

AUTOMATIC ARRHYTHMIAS DETECTION USING VARIOUS TYPES OF ARTIFICIAL NEURAL NETWORK BASED LEARNING VECTOR QUANTIZATION

Diane Fitria¹, M. Anwar Ma'sum¹, Elly Matul Imah², and Alexander A.S. Gunawan³

¹Faculty of Computer Science, Universitas Indonesia, Kampus UI Depok, 16424, Indonesia

²Mathematics Department, State University of Surabaya, Jl. Ketintang, Surabaya, 60231, Indonesia

³Faculty of Computer Science, Bina Nusantara University, Jl. K H Syahdan No. 9 Jakarta, 11480, Indonesia

E-mail:diane.fitria@ui.ac.id

Abstract

An automatic Arrhythmias detection system is urgently required due to small number of cardiologists in Indonesia. This paper discusses only about the study and implementation of the system. We use several kinds of signal processing methods to recognize arrhythmias from ecg signal. The core of the system is classification. Our LVQ based artificial neural network classifiers based on LVQ, which includes LVQ1, LVQ2, LVQ2.1, FNLVQ, FNLVQ MSA, FNLVQ-PSO, GLVQ and FNGLVQ. Experiment result show that for non round robin dataset, the system could reach accuracy of 94.07%, 92.54%, 88.09% , 86.55% , 83.66%, 82.29 % , 82.25%, and 74.62% respectively for FNGLVQ, FNLVQ-PSO, GLVQ, LVQ2.1, FNLVQ-MSA, LVQ2, FNLVQ and LVQ1. Whereas for round robin dataset, system reached accuracy of 98.12%, 98.04%, 94.31%, 90.43%, 86.75%, 86.12 % , 84.50%, and 74.78% respectively for GLVQ, LVQ2.1, FNGLVQ, FNLVQ-PSO, LVQ2, FNLVQ-MSA, FNLVQ and LVQ1.

Keywords: *Automatic Arrhythmias detection, ECG, Classification, LVQ1, LVQ2, LVQ2.1, FNLVQ, FNLVQ MSA, FNLVQ-PSO, GLVQ, FNGLVQ*

Abstrak

Sistem deteksi aritmia otomatis sangat diperlukan karena keterbatasan dokter spesialis jantung di Indonesia. Paper ini akan mendiskusikan secara lengkap tentang studi dan implementasi dari sistem tersebut. Kami menggunakan berbagai macam metode pengolahan sinyal untuk mengenali aritmia berdasarkan sinyal ekg. Bagian utama dari sistem adalah klasifikasi. Kami menggunakan jaringan syaraf tiruan berbasis LVQ yang meliputi LVQ1, LVQ2, LVQ2.1, FNLVQ, FNLVQ MSA, FNLVQ-PSO, GLVQ dan FNGLVQ. Hasil eksperimen menunjukkan untuk data non round robin tingkat akurasi sistem mencapai 94.07%, 92.54%, 88.09% , 86.55% , 83.66%, 82.29 % , 82.25%, dan 74.62% berturut-turut untuk FNGLVQ, FNLVQ-PSO, GLVQ, LVQ2.1, FNLVQ-MSA, LVQ2, FNLVQ dan LVQ1. Sedangkan untuk data round robin tingkat akurasi sistem mencapai 98.12%, 98.04%, 94.31%, 90.43%, 86.75%, 86.12 % , 84.50%, dan 74.78% berturut-turut untuk GLVQ, LVQ2.1, FNGLVQ, FNLVQ-PSO, LVQ2, FNLVQ-MSA, FNLVQ dan LVQ1.

Kata Kunci: *Deteksi aritmia otomatis, EKG, klasifikasi, LVQ1, LVQ2, LVQ2.1, FNLVQ, FNLVQ MSA, FNLVQ-PSO, GLVQ, FNGLVQ*

1. Introduction

Cardiac dysrhythmia or arrhythmia is an abnormality of the heart rhythm. In this disorder, the heart rate may be too fast or just too slow compared to normal heart rate, which is about 60-80 beats per minute. In this disorder heart rhythm also could be irregular [1]. Some types of arrhythmia are not dangerous even sometimes patients are unaware of the condition. In several cases, patients may experience clinical symptoms such as

palpitations, dizziness, chