into two steps: depolarization of the septum followed by depolarization of the ventricular free walls because the Purkinje fibers are located just beneath the endocardium, activation of the ventricular walls spreads from the endocardium to the epicardium. The left aspect of the septum is the first part of the ventricles to depolarize. Normal septal depolarization occurs in a left-to-right direction. This results in the small septal R wave in the right precordial lead V1 and the small septal Q wave in V6. Depolarization of left and right ventricular free walls normally occurs simultaneously. The left ventricular free wall is composed of the area of the left ventricular wall not in contact with the interventricular septum and is not part of the apex. The right-to-left depolarization in the larger and thicker left ventricle comprises the dominating vector force. The ECG interprets this depolarization as a right-to-left force even though depolarization in the right ventricle slightly opposes this force. This dominant vector accounts for the large S wave in V1 that transitions in V3/V4 to become the large R wave in the left precordial leads (V6) [30, 14, 20].

2.6 Waves, Intervals, and Segments

The P Wave. Atria is typically activated in a right-to-left direction as the electrical impulse spreads from the sinus node in the right atrium to the left atrium. The first half of the P wave therefore represents activation of the right atrium. In normal sinus rhythm, P waves should be upright in the inferior leads (reflecting the superior to inferior direction of the impulse from sinus to AV node). The P wave in V1 is upright or biphasic. The QRS Wave represents rapid ventricular depolarization and corresponds to phase 0 of the action potential. The QRS complex widens by a delay in the intra-ventricular conduction system and ventricular hypertrophy. The T Wave corresponds to rapid repolarization (phase 3 of the action potential). Repolarization of the epicardium follows by repolarization of the endocardium. The axis of the T wave should parallel that of the QRS wave when depolarization is normal. The U Wave may by absent in the normal electrocardiogram. The source of the U wave is unclear but may represent His bundle/Purkinje fibers repolarization [30, 14, 20].

The PR Interval represents the time for an impulse to travel from the atria to the ventricles, including the time it takes to travel through the AV node and bundle of His. PR prolongation most often results from delayed conduction within the AV node. PR shortening classically occurs when an impulse travels from the atrium to the ventricle through an accessory pathway that bypasses the delay in conduction that occurs in the AV node. The QT interval represents ventricular depolarization and repolarization, corresponding to phases 0 to 3 of the action potential and ventricular systole. QT prolongation often results from delay in repolarization. The R-R interval corresponds to a complete cardiac cycle [30, 14, 20].

The time it takes for an impulse to travel from the AV node, through the His bundle, and to the ventricles is represented by the PR segment. During phase 2 or plateau phase of the action potential, the influx of intracellular calcium and subsequent release of intracellular calcium stores allow for ventricular contraction. This phase is represented by the ST segment [30, 14, 20].

2.7 Autoregressive Models, Shannon Entropy and Multifractal Analysis Wavelets

Many observed time series exhibit serial autocorrelation; that is, linear association between lagged observations. This suggest past observations might predict current observations. The autoregressive process models the conventional mean of y_t as a function of past observations, $y_{(t-1)},y_{(t-2)},\ldots,y_{(t-p)}$. An AR process that depends on p past observations is called an AR model of degree p, denoted by AR(p).

Information entropy is the average rate at which information is produced by a stochastic source of data. The measure of information entropy associated with each possible data value is the negative logarithm of the probability mass function for the value, see equation 2.1.

$$S = -\sum_{i} P_i \log P_i \tag{2.1}$$

When the data source has a lower probability value, the event carries more information than when the source data has a higher probability value. Generally, entropy refers to disorder or uncertainty, and the definition of entropy used in information theory is directly analogous to the definition used in statistical thermodynamics. The concept of information entropy was introduced by Claude Shannon.

Multifractal analysis is quasi-systematically performed using the coefficients of continuous or discrete wavelet transforms. Wavelet coefficients consist of quantities that mostly concentrate around 0, rendering the numerical computation of negative q moments extremely unstable or even theoretically infinite [17].

Recently, an alternative approach has been proposed the wavelet leader (WL). This method is theoretically backed up by a strong mathematical framework. Also, its being defined from an discrete wavelet transform (DWT) [17].

2.8 Fuzzy K-Nearest Neighbor algorithm

The fuzzy algorithm is similar to the crisp K-Nearest Neighbor. The Fuzzy K-Nearest Neighbor algorithm assigns class membership to a sample vector rather than assigning the vector to a particular class. The basis of the Fuzzy KNN algorithm is to assign membership as a function of the vector's distance from its K-Nearest Neighbors and those neighbors' memberships in the possible classes. The Fuzzy KNN algorithm is as follows:

BEGIN

Input x, of unknown classification

Set K, 1 <= K <= n.

Initialize i=1

DO UNTIL (K-nearest neighbors to x found)

Compute distances Euclidean from x to xi

IF (i <= K) **THEN**

Include xi in set of K-nearest neighbor

ELSE IF (xi closer to x than any previous nearest)

THEN

Delete the farthest of the K-nearest neighbors

Include xi in the set of K-nearest neighbors

END IF

END DO UNTIL

Initialize i=1

DO UNTIL (x assigned membership in all classes (2.2))

Compute ui(x)

END DO UNTIL

END

The equation 2.2 as a function of sample's distance from its KNN training samples:

$$u_i(x) = \frac{\sum_{j=1}^{k} u_{ij} \left(1/\left\|x - x_j\right\|^{2/(m-1)}\right)}{\sum_{j=1}^{k}\left(1/\left\|x - x_j\right\|^{2/(m-1)}\right)} \qquad (2.2)$$

The memberships of the set of training samples are assign based on the distance from their class mean. Where *m* represents the scaling parameter with values between 1 and 2. The memberships are calculated by equation 2.2 for the test sample and assign to the class with major membership.

2.9 Euclidean Distance

The Euclidean distance between points p and q is the length of the line segment connecting them (pq).

In Cartesian coordinates, if p=(p1, p2 …, pn) and q= (q1, q2,…, qn) are two points in Euclidean n-space (d) from p to q, or from q to p is given by the Pythagorean formula in equation 2.3, then:

$$d(p,q) = d(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \tag{2.3}$$

2.10 Hamming Distance

The Hamming distance is a metric expressing the distance between two objects by the number of mismatches among their pairs of variables. We present the Hamming distance in equation 2.4.

$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

where (2.4)

$$x = y \Rightarrow D = 0$$

2.11 Cosine Similarity Distance

The cosine similarity between two vectors is a measure that calculates the cosine of the angle between them. This metric is a measurement of orientation and not magnitude, in equation 2.5.

$$\vec{a} \cdot \vec{b} = \|\vec{a}\|\|\vec{b}\| \cos\theta$$
$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|\|\vec{b}\|} \tag{2.5}$$

2.11 City Block Distance

The city block distance between two points, a and b, with k dimensions is calculated, in equation 2.6 as:

$$d = \sum_{j=1}^{k} \left| a_j - b_j \right| \tag{2.6}$$

These basic concepts represents the fundamental basis as background theory to understand the rest of this thesis respect to the proposed method and as well as the context where it was applied.

Chapter 3
Proposed method

3.1 A new Modular approach for forming classifier models based on type-2 Fuzzy Logic and Neural Networks

In this work, we propose a new modular approach for forming classifier models based on type-2 Fuzzy Logic and Neural Networks. The main goal is to create a new classifier with a modular approach using type-2 Fuzzy Logic and Neural Networks to solve complex classification problems. Propose as a based on the Fuzzy K-Nearest Neighbors algorithm and Radial Basis Functions Neural Networks using type-2 Fuzzy Logic for the creation of expert modules. Create an adaptive mechanism for determining the number of expert modules and their configuration, which will be used in model formation. Integrate the results of the expert modules through a Mamdani type-2 Fuzzy Inference System, to finally obtain the global classification. Use different databases to testing our proposed model and different preprocessing methods for the databases. In the Figure 3.1, we show the Architecture of Modular Model of classification.

In order to perform this work, we have applied the development of this thesis to arrhythmias classification problems through of study cases presented in details in the sections of chapter 4.

The first case of study is related of 2-lead cardiac arrhythmia classification using two basic module units, where each basic module unit represents a lead and is composite by three different classifiers: Fuzzy KNN algorithm, MLP-GDM, MLP SCG, the final decision of the basic module unit is determined by type-1 and type-2 Fuzzy Inference System. Each basic module unit is an expert for specific lead and all classes included in the MIT-BIH database. Finally, the global decision of the hybrid intelligent system considering both basic module units is combined by a Fuzzy Inference System [53, 54].

In this thesis, the main goal is to improve the global classification rate for arrhythmia classification, and we propose an approach using different computational intelligence techniques to form a hybrid model as a classification method for 2-lead cardiac arrhythmias such as artificial neural networks and fuzzy logic. The hybrid model is composed by two basic module units. To perform the classification for each signal lead the basic module unit is used. Each basic module unit is composed by three classifiers based on the fuzzy KNN algorithm and MLP-GDM and MLP-SCG. The output results of the classifiers in each basic module unit are combined with a type-1 FIS. Finally, we used type-1 FIS to combine the outputs of both basic module units achieving an improvement in the global classification rate of the proposed hybrid model. In addition, we have performed experiments using IT2FIS and we found out better results than type-1 FIS. This hybrid model can be extended to use 12 lead and other ECG databases for a complete medical diagnosis [9, 11, 14, 20].

The hybridization of methods can increase the performance in a system and take advantage of the benefits offered by such techniques in solving complex problems. The interpretation of electrocardiograms is a useful task for physicians, but when it comes to reviewing more than 24 hours of information, it becomes a laborious task for them. For this reason, design a computational model that helps such a task, is very useful for the timely medical diagnosis [9, 11, 14, 20].

In the second case of study, we present a 12-lead cardiac arrhythmia classification solution. We used PTB Diagnosis ECG database with 9 classes included. In this hybrid model is composite by 12 expert modules, where each one is expert in a specific lead*: i, ii, iii, avf, avr, avl, v1, v2, v3, v4, v5 and v6*. Finally, the global decision is determined by a type-1 and type-2 Fuzzy Inference System. The Fuzzy Inference Systems are based on ECG regions knowledge expert as medical criteria [30, 55].

3.2 MIT-BIH arrhythmias database

There are 48 electrocardiogram records of half-hour excerpts with two channels in the MIT-BIH arrhythmia database of 47 patients. The ECG records were obtained with a Holter device in Beth Israel Hospital. The MIT-BIH arrhythmia database includes annotations by cardiologists.

3.3 PTB Diagnostic ECG database

The PTB Diagnostic ECG Database contains 549 ECG records from 290 subjects, the subjects aged 17 to 87 years old. One to five ECG records represent each subject. Each ECG record includes 15 measured signals, the conventional 12 leads: i, ii, iii, avr, avl, avf, v1, v2, v3, v4, v5, v6, and three Frank lead such as vx, vy and vz. The diagnosis classes included in the database are Myocardial Infarction, Cardiomyopathy, Bundle Branch Block, Dysrhythmia, Myocardial Hypertrophy, Valvular Heart Disease, Myocarditis, Miscellaneous, and Healthy Controls.

3.4 Preprocessing Stage

In this section, we describe in detail to the preprocessing stage for both study cases. In the first case of study, we have taken the samples of the heartbeats of the electrocardiograms of the MIT-BIH arrhythmia database, the segmentation was performed manually and the transformation process consists in rearranging the voltage values in the heartbeat signal, selecting only the 70 larger and 70 smaller voltage values, because the size of each heartbeat is different. This means that each heartbeat is represented by a vector with 140 voltage values [12, 54, 55, 56]. This transformation process simplifies the signal of the heartbeat to be classified of the hybrid model. The main goals of this work are the hybrid model as well as improving the global classification rate. In Figure 2.3, the architecture of modular model of classification are presented.

For the second case of study, the PTB Diagnosis ECG, we applied a feature extraction process in order to reduce or simply the information of the selected samples of the electrocardiograms as well as for the classifiers captures the differences between the classes to improve the classification rate. We created a

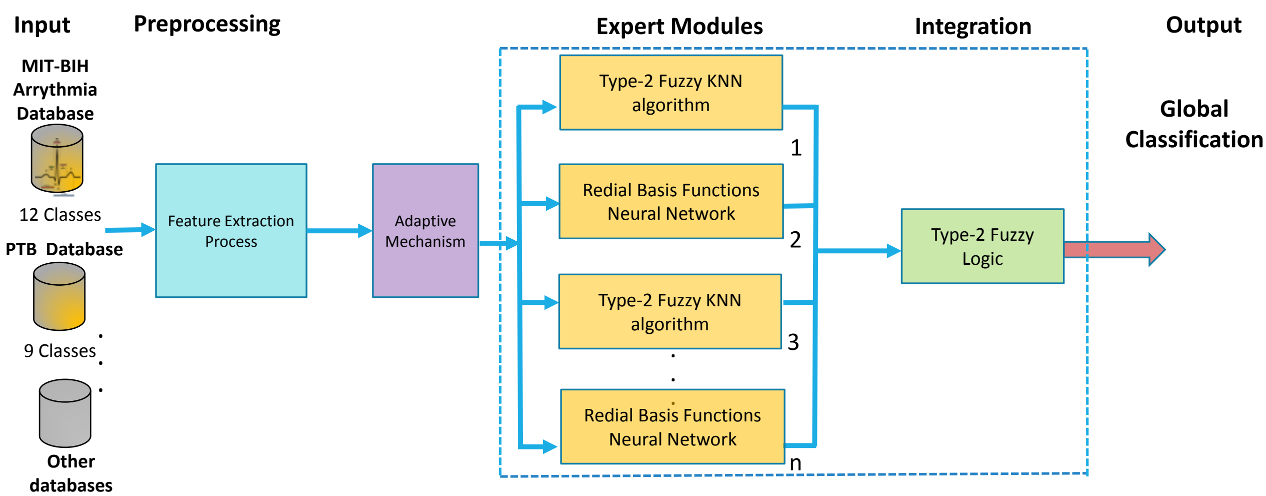


Figure 3.1 Architecture of Modular Model of Classification

set of vectors that represent a complete signal or lead in the electrocardiogram; in other words, each electrocardiogram has its twelve feature vectors to be learning for the expert modules in the proposed hybrid intelligent system. The feature vectors are built with autoregressive models, Shannon entropies, and wavelets.

3.5 Adaptive Mechanism

The adaptation feature represents the ability of the systems to learn about new situations to adjust itself in a flexible way to new environment conditions. The adaptive mechanism in the proposed work is based on adaptation due to structural update or dynamic changing structure of the architecture of Modular Model of Classification. The main goal of the adaptive mechanism is to determine the number of expert modules and their configuration.

3.6 Classification Methods

Three classifier models are used: fuzzy KNN algorithm, MLP-GDM, and MLP-SCG. Finally, the output results of the classifiers are combined in two ways, firstly with a type-1 FIS, as well as with an IT2FIS, and we want to show a comparison of the results with type-1 and type-2 fuzzy logic. Figure 3.2 illustrates the basic module unit for cardiac arrhythmia classification.

Every classifier receives the same set of vectors that corresponds to the selected heartbeats. Each classifier produces an output that represents the specific classification of the heartbeat.

3.7 Combined by a Fuzzy Inference System

In the first case of study, the outputs of the classifiers form the integration matrix. The integration matrix is the input of both above-mentioned fuzzy systems. Finally, the fuzzy systems (type-1 FIS and IT2FIS) produce the final classification of the basic module unit for a ECG's signal electrode. Finally the global classification is obtain using a Fuzzy Inference System [53, 54].

In the second case of study, the outputs of the expert modules form the integration matrix, and represent the input for the global classification obtained by the fuzzy systems (type-1 and type-2) FIS [55].

3.8 List of Production Research

- Patricia Melin, Eduardo Ramírez, German Prado-Arechiga: **Toward improving the Fuzzy KNN algorithm based on Takagi-Sugeno Fuzzy Inference System.** NAFIPS 2020.

- Eduardo Ramírez, Patricia Melin, German Prado-Arechiga: **Hybrid Model Based on Neural Networks and Fuzzy Logic for 2-Lead Cardiac Arrhythmia Classification.** Hybrid Intelligent Systems in Control, Pattern Recognition and Medicine 2020: 193-217.

- Eduardo Ramirez, Patricia Melin, German Prado-Arechiga: **A Modular Neural Network Approach for Cardiac Arrhythmia Classification.** Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms 2020: 211-223.

- Eduardo Ramirez, Patricia Melin, German Prado-Arechiga: **Hybrid model based on neural networks, type-1 and type-2 fuzzy systems for 2-lead cardiac arrhythmia classification.** Expert Syst. Appl. 126: 295-307 (2019).

- Patricia Melin, Eduardo Ramirez, German Prado-Arechiga: **A new variant of Fuzzy K-Nearest Neighbor using Interval Type-2 Fuzzy Logic**. FUZZ-IEEE 2018: 1-7.

- Patricia Melin, Eduardo Ramirez, German Prado-Arechiga. **Cardiac arrhythmia classification usign computational intelligence: neural networks and fuzzy logic techniques.** European Heart Journal 2017, OXFORD academic, 38, P6388.

Chapter 4
Study Cases

In this chapter, we present the different study cases used to form the hybrid models as a new classifier with a modular approach using Type-2 Fuzzy Logic and Neural Networks to solve complex classification problems developed in this thesis work. In Section 3 the general methodology of the proposal is shown, this methodology was applied to the following two study cases: 2-lead cardiac arrhythmia classification and 12-lead cardiac arrhythmia classification problems.

4.1 2-lead cardiac arrhythmia classification

There are 48 electrocardiogram records of half-hour excerpts with two channels or 2-lead in the MIT-BIH arrhythmia database of 47 patients. The ECG records were obtained with a Holter device in Beth Israel Hospital. The MIT-BIH arrhythmia database includes annotations by cardiologists.

The samples of the classes are normal, and the different arrhythmias such as right bundle branch block, left bundle branch block, premature ventricular contraction, fusion paced and normal, atrial premature, aberrated atrial premature, fusion of ventricular and normal, ventricular escape, paced named N, LBB, RBB, PVC, FPN, AP, AAP, FVN, VE and P respectively. In Figure 4.1, we present some examples of normal beat and arrhythmias of the MIT-BIH arrhythmia database.

4.1.1 Hybrid model based on neural networks, type-1 and type-2 fuzzy systems for 2-lead cardiac arrhythmia classification

This case of study describes an approach using computational intelligence methods to form a hybrid model as a classification method for 2-lead cardiac arrhythmias. The hybridization of methods can increase the performance in a system and take advantage of the benefits offered by such techniques in solving complex problems. The interpretation of electrocardiograms is a useful task for physicians, but when it comes to reviewing more than 24 hours of information, it becomes a laborious task for them [9, 11, 14, 20]. For this reason, the design a computational model that helps in such a task is very useful for the timely medical diagnosis. The hybrid model is built using artificial neural networks and fuzzy logic. Training and testing of the hybrid model was with the Massachusetts Institute of Technology and Beth Israel Hospital (MIT-BIH) arrhythmia database. The heartbeats are preprocessed to improve results of classification. Ten different classes of normal and arrhythmia signals for building the hybrid model are considered. We used two electrode signals or leads included in the MIT-BIH arrhythmia database, MLII and V1, V2, or V3 as second electrode signal. The hybrid model is composed by two basic module units, as described below. A basic module unit to perform the classification for each signal lead is used. Each basic module unit is composed of three different classifiers based on the following models: fuzzy KNN algorithm, multilayer perceptron with gradient descent and momentum (MLP-GDM), and multilayer perceptron with scaled conjugate gradient backpropagation (MLP-SCG). The outputs from the classifiers are combined using a fuzzy system for integration of results. We designed two fuzzy systems, Mamdani type-1 fuzzy system (type-1 FIS) and an interval type-2 fuzzy system (IT2FIS). The reason is to perform a comparison between type-1 FIS and IT2FIS in the hybrid model. We have obtained best results in the classification rate using IT2FIS instead of type-1 FIS in the basic units. Finally, a type-1 FIS is used to determine the global classification for the 2 basic units in a hybrid model. We obtained a good classification rate in each basic module unit, 92.90% and 92.70% of classification rate for basic modules unit 1 and unit 2 respectively. Finally, we obtained a 93.80% when used type-1 FIS and 94.20% of classification rate used IT2FIS combining both basic module units. In the results presented, we improve the global classification in the proposed hybrid model combining neural networks and fuzzy logic used both signal lead included in MIT-BIH arrhythmia database. The proposed hybrid model maybe extended to use multi-lead arrhythmia classification using other databases that contain 12 leads to be able to make a complete medical diagnosis.

The hybrid model is composed of two basic module units, where each module represents one electrode signal or lead of the electrocardiogram signal included in the MIT-BIH arrhythmia database. In Figure 4.2 is presented the proposed hybrid model. This hybrid model can be extended to 12-lead cardiac arrhythmia

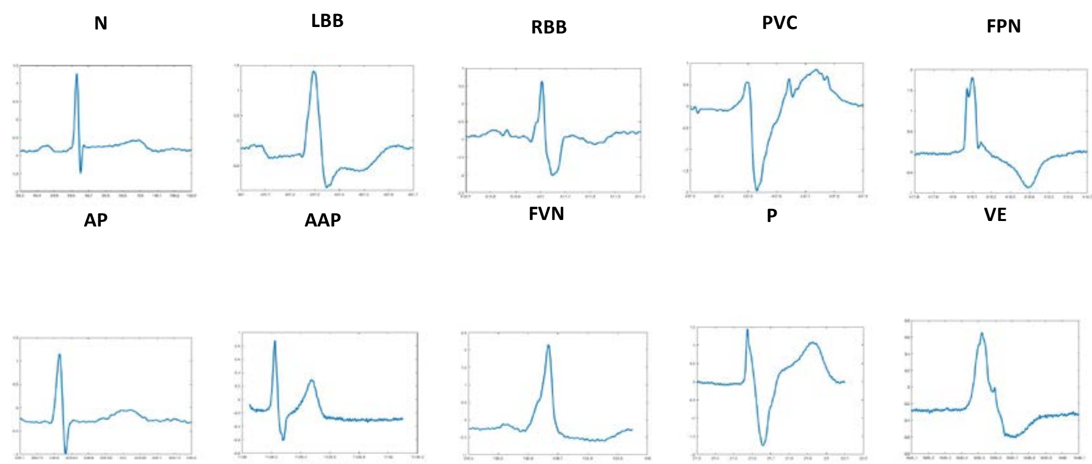


Figure 4.1 Normal beat and arrhythmias of the MIT-BIH Arrhythmia Database

classification problem. We used an IT2FIS to perform the global arrhythmia

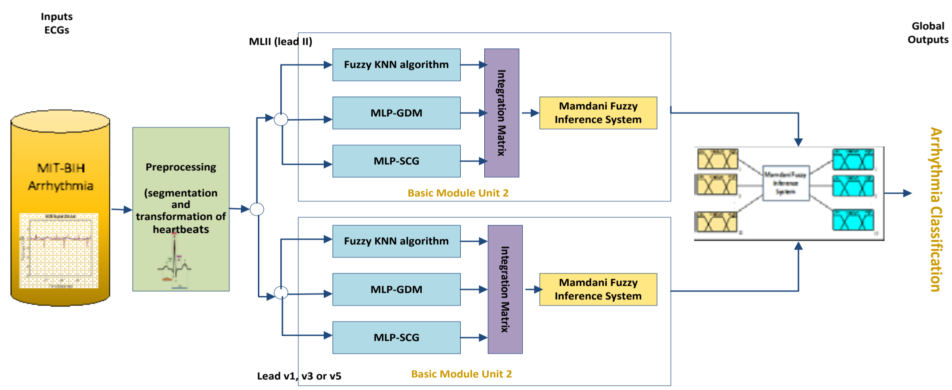


Figure 4.2 Hybrid model for 2-lead cardiac arrhythmia classification

classification of the hybrid model using the two basic module units.

4.1.2 Preprocessing stage

In this section, we describe in detail to the preprocessing stage. We have taken the samples of the heartbeats of the electrocardiograms of the MIT-BIH arrhythmia database, the segmentation was performed manually and the transformation process consists in rearranging the voltage values in the heartbeat signal, selecting only the 70 larger and 70 smaller voltage values, because the size of each heartbeat is different. This means that each heartbeat is represented by a vector with 140 voltage values [12, 56, 4]. This transformation process simplifies the signal of the heartbeat to be classified of the hybrid model. The main goals of this work are the hybrid model as well as improving the global classification rate.

4.1.3 Basic module unit

A basic module unit is used to solve the classification for each signal lead. Each module unit is composed of three different classifiers based on the following models: fuzzy KNN algorithm [23], MLP-GDM [2], and MLP-SCG [2]. The output results from the three classifiers are combined using a fuzzy system. We designed two fuzzy systems for this task, Mamdani type-1 FIS and an IT2FIS. The hybrid model used two basic module units. The basic module unit 1 learns the patterns of the classes for the specific MLII signal lead included in the MIT-BIH arrhythmia database. The basic module unit 2 learns the patterns of the classes for the second lead with the V1, V2 or V3 electrode signal.

4.1.4 Classifiers

Three classifier models are used: fuzzy KNN algorithm, MLP-GDM, and MLP-SCG. Finally, the output results of the classifiers are combined in two ways, firstly with a type-1 FIS, as well as with an IT2FIS, and we want to show a comparison of the results with type-1 and type-2 fuzzy logic. Figure 4.3 illustrates the basic module unit for cardiac arrhythmia classification.

Every classifier receives the same set of vectors that corresponds to the selected heartbeats. Each classifier produces an output that represents the specific classification of the heartbeat. The outputs of the classifiers form the integration matrix. The integration matrix is the input of both above-mentioned fuzzy systems. Finally, the fuzzy systems (type-1 FIS and IT2FIS) produce the final classification of the basic module unit for an ECG's signal electrode.

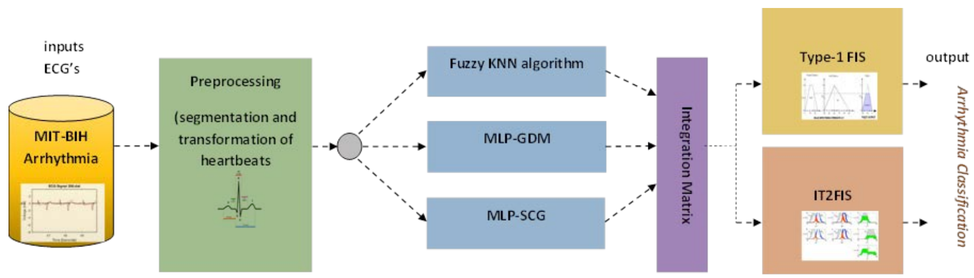


Figure 4.3 Basic module unit for cardiac arrhythmia classification

4.1.5 Type-1 Fuzzy System for basic module unit

We used a type-1 FIS to combine the outputs of the classifiers in the basic module unit and details of the results for the experiments are presented in Section 3. The specifications for the type-1 FIS are: Mamdani with 30 inputs; 10 inputs belonging to the first classifier fuzzy KNN algorithm, the next 10 inputs correspond for the second classifier MLP-GDM, the last 10 inputs corresponding for the third classifier MLP-SCG, the inputs and outputs are trapezoidal functions with linguistic variables Low_Classification, Medium_Classification, and High_Classification. The defuzzification method is by centroid. We have 260 rules to combine the outputs of the classifiers regarding the membership of the fuzzy KNN algorithm and the activation of the both MLPs, in the next part of this section we present some examples of the fuzzy rules. We show the type-1 FIS in Figure 4.4.

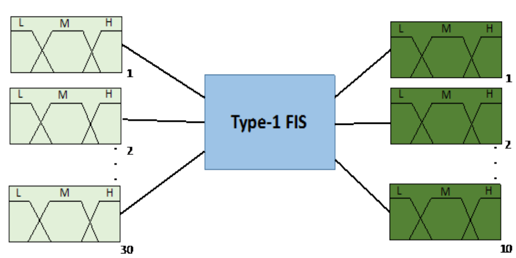


Figure 4.4 Mamdani type-1 Fuzzy System

4.1.5.1 Input and output variables

The membership function parameters for the inputs and outputs are expressed in equation 4.1, 4.2, 4.3 as:

$$\mu_{\text{Low}}(x)=\begin{cases} 0, & x\leq -0.0529 \\ \frac{x+0.0529}{0.05033}, & -0.0529\leq x\leq -0.00257 \\ 1, & -0.00257\leq x\leq 0.2701 \\ \frac{0.322-x}{0.1149}, & 0.2071\leq x\leq 0.322 \\ 0, & 0.322\leq x \end{cases} \tag{4.1}$$

$$\mu_{\text{Medium}}(x)=\begin{cases} 0, & x\leq 0.323 \\ \frac{x-0.323}{0.134}, & 0.323\leq x\leq 0.457 \\ 1, & 0.457\leq x\leq 0.552 \\ \frac{0.705-x}{0.153}, & 0.552\leq x\leq 0.705 \\ 0, & 0.705\leq x \end{cases} \tag{4.2}$$

$$\mu_{\text{High}}(x)=\begin{cases} 0, & x\leq 0.702 \\ \frac{x-0.702}{0.103}, & 0.702\leq x\leq 0.805 \\ 1, & 0.805\leq x\leq 1.004 \\ \frac{0.705-x}{0.046}, & 1.004\leq x\leq 1.05 \\ 0, & 1.05\leq x \end{cases} \tag{4.3}$$

The values for the membership functions were defined considering the three possible states such as Low, Medium and High, assigned and adjusted in a manual way. We also carried out experiments using other types of membership functions such as triangular, Gaussian, bell, and trapezoid. We obtained better results using trapezoidal function to combine the output results of the classifiers mentioned.

4.1.5.1 Fuzzy rules

There are 260 fuzzy if-then rules in the type-1 fuzzy inference system, 26 rules represent the specific basic rules for each class. The basic rules are the following:

1. IF (Normal_KNN IS HIGH_CLASSIFICATION) AND (Normal_MLP1 IS HIGH_CLASSIFICATION) AND (Normal_MLP2 IS HIGH_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

2. IF (Normal_KNN IS HIGH_CLASSIFICATION) AND (Normal_MLP1 IS HIGH_CLASSIFICATION) AND (Normal_MLP2 IS LOW_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

3. IF (Normal_KNN IS HIGH_CLASSIFICATION) AND (Normal_MLP1 IS HIGH_CLASSIFICATION) AND (Normal_MLP2 IS MEDIUM_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

4. IF (Normal_KNN IS HIGH_CLASSIFICATION) AND (Normal_MLP1 IS LOW_CLASSIFICATION) AND (Normal_MLP2 IS HIGH_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

5. IF (Normal_KNN IS HIGH_CLASSIFICATION) AND (Normal_MLP1 IS MEDIUM_CLASSIFICATION) AND (Normal_MLP2 IS HIGH_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

6. IF (Normal_KNN IS LOW_CLASSIFICATION) AND (Normal_MLP1 IS HIGH_CLASSIFICATION) AND (Normal_MLP2 IS HIGH_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

7. IF (Normal_KNN IS MEDIUM_CLASSIFICATION) AND (Normal_MLP1 IS HIGH_CLASSIFICATION) AND (Normal_MLP2 IS HIGH_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

8. IF (Normal_KNN IS HIGH_CLASSIFICATION) AND (Normal_MLP1 IS MEDIUM_CLASSIFICATION) AND (Normal_MLP2 IS MEDIUM_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

9. IF (Normal_KNN IS HIGH_CLASSIC) AND (Normal_MLP1 IS MEDIUM_CLASSIFICATION) AND (Normal_MLP2 IS LOW_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

10. IF (Normal_KNN IS HIGH_CLASSIFICATION) AND (Normal_MLP1 IS LOW_CLASSIFICATION) AND (Normal_MLP2 IS LOW_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

11. IF (Normal_KNN IS HIGH_CLASSIFICATION) AND (Normal_MLP1 IS LOW_CLASSIFICATION) AND (Normal_MLP2 IS MEDIUM_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

12. IF (Normal_KNN IS MEDIUM_CLASSIFICATION) AND (Normal_MLP1 IS HIGH_CLASSIFICATION) AND (Normal_MLP2 IS MEDIUM_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

13. IF (Normal_KNN IS MEDIUM_CLASSIFICATION) AND (Normal_MLP1 IS MEDIUM_CLASSIFICATION) AND (Normal_MLP2 IS HIGH_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

14. IF (Normal_KNN IS MEDIUM_CLASSIFICATION) AND (Normal_MLP1 IS HIGH_CLASSIFICATION) AND (Normal_MLP2 IS LOW_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

15. IF (Normal_KNN IS MEDIUM_CLASSIFICATION) AND (Normal_MLP1 IS LOW_CLASSIFICATION) AND (Normal_MLP2 IS HIGH_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

16. IF (Normal_KNN IS LOW_CLASSIFICATION) AND (Normal_MLP1 IS MEDIUM_CLASSIFICATION) AND (Normal_MLP2 IS HIGH_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

17. IF (Normal_KNN IS MEDIUM_CLASSIFICATION) AND (Normal_MLP1 IS MEDIUM_CLASSIFICATION) AND (Normal_MLP2 IS MEDIUM_CLASSIFICATION) THEN (Normal IS MEDIUM_CLASSIFICATION).

18. IF (Normal_KNN IS MEDIUM_CLASSIFICATION) AND (Normal_MLP1 IS MEDIUM_CLASSIFICATION) AND (Normal_MLP2 IS LOW_CLASSIFICATION) THEN (Normal IS MEDIUM_CLASSIFICATION).

19. IF (Normal_KNN IS MEDIUM_CLASSIFICATION) AND (Normal_MLP1 IS LOW_CLASSIFICATION) AND (Normal_MLP2 IS MEDIUM_CLASSIFICATION) THEN (Normal IS MEDIUM_CLASSIFICATION).

20. IF (Normal_KNN IS LOW_CLASSIFICATION) AND (Normal_MLP1 IS MEDIUM_CLASSIFICATION) AND (Normal_MLP2 IS MEDIUM_CLASSIFICATION) THEN (Normal IS MEDIUM_CLASSIFICATION).

21. IF (Normal_KNN IS LOW_CLASSIFICATION) AND (Normal_MLP1 IS LOW_CLASSIFICATION) AND (Normal_MLP2 IS MEDIUM_CLASSIFCATION) THEN (Normal IS MEDIUM_CLASSIFICATION).

22. IF (Normal_KNN IS MEDIUM_CLASSIFICATION) AND (Normal_MLP1 IS LOW_CLASSIFICATION) AND (Normal_MLP2 IS LOW_CLASSIFICATION) THEN (Normal IS MEDIUM_CLASSIFICATION).

23. IF (Normal_KNN IS LOW_CLASSIFICATION) AND (Normal_MLP1 IS MEDIUM_CLASSIFICATION) AND (Normal_MLP2 IS LOW_CLASSIFICATION) THEN (Normal IS MEDIUM_CLASSIFICATION).

24. IF (Normal_KNN IS LOW_CLASSIFICATION) AND (Normal_MLP1 IS LOW_CLASSIFICATION) AND (Normal_MLP2 IS LOW_CLASSIFICATION) THEN (Normal IS LOW_CLASSIFICATION).

25. IF (Normal_KNN IS LOW_CLASSIFICATION) AND (Normal_MLP1 IS LOW_CLASSIFICATION) AND (Normal_MLP2 IS HIGH_CLASSIFICATION) THEN (Normal IS HIGH_CLASSIFICATION).

26. IF (Normal_KNN IS LOW_CLASSIFICATION) AND (Normal_MLP1 IS LOW_CLASSIFICATION) AND (Normal_MLP2 IS MEDIUM_CLASSIFICATION) THEN (Normal IS MEDIUM_CLASSIFICATION).

In the fuzzy if-then rules we take into consideration for a specific class the outputs of the classifiers regarding the membership degrees from fuzzy KNN algorithm as the first classifier and the activations of both neural networks (MLP-GDM and MLP-SCG) as second and third classifier respectively. We built the fuzzy rules in a manual way using as criteria the different expertise or knowledge of the classifiers by different experts with respect to the classification. We summarize the criteria used to develop the fuzzy rules as follows: first, if the three classifiers obtain the same opinion with respect to classify a specific sample then we keep the same fuzzy value; second, if two classifiers obtain the same linguistic value with respect to classify a specific sample and the linguistic value of the different opinion classifier represent major value, then the consequent obtain the linguistic value of the different opinion classifier; third, If two classifiers obtain the same linguistic value with respect to classify a specific sample and the linguistic value of the different opinion classifier represent lower linguistic value, then the consequent obtains the linguistic value of the two classifiers with the same opinion; if the three classifiers differ in opinion then the consequent obtains the higher linguistic value. We apply these three criteria to form the all fuzzy rules for the fuzzy system to combine all the output results of the classifiers in the basic module unit for the hybrid model.

4.1.6 Interval Type-2 Fuzzy System for basic module unit

In this work, we also used a type-2 fuzzy system to compare the global classification rate with respect to type-1 fuzzy logic. We propose using an Interval Type-2 Fuzzy (IT2FIS). We used the similar specifications than type-1 FIS above mentioned. The specifications of the IT2FIS are presented: Mamdani, 30 inputs, 10 outputs. The inputs and outputs use trapezoidal functions for the fuzzy variables.

The fuzzy rules are a total of 260, and the centroid as a defuzzification method. Figure 4.5 illustrates the IT2FIS structure.

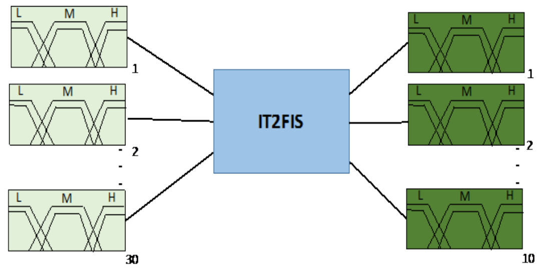


Figure 4.5 Interval Type-2 Fuzzy System

4.1.6.1 Input and output variables

The parameters of the membership functions for the inputs and outputs can be expressed as:

•Low_Classification= [-0.4025; 0.4025; 0.08335; 0.08335; 0.07798; 0.8].

•Medium_Classification= [0.09747; 0.9025; 0.08333; 0.08335; 0.07796; 0.8].

•High_Classification= [0.5975; 1.402; 0.08335; 0.0835; 0.07798; 0.8].

4.1.6.2 Fuzzy rules

We used the same above-mentioned set of fuzzy rules of the type-1 FIS to evaluate the proposed IT2FIS. We built the fuzzy rules in a manual way using the same criteria.

4.1.7 Type-1 Fuzzy System and IT2FIS for Hybrid model

In the hybrid model, a type-1 FIS is used to combine the results of two basic module units obtaining finally a global rate for cardiac arrhythmia classification and solving a 2-lead arrhythmia classification problem. The specifications of the type-1 FIS are mentioned below.

4.1.7.1 Input and output variables

The parameters of membership functions for inputs and outputs can be expressed in equation 4.4, 4.5, 4.6 as:

$$\mu_{Low}(x)=\begin{cases} 0, & x\leq -0.36 \\ \frac{x+0.36}{0.32}, & -0.36\leq x\leq -0.04 \\ 1, & -0.04\leq x\leq 0.04 \\ \frac{0.04-x}{0.32}, & 0.04\leq x\leq 0.36 \\ 0, & 0.36\leq x \end{cases} \tag{4.4}$$

$$\mu_{Medium}(x)=\begin{cases} 0, & x\leq 0.14 \\ \frac{x-0.14}{0.32}, & 0.14\leq x\leq 0.46 \\ 1, & 0.46\leq x\leq 0.54 \\ \frac{0.86-x}{0.32}, & 0.54\leq x\leq 0.86 \\ 0, & 0.86\leq x \end{cases} \tag{4.5}$$

$$\mu_{High}(x)=\begin{cases} 0, & x\leq 0.64 \\ \frac{x-0.64}{0.32}, & 0.64\leq x\leq 0.96 \\ 1, & 0.96\leq x\leq 1.04 \\ \frac{1.36-x}{0.32}, & 1.04\leq x\leq 1.36 \\ 0, & 1.36\leq x \end{cases} \tag{4.6}$$

The values for the membership functions were defined considering the three possible states such as Low, Medium and High, assigned and adjusted in a manual way.

4.1.7.2 Fuzzy rules

In the hybrid model a type-1 FIS and IT2FIS are used to combine the outputs of two leads for cardiac arrhythmia classification. In other words, to combine the two basic module units. We built the fuzzy if-then rules in a manual way using the same criteria mentioned above. We have 90 fuzzy rules in total; 9 fuzzy rules are the basic rules for each class to represent the knowledge in the fuzzy system. The basic fuzzy rules are listed as follows:

1. IF (NormalUnit1 is LOW_CLASSFICATION) AND (NormalUnit2 is LOW_CLASSIFICATION) THEN (Normal is LOW_CLASSIFICATION).

2. IF (NormalUnit1 is MEDIUM_CLASSIFICATION) AND (NormalUnit2 is MEDIUM_CLASSIFICATION) THEN (Normal is MEDIUM_CLASSIFICATION).

3. IF (NormalUnit1 is HIGH_CLASSIFICATION) AND (NormalUnit2 is HIGH_CLASSIFICATION) THEN (Normal is HIGH_CLASSIFICATION).

4. IF (NormalUnit1 is LOW_CLASSIFICATION) AND (NormalUnit2 is MEDIUM_CLASSIFICATION) THEN (Normal is MEDIUM_CLASSIFICATION).

5. IF (NormalUnit1 is LOW_CLASSIFICATION) AND (NormalUnit2 is HIGH_CLASSIFICATION) THEN (Normal is HIGH_CLASSIFICATION).

6. IF (NormalUnit1 is MEDIUM_CLASSIFICATION) AND (NormalUnit2 is LOW_CLASSIFICATION) THEN (Normal is MEDIUM_CLASSIFICATION).

7. IF (NormalUnit1 is HIGH_CLASSIFICATION) AND (NormalUnit2 is LOW_CLASSIFICATION) THEN (Normal is HIGH_CLASSIFICATION).

8. IF (NormalUnit1 is MEDIUM_CLASSIFICATION) AND (NormalUnit2 is HIGH_CLASSIFICATION) THEN (Normal is HIGH_CLASSIFICATION).

9. IF (NormalUnit1 is HIGH_CLASSIFICATION) AND (NormalUnit2 is MEDIUM_CLASSIFICATION) THEN (Normal is HIGH_CLASSIFICATION).

4.1.8 Experiments

We used the 2 lead included in MIT-BIH arrhythmia database, the MLII signal and v1, v3 or v5 signal. The records used for the classes were: N 113, 115, and 122, LBBB 109, 111, and 214, RBBB 118, 124, and 212, PVC 106, 200 and

208 FPN 217, APB 209 and 222, AAPB 201, 202, and 210, FVNB 208 and 214, VEB 207, and PB 217.

We used the Matlab R2016a programming language, Fuzzy Logic Toolbox and Interval Type-2 Fuzzy Logic Toolbox to do experiments for the proposed hybrid model. The experiments were performed using a Mac Pro 6.1, 6-Core Intel Xeon E5 3.5 Ghz and 16 GB RAM.

We have performed experiments using 10-fold cross validation to evaluate the hybrid model, that consist on partitioning the original set of samples of heartbeats into training set to train each classifier, and test set, herein for validation. There are 1000 samples of heartbeats in the dataset. Recall that each class represents 10% of the data, 100 samples per each class. We apply 10-fold cross validation. We created 10 disjoint random subsamples. In the first iteration, one subsample for validation and the other nine for training of the hybrid model are used. In the second iteration, a second subsample to validate and the rest for training and we repeated this process to complete 10 iterations of the 10-fold cross validation process. In each basic module unit, we have trained the three classifiers, selected different parameters and tested. For the first classifier, Fuzzy KNN algorithm, we used k= 1, 3, 4, and 5. For the second and third classifiers, MLP with gradient descent and momentum as training algorithm (MLP-GDM), we used 140 input neurons, 50, 100 and 150 hidden neurons, log sigmoid transfer function, 10000 epochs, learning rate 0.3 and momentum 0.5, 10 output neurons. For the third classifier, MLP with scale conjugate gradient backpropagation as training algorithm (MLP-SCG), 50, 100 and 150 hidden neurons, log sigmoid transfer function, 10000 epochs, learning rate 0.001, 10 input neurons. In this case, the output neurons represent the classes of arrhythmias. We selected the best results for each mentioned architecture to represent the classifiers. The outputs of the three classifiers form the integration matrix, herein the input for the type-1 FIS to combine the output results of the classifiers. Also we used IT2FIS to combine the output results of the classifiers. The outputs of the classifiers are evaluated for type-1 FIS and IT2FIS. In other words, both fuzzy systems, type-1 and IT2FIS receive the same data vector of the integration matrix in their inputs. Around 7 minutes it took the training of the IT2FIS from Unit 1 applied cross validation. We do the same process for both basic module units. Finally, we used other type-1 FIS to combine the output results for the two basic module units in the hybrid model.

The equation 4.7 for the calculated classification rate coefficient is as follows:

$$Classification\,rate = \left(\frac{TP + TN}{TP + TN + FN + FP}\right) * 100 \tag{4.7}$$

Where TP is true positive prediction or correct positive prediction,

TN is true negative prediction or correct negative prediction,

FN is false negative prediction or incorrect negative prediction and

FP is false positive prediction or incorrect positive prediction.

4.1.9 Results

In this part, we present the results for the both basic module units separately, and the global results for the hybrid model proposed. In the last part of this section, we are present a statistical analysis to compare the results of the fuzzy systems.

4.1.9.1 Results for the basic module unit 1

Firstly, we are present the results for the first classifier into basic module unit 1 we found out that the best result is with k=4 achieving a 90.3% of classification rate, as can be seen in confusion matrix in Table 4.1. For MLP-GDM, second classifier, with 150 hidden neurons, we obtained 83.9% of classification rate, see Table 4.2, and for the last classifier MLP-SCG we obtained best result used 50 hidden neurons with a 92.80% of classification rate, see details in Table 4.3. The results for the type-1 FIS and IT2FIS are shown in Table 4.4 and Table 4.5 respectively. The results for the basic module unit 1 are shown an increase in the final classification rate of the basic module unit 1 of 92.90% used type-1 FIS and 93.6% used IT2FIS.

In the last part of this section the results for the basic module unit 2 are presented.

Table 4.1 Results for cross validation of fuzzy KNN algorithm, k=4, first classifier in basic module unit 1

Class	N	LBB	RBB	PVC	FPN	AP	AAP	FVN	VE	P
N	97	0	0	0	0	1	0	2	0	0
LBB	0	92	1	0	0	1	2	2	2	0
RBB	0	2	96	0	1	0	1	0	0	0
PVC	0	3	0	83	3	0	1	2	2	6
FPN	1	0	2	2	85	0	1	6	0	3
AP	0	1	0	0	0	92	7	0	0	0
AAP	0	4	0	0	0	10	80	1	5	0
FVN	3	4	1	2	2	1	1	86	0	0
VE	0	0	0	0	0	0	5	0	95	0
PB	0	0	0	1	2	0	0	0	0	97

Classification rate= 90.3%

Table 4.2 Results for cross validation of MLP-GDM, 150 hidden neurons, second classifier in basic module unit 1

Class	N	LBB	RBB	PVC	FPN	AP	AAP	FVN	VE	P
N	97	0	0	0	0	0	0	3	0	0
LBB	1	69	0	0	0	0	22	4	4	0
RBB	4	3	73	0	18	1	0	0	1	0
PVC	0	3	0	80	4	2	1	0	3	7
FPN	0	1	3	2	90	1	0	1	0	2
AP	0	0	0	0	0	99	1	0	0	0
AAP	0	5	3	0	1	18	67	0	6	0
FVN	2	20	0	3	0	2	0	73	0	0
VE	0	0	0	0	0	0	5	0	95	0
P	0	0	1	1	2	0	0	0	0	96

Classification rate= 83.9%

Table 4.3 Results for cross validation of MLP-SCG 50 hidden neurons, third classifier in basic module unit 1

Class	N	LBB	RBB	PVC	FPN	AP	AAP	FVN	VE	P
N	99	0	0	0	0	0	0	1	0	0
LBB	3	87	3	0	1	0	1	3	2	0
RBB	0	1	97	0	0	0	0	1	1	0
PVC	0	4	0	86	2	0	3	2	1	2
FPN	0	1	1	1	93	0	1	0	0	3
AP	0	0	0	0	0	97	3	0	0	0
AAP	1	2	2	0	0	5	88	0	2	0
FVN	1	1	0	2	1	0	1	94	0	0
VE	3	0	0	0	0	0	7	1	89	0
P	0	0	0	0	2	0	0	0	0	98

Classification rate= 92.80%

We selected the best parameters values and structures representation of the classifiers above mentioned for complete the basic module unit with type-1 FIS and IT2FIS. The parameters values in the basic module unit 1 for the Fuzzy KNN algorithm is k= 4 neighbors, for the MLP-GDM with 150 hidden neurons and for the MLP-SCG with 50 hidden neurons. Finally, with the type-1 FIS and IT2FIS we combined the outputs of the classifiers and the classification rate in basic module unit 1 is 92.20%, see Table 4.4. In Table 4.5 are presented the results of IT2FIS with a 93.80%. We have the global classification rates of the methods in Figure 4.6.

Table 4.4 Results for cross validation of type-1 FIS, combining the outputs of the three classifiers in basic module unit 1

Class	N	LBB	RBB	PVC	FPN	AP	AAP	FVN	VE	P
N	99	0	0	0	0	0	0	1	0	0
LBB	0	91	0	0	1	1	2	3	2	0
RBB	0	2	96	0	1	0	1	0	0	0
PVC	0	4	0	86	3	0	3	1	0	3
FPN	0	0	2	1	92	0	1	2	0	2
AP	0	0	0	0	0	94	6	0	0	0
AAP	0	4	0	0	0	6	84	1	5	0
FVN	1	2	0	1	1	1	0	94	0	0
VE	0	0	0	0	0	0	4	0	96	0
P	0	0	0	0	3	0	0	0	0	97

Classification rate= 92.9%

Table 4.5 Results for cross validation of IT2FIS, combining the outputs of the three classifiers in basic module unit 1

Class	N	LBB	RBB	PVC	FPN	AP	AAP	FVN	VE	P
N	99	0	0	0	0	0	0	1	0	0
LBB	0	94	0	0	0	1	1	2	2	0
RBB	0	2	96	0	1	0	1	0	0	0
PVC	0	4	0	86	3	0	2	2	0	3
FPN	0	0	2	1	91	0	1	2	0	3
AP	0	0	0	0	0	95	5	0	0	0
AAP	0	3	0	0	0	5	87	1	4	0
FVN	0	1	0	1	1	0	1	96	0	0
VE	0	0	0	0	0	0	4	0	96	0
P	0	0	0	0	2	0	0	0	0	98

Classification rate= 93.8%

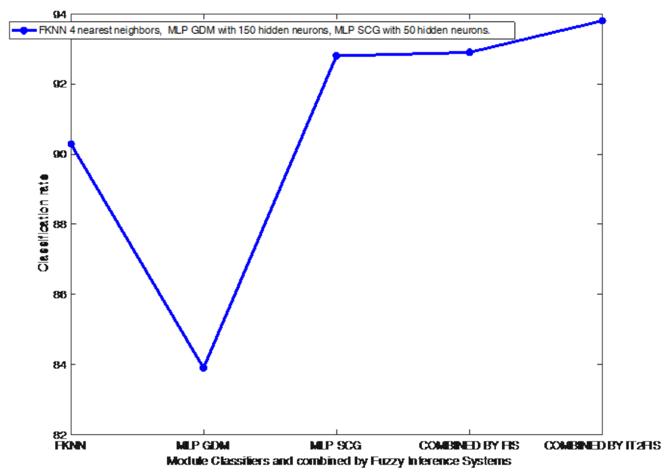


Figure 4.6 Classification rate of basic module unit 1 using MIT-BIH arrhythmia database's MLII electrode signal

4.1.9.2 Results for the basic module unit 2

In this part of this section, we show the results of the basic module unit 2. In the results for the first classifier into basic module unit 2 we found out that the best result used with k=3 nearest neighbors with 89.10% of classification rate, see confusion matrix in Table 4.6. For the second classifier, MLP-GDM with 150 hidden neurons, we obtained 76.10% of classification rate, see Table 4.7, and for the third classifier MLP-SCG, we found out that best result is with 50 hidden neurons with an 91.20% of classification rate, see details in Table 4.8. The results for the type-1 FIS and ITSFIS are shown in Table 4.9 and Table 4.10 respectively.

Table 4.6 Results for cross validation of fuzzy KNN algorithm, k=3, first classifier in basic module unit 2

Class	N	LBB	RBB	PVC	FPN	AP	AAP	FVN	VE	P
N	91	0	1	0	0	1	6	0	1	0
LBB	0	93	0	2	1	0	0	3	1	0
RBB	0	0	99	0	0	0	0	0	1	0
PVC	2	9	1	67	3	8	5	2	3	0
FPN	0	1	0	0	97	0	1	0	0	1
AP	6	0	0	2	0	87	3	1	1	0
AAP	6	0	0	2	2	4	86	0	0	0
FVN	2	13	1	3	2	2	0	77	0	0
VE	1	1	1	0	0	0	0	0	95	2
P	0	0	0	0	1	0	0	0	0	99

Classification rate= 89.10%

Table 4.7 Results for cross validation of MLP-GDM, 150 hidden neurons, second classifier in basic module unit 2

Class	N	LBB	RBB	PVC	FPN	AP	AAP	FVN	VE	P
N	72	0	0	0	0	23	0	0	5	0
LBB	0	90	3	0	7	0	0	0	0	0
RBB	2	0	97	0	0	0	0	0	1	0
PVC	6	41	1	18	2	14	6	5	7	0
FPN	0	0	0	1	97	0	0	0	0	2
AP	10	0	0	0	0	89	1	0	0	0
AAP	15	1	10	0	12	14	46	0	2	0
FVN	0	25	1	0	5	11	3	55	0	0
VE	0	0	2	0	0	0	1	0	97	0
P	0	0	0	0	0	0	0	0	0	100

Classification rate= 76.10%

Table 4.8 Results for cross validation of MLP-SCG 50 hidden neurons, third classifier in basic module unit 2

Class	N	LBB	RBB	PVC	FPN	AP	AAP	FVN	VE	P
N	89	0	1	1	0	3	4	0	2	0
LBB	0	99	0	0	0	0	0	1	0	0
RBB	0	0	98	0	0	0	1	0	1	0
PVC	0	4	2	70	0	6	9	4	5	0
FPN	0	1	0	1	95	0	0	1	0	2
AP	3	1	0	0	0	95	1	0	0	0
AAP	0	0	0	3	2	5	90	0	0	0
FVN	0	1	1	2	1	2	1	92	0	0
VE	0	1	1	0	0	0	0	0	97	1

P	0	0	0	0	2	0	0	1	10	87

Classification rate= 91.20%

We selected the best parameters values and structures representation of the classifiers mentioned above for complete the basic module unit with type-1 FIS and IT2FIS. The parameters values in the basic module unit 2 for the Fuzzy KNN algorithm is k= 3 neighbors, for the MLP-GDM with 150 hidden neurons and for the MLP-SCG with 50 hidden neurons. Finally, with the type-1 FIS and IT2FIS we combined the outputs of the classifiers and the classification rate of the basic module unit 2 is of 92.40%, see Table 4.9. In Table 4.10 are presented the results of IT2FIS with a 92.70%. We have the global classification rates of the methods in Figure 4.7.

Table 4.9 Results for cross validation of type-1 FIS, combining the outputs of the three classifiers in basic module unit 2

Class	N	LBB	RBB	PVC	FPN	AP	AAP	FVN	VE	P
N	96	0	0	1	0	1	2	0	0	0
LBB	0	94	1	1	1	0	0	3	0	0
RBB	0	0	99	0	0	0	0	0	1	0
PVC	1	6	1	73	2	4	5	3	5	0
FPN	0	1	0	1	96	0	0	1	0	1
AP	4	0	0	0	0	96	0	0	0	0
AAP	2	0	0	4	2	3	88	0	1	0
FVN	1	2	1	4	1	1	0	90	0	0
VE	1	1	1	0	0	0	0	0	95	2
P	0	0	0	0	3	0	0	0	0	97

Classification rate= 92.40%

Table 4.10 Results for cross validation of IT2FIS, combining the outputs of the three classifiers in basic module unit 2

Class	N	LBB	RBB	PVC	FPN	AP	AAP	FVN	VE	P
N	96	0	1	0	0	1	2	0	0	0
LBB	0	95	0	2	1	0	0	1	1	0
RBB	0	0	99	0	0	0	0	0	1	0
PVC	1	6	1	72	2	7	4	2	5	0
FPN	0	1	0	1	97	0	0	0	0	1
AP	3	0	0	0	0	96	0	0	1	0
AAP	4	0	0	2	2	5	87	0	0	0
FVN	1	2	1	2	1	2	1	90	0	0
VE	1	1	1	0	0	0	0	0	95	2
P	0	0	0	0	0	0	0	0	0	100

Classification rate= 92.7%

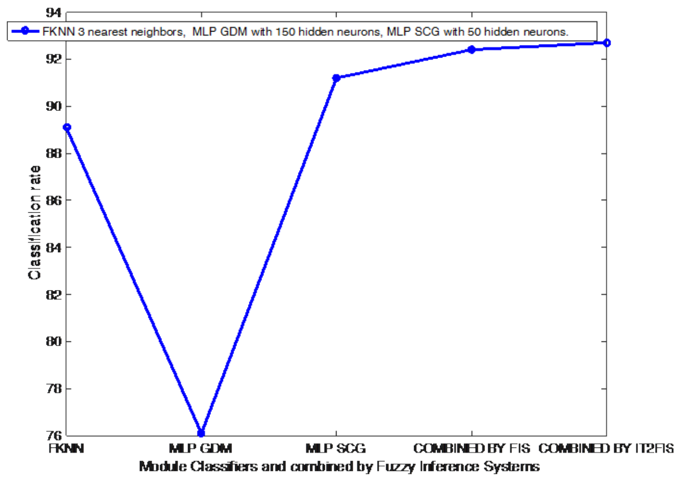


Figure 4.7 Classification rate of basic module unit 2 using MIT-BIH arrhythmia database's (v1, v2 or v3) electrode signal

4.1.9.3 Results for the hybrid model.

Finally, the confusion matrix of the results of the hybrid model using type-1 FIS are presented in Table 4.11, we presented a brief summary results of the complete proposed hybrid model in this work in Table 4.12.

Table 4.11 Results for 10-cross validation of hybrid model using type-1 FIS

Class	N	LBB	RBB	PVC	FPN	AP	AAP	FVN	VE	P
N	97	0	0	0	0	0	2	1	0	0
LBB	0	95	0	0	0	0	1	2	2	0
RBB	0	2	97	0	0	0	1	0	0	0
PVC	0	5	1	84	0	2	4	2	1	1
FPN	0	1	0	0	97	0	0	2	0	0
AP	1	0	0	0	0	92	7	0	0	0
AAP	1	3	0	1	0	1	91	1	2	0
FVN	2	3	1	0	0	0	0	94	0	0
VE	1	0	1	0	0	0	2	0	95	1
P	0	0	0	0	4	0	0	0	0	96

Classification rate= 93.8%

Table 4.12 Results for 10-cross validation of complete hybrid model

	Unit 1					Unit 2					Hybrid **Model**		
	FKNN	MLP1	MLP2	type-1 FIS	IT2FIS	FKNN	MLP1	MLP2	type-1 FIS	IT2FIS	type-1 FIS	IT2FIS A	IT2FIS B
Classification rate=	90.3	83.90	92.80	92.90	93.80	89.10	76.10	91.20	92.40	92.70	93.80	94.20	94.30

IT2FIS A= type-1 FIS from Unit1 + type-1 FIS from Unit2.
IT2FIS B= IT2FIS from Unit1 + ITFIS from Unit2.

4.1.9.4 Statistical Analysis

In this section, we are presenting a comparison of results for the statistical analysis between type-1 FIS and IT2FIS for the basic module unit 1. Both Fuzzy Systems were used to combine the outputs of the three classifiers. These results were obtained by a 10-cross validation process and reflected in the classification rate for the basic module unit 1. In Table 4.13, the results of the hypothesis testing are presented and illustrated in Figure 4.8.

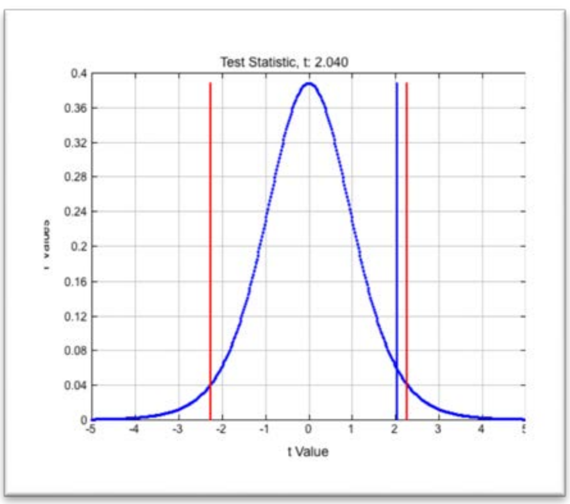


Figure 4.8 Hypothesis testing of type-1 FIS and IT2FIS for basic

Table 4.13 Results for Hypothesis Testing of type-1 FIS and IT2FIS for basic module unit 1

Alternative Hypothesis:	**μ not equal μ(hyp)**
Size of sample, n:	10
Difference of means, d:	1.4
Difference of st. dev., sd:	2.170509
Statistic, t:	2.0397
Critical value of t:	±2.262
Value of -P:	0.0718
95% Confidence interval:	-0.1526873 < μd < 2.952687

Null hypothesis not rejected
Sample does not provides sufficient statistical evidence to reject null hypothesis

4.1.9.5 Comparison with other works

We presented brief results comparison between proposed hybrid model and some other works, in literature, see Table 4.14.

Table 4.14 Results comparison between proposed hybrid model and other works

Work	Classifier	Effectiveness
Ceylan, R., et. al.	Type-2 fuzzy clustering neural network	99%
Amezcua, J., Melin, P.	modular LVQ neural network	98.89%
Hu et. al.	Mixture of experts	94%
Ince et. al.	MDPSO	95.58%
Martis et. al.	SVM-RBF	93.48%
De Chazal et al.	Weighted LD	83%
Soria and Martinez	Weighted LD	90%
Llamedo and Martinez	Weighted LD	93%
Ye et al.	SVM	86.4%
Lin and Yang	Weighted LD	93%
Zhang and Luo	Combined SVM	87%
Proposed Hybrid Model	**Fuzzy logic and neural networks**	**94.30%**

4.1.9.6 Optimization of the type-1 Fuzzy Inference Systems in proposed Hybrid Model

We have optimized the type-1 Fuzzy Inference Systems in the proposed hybrid model used Bee Colony Optimization algorithm. We found a structure for the type-1 FIS that gave us better results. As we can observe, the classification rate for the basic module unit 1 increased from 92.90 to 93.20 and 92.40 to 93. With respect to the combination of both basic module units in the hybrid model, we also achieved an increase in the global classification rate from 93.80 to 95.80 using type-1 fuzzy logic. In Table 4.15, we present the optimization results of the type-1 FIS. Figure 4.9 shows the membership functions that corresponding to the optimized type-1 FIS for the basic module units.

Table 4.15 Optimization results for the type-1 FIS in hybrid model used BCO

	Unit 1		**Unit 2**		**Hybrid Model**	
	type-1 FIS	**type-1 FIS+ BCO**	**type-1 FIS**	**type-1 FIS+ BCO**	**type-1 FIS**	**type-1 FIS+BCO**
Classification rate=	**92.90**	**93.20**	**92.40**	**93**	**93.80**	**95.80**

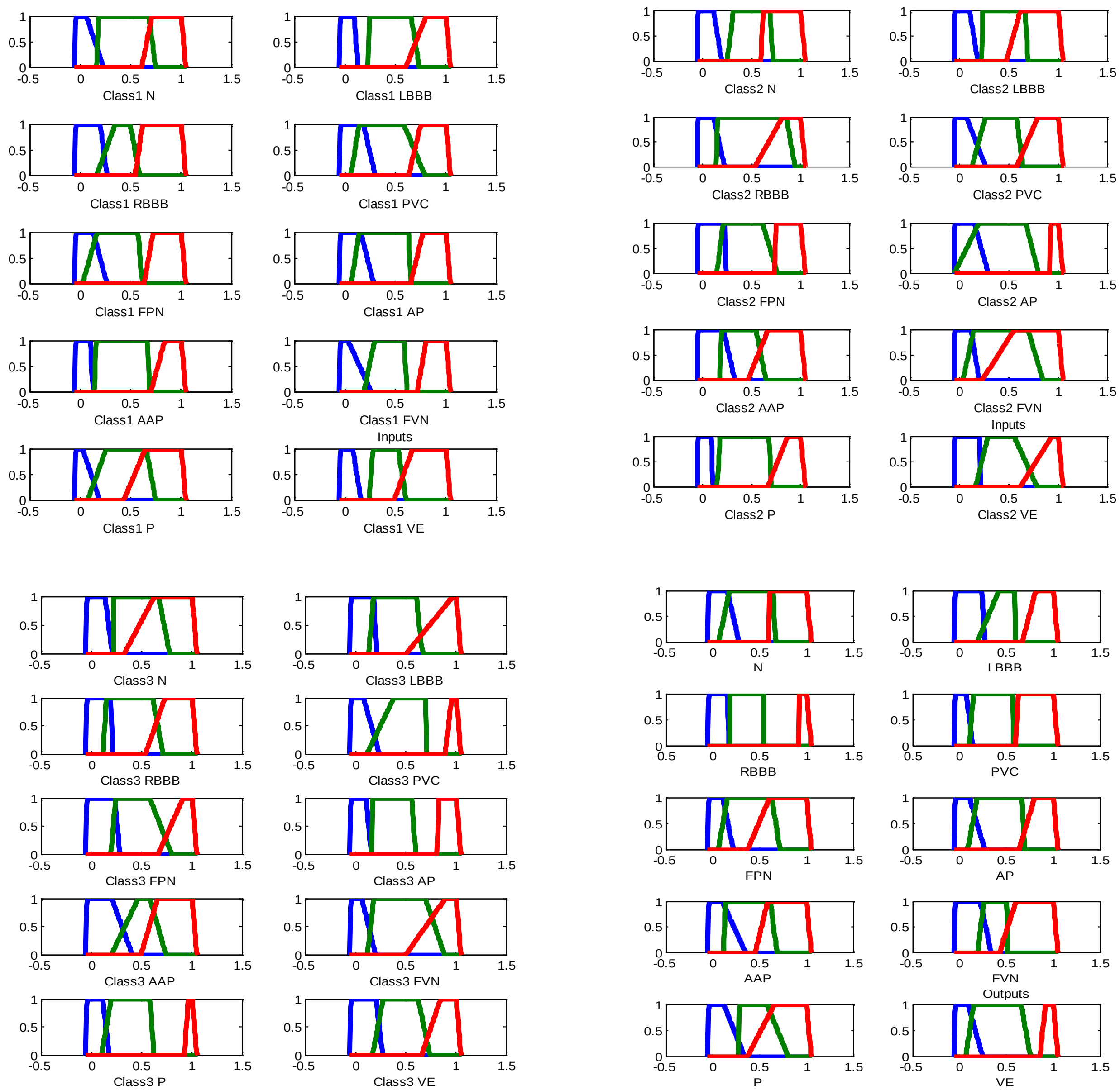


Figure 4.9 Memberships functions for inputs and outputs of the optimized type-1 FIS for the basic module units in hybrid model

4.1.10 Discussions of results

In the results presented with respect to the performed experiments, we conclude that we have obtained a good classification rate with the classifiers in the two basic module units, and even more combining their outputs using fuzzy logic, we have obtained even better results; in this part of the hybrid system, we used two options to combine the results of the classifiers in the basic module units and we have presented a comparison between type-1 FIS and IT2FIS with hypothesis testing. We obtained the best results using an IT2FIS with a 93.80% against 92.90% of classification rate for the type-1 FIS in basic module unit 1. We improved the classification rate in basic module unit achieving a 93.80% with 10-fold cross validation using a type-2 fuzzy system. In the comparison using hypothesis testing, although the current results do not provide sufficient statistical evidence to reject the null hypothesis, we find out that using type-2 fuzzy logic, we can achieve better results, and we can still improve the interval type-2 fuzzy system with optimization methods. Referring to the basic module unit 2, we obtained 92.40% of classification rate used type-1 FIS combining the results of the classifiers above mentioned, and 92.70% of classification rate used type-2 fuzzy system.

We used two basic module units in the hybrid model, where each one was trained with electrode signals of the MIT-BIH arrhythmia database and we combined the output results of the two basic module units using a type-1 FIS, the results show the improvement of the global classification rate in the hybrid model to achieve a 93.80%. We found out that some samples of heartbeats were misclassified by one basic module unit. However, when combining the outputs with 2-lead using two basic module units some samples with misclassification were fixed and classified correctly by the other basic module unit using a fuzzy system, and in this form, we have obtained better performance and classification rate using a hybrid model to resolve cardiac arrhythmia classification problem.

Based on the obtained results, we found out that IT2FIS from Unit 1 achievement obtain the same classification rate than the global hybrid model used type-1 FIS, but also we have included an IT2FIS to use in the global hybrid model. Therefore, we can show that global IT2FIS provided higher accuracy with 94.20% combining type-1 FIS from Unit1 and type-1 from Unit2) and 94.30% combining IT2FIS from Unit1 and IT2FIS from Unit2 than global type-1 FIS 93.80% of classification rate. In other words, implementing type-2 instead type-1 fuzzy logic improves overall accuracy of the proposed hybrid model.

Some advantages about the proposed hybrid model are that we can combine several computational intelligence methods using fuzzy systems to increase the overall accuracy of the hybrid model. We can separate into basic module units of expert modules for the electrode signal or lead of an ECG to consider the different perspectives that offers each lead and helps the classification of cardiac arrhythmias. One disadvantage that we can mention with respect to the proposed hybrid model relates to the total number of fuzzy if-then rules for the fuzzy systems, and for this reason is necessary to optimize, for example, with genetic algorithms or other technique in order to reduce the total of fuzzy if-then rules in the fuzzy systems.

We will work to extend the hybrid system to use multi-lead arrhythmia classification using other databases that contain 12 leads to be able to make a complete medical diagnosis.

4.2 12-lead cardiac arrhythmia classification

The conventional 12-lead ECG used 10 electrodes simply by comparing electrical potentials formed the leads such as i, ii, iii, avf, avr, avl, v1, v2, v3, v4, v5, and v6.

The PTB Diagnostic ECG Database contains 549 ECG records from 290 subjects, the subjects aged 17 to 87 years old. One to five ECG records represent each subject. Each ECG record includes 15 measured signals, the conventional 12 leads: *i, ii, iii, avr, avl, avf, v1, v2, v3, v4, v5, v6*, and three Frank lead such as *vx, vy* and *vz*. The diagnosis classes included in the database are Myocardial Infarction, Cardiomyopathy, Bundle Branch Block, Dysrhythmia, Myocardial

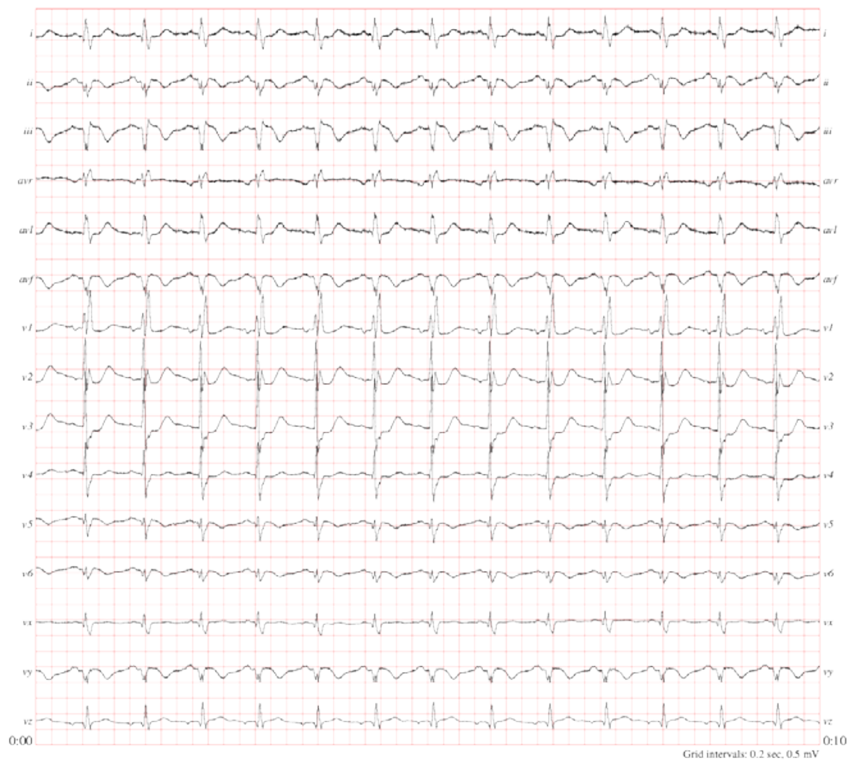
Hypertrophy, Valvular Heart Disease, Myocarditis, Miscellaneous, and Healthy Controls.

This work is related to cardiac arrhythmia classification using the Physikalisch-Technische Bundesanstalt (PTB) Diagnostic ECG database [3, 13] achieved through the hybridization of two computational intelligence techniques such as fuzzy logic and artificial neural networks. The purpose is to design a hybrid intelligent system to be able combine different computational intelligence techniques considering the conventional 12 leads from the electrode signals of a complete electrocardiogram to classify cardiac arrhythmias to support a complete medical diagnosis.

4.2.1 Hybrid model based on neural networks, type-1 and type-2 fuzzy systems for 12-lead cardiac arrhythmia classification

We extracted different samples of electrocardiograms from the PTB Diagnostic ECG database. The selected samples of electrocardiograms were preprocessing used autoregressive model coefficients, Shannon entropy and multifractal wavelets to create the feature vectors that represent each sample of electrocardiogram. The feature vectors are the inputs for the expert modules of the proposed hybrid intelligent system. The classes that were included in the PTB Diagnostic ECG database are Myocardial Infarction, Cardiomyopathy, Bundle Branch Block, Dysrhythmia, Myocardial Hypertrophy, Valvular Heart Disease, Myocarditis, Miscellaneous and Healthy Controls. In Figure 4.10, 4.11, and 4.12, some examples of different diagnostic classes of the PTB Diagnostic ECG database are presented.

We selected 280 electrocardiograms of different patients. We are working on 9 classes: Myocardial infarction, Cardiomyopathy, Bundle branch block, Dysrhythmia, Myocardial hypertrophy, Valvular heart disease, Myocarditis, Miscellaneous and Healthy controls, 466 samples segments of electrocardiograms. The hybrid intelligent system learns the 9 classes using samples of each class. We used 70% for training and 30% for testing.



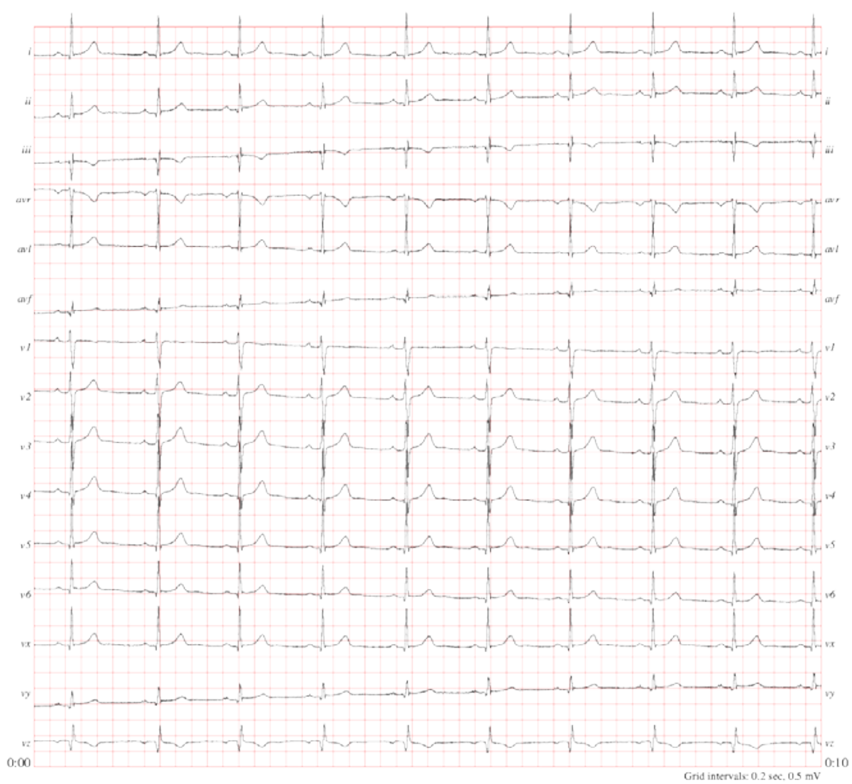


Figure 4.10 Examples of the classes Myocardial Infaction and Miocarditis respectively, in the PTB Diagnosis ECG database

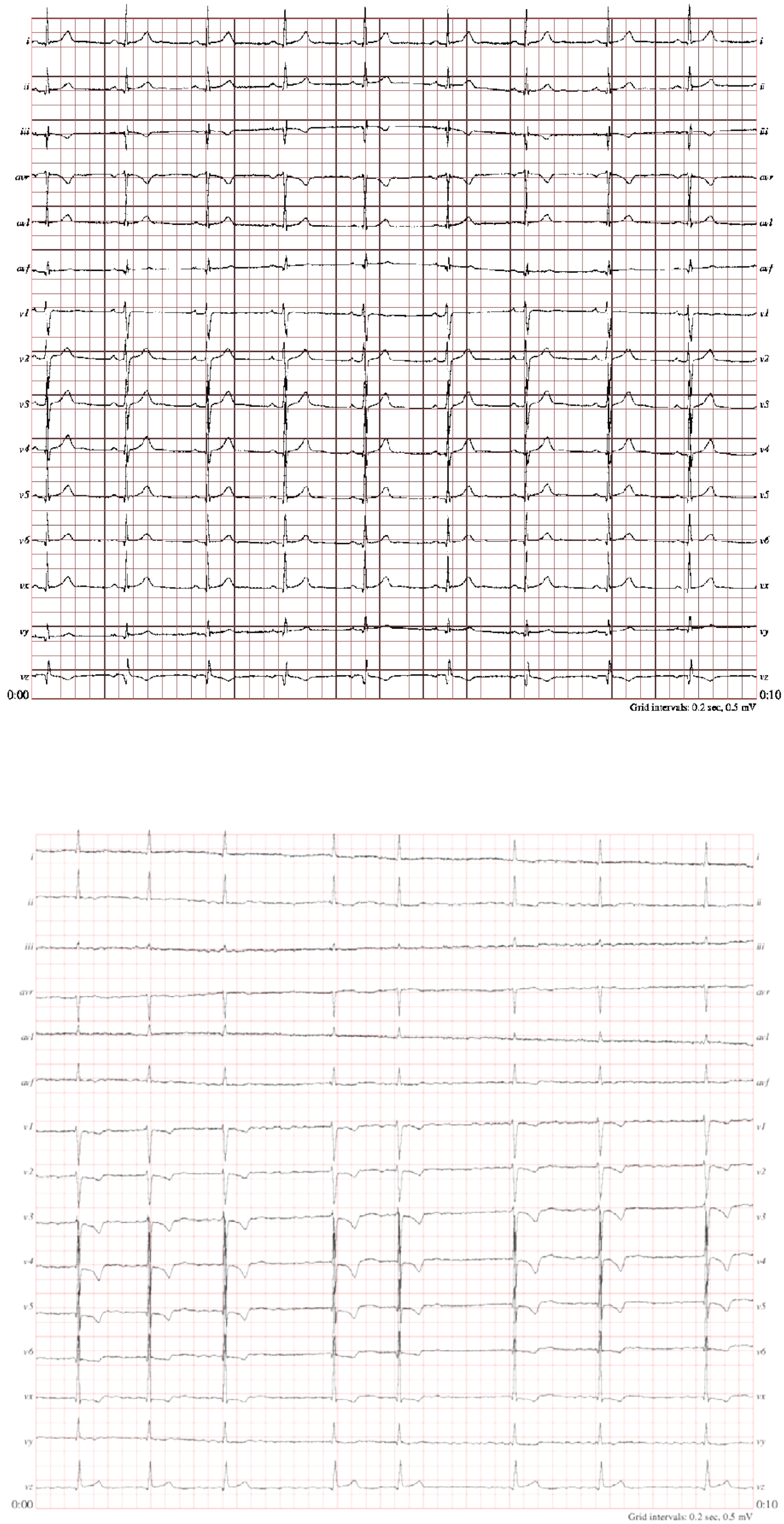


Figure 4.11 Examples of the classes Myocardial hypertrophy and Cardiomyopathy respectively, in the PTB Diagnostic ECG database

Figure 4.12 Examples of the classes Dysrhythmia and Healthy controls respectively, in the PTB Diagnostic ECG database

In this work, we used the conventional twelve leads: i, ii, iii, avr, avl, avf, v1, v2, v3, v4, v5 and v6 included in the electrocardiograms of the PTB Diagnostic ECG database, in Figure 4.13, 4.14 and 4.15, we show some examples of the conventional leads. The electrocardiograms used for the classes were belonging of the patients 1 to 103, 108, 111, 120, 128, 138 to 142, 145, 148, 149, 152, 158, 163, 183, 193, 195, 197, 207, 211, 223, 226, 230, 231, 259, 261, 265, 268, 270, 273, 274, 280, 282, 283, 287, 290 to 294 for Myocardial infarction. For the class of Cardiomyopathy 127, 129, 136, 201, 215, 222, 232, 253, 254, 256, 257, 262, 288 and 289. For the class of Bundle branch block 171, 175, 199, 202, 203, 204, 206, 208, 209, 213, 217, 219, 220, 225 and 228. For the class of Dysrhythmia 109, 112, 113, 133, 147, 151, 157, 168, 177, 187, 218, 258 and 286. For the class of Hypertrophy 159, 210, 212, 216, 221 and 227. For the class of Valvular heart disease 106, 107, 110, 114, 188 and 224. For the class of Myocarditis 249, 269, 271 and 272. For the class of Miscellaneous 119, 125, 126, 130, 136, 143, 144, 146, 153, 154, 162, 164, 176, 178, 179, 181, 186, 190, 191, 192, 194, 196, 200, 275, 278, 281 and 285. Finally, the records of the patients 104, 105, 116, 117, 121, 122, 131, 155, 156, 165, 166, 169 to 174, 180, 182, 184, 185, 198, 214, 229, 233 to 248, 251, 252, 255, 260, 263, 264, 266, 267, 276, 277, 279, 284 for healthy controls, in some cases more than one electrocardiogram of the same patient was used available in the PTB diagnosis ECG database.

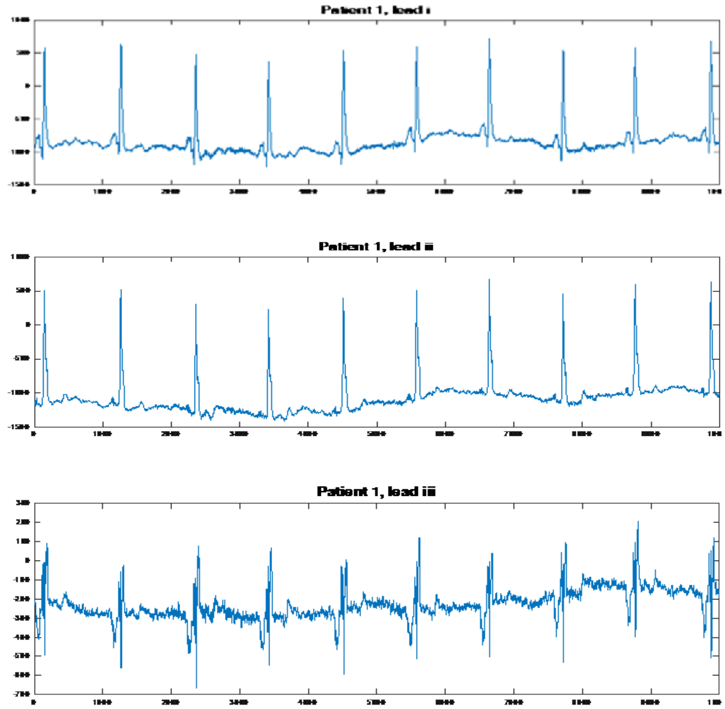


Figure 4.13 The conventional leads *i, ii, iii* in electrocardiograms of the PTB Diagnostic database

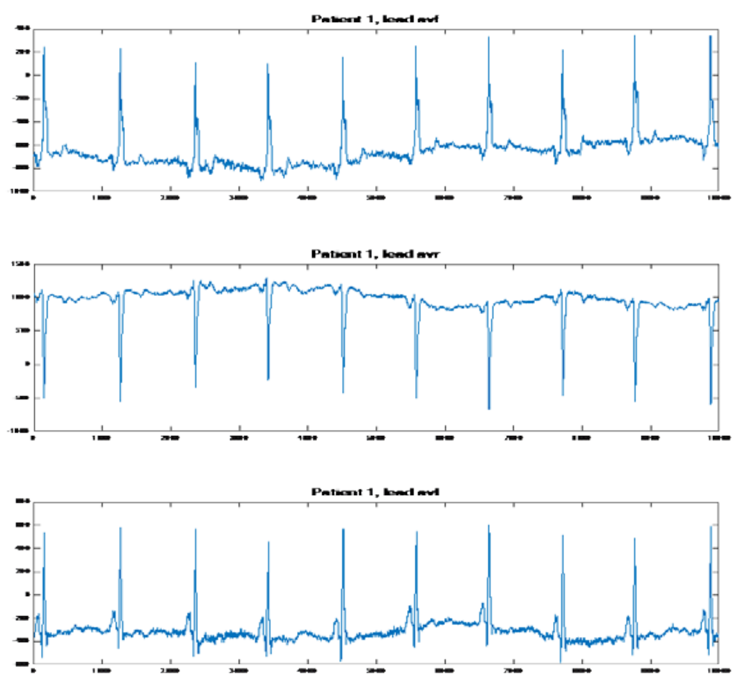


Figure 4.14 The conventional leads *avf, avr, avl* in electrocardiograms of the PTB Diagnostic database

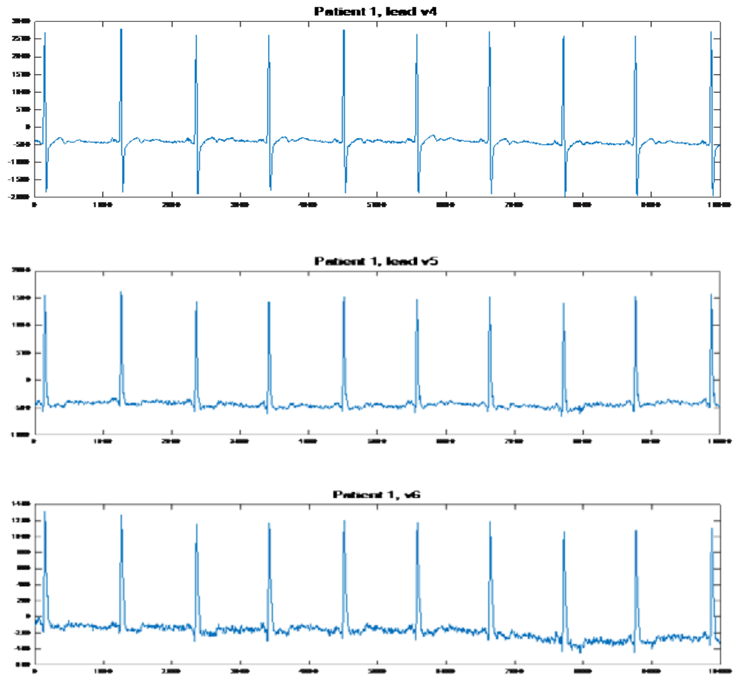


Figure 4.15 The conventional leads *v4, v5, v6* in electrocardiograms of the PTB Diagnostic database

The hybrid intelligent system is composed of twelve expert modules, where each module is related to an electrode signal or lead in the electrocardiogram being 12 Leads cardiac arrhythmia classification solution. We used an interval type-1 and interval type-2 fuzzy inference system to determine the global classification for the hybrid intelligent system using the twelve expert modules. In Figure 4.16, we are presented the architecture of the hybrid intelligent system.

In the preprocessing phase, we applied a feature extraction process in order to reduce or simply the information of the selected samples of the electrocardiograms as well as for the classifiers captures the differences between the classes to improve the classification rate. We created a set of vectors that represent a complete signal or lead in the electrocardiogram; in other words, each electrocardiogram has its twelve feature vectors to be learning for the expert modules in the proposed hybrid intelligent system. The feature vectors are built with autoregressive models, Shannon entropies, and wavelets.

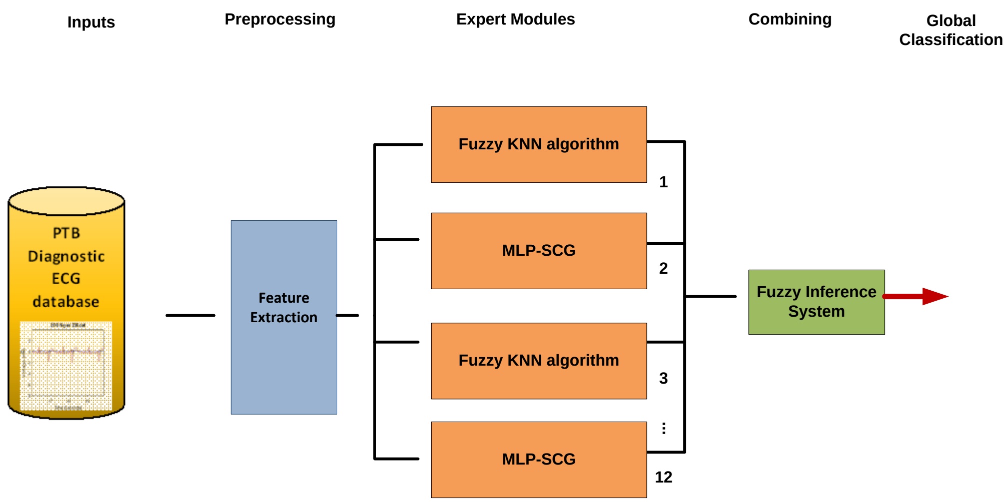


Figure 4.16 Hybrid Intelligent System for 12-leads arrhythmia classification

We combined the output results of the twelve expert modules with the type-1 and interval type-2 fuzzy inference systems. The type-1 and interval type-2 fuzzy inference system has the follow structure: Mamdani type, 108 inputs, and 9 outputs; the first 9 inputs belong for the expert module of the *i* lead, the second 9 inputs corresponding for the expert module of the *ii* lead, the third 9 inputs for the expert module of *iii* lead, and so on until the expert module *v6* lead. The outputs corresponding to the classes learned for the Fuzzy Inference System. We used trapezoid functions for the inputs and outputs. The fuzzy variables *Low, Medium, and High* were used in each input and output. Centroid method is used as defuzzification method. The fuzzy rules represent the basic knowledge specific class. We show type-1 and interval type-2 fuzzy inference systems in Figure 4.17.

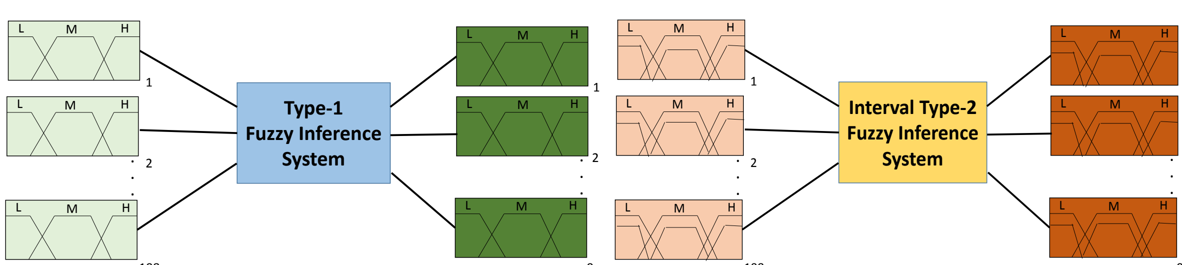


Figure 4.17 type-1 and interval type-2 fuzzy inference system

We present the parameters used for inputs and outputs for type-1 fuzzy inference system, see equation 4.8, 4.9 and 4.10. In equation 4.11, 4.12 and 4.13 for interval type-2 fuzzy inference system:

$$\mu_{Low}(x) = \begin{cases} 0, & x \leq -1.218 \\ \frac{x+1.218}{0.112}, & -1.218 \leq x \leq -1.106 \\ 1, & -1.106 \leq x \leq -0.6361 \\ \frac{-0.6361-x}{0.112}, & -0.6361 \leq x \leq -0.3787 \\ 0 & -0.3787 \leq x \end{cases} \tag{4.8}$$

$$\mu_{Medium}(x) = \begin{cases} 0, & x \leq -0.3742 \\ \frac{x+-0.3742}{0.30012}, & -0.3742 \leq x \leq -0.07408 \\ 1, & -0.07408 \leq x \leq 0.1374 \\ \frac{0.4814-x}{0.1149}, & 0.1374 \leq x \leq 0.4814 \\ 0 & 0.4814 \leq x \end{cases} \tag{4.9}$$

$$\mu_{High}(x) = \begin{cases} 0, & x \leq 0.4725 \\ \frac{x-0.4725}{0.2307}, & 0.4725 \leq x \leq 0.7032 \\ 1, & 0.7032 \leq x \leq 1.149 \\ \frac{1.252-x}{0.2307}, & 1.149 \leq x \leq 1.252 \\ 0 & 1.252 \leq x \end{cases} \tag{4.10}$$

$$\mu_{Low}(x) = \begin{cases} 0, & x \leq -0.36 \\ \frac{x+0.36}{0.32}, & -0.36 \leq x \leq -0.04 \\ 1, & -0.04 \leq x \leq 0.04 \\ \frac{0.04-x}{0.32}, & 0.04 \leq x \leq 0.36 \\ 0, & 0.36 \leq x \end{cases} \tag{4.11}$$

$$\mu_{Medium}(x) = \begin{cases} 0, & x \leq 0.14 \\ \frac{x-0.14}{0.32}, & 0.14 \leq x \leq 0.46 \\ 1, & 0.46 \leq x \leq 0.54 \\ \frac{0.86-x}{0.32}, & 0.54 \leq x \leq 0.86 \\ 0, & 0.86 \leq x \end{cases} \tag{4.12}$$

$$\mu_{High}(x)=\begin{cases} 0, & x\leq 0.64 \\ \frac{x-0.64}{0.32}, & 0.64\leq x\leq 0.96 \\ 1, & 0.96\leq x\leq 1.04 \\ \frac{1.36-x}{0.32}, & 1.04\leq x\leq 1.36 \\ 0, & 1.36\leq x \end{cases} \quad (4.13)$$

We have 224 fuzzy rules in type-1 and interval type-2 fuzzy inference systems of which represent a set of basic rules for combining from expert module responses considering the specific class and the ECG regions of the heart are presented in Table 4.16.

Table 4.16 Fuzzy rules of type-1 and interval type-2 Fuzz Inference Systems based on ECG regions of the heart

Leads	*ECG Regions*	*Fuzzy Rules*
ii, iii, avr	Inferior	21
i, avl	High Lateral	9
v5, v6	Low Lateral	9
i, avl, v5, v6	Free Wall	31
v1, v2	Middle Septal	9
v3, v4	Inferior Septal	9
v1 to v4	Antero septal	31
v1 to v6	Anterior Extension	105

The 21 basic fuzzy rules for inferior ECG region of the heart used ii, iii and avr leads are listed as follows:

1.- IF (MI_ii IS High) AND (MI_iii IS High) AND (MI_avr IS High) THEN (MI IS High).

2.- IF (MI_ii IS Low) AND (MI_iii IS Medium) AND (MI_avr IS Low) THEN (MI IS Low).

3.- IF (MI_ii IS Low) AND (MI_iii IS Medium) AND (MI_avr IS Low) THEN (MI IS Low).

4.- IF (MI_ii IS Medium) AND (MI_iii IS Low) AND (MI_avr IS Low) THEN (MI IS Low).

5.- IF (MI_ii IS Low) AND (MI_iii IS Low) AND (MI_avr IS High) THEN (MI IS Low).

6.- IF (MI_ii IS Low) AND (MI_iii IS High) AND (MI_avr IS Low) THEN (MI IS Low).

7.- IF (MI_ii IS High) AND (MI_iii IS Low) AND (MI_avr IS Low) THEN (MI IS Low).

8.- IF (MI_ii IS Medium) AND (MI_iii IS Medium) AND (MI_avr IS Medium) THEN (MI IS Medium).

9.- IF (MI_ii IS Medium) AND (MI_iii IS Medium) AND (MI_avr IS Low) THEN (MI IS Medium).

10.-IF (MI_ii IS Medium) AND (MI_iii IS Low) AND (MI_avr IS Medium) THEN (MI IS Medium).

11.-IF (MI_ii IS Low) AND (MI_iii IS Low) AND (MI_avr IS Medium) THEN (MI IS Medium).

12.-IF (MI_ii IS Medium) AND (MI_iii IS Medium) AND (MI_avr IS High) THEN (MI IS Medium).

13.-IF (MI_ii IS Medium) AND (MI_iii IS High) AND (MI_avr IS Medium) THEN (MI IS Medium).

14.-IF (MI_ii IS High) AND (MI_iii IS Medium) AND (MI_avr IS Medium) THEN (MI IS Medium).

15.-IF (MI_ii IS High) AND (MI_iii IS High) AND (MI_avr IS High) THEN (MI IS high).

16.-IF (MI_ii IS High) AND (MI_iii IS High) AND (MI_avr IS Low) THEN (MI IS High).

17.-IF (MI_ii IS High) AND (MI_iii IS Low) AND (MI_avr IS High) THEN (MI IS High).

18.-IF (MI_ii IS Low) AND (MI_iii IS High) AND (MI_avr IS High) THEN (MI IS High).

19.-IF (MI_ii IS High) AND (MI_iii IS High) AND (MI_avr IS Medium) THEN (MI IS High).

20.-IF (MI_ii IS High) AND (MI_iii IS Medium) AND (MI_avr IS High) THEN (MI IS High).

21.-IF (MI_ii IS Medium) AND (MI_iii IS High) AND (MI_avr IS High) THEN (MI IS High).

In the following part of this section, we describe a brief summary with some important concepts referent of the techniques used in this proposed work.

4.2.2 Experiments

We performed experiments used randomly 70% of the selected electrocardiograms to train the expert modules in proposed hybrid intelligent system and the rest 30% for testing. We have trained classifiers based on the Fuzzy KNN algorithm and MLP-SCG to select the better that represent each expert module by lead or electrode signal from electrocardiogram. The parameters for Fuzzy KNN algorithm were used k=3 to 35 nearest neighbor. The structure for the MLP-SCG's were 50, 100, 150 to 850 hidden neurons, 190 input neurons, 9 output neurons, 10000 epochs, learning rate 0.001. Where the output neurons are the classes used to train the hybrid intelligent system such as Myocardial Infarction, Cardiomyopathy, Bundle Branch Block, Dysrhythmia, Myocardial Hypertrophy, Valvular Heart Disease, Myocarditis, Miscellaneous and Healthy Controls. We selected the better-trained classifier in order to form the expert module representation in hybrid intelligent system. The outputs of the selected expert modules were combining using a type-1 and interval type-2 Fuzzy Inference System to determine the global classification of the proposed hybrid intelligent system. The rules of Fuzzy Inference Systems are based on ECG regions such as inferior, high Lateral, low lateral, free wall, middle septal, inferoseptal, anteroseptal, anterior extension.

4.2.3 Results

In this part, we present the results for the expert modules separately and the global results for the hybrid model proposed. In the last part of this section, we are present a statistical analysis to compare the results of the fuzzy systems.

4.2.3.1 Expert Modules. Classifiers: Fuzzy KNN algorithm and MLP-SCG.

Firstly, we present the results for the classifiers used to form the expert modules in hybrid intelligent system. The results of the classifiers Fuzzy KNN algorithm are shown in Table 4.17 and 4.18, MLP-SCG above mentioned in the experiments section are show in Table 4.19, 4.20, 4.21. We can observe the better results represented for each lead, which were selected to form part of the expert modules in the proposed hybrid intelligent system.

For the Fuzzy KNN algorithm, the best results for 12 leads were used k=3, 4 and 5 nearest neighbors. In Figure 4.18 and 4.19 are presented the comparative results of classification rate per each lead used Fuzzy KNN algorithm.

Table 4.17 Results for testing of Fuzzy KNN algorithm

Lead	*K*=3	*K*=4	*K*=5	*K*=6	*K*=7	*K*=8	*K*=9	*K*=10	*K*=11	*K*=12	*K*=13	*K*=14	*K*=15	*K*=16	*K*=17	*K*=18
i	92.14	90.71	90.71	89.28	89.28	89.28	89.28	89.28	89.28	89.28	89.28	89.28	89.28	89.28	89.28	89.28
ii	84.28	85	85.71	56.42	56.42	56.42	56.42	56.42	56.42	56.42	56.42	55.71	55.71	55.71	55.71	55.71
iii	82.14	82.85	82.85	70	70	70	70	70	69.28	70	70	70	69.28	69.28	69.28	69.28
avf	82.14	82.85	80.70	52.85	54.28	53.57	50.71	50	50	49.28	49.28	49.28	49.28	49.28	49.28	49.28
avr	85.71	85.71	85	69.28	69.28	68.57	68.57	68.57	67.14	67.14	66.42	67.85	67.85	67.85	67.85	67.85
avl	87.14	88.57	87.14	86.42	86.42	86.42	85.71	85.71	85.71	85.71	85.71	85.71	85.71	85.71	85.71	85.71
v1	87.85	86.42	85.71	70.71	71.42	70.71	71.42	71.42	70.71	70.71	70.71	70.71	70.71	70.71	70.71	70
v2	91.42	90.71	90.71	82.85	81.42	8.71	80.71	80.71	80.71	81.42	81.42	81.42	81.42	80.71	80.71	80.71
v3	92.85	92.85	92.85	57.85	58.57	58.57	58.57	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14
v4	94.28	94.28	93.57	75.71	75.71	75	74.28	74.28	74.28	74.28	74.28	74.28	74.28	74.28	74.28	74.28
v5	94.28	94.28	94.28	85	85.71	85.71	85.71	85.71	85.71	85.71	85	85	84.28	84.28	84.28	84.28
v6	93.57	93.57	92.85	83.57	83.57	84.28	83.57	82.85	82.85	82.85	82.85	82.85	83.57	83.57	83.57	83.57

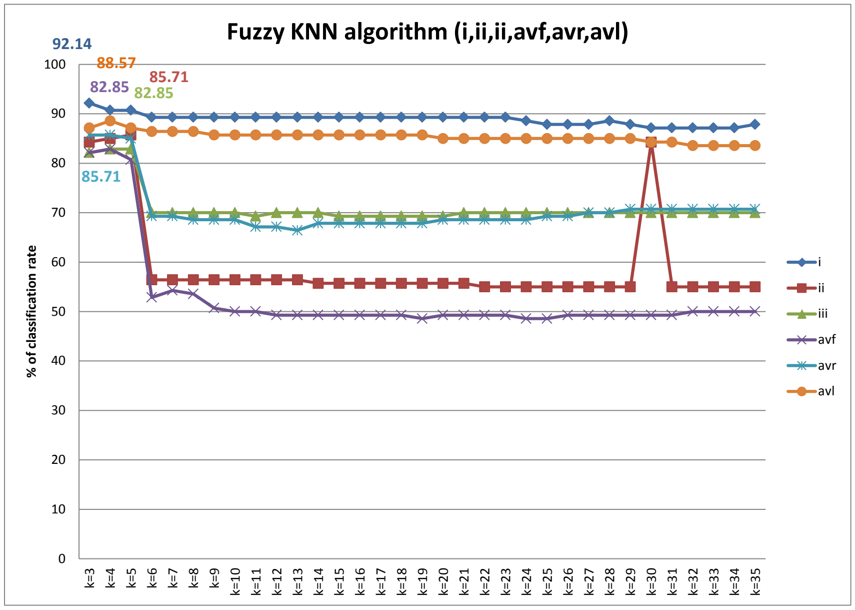


Figure 4.18 Comparative results of different values of *k* for Fuzzy KNN algorithm classifier for specific lead (*i, ii, iii, avf, avr, avl*)

Table 4.18 Results for testing of Fuzzy KNN algorithm

Lead	*K*=19	*K*=20	*K*=21	*K*=22	*K*=23	*K*=24	*K*=25	*K*=26	*K*=27	*K*=28	*K*=29	*K*=30	*K*=31	*K*=32	*K*=33	*K*=34	*K*=35
i	89.28	89.28	89.28	89.28	89.28	88.57	87.85	87.85	87.85	88.57	87.85	87.14	87.14	87.14	87.14	87.14	87.85
ii	55.71	55.71	55.71	55	55	55	55	55	55	55	55	84.28	55	55	55	55	55
iii	69.28	69.28	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70
avf	48.57	49.28	49.28	49.28	49.28	48.57	48.57	49.28	49.28	49.28	49.28	49.28	49.28	50	50	50	50
avr	67.85	68.57	68.57	68.57	68.57	68.57	69.28	69.28	70	70	70.71	70.71	70.71	70.71	70.71	70.71	70.71
avl	85.71	85	85	85	85	85	85	85	85	85	85	84.28	84.28	83.57	83.57	83.57	83.57
v1	70	70	70	70	70	70	69.28	69.28	69.28	69.28	69.28	70.71	70.71	70.71	70.71	70.71	70.71
v2	80.71	80.71	80.71	80.71	80.71	80.71	80.71	80.71	80.71	80.71	80.71	80.71	80.71	80.71	80.71	80.71	80.71
v3	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14
v4	74.28	74.28	75	75	75	75	74.28	74.28	74.28	74.28	74.28	74.28	74.28	73.57	73.57	73.57	73.57
v5	83.57	84.28	84.28	85	84.28	84.28	84.28	84.28	84.28	84.28	84.28	84.28	84.28	84.28	84.28	84.28	84.28
v6	83.57	83.57	83.57	83.57	83.57	83.57	83.57	83.57	83.57	83.57	84.28	84.28	84.28	84.28	84.28	84.28	84.28

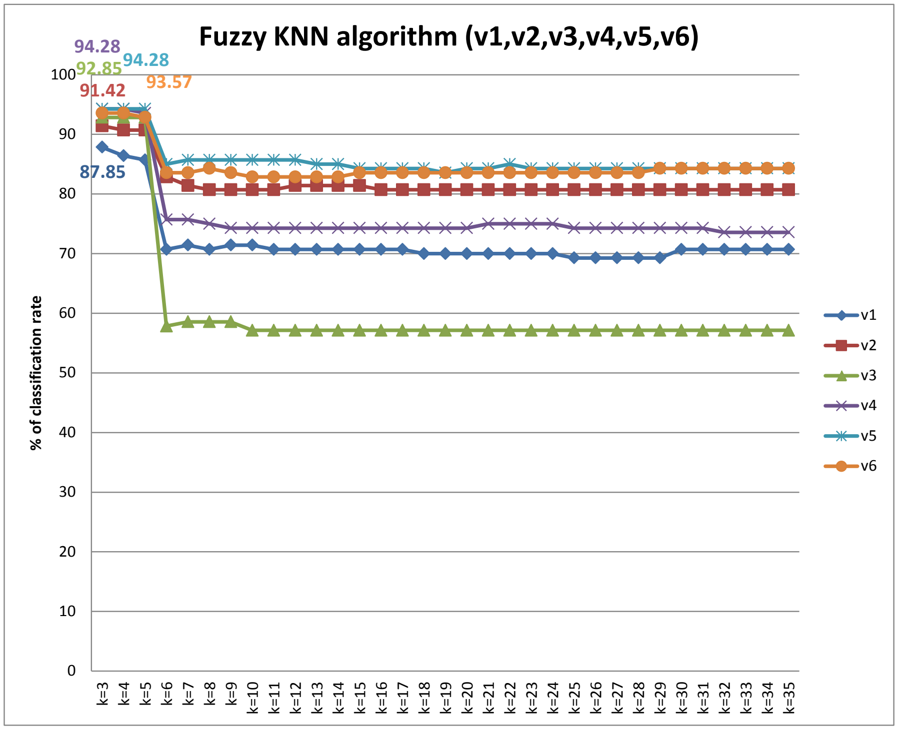


Figure 4.19 Comparative results of different values of *k* for Fuzzy KNN algorithm classifier for specific lead (*v1, v2, v3, v4, v5, v6*)

We perform different architectures for the MLP-SCG varying the hidden neurons and the count of hidden layers, and the results tending to improve with these specifications. In Table 4.19, 4.20 and 4.21, we can observe the results of the experiments to perform for the 12 leads used from 50 to 850 hidden neurons and 1 to 3 hidden layers respectively. In Figure 4.20 and 4.21 presented the comparative results of different values of hidden neurons and layers for MLP-SCG classifier for specific lead.

Table 4.19 Results for testing of MLP-SCG

Lead	*50 neurons*			*100 neurons*			*150 neurons*			*200 neurons*			*250 neurons*			*300 neurons*		
	1L	*2L*	*3L*	*1L*	*2L*	*3L*	*1L*	*2L*	*3L*	*1L*	*2L*	*3L*	*1L*	*2L*	*3L*	*1L*	*2L*	*3L*
i	55.71	60	9.28	70.71	81.42	80.71	72.14	82.14	69.28	77.85	84.28	88.57	85.71	85.71	85.71	82.85	86.42	86.42
ii	45	60.71	44.28	56.42	79.28	60.71	72.14	75.71	79.28	82.85	81.42	81.42	81.42	85	76.42	77.85	84.28	84.28
iii	55	64.28	52.85	67.85	74.28	61.42	75.71	69.28	77.85	80.71	85.71	85	80.71	87.85	84.28	85.71	85.71	87.85
avf	51.42	45	31.42	63.57	65	67.14	70	72.14	68.57	77.14	80.71	85.71	73.57	77.85	80.71	81.14	79.28	80
avr	67.85	62.85	13.57	80.71	78.57	77.85	68.57	84.28	83.57	84.28	85.71	85.71	85	86.42	85.71	86.42	85.71	86.42

avl	61.42	55.71	30	73.57	67.85	64.28	82.14	76.42	78.57	80.71	85.71	73.53	80.71	84.28	82.85	73.57	86.42	82.14
v1	63.57	53.57	55.71	75.71	77.14	70.71	73.57	79.28	70	86.42	90	83.57	85	89.28	87.85	86.42	90	88.57
v2	58.57	50.71	42.85	76.42	67.85	39.28	71.42	77.85	70.71	76.48	85	78.57	85	85	78.85	84.28	87.14	82.85
v3	64.28	52.85	35.71	63.57	59.28	58.57	81.42	86.42	83.57	77.14	85.71	83.57	87.14	88.57	66.42	84.28	91.42	88.57
v4	55.71	49.28	46.42	75	71.42	72.14	79.28	79.28	71.42	78.57	82.85	85	75	87.85	80	77.14	88.57	90.71
v5	63.57	12.85	57.14	75.71	82.14	77.85	74.28	83.57	85	71.42	87.14	77.85	90	87.85	85.71	82.85	88.57	87.14
v6	81.42	62.85	63.57	74.28	80	13.57	86.42	83.57	90.71	92.14	87.85	79.28	90.71	90	90.71	75	90.71	88.57

Table 4.20 Results for testing of MLP-SCG

Lead	*350 neurons*			*400 neurons*			*450 neurons*			*500 neurons*			*550 neurons*			*600 neurons*		
	1L	*2L*	*3L*	*1L*	*2L*	*3L*	*1L*	*2L*	*3L*	*1L*	*2L*	*3L*	*1L*	*2L*	*3L*	*1L*	*2L*	*3L*
i	85.71	88.57	88.57	84.28	87.85	88.57	87.85	87.85	90	89.28	87.85	87.14	90.71	87.85	85.71	87.85	88.57	88.57
ii	84.28	82.85	84.28	84.28	85	82.85	76.42	84.28	86.42	77.85	82.14	84.28	82.85	83.57	83.57	85.71	85.71	83.57
iii	85	87.14	86.42	78.57	87.85	87.14	85.71	87.14	88.57	85.71	89.28	87.85	87.14	89.28	90	84.28	87.85	88.57
avf	72.14	82.85	85	78.57	86.42	83.57	79.28	82.85	83.57	77.85	88.57	84.28	80.71	81.42	85	85	85.71	84.28
avr	85	87.14	86.42	86.42	87.14	87.14	83.57	88.57	87.14	82.85	87.14	88.57	85.71	87.14	87.14	85.71	87.14	86.42
avl	79.28	87.14	85.71	82.14	88.57	89.28	86.42	88.57	88.57	85.71	88.57	86.42	79.28	85.71	87.85	84.28	86.42	87.14
v1	88.57	90.71	89.28	88.57	87.85	83.57	87.85	90.71	86.42	84.28	91.42	87.85	90	90.71	87.85	87.85	87.85	92.14
v2	90.71	87.14	86.42	81.42	88.57	90.71	85.71	85.71	88.57	90	90.71	86.42	90	90	90	90.71	90	91.42
v3	88.57	91.42	92.85	91.42	90.71	92.14	86.42	92.14	91.42	90.71	92.14	91.42	88.57	91.42	91.42	87.14	92.14	92.14
v4	88.57	89.28	86.42	87.14	84.28	90.71	82.14	82.85	92.14	80	92.14	92.14	82.85	92.14	91.42	92.14	93.57	91.42
v5	85	87.85	88.57	85.71	90.71	87.14	88.57	86.42	87.85	87.14	90	89.28	87.14	88.57	86.42	87.85	87.14	87.85
v6	82.14	92.85	92.85	85	92.14	92.85	83.57	93.57	92.14	87.14	92.14	92.85	90.71	92.14	92.85	92.85	92.14	95

In Table 4.22, we can see the worst, best; standard deviation and average of 30 experiments for the 12-leads of the electrocardiograms included in PTB Diagnostic ECG database used MLP-SCG. In the performed experiments shown in lead i, we found the best result is used 550 hidden neurons and 1 hidden layer with 90.71% of classification rate. For the lead ii, we found out the best result is used 750 hidden neurons and 2 hidden layers obtained an 87.85% of classification rate. For the iii lead, we can see the best result is used 550 hidden neurons and 3 hidden layers with 90% of classification rate. For avf, avr and avl leads the best results are used 500 hidden neurons and 2 hidden layers, 850 hidden neurons and 2 hidden layers, 400 hidden neurons and 3 hidden layers with 88.57%, 89.28%, and 89.28% of classification rate respectively. For v1, v2 and v3 leads, we found out the best results obtained are used with 800 hidden neurons and 2 hidden layers, 850 hidden neurons and 3 hidden layers, 650 hidden neurons and 2 hidden layers with a 92.85%, 93.57% and 92.85% of classification rate. For the v4, v5 and v6 leads, we found out the best results are used with 750 hidden neurons and 1 hidden layer, 400 hidden neurons and 2 hidden layers, 600 hidden neurons and 3 hidden layers with a 94.28%, 90.71% and 95% of classification rate respectively. The leads that obtained the highest percent of classification rate were v1, v2, v3, v4 and v6. The leads that registered the lowest percent of classification rate were ii, avf, and avl.

Table 4.21 Results for testing of MLP-SCG

Lead	*650 neurons*			*700 neurons*			*750 neurons*			*800 neurons*			*850 neurons*		
	1L	*2L*	*3L*	*1L*	*2L*	*3L*	*1L*	*2L*	*3L*	*1L*	*2L*	*3L*	*1L*	*2L*	*3L*
I	89.28	87.14	87.85	86.42	88.57	90	87.85	85.71	87.14	87.14	87.85	88.57	87.14	87.14	87.85
Ii	84.28	85	85.71	86.42	87.14	87.14	83.57	87.85	85.71	85	87.14	85	84.28	86.42	86.42
Iii	83.57	88.57	88.57	77.85	87.85	89.28	75.71	90	90	90	89.28	90	80	88.57	87.85
Avf	80.71	87.14	82.14	82.85	85	85.71	82.14	85.71	85	81.42	85	83.57	82.85	86.42	87.14
Avr	77.14	87.85	87.14	86.42	87.85	87.85	76.42	86.42	86.42	87.85	86.42	88.57	87.55	89.28	87.85
Avl	87.85	87.14	87.14	80	87.14	86.42	78.57	86.42	86.42	86.42	87.85	87.85	85.71	87.14	87.85
v1	87.85	91.42	90	75.71	91.42	87.85	74.28	91.42	90	90	92.85	89.28	92.14	90	90
v2	88.57	90	88.57	89.28	90.71	90.71	80	90	92.14	90	90	90	91.42	90	93.57
v3	89.28	92.85	92.14	90.71	92.85	91.42	91.42	90.71	92.14	90.71	91.42	92.14	88.57	92.85	92.14
v4	72.85	92.14	92.14	92.85	93.57	92.14	94.28	90.71	91.42	81.42	92.14	93.57	90	92.85	94.28
v5	81.42	90.71	90.71	89.28	90	90	88.57	90.71	88.57	89.28	89.28	89.28	89.28	89.28	89.28
v6	92.14	92.14	93.57	90.57	92.85	92.85	91.42	92.85	92.85	93.57	91.42	91.42	93.57	93.57	92.85

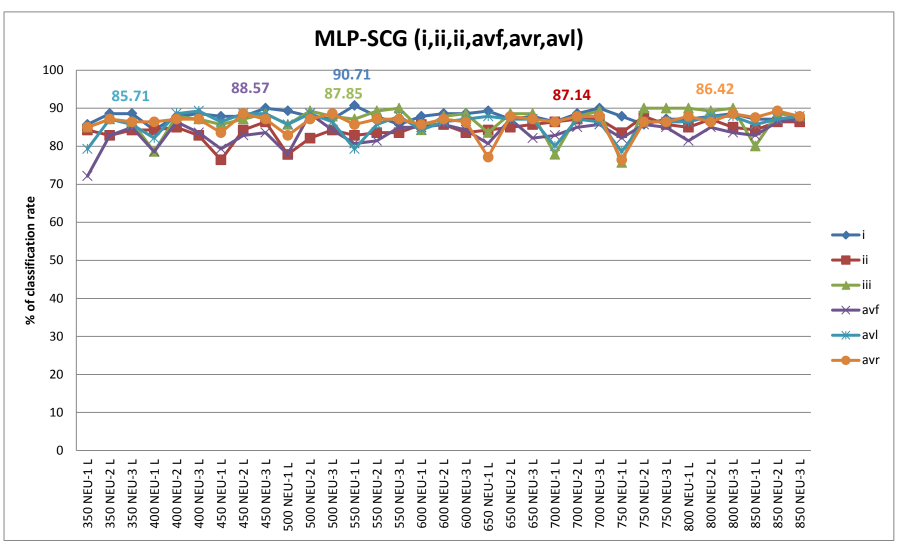


Figure 4.20 Comparative results of different values of *hidden neurons and layers* for MLP-SCG classifier for specific lead (*i, ii, iii, avf, avr, avl)*

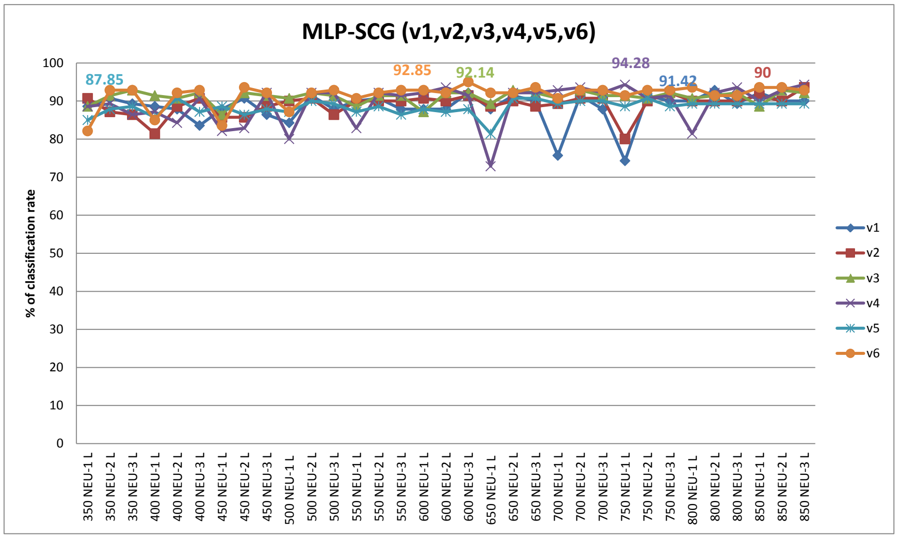


Figure 4.21 Comparative results of different values of *hidden neurons and layers* for MLP-SCG classifier for specific lead (*v1, v2, v3, v4, v5, v6*)

Table 4.22 Results for testing of the 33 experiments used by the Fuzzy KNN algorithm and MLP-SCG

Lead	*Classifier*	*Worst*	*Best*	*Std. Dev.*	*Average*
i	**Fuzzy KNN**	**87.14**	**92.14**	**1.129662393**	**88.8693939**
	MLP-SCG	85.71	90.71	1.330938949	87.8312121
ii	Fuzzy KNN	55	85.71	9.706412165	59.1306060
	MLP-SCG	**84.28**	**87.85**	**2.376372012**	**84.4542424**
iii	Fuzzy KNN	69.28	82.85	3.744379905	70.9939393
	MLP-SCG	**85**	**90**	**3.6937905**	**86.6848484**
avf	Fuzzy KNN	48.57	82.85	9.455843472	52.7448484
	MLP-SCG	**72.14**	**88.57**	**3.213136065**	**83.3293939**
avr	Fuzzy KNN	66.42	85.71	4.971103335	70.4506060
	MLP-SCG	**85**	**89.28**	**2.781743036**	**86.2845454**
avl	**Fuzzy KNN**	**83.57**	**88.57**	**1.086579112**	**85.3866666**
	MLP-SCG	79.28	89.28	2.866876963	85.9690909
v1	Fuzzy KNN	69.28	87.85	4.800434403	71.8572727
	MLP-SCG	**88.57**	**92.85**	**4.047687705**	**88.4151515**
v2	Fuzzy KNN	80.71	91.42	2.966763267	81.8130303
	MLP-SCG	**90.71**	**93.57**	**2.786976204**	**89.0663636**
v3	Fuzzy KNN	57.14	92.85	10.3851561	60.5378787
	MLP-SCG	**88.57**	**92.85**	**1.617754008**	**91.0769697**
v4	Fuzzy KNN	73.57	94.28	5.757801674	76.1863636
	MLP-SCG	**88.57**	**94.28**	**5.038525386**	**89.3245454**
v5	Fuzzy KNN	83.57	94.28	2.875146867	85.5148484
	MLP-SCG	**85**	**90.71**	**1.92122904**	**88.3290909**
v6	Fuzzy KNN	82.85	93.57	2.866518451	84.5203030
	MLP-SCG	**82.14**	**95**	**2.900961341**	**91.5972727**

We have selected the best representations for the expert modules based on Fuzzy KNN algorithm and MLP-SCG considering the results of the average and standard deviation of the 33 experiments performed presented in the Table 4.23, shown above.

4.2.4 Hybrid Intelligent System. Type-1 and Interval Type-2 Fuzzy Inference Systems.

Finally, we combined the output results for the twelve selected expert modules to be integrated through a type-1 and interval type-2 fuzzy inference systems and obtained an 86.42% and 97.14% respectively of global classification rate in the hybrid intelligent system based on ECG regions. In Table 4.23, we present more details in confusion matrix of interval type-2 fuzzy inference system that represents the global classification for the proposed hybrid intelligent system.

Table 4.23 Results for global interval type-2 fuzzy inference system, combining the outputs of the twelve expert modules in proposed hybrid intelligent system

Class	Myocardial Infarction	Cardiomyopathy	Bundle Branch Block	Dysrhythmia	Myocardial Hypertrophy	Valvular Heart Disease	Myocarditis	Miscellaneous	Healthy Controls
Myocardial Infarction	**18**	0	0	1	0	0	0	0	0
Cardiomyopathy	0	**21**	0	0	0	0	0	0	0
Bundle Branch Block	0	0	**23**	0	0	0	0	0	0
Dysrhythmia	0	0	0	**11**	0	0	0	0	0
Myocardial Hypertrophy	0	0	0	0	**15**	0	0	0	0
Valvular Heart Disease	0	0	0	0	0	**2**	0	0	0
Myocarditis	0	0	0	0	0	0	**9**	0	1
Miscellaneous	0	0	0	2	0	0	0	**18**	0
Healthy Controls	0	0	0	0	0	0	0	0	**19**

Classification rate= **97.14%**

.

4.2.5 Discussion of results

In the performed experiments, we found out the feature extraction with AR, Shannon entropy and wavelets of signals ECGs for the 12-lead captures the difference between the classes of the PTB database. We used classifiers based on Fuzzy KNN algorithm and MLP-SCG to form the expert modules in the hybrid intelligent system. We obtained better results using interval type-2 fuzzy inference system compare with type-1 fuzzy inference system to combine the outputs of the expert module classifiers. The strategy we have implemented in this model to represent the medical knowledge used by cardiologist physicians for ECG interpretation helped the proposed Hybrid Intelligent System to increase the overall classification rate. In future work, we will optimize the expert modules and fuzzy systems with genetic algorithms or other optimization method in order to improve the global classification in the proposed hybrid intelligent system.

4.3. A new variant of Fuzzy K-Nearest Neighbor using Interval Type-2 Fuzzy Logic.

The K-Nearest Neighbor algorithm (KNN) is a widely used method to solve different classification problems [67, 68, 69]. Several variants based on Fuzzy Logic have been proposed, such as Type-1 Fuzzy Sets, Type-2 Fuzzy Sets, Possibilistic Methods, Intuitionistic Fuzzy Sets, Fuzzy Rough Sets, and Preprocessing methods via data reduction, that consider the aspects of member, distance, voting, independence of k, preprocessing, and center based.

It is well known that in different applications, the implementation of type-2 fuzzy logic obtains a better result when compared to type-1 fuzzy logic [70, 38, 71]. We propose the idea of implementing an Interval Type-2 Mamdani Fuzzy Inference System to improve the performance of the original Fuzzy K-Nearest Neighbor algorithm offering a new alternative to use on complex classification problems.

We used the MIT-BIH arrhythmia database to validate the proposed method. This database contains 48 half hour excerpts of two channel ambulatory electrocardiograms recordings belonging to 47 patients. The heartbeats are segmented and preprocessed [4, 39, 56].

The new variant of the Fuzzy KNN algorithm using an Interval Type-2 Fuzzy Inference System, called IT2FISKNN, we consider it as a new variant of the fuzzy KNN algorithm, and instead of calculating the inverse of Euclidean distance to represent the membership degree of the unknown sample, firstly calculate the Euclidean, Hamming, cosine similarity and city block measures of the distances. The IT2FISKNN uses these measures of distances as inputs, and through the fuzzy rules define the final distance that will replace the Euclidean distance used in the original Fuzzy KNN algorithm.

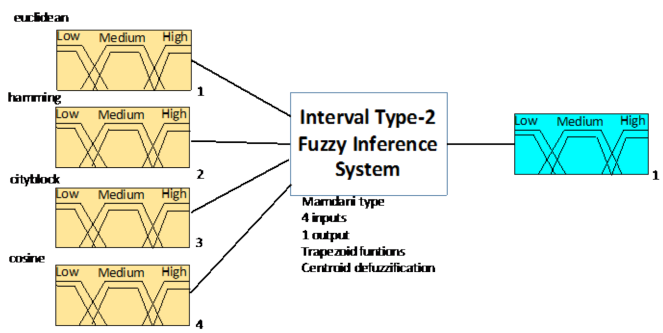


Figure 4.22 Interval Type-2 Fuzzy System

The IT2FISKNN uses 4 inputs with trapezoidal functions (Low, Medium, and High), 47 rules, and 1 output with trapezoidal functions (Low, Medium, and High), and centroid defuzzification method, see Figure 4.22. The rules for the IT2FISKNN are presented below. The output of the IT2FISKNN is used in the Fuzzy KNN algorithm. The rest of the steps of the Fuzzy KNN are executed as they are mentioned in the above section.

The parameters for the Euclidean distance input are:

- Low= [-5.267 6.267 1.29 1.29 1.269 0.8].
- Medium= [2.482 14.03 1.292 1.295 1.27 0.8].
- High= [10.24 21.76 1.293 1.3 1.272 0.8].

The parameters for the Hamming distance input are:

- Low= [0.8628 0.9372 0.0083 0.0083 0.00656 0.8].
- Medium= [0.9128 0.9872 0.0083 0.00835 0.00656 0.8].
- High= [0.9628 1.038 0.00835 0.0085 0.00648 0.8].

The parameters for the city block distance input are:

- Low= [-54.09 66.88 13.49 13.49 8.535 0.8].
- Medium= [27.21 148.2 13.5 13.6 8.536 0.8].
- High= [108.5 230.8 13.58 13.8 8.44 0.8].

The parameters for the similarity distance input are:

- Low= [-0.4622 0.4621 0.1031 0.1031 0.05216 0.8].
- Medium= [0.159 1.083 0.1031 0.1038 0.05216 0.8].
- High= [0.78 1.715 0.1037 0.1055 0.0516 0.8].

The parameters for the output are:

- Low= [-5.267 6.267 1.29 1.29 1.269 0.8].
- Medium= [2.482 14.03 1.292 1.295 1.27 0.8].
- High= [10.24 21.76 1.293 1.3 1.272 0.8].

The fuzzy if-then rules of IT2FISKNN are listed as follows:

1. IF (euclidean is LOW) and (hamming is LOW) and (cityBlock is LOW) and (similarity is LOW) THEN (distance is LOW)

2. IF (euclidean is MEDIUM) and (hamming is MEDIUM) and (cityBlock is MEDIUM) and (similarity is MEDIUM) THEN (distance is MEDIUM)

3. IF (euclidean is HIGH) and (hamming is HIGH) and (cityBlock is HIGH) and (similarity is HIGH) THEN (distance is HIGH)

4. IF (euclidean is LOW) and (hamming is LOW) and (cityBlock is LOW) and (similarity is MEDIUM) THEN (distance is LOW)

5. IF (euclidean is LOW) and (hamming is LOW) and (cityBlock is LOW) and (similarity is HIGH) THEN (distance is LOW)

6. IF (euclidean is LOW) and (hamming is LOW) and (cityBlock is MEDIUM) and (similarity is LOW) THEN (distance is LOW)

7. IF (euclidean is LOW) and (hamming is LOW) and (cityBlock is HIGH) and (similarity is LOW) THEN (distance is LOW)

8. IF (euclidean is LOW) and (hamming is MEDIUM) and (cityBlock is LOW) and (similarity is LOW) THEN (distance is LOW)

9. IF (euclidean is LOW) and (hamming is HIGH) and (cityBlock is LOW) and (similarity is LOW) THEN (distance is LOW)

10. IF (euclidean is MEDIUM) and (hamming is LOW) and (cityBlock is LOW) and (similarity is LOW) THEN (distance is LOW)

11. IF (euclidean is HIGH) and (hamming is LOW) and (cityBlock is LOW) and (similarity is LOW) THEN (distance is LOW)

12. IF (euclidean is LOW) and (hamming is LOW) and (cityBlock is MEDIUM) and (similarity is MEDIUM) THEN (distance is MEDIUM)

13. IF (euclidean is LOW) and (hamming is MEDIUM) and (cityBlock is LOW) and (similarity is MEDIUM) THEN (distance is MEDIUM)

14. IF (euclidean is MEDIUM) and (hamming is LOW) and (cityBlock is LOW) and (similarity is MEDIUM) THEN (distance is MEDIUM)

15. IF (euclidean is LOW) and (hamming is MEDIUM) and (cityBlock is MEDIUM) and (similarity is LOW) THEN (distance is MEDIUM)

16. IF (euclidean is MEDIUM) and (hamming is LOW) and (cityBlock is MEDIUM) and (similarity is LOW) THEN (distance is MEDIUM)

17. IF (euclidean is MEDIUM) and (hamming is LOW) and (cityBlock is LOW) and (similarity is LOW) THEN (distance is MEDIUM)

18. IF (euclidean is LOW) and (hamming is LOW) and (cityBlock is HIGH) and (similarity is HIGH) THEN (distance is HIGH)

19. IF (euclidean is LOW) and (hamming is HIGH) and (cityBlock is LOW) and (similarity is HIGH) THEN (distance is HIGH)

20. IF (euclidean is MEDIUM) and (hamming is MEDIUM) and (cityBlock is HIGH) and (similarity is MEDIUM) THEN (distance is MEDIUM)

21. IF (euclidean is HIGH) and (hamming is HIGH) and (cityBlock is LOW) and (similarity is HIGH) THEN (distance is HIGH)

22. IF (euclidean is LOW) and (hamming is HIGH) and (cityBlock is LOW) and (similarity is HIGH) THEN (distance is HIGH)

23. IF (euclidean is HIGH) and (hamming is HIGH) and (cityBlock is LOW) and (similarity is HIGH) THEN (distance is HIGH)

24. IF (euclidean is MEDIUM) and (hamming is MEDIUM) and (cityBlock is MEDIUM) and (similarity is LOW) THEN (distance is HIGH)

25. IF (euclidean is MEDIUM) and (hamming is MEDIUM) and (cityBlock is MEDIUM) and (similarity is HIGH) THEN (distance is MEDIUM)

26. IF (euclidean is HIGH) and (hamming is HIGH) and (cityBlock is HIGH) and (similarity is MEDIUM) THEN (distance is HIGH)

27. IF (euclidean is MEDIUM) and (hamming is MEDIUM) and (cityBlock is LOW) and (similarity is MEDIUM) THEN (distance is MEDIUM)

28. IF (euclidean is HIGH) and (hamming is HIGH) and (cityBlock is LOW) and (similarity is HIGH) THEN (distance is HIGH)

29. IF (euclidean is MEDIUM) and (hamming is MEDIUM) and (cityBlock is LOW) and (similarity is MEDIUM) THEN (distance is MEDIUM)

30. IF (euclidean is HIGH) and (hamming is HIGH) and (cityBlock is LOW) and (similarity is HIGH) THEN (distance is HIGH)

31. IF (euclidean is MEDIUM) and (hamming is MEDIUM) and (cityBlock is HIGH) and (similarity is MEDIUM) THEN (distance is MEDIUM)

32. IF (euclidean is HIGH) and (hamming is HIGH) and (cityBlock is LOW) and (similarity is HIGH) THEN (distance is HIGH)

33. IF (euclidean is MEDIUM) and (hamming is LOW) and (cityBlock is MEDIUM) and (similarity is MEDIUM) THEN (distance is MEDIUM)

34. IF (euclidean is HIGH) and (hamming is LOW) and (cityBlock is HIGH) and (similarity is HIGH) THEN (distance is HIGH)

35. IF (euclidean is MEDIUM) and (hamming is HIGH) and (cityBlock is MEDIUM) and (similarity is MEDIUM) THEN (distance is MEDIUM)

36. IF (euclidean is HIGH) and (hamming is LOW) and (cityBlock is HIGH) and (similarity is HIGH) THEN (distance is HIGH)

37. IF (euclidean is LOW) and (hamming is MEDIUM) and (cityBlock is MEDIUM) and (similarity is MEDIUM) THEN (distance is MEDIUM)

38. IF (euclidean is MEDIUM) and (hamming is HIGH) and (cityBlock is HIGH) and (similarity is HIGH) THEN (distance is HIGH)

39. IF (euclidean is MEDIUM) and (hamming is MEDIUM) and (cityBlock is HIGH) and (similarity is HIGH) THEN (distance is HIGH)

40. IF (euclidean is MEDIUM) and (hamming is HIGH) and (cityBlock is MEDIUM) and (similarity is HIGH) THEN (distance is HIGH)

41. IF (euclidean is HIGH) and (hamming is MEDIUM) and (cityBlock is HIGH) and (similarity is MEDIUM) THEN (distance is HIGH)

42. IF (euclidean is LOW) and (hamming is HIGH) and (cityBlock is HIGH) and (similarity is LOW) THEN (distance is HIGH)

43. IF (euclidean is MEDIUM) and (hamming is HIGH) and (cityBlock is MEDIUM) and (similarity is HIGH) THEN (distance is HIGH)

44. IF (euclidean is HIGH) and (hamming is LOW) and (cityBlock is HIGH) and (similarity is LOW) THEN (distance is HIGH)

45. IF (euclidean is LOW) and (hamming is MEDIUM) and (cityBlock is LOW) and (similarity is MEDIUM) THEN (distance is MEDIUM)

46. IF (euclidean is HIGH) and (hamming is MEDIUM) and (cityBlock is HIGH) and (similarity is MEDIUM) THEN (distance is HIGH)

47. IF (euclidean is HIGH) and (hamming is HIGH) and (cityBlock is MEDIUM) and (similarity is MEDIUM) THEN (distance is HIGH).

The fuzzy if-then rules were designed thinking about combining the measures of the distances from the different perspectives provided by each metric with respect to the vector unknown classification. For example, if the Euclidean, Hamming, city block and similarity measures are the same perspective, then the consequent will be the same perspective. If the three measures have the same perspective, then will be that perspective as the consequent. In others words, we consider the majority perspective of the measures to represent the consequent. We have not considered all the possible options of the fuzzy rules that can be created, instead of this, the most important fuzzy rules are considered.

4.3.1. Experiments and Results.

In this section, we describe details for the MIT-BIH Arrhythmia database, experimental setup, and simulation results.

For this work, we used the MIT-BIH Arrhythmia Database to validate the proposed method. A set of samples of heartbeat records from this database are extracted. The heartbeats are segmented and transformed in the preprocessing stage [4]. The samples of heartbeats are normal beat (NB), left bundle branch block beat (LBBB), right bundle branch block beat (RBBB), premature ventricular contraction beat (PVC), fusion paced and normal beat (FPN), atrial premature beat (APB), aberrated atrial premature beat (AAPB), fusion of ventricular and normal beat (FVNB), ventricular escape beat (VEB) and paced beat (PB). In Figure 4.23, some examples of the MIT-BIH arrhythmia database are presented. The MIT-BIH arrhythmia database contains 48 half-hour excerpts of two channel ambulatory electrocardiogram recordings belonging to 47 patients, 22 women and 25 men aged between 23 and 89 years. In Table 4.24, results of cross validation using 3 nearest neighbors, we obtained best classification rate in fold 8 with a 93% for ITSFISKNN compared with 91% for Fuzzy KNN. In Table 4.25, results of cross validation are listed with 4 nearest neighbors, we obtained the best classification rate in fold 1 with a 93% for IT2FISKNN compared with a 92% for Fuzzy KNN. In Table 4.26, results of cross validation are listed with 4 nearest neighbors, we obtained best classification rate in fold 3 with a 91% for IT2FISKNN.

Table 4.24 Results for Cross Validation K Fold 10, K=3

	K=3	
	Fuzzy KNN	***IT2FISKNN***
Fold 1	91	89
Fold 2	95	94
Fold 3	90	90
Fold 4	90	86
Fold 5	86	85
Fold 6	90	88
Fold 7	92	89
Fold 8	91	93
Fold 9	87	85
Fold 10	90	87
Classification rate	90.2	88.6

Table 4.25 Results for Cross Validation K Fold 10, K=4

	K=4	
	Fuzzy KNN	***IT2FISKNN***
Fold 1	92	93
Fold 2	95	95
Fold 3	91	91
Fold 4	90	87
Fold 5	87	85
Fold 6	89	89
Fold 7	92	91
Fold 8	93	93
Fold 9	87	87
Fold 10	93	90
Classification rate	90.9	90.1

Table 4.26 Results for Cross Validation K Fold 10, K=5

	K=5	
	Fuzzy KNN	***IT2FISKNN***
Fold 1	91	89
Fold 2	95	93
Fold 3	90	91
Fold 4	86	83
Fold 5	86	86
Fold 6	89	89
Fold 7	92	90
Fold 8	92	90
Fold 9	86	83
Fold 10	93	89
Classification rate	90	88.3

Finally, the confusion matrix of the Fuzzy KNN algorithm and IT2FISKNN are presented in Tables 4.27 and 4.28. The results are organized by class in both confusion matrices. We have decided to show only the results with 4 nearest neighbors because it is the best result so far. We can observe the classification rate of LBBB is slightly better for IT2FISKNN than Fuzzy KNN. In the rest of the classes, the classification rates are tied and in others, the Fuzzy KNN algorithm is better than IT2FISKNN.

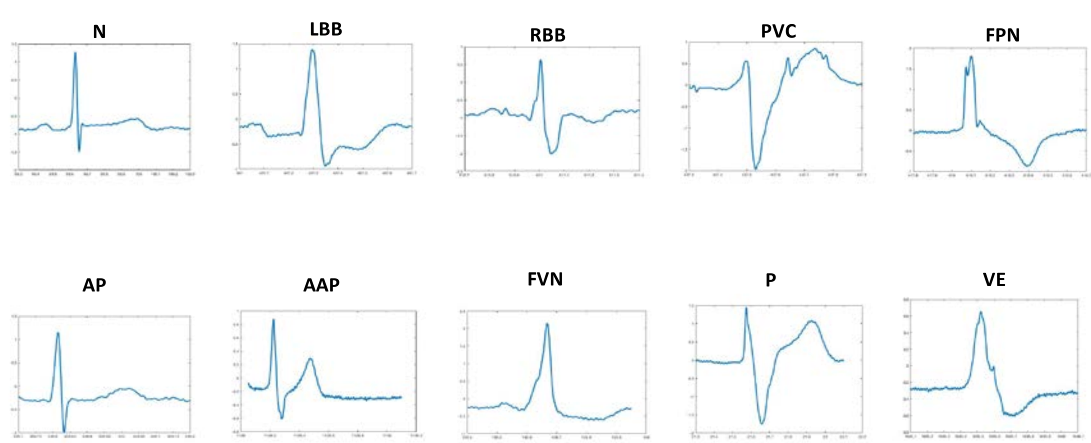


Figure 4.23 Normal beat and arrhythmias of the MIT-BIH Arrhythmia Database

Table 4.27 Confusion Matrix of Fuzzy KNN Algorithm, K=4

	K=4									
	NB	**LBBB**	**RBBB**	**PVC**	**FPN**	**APB**	**AAPB**	**FVNB**	**VEB**	**PB**
NB	**97**	0	0	0	0	1	0	2	0	0
LBBB	0	**91**	1	0	0	1	2	3	2	0
RBBB	1	2	**95**	0	1	0	1	0	0	0
PVC	0	3	0	**85**	2	0	1	2	1	6
FPN	0	0	1	2	**87**	0	1	6	0	3
APB	0	1	0	0	1	**93**	5	0	0	0
AAPB	0	3	0	1	0	11	**80**	1	4	0
FVNB	1	4	1	1	3	1	0	**89**	0	0
VEB	0	1	0	0	0	0	5	0	**94**	0
PB	0	0	0	0	2	0	0	0	0	**98**

Table 4.28 Confusion Matrix of IT2SFISKNN, K=4

	K=4									
	NB	**LBBB**	**RBBB**	**PVC**	**FPN**	**APB**	**AAPB**	**FVNB**	**VEB**	**PB**
NB	**96**	0	0	0	0	1	0	3	0	0
LBBB	0	**92**	0	0	0	2	1	3	2	0
RBBB	2	2	**94**	0	1	0	1	0	0	0
PVC	0	3	0	**84**	2	1	1	2	1	6
FPN	0	1	2	1	**86**	0	1	5	0	4
APB	0	0	0	0	1	**93**	6	0	0	0
AAPB	0	4	0	0	0	14	**78**	1	3	0
FVNB	3	4	1	1	4	1	0	**86**	0	0
VEB	0	1	0	0	0	0	5	0	**94**	0
PB	0	0	0	0	2	0	0	0	0	**98**

4.3.2. Discussion of results

In this work, we have proposed a new variant of the Fuzzy K-Nearest Neighbor algorithm using Type-2 Fuzzy Logic and we found out the possibility to use an interval type-2 fuzzy inference system and several measures to calculate the distances between the neighbors and the vectors to classify. In the performed experiments using cross validation some folds with better classification rate than the Fuzzy KNN algorithm. Although until now, we have not obtained a better global classification rate using IT2FISKNN that exceeds the Fuzzy KNN, 90.1% and 90.9% of classification rate, respectively. We continue to perform experiments and making changes to the Interval type-2 fuzzy inference system to improve the results in the classification rate.

Chapter **5**

Conclusions and Future works

In the results presented of the first study case in the model for 2-lead cardiac arrhythmia classification with respect to the performed experiments, we conclude that we have obtained a good classification rate with the classifiers in the two basic module units, and even more combining their outputs using fuzzy logic we have obtained even better results; in this part of the hybrid system, we used two options to combine the results of the classifiers in the basic module units and we have presented a comparison between type-1 FIS and IT2FIS with hypothesis testing. We obtained the best results using an IT2FIS than type-1 FIS in basic module unit 1. We improved the classification rate in the basic module unit with 10-fold cross validation using a type-2 fuzzy system. In the comparison using hypothesis testing, although the current results do not provide sufficient statistical evidence to reject the null hypothesis, we find out that using type-2 fuzzy logic, we can achieve better results, and we can still improve the interval type-2 fuzzy system with optimization methods. Referring to the basic module unit 2, we obtained better results used type-2 fuzzy system comparing with type-1 FIS combining the results of the classifiers above mentioned.

We used two basic module units in the hybrid model, where each one was trained with electrode signals of the MIT-BIH arrhythmia database and we combined the output results of the two basic module units using a type-1 FIS, the results show the improvement of the global classification rate in the hybrid model. We found out that some samples of heartbeats were misclassified by one basic module unit. However, when combining the outputs with 2-lead using two basic module units some samples with misclassification were fixed and classified correctly by the other basic module unit using a fuzzy system, and in this form we have obtained better performance and classification rate using a hybrid model to resolve cardiac arrhythmia classification problem.

Based on the obtained results, we found out that IT2FIS from Unit 1 achievement obtain the same classification rate than the global hybrid model used type-1 FIS, but also we have included an IT2FIS to use in the global hybrid model. Therefore, we can show that global IT2FIS provided higher accuracy combining type-1 FIS. In other words, implementing type-2 instead of type-1 fuzzy logic improves overall accuracy of the proposed hybrid model.

Some advantages about the proposed hybrid model are that we can combine several computational intelligence methods using fuzzy systems to increase the overall accuracy of the hybrid model. We can separate into basic module units of expert modules for the electrode signal or lead of an ECG to consider the different perspectives that offers each lead and helps the classification of cardiac arrhythmias. One disadvantage that we can mention with respect to the proposed hybrid model relates to the total number of fuzzy rules for the fuzzy systems, and for this reason is necessary to optimize, for example, with genetic algorithms or other technique in order to reduce the total of fuzzy rules in the fuzzy systems.

In the performed experiments of the second study case, for the model of 12-leads cardiac arrhythmia classification we found out the feature extraction with AR, Shannon entropy and wavelets of signals ECGs for the 12-lead captures the difference between the classes of the PTB database. We used classifiers based on the Fuzzy KNN algorithm and MLP-SCG to form the expert modules in the hybrid intelligent system. We obtained better results used interval type-2 fuzzy inference systems compare type-1 fuzzy inference systems to combine the outputs of the expert module classifiers. The strategy we have implemented in this model to represent the medical knowledge used by cardiologist physicians for ECG interpretation helped the proposed Hybrid Intelligent System to increase the overall classification rate. In future works, we will optimize the expert modules and fuzzy systems with genetic algorithms or other optimization methods in order to improve the global classification in the proposed hybrid intelligent system.

References

[1] Amezcua, J., & Melin, P. (2014). A modular LVQ neural network with fuzzy response integration for arrhythmia classification, IEEE Conference on Norbert Wiener in the 21st Century, Boston, USA.

[2] Bishop, C.M. (1996). Neural network for pattern recognition, Oxford, U.K., Clarendon Press.

[3] Bousseljot, R., Kreiseler, D., & Nutzung, A., (1995). der EKG-Signaldatenbank CARDIODAT der PTB uber das Internet. Biomedizinische Technik, Band 40, Erganzungsband 1 S 317.

[4] Castillo, O., Melin, P., Ramirez, E., & Soria, J. (2012). Hybrid intelligent system for cardiac arrhythmia classification with fuzzy K-Nearest Neighbors and neural networks combined with a fuzzy system, Expert system with applications, ElSevier, 39, 2947-2955.

[5] Ceylan, R., Ozbay, Y., & Karlik, B. (2009). A novel approach for classification of ECG arrhythmias: type-2 fuzzy clustering neural network, Expert Systems with Applications, ElSevier, 36, 6721-6727.

[6] Chua, T.W., & Tan, W.W. (2009). Interval type-2 fuzzy system for ECG Arrhythmia classification, Fuzzy systems in Bioinformatics and computational biology, ISBN 978-3-540-89968-6. Springer, 242, 297-314.

[7] Conference on Biomedical Engineering and Sciences (IECBES), Kuala Lumpur, Malaysia.

[8] De Chazal, P., O'Dwyer, M., Reilly, R. (2004), Automatic classification of heartbeats using ECG morphology and heartbeat interval features, IEEE Trans. Biomed. Eng., 51, 1196-1206.

[9] Edward, J.S., Ramu, P., & Swaminathan, R. (2016). Imperceptibility-Robustness tradeoff studies for ECG steganography using Continuous Ant Colony Optimization. Experts Systems with Applications, ElSevier, 49, 123-135.

[10] Elhaj, F.A., Salim, N., Harris, A.R., Swee, T.T., & Ahmed, T. (2016). Arrhytmia recognition and classification using combined linear and nonlinear feactures of ECG signals, Computer methods and programs in Biomedicine, Elsevier, 127, 52-63.

[11] Gacek, A., & Witold, P. (2012). Classification and Interpretation, a comprehensive framework of computational intelligence, ISBN 978-0-85729-867-6, Springer, 47-77.

[12] Gaetano, D., Panunzi, S., Risi, A., & Sciandrone, M. (2009). A patient adaptable ECG beat classifier based on neural networks, Applied mathematics and computation, Elsevier, 213, 243-249.

[13] Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P., Mark, R., Mietus, J., Moody, G., Peng, C., & Stanley, H. (2003), PhysioBank, PhysioToolkit, and Phisionet: Components of a New Research Resource for Complex Physiologic Signals, Circulation, 101(23):e215-e220.

[14] Hampton, J.R., & Adlam, D. (2013). The ECG in Practice, sixth edition, ISBN 978-0-7020-4643-8, Churchill Livingstone, Elsevier.

[15] Homaeinezhad, M.R., Atyabi, S.A., Tavakkoli, E., Toosi, H.N., Ghaffari, A., & Ebrahimpour, R. (2012). ECG arrhythmia recognition via a neuro-SVM-KNN hybrid classifier with virtual QRS image-based geometrical features, Expert Systems with Applications, Elsevier, 39, 2047-2058.

[16] Hu, Y., Palreddy, S., Tompkins, W. (1997), A patient-adaptable ECG beat classifier using mixture of experts approach, Biomed. Eng. IEEE Trans., 44, 891-900.

[17] Ince, T., Kiranyaz, S., Gabbouj, M. (2009), A generic and roboust system for automated patient-specific classification of ECG signals, Biomed. Eng. IEEE Trans., 56, 1415-1426.

[18] Jadhav, S.M., Nalbalwar, S.L., & Ghatol, A.A. (2010). ECG arrhythmia classification using modular neural network model, 2010 IEEE EMBS

[19] Javadi, M., Asgnar, A., Sajedin, A., & Ebrahimpour, R. (2013). Classification of ECG arrhythmia by a modular neural network based on Mixture of Experts and Negatively Correlated Learning, Biomedical Signal Processing and Control, Elsevier, 8, 289-296.

[20] Jayasinghe, R. (2012). , ECG Workbook, ISBN 978-0-7295-4109-1, Churchill Livingstone, Elsevier.

[21] Jovic, N., & Bogunovic (2012). Evaluating and comparing performance of feature combinations of heart rate variability measures for cardiac rhythm classification, Biomedical Signal Processing and Control, Elsevier, 7, 245-255.

[22] Kalaf, A.F., Owis M.L., & Yassine, I.A. (2015). A novel technique for cardiac arrhythmia classification usign spectral correlation and support vector machines, Expert systems with applications, Elsevier, 42, 8361-8368.

[23] Keller, J.M., Gray, M.R., & Givens, J.A. (1985). A fuzzy K-Nearest neighbor algorithm, IEEE Transactions on fuzzy systems, Man and cybernetics, 15, 580-585.

[24] Kitin, A.D. (1949)., Obsercations on the anatomy of the atrioventricular bundle (bundle of His) and the question of other muscular atrioventricular connections in normal human hearts. American Heart Journal, 37, 847-67.

[25] Koruek, M., & Dogan, B. (2010). ECG beat classification using particle swarm optimization and radial basis function neural network, Expert Systems with Applications, ElSevier, 37, 7563-7569.

[26] Lin, C., Yang, C. (2014), Heartbeat classification using normalized RR intervals and morphological features, Math. Problem Eng., 2014, 1-11.

[27] Llamedo, M., Martinez, J. (2011), Heart classification using feature selection driven by database generalization criteria, IEEE Trans. Biomed. Eng., 58, 616-625.

[28] Luis, E.J., Nunes, T.M., & Alburquerque, V.C. (2013). ECG arrhythmia classification based on optinum-path forest, Expert Systems with Applications, ElSevier, 40, 3561-3573.

[29] Luz, E.J., Schwartz, W.R., Camara, G.C., & Menotti, D. (2015). ECG-based heartbeat classification for arrhythmia detection: a survey, Computer methods and programs in biomedecine, Elsevier, 127, 144-164.
[30] Martindale, J.L., Brown, D.F.M. (2017). A visual guide to ECG interpretation, second edition, ISBN 978-1-4963-2153-4. Wolters Kluwer.
[31] Martis, R., Acharya, U., Mandana, K., Ray, A., Chakraborty, C. (2013), Cardiac decision making using order spectra, Biomed. Signal Process. Control, 8, 193-203.
[32] Martis, R.J., Acharya, U.R., Mandana, K.M., Ray, A.K., & Chakraborty, C. (2012). Application of principal component analysis to ECG signals for automated diagnosis of cardiac health, Expert Systems with Applications, ElSevier, 39, 11792-11800.
[33] Martis, R.J., Achayra, U.R., & Min, L.C. (2013). ECG beat classification using PCA, LDA, ICA, and Discrte Wavelet Transform, Biomedical Signal Processing and Control, Elsevier, 8, 437-448.
[34] Martis, R.J., Achayra, U.R., Prasad, H., & Chua, K.C. (2013). Application of higher order statistics for atrial arrhythmia classification, Biomedical Signal Processing and Control, Elsevier, 8, 888-900.
[35] Megat, M.S.A., Jahidin, A.H., & Norall, A.N. (2012). Hybrid multilayered perceptron network classification of bundle branch blocks, Biomedical engineering Icobe, IEEE 2012 International conference, Penang, Malaysia.
[36] Melin, P., Ramirez, E., & Prado-Arechiga, G. (2018). A new variant of Fuzzy K-Nearest Neighbor using Interval Type-2 Fuzzy Logic, FUZZ-IEEE, 1-7.
[37] Melin, P., & Castillo, O. (2013). A review of the applications of type-2 fuzzy logic in classification and pattern recognition, Expert systems with applications, ElSevier, 40, 5413-5423.
[38] Melin, P., & Castillo, O. (2013). A review on the applications of type-2 fuzzy logic in classification and pattern recognition, Expert Systems with Applications. ElSevier, 40, 5413-5423.
[39] Melin, P., Amezcua, J., Valdez, F., & Castillo, O. (2014). A new neural network model based on the LVQ algorithm for multi-class classification of arrhythmias, Information Sciences, ElSevier, 279, 483-497.
[40] Melin, P., Prado, G., Miramontes, I., & Medina, M. (2016). A hybrid intelligent model based on modular neural network and fuzzy logic for hypertension risk diagnosis, Journal of Hypertension, 34, e153.
[41] Melin, P., Ramirez, E., & Prado, G. (2017). Cardiac arrhythmia classification using computational intelligence: neural networks and fuzzy logic techniques. European Heart Journal, OXFORD academic, 38, P6388.
[42] Merone, M., Soda, P., & Sansone, M. (2017). ECG databases for biometric systems: A systematic review, Expert Systems with Applications, ElSevier, 67, 189-202.
[43] Moavenian, M., & Khorrami, H. (2010). A qualitative comparasion of Artifical Neural Networks and Support Vector Machines in ECG arrhythmias classification, Expert Systems with Applications, ElSevier, 37, 3088-3093.

[44] Nazmy, T.M., EL-Messiry, H., & AL-Bokhity, B. (2010). Classification of Cardiac Arrhythmia based on Hybrid System, International Journal of Computer Applications, 2, 0975-8887.

[45] Nien, S., & Chou, K. (2009). Selection of significant independent components for ECG beat classification, Expert Systems with Applications, ElSevier, 36, 2088-2096.

[46] Osowki, S., Markiewicz, T., & Hoal, L.T. (2018). Recognition and classification systems of arrhythmia using ensemble of neural networks, Mearurement, Elsevier, 41, 610-617.

[47] Osowksi, K., Siwek, K., & Siroic, R. (2011). Neural system for heartbeats recognition using genetically integrated ensemble of classifiers, Computers in biology and medicine, Elsevier, 41, 173-180.

[48] Ozbay, Y., & Tezel, G. (2010). A new method for classification of ECG arrhythmias using neural network with adaptive activation function, Digital Signal Processing, Elsevier, 20, 1040-1049.

[49] Ozbay, Y., Ceylan, R., & Karlik, B. (2005). A fuzzy clustering neural network architecture for classification of ECG arrhythmias, Computers in Biology and Medicine, Elsevier, 36, 376-388.

[50] Plawiak, P. (2018). Novel genetic ensembles of classifiers applied to myocardium dysfunction recognition based on ECG signals, Swarm and Evolutionary Computation, 39, 192-208.

[51] Plawiak, P. (2018). Novel methodology of cardiac health recognition based on ECG signals and evolutionary-neuronal system, Expert Systems with Applications, Elsevier, 92, 334-349.

[52] Rahhal, M.M., Bazi, Y., Alhichri, H., Alajlan, N., Melgani, F., & Yager, R.R. (2016). Deep learning approach for active classification of electrocardiogram signals, Information Sciences, Elsevier, 345, 340-354.

[53] Ramirez, E., Melin, P., Prado-Arechiga, G. (2019). Hybrid model based on neural networks, type-1 and type-2 fuzzy systems for 2-lead cardiac arrhythmia classification, Expert Systems with Applications, ElSevier, 126, 295-307.

[54] Ramirez, E., Melin, P., Prado-Arechiga, G. (2020). Hybrid Model Based on Neural Networks and Fuzzy Logic for 2-Lead Cardiac Arrhythmia Classification, Hybrid Intelligent Systems in Control, Pattern Recognition and Medicine, Springer, 193-217.

[55] Ramirez, E., Melin, P., Prado-Arechiga, G. (2020). A Modular Neural Network Approach for Cardiac Arrhythmia Classification, Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms, Springer, 211-223.

[56] Ramirez, E., Castillo, O., Soria, J. (2010). Hybrid system for cardiac arrhythmia classification with Fuzzy K-Nearest Neighbors and neural networks combined by a fuzzy inference system. Soft Computing for Recognition Based on Biometrics. ISBN 978-3-642-15110-1, Springer, 312, 37-53.

[57] Shao, Y.E., Hou, C.D., & Chiu, C.C. (2014). Hybrid intelligent modeling schemes for heart disease classification, Applied Soft Computing, Elsevier, 14, 47-52.

[58] Soria, M., Martinez, J. (2009), Analisys of multidomain features for ECG classification, Comput. Cardiol, 561-564.

[59] Sufi, F., Khalil, I., & Naser, A. A clustering based system for instant detection of cardiac abnormalities from compressed ECG, Expert Systems with Applications, ElSevier, 38, 4705-4713.

[60] Tan, W.W., FOO, C.L., & Chua, T. (2007). Type-2 fuzzy system for ECG arrhythmic classification. 2007 IEEE International Fuzzy Systems Conference, FYZZ-IEEE, London, UK.

[61] Wang, J.S., Chiang, W.C., Hsu, Y.L., & Yang, Y.T. (2013). ECG arrhythmia classification using a probabilistic neural network with a feature reduction method, Neurocomputing, Elsevier, 116, 38-45.

[62] Wozniak, M., Grana, M. & Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems, Information Fusion, Elsevier, 16, 3-1.

[63] Ye, C., Kumar, B., Coimbra, M. (2012), Combining general multi-class and specific two-class classifiers for improved customized ECG heartbeat classification, International Conference on Pattern Recognition (ICPR), 2428-2431.

[64] Yildirim, O., Plawiak, P., Tan, R., & Acharya, U. (2018). Arrhythmia detection using deep convolutional neural network with long duration ECG signals, 102, 411-420.

[65] Yu, S., & Chou, K. (2008). Integration of independent component analysis and neural networks for ECG beat classification, Expert Systems with Applications, ElSevier, 34, 2841-2846.

[66] Zopounidis, C., & Doumpos, M. (2002). Multicriteria classification and sorting methods: A literature review, European Journal of Operational Research, Elsevier, 138, 229-246.

[67] D.W. Aha, D. Kibler, M.K. Albert, “Instance-based learning algorithms, Machine Learning” vol. 6, pp. 37-66, 1991.

[68] S. Garcia, J. Derrac, J.R. Cano, F. Herrera, “Prototype selection for nearest neighbor classification: taxonomy and empirical study”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, pp.417-435, 2012.

[69] J. Derrac, S. Garcia, F. Herrera, “Fuzzy nearest neighbor algorithms: Taxonomy, experimental analysis and prospects”, Information Sciences, vol. 260, pp. 98-119, 2014.

[70] J.M. Mendel. “Explaining the Performance Potential of Rule-Based Fuzzy Systems as a Greater Sculpting of the State Space.” IEEE Transactions on Fuzzy Systems, pp. 2199-2216, 2017.

[71] O. Castillo, L. Amador-Angulo, J.R. Castro, M. Garcia-Valdez, “A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems”, Information Sciences, vol. 354, pp. 257-274, 2016.

Appendix

A.1. Function MLPPTB

```
function [matrizConfusion,order_matrizConfusion,ECGNet10000, Activaciones, ClasificacionMLP
]=MLPPTB(TrainSet, TestSet, neuronasCapaOculta)

clasificacionEntrenamiento_Conocida = TrainSet(:,end);
TrainSet = TrainSet(:,1:end-1);

clasificacionPrueba_Conocida = TestSet(:,end);
clasificacionPrueba_Conocida = clasificacionPrueba_Conocida';
TestSet = TestSet(:,1:end-1);

TrainOut = zeros (size(TrainSet,1), 9);
for i=1:1:length(clasificacionEntrenamiento_Conocida)
   switch clasificacionEntrenamiento_Conocida(i)
      case 1
        TrainOut(i,1) = 1;
      case 2
        TrainOut(i,2) = 1;
      case 3
        TrainOut(i,3) = 1;
      case 4
        TrainOut(i,4) = 1;
      case 5
        TrainOut(i,5) = 1;
      case 6
        TrainOut(i,6) = 1;
      case 7
        TrainOut(i,7) = 1;
      case 8
        TrainOut(i,8) = 1;
      case 9
        TrainOut(i,9) = 1;

   end
end


TrainSet = TrainSet';
TrainOut = TrainOut';
TestSet = TestSet';


datos = TrainSet;
datos(:,(end+1):(end + size(TestSet,2))) = TestSet;
Ranges = minmax(datos);

Arch = [neuronasCapaOculta 9];

ActFunc = {'logsig', 'logsig'};
ECGNet = newff(Ranges, Arch, ActFunc, 'traingd');
```

```
ECGNet.trainParam.epochs = 10000;
ECGNet.trainParam.lr = 0.001;
ECGNet.layerWeights{1,2}.trainParam.lr = 0.001;
ECGNet.layerWeights{1,2}.trainParam.learnFcn = 'learngdm';
ECGNet.layerWeights{1,2}.trainParam.mc = 0.5;
ECGNet10000 = train(ECGNet, TrainSet, TrainOut);

out = sim(ECGNet10000, TestSet);

revisa = out;
revisa(10,:) = clasificacionPrueba_Conocida;



for i=1:1:size(out,2)
   maximo=0;
   for j=1:1:size(out,1)
      if out(j,i) > maximo(1,1)
         maximo(1,1) = out(j,i);

         maximo(1,2) = j;

      end
   end
   ClasificacionMLP(i) = maximo(1,2);
   clear maximo;
end

 Activaciones = out;
%matriz de confusion
g1= clasificacionPrueba_Conocida;
g2 = ClasificacionMLP;

[matrizConfusion,order_matrizConfusion] = confusionmat(g1,g2);
```

A.2. Function MLPPTB2

```
function [matrizConfusion,order_matrizConfusion,ECGNet10000, Activaciones, ClasificacionMLP
]=MLPPTB2(TrainSet, TestSet, neuronasCapaOculta)

clasificacionEntrenamiento_Conocida = TrainSet(:,end);
TrainSet = TrainSet(:,1:end-1);

clasificacionPrueba_Conocida = TestSet(:,end);
clasificacionPrueba_Conocida = clasificacionPrueba_Conocida';
TestSet = TestSet(:,1:end-1);

TrainOut = zeros (size(TrainSet,1), 3);
for i=1:1:length(clasificacionEntrenamiento_Conocida)
   switch clasificacionEntrenamiento_Conocida(i)
      case 1
        TrainOut(i,1) = 1;
      case 2
        TrainOut(i,2) = 1;
      case 3
```

```
        TrainOut(i,3) = 1;
      case 4
        TrainOut(i,4) = 1;
      case 5
        TrainOut(i,5) = 1;
      case 6
        TrainOut(i,6) = 1;
      case 7
        TrainOut(i,7) = 1;
      case 8
        TrainOut(i,8) = 1;
      case 9
        TrainOut(i,9) = 1;

   end
end


TrainSet = TrainSet';
TrainOut = TrainOut';
TestSet = TestSet';


datos = TrainSet;
datos(:,(end+1):(end + size(TestSet,2))) = TestSet;
Ranges = minmax(datos);

Arch = [neuronasCapaOculta  3];

ActFunc = {'logsig', 'logsig'};
ECGNet = newff(Ranges, Arch, ActFunc, 'trainscg');


ECGNet.trainParam.epochs = 10000;
ECGNet.trainParam.lr = 0.001;
ECGNet.layerWeights{1,2}.trainParam.lr = 0.001;
ECGNet10000 = train(ECGNet, TrainSet, TrainOut);

out = sim(ECGNet10000, TestSet);

revisa = out;
revisa(4,:) = clasificacionPrueba_Conocida;

for i=1:1:size(out,2)
   maximo=0;
   for j=1:1:size(out,1)
      if out(j,i) > maximo(1,1)
         maximo(1,1) = out(j,i);

         if j==6
             maximo(1,2) = 9;
         else
             maximo(1,2) = j;
         end
```

```
        end
    end
    ClasificacionMLP(i) = maximo(1,2);
    clear maximo;
end

 Activaciones = out;
%matriz de confusion
g1= clasificacionPrueba_Conocida;
g2 = ClasificacionMLP;

[matrizConfusion,order_matrizConfusion] = confusionmat(g1,g2);
```

A.3. Function FuzzyKNNPTB

```
function [predicted,memberships, numhits, matrizConfusion,order_matrizConfusion]=
FuzzyKnnPTB(datosEntrenamiento, datosPrueba, k_values, info, fuzzy)

data= datosEntrenamiento(:,1:size(datosEntrenamiento,2)-1);
labels(:,1)= datosEntrenamiento(:,size(datosEntrenamiento,2));


test= datosPrueba(:,1:size(datosPrueba,2)-1);
testlabels(:,1)= datosPrueba(:,size(datosPrueba,2));

[predicted,memberships, numhits] = fknn(data, labels, test, testlabels, k_values, info, fuzzy);

ClasficacionFKNN = predicted;

ClasesDatosValidacion= testlabels;
 %matriz de confusion
 g1= ClasesDatosValidacion; %known groups
 g2 =ClasficacionFKNN'; %predicted groups

 [matrizConfusion,order_matrizConfusion] = confusionmat(g1,g2);

  %

 clear g1;
 clear g2;
```

A.4. Function HerlperRandomSplit

```
function [trainData, testData, trainLabels, testLabels]= helperRandomSplit(percent_train,
ECGDataPTB)

totalECG= size(ECGDataPTB.Data.Data_i,1);
tamanoEntrenamiento= floor(totalECG*percent_train/100);

idx = randperm(totalECG);
indexToTrain = (idx<=tamanoEntrenamiento);
indexToTest = (idx>tamanoEntrenamiento);

trainData.trainData_i = ECGDataPTB.Data.Data_i(indexToTrain,:);
```

```
testData.trainData_i = ECGDataPTB.Data.Data_i(indexToTest,:);

trainData.trainData_ii = ECGDataPTB.Data.Data_ii(indexToTrain,:);
testData.trainData_ii = ECGDataPTB.Data.Data_ii(indexToTest,:);

trainData.trainData_iii = ECGDataPTB.Data.Data_iii(indexToTrain,:);
testData.trainData_iii = ECGDataPTB.Data.Data_iii(indexToTest,:);

trainData.trainData_avr = ECGDataPTB.Data.Data_avr(indexToTrain,:);
testData.trainData_avr = ECGDataPTB.Data.Data_avr(indexToTest,:);

trainData.trainData_avl = ECGDataPTB.Data.Data_avl(indexToTrain,:);
testData.trainData_avl = ECGDataPTB.Data.Data_avl(indexToTest,:);

trainData.trainData_avf = ECGDataPTB.Data.Data_avf(indexToTrain,:);
testData.trainData_avf = ECGDataPTB.Data.Data_avf(indexToTest,:);

trainData.trainData_v1 = ECGDataPTB.Data.Data_v1(indexToTrain,:);
testData.trainData_v1 = ECGDataPTB.Data.Data_v1(indexToTest,:);

trainData.trainData_v2 = ECGDataPTB.Data.Data_v2(indexToTrain,:);
testData.trainData_v2 = ECGDataPTB.Data.Data_v2(indexToTest,:);

trainData.trainData_v3 = ECGDataPTB.Data.Data_v3(indexToTrain,:);
testData.trainData_v3 = ECGDataPTB.Data.Data_v3(indexToTest,:);

trainData.trainData_v4 = ECGDataPTB.Data.Data_v4(indexToTrain,:);
testData.trainData_v4 = ECGDataPTB.Data.Data_v4(indexToTest,:);

trainData.trainData_v5 = ECGDataPTB.Data.Data_v5(indexToTrain,:);
testData.trainData_v5 = ECGDataPTB.Data.Data_v5(indexToTest,:);

trainData.trainData_v6 = ECGDataPTB.Data.Data_v6(indexToTrain,:);
testData.trainData_v6 = ECGDataPTB.Data.Data_v6(indexToTest,:);

trainData.trainData_vx = ECGDataPTB.Data.Data_vx(indexToTrain,:);
testData.trainData_vx= ECGDataPTB.Data.Data_vx(indexToTest,:);

trainData.trainData_vy = ECGDataPTB.Data.Data_vy(indexToTrain,:);
testData.trainData_vy = ECGDataPTB.Data.Data_vy(indexToTest,:);

trainData.trainData_vz = ECGDataPTB.Data.Data_vz(indexToTrain,:);
testData.trainData_vz = ECGDataPTB.Data.Data_vz(indexToTest,:);


trainLabels= ECGDataPTB.Labels(indexToTrain,:);
testLabels= ECGDataPTB.Labels(indexToTest,:);
```

A.5. Function Main

```
% FKNN

    knn= 3;
```

```
    [predicted_Fknn_3,memberships_Fknn_3,
numhits_Fknn_3,matrizConfusionFknn_3,order_matrizConfusion_Fknn_3]=
FuzzyKnnPTB(TrainSet, TestSet ,knn, 0, true, clasificacionPrueba_Conocida);


    PorcentajeClasificacion_Fknn_3=0;
    TotalCorrectos_Fknn_3=0;
    TotalErrores_Fknn_3=0;

    for iclass=1:1:size(order_matrizConfusion_Fknn_3,1)

      TotalCorrectos_Fknn_3=TotalCorrectos_Fknn_3 + matrizConfusionFknn_3(iclass,iclass);

    end

    TotalErrores_Fknn_3= size( clasificacionPrueba_Conocida,1) - TotalCorrectos_Fknn_3;
    PorcentajeClasificacion_Fknn_3= (TotalCorrectos_Fknn_3*100)/  size(
clasificacionPrueba_Conocida,1);


%200
 neuronasCapaOculta = 200;
[matrizConfusionMLP2_200,order_matrizConfusionMLP2_200, ECGNet10000MLP2_200,
ActivacionesMLP2_200, ClasificacionMLP2_200  ]=MLPPTB_Mod(TrainSet, TestSet,
neuronasCapaOculta);



  PorcentajeClasificacion_MLP2_200=0;
  TotalCorrectosMLP2_200=0;
  TotalErroresMLP2_200=0;

  for iclass=1:1:size(order_matrizConfusionMLP2_200,1)

    TotalCorrectosMLP2_200=TotalCorrectosMLP2_200 +
matrizConfusionMLP2_200(iclass,iclass);

  end

  TotalErroresMLP2_200= size( clasificacionPrueba_Conocida,1) - TotalCorrectosMLP2_200;
  PorcentajeClasificacion_MLP2_200= (TotalCorrectosMLP2_200*100)/  size(
clasificacionPrueba_Conocida,1);

 %50
 neuronasCapaOculta = 50;
[matrizConfusionMLP2_50_2L,order_matrizConfusionMLP2_50_2L, ECGNet10000MLP2_50_2L,
ActivacionesMLP2_50_2L, ClasificacionMLP2_50_2L  ]=MLPPTB_Mod(TrainSet, TestSet,
neuronasCapaOculta);


  PorcentajeClasificacion_MLP2_50_2L=0;
  TotalCorrectosMLP2_50_2L=0;
  TotalErroresMLP2_50_2L=0;

  for iclass=1:1:size(order_matrizConfusionMLP2_50_2L,1)
```

```
      TotalCorrectosMLP2_50_2L=TotalCorrectosMLP2_50_2L +
matrizConfusionMLP2_50_2L(iclass,iclass);

   end

   TotalErroresMLP2_50_2L= size( clasificacionPrueba_Conocida,1) - TotalCorrectosMLP2_50_2L;
   PorcentajeClasificacion_MLP2_50_2L= (TotalCorrectosMLP2_50_2L*100)/  size(
clasificacionPrueba_Conocida,1);

 %IntegrationMatrix


MatrizIntegracion(1:2,:)= ActivacionesMLP2_150_i(1:2,1:end);
MatrizIntegracion(3:4,:)= ActivacionesMLP2_50_ii(1:2,1:end);
MatrizIntegracion(5:6,:)= ActivacionesMLP2_150_iii(1:2,1:end);
MatrizIntegracion(7:8,:)= ActivacionesMLP2_150_avf(1:2,1:end);
MatrizIntegracion(9:10,:)= membershipsFKNN_4_avr(1:end,1:2)';
MatrizIntegracion(11:12,:)= membershipsFKNN_5_avl(1:end,1:2)';
MatrizIntegracion(13:14,:)= membershipsFKNN_3_v1(1:end,1:2)';
MatrizIntegracion(15:16,:)= membershipsFKNN_4_v2(1:end,1:2)';
MatrizIntegracion(17:18,:)= membershipsFKNN_5_v3(1:end,1:2)';
MatrizIntegracion(19:20,:)=  ActivacionesMLP2_150_v4(1:2,1:end);
MatrizIntegracion(21:22,:)=  ActivacionesMLP2_50_v5(1:2,1:end);
MatrizIntegracion(23:24,:)=  ActivacionesMLP2_100_v6(1:2,1:end);

MatrizIntegracion= MatrizIntegracion';


%CombiningFuzzyInferenceSystem

fismat= readfis('Fis_PTB.fis');
[ClasificacionArritmiasFinal]=evalfis(MatrizIntegracion, fismat);


outputValidacion= ClasificacionArritmiasFinal';

for i=1:1:size(outputValidacion,2)
   maximo=0;
   for j=1:1:size(outputValidacion,1)
      if outputValidacion(j,i) > maximo(1,1)
         maximo(1,1) = outputValidacion(j,i);
         maximo(1,2) = j;

      end
   end
   ClasificacionValidacion_Fis(i) = maximo(1,2);
   clear maximo;

end

ClasificacionValidacion_Fis=  ClasificacionValidacion_Fis';

%matriz de confusion
 g1= testLabels; %known groups
 g2 =ClasificacionValidacion_Fis; %predicted groups
```

```
[matrizConfusion,order_matrizConfusion] = confusionmat(g1,g2);
```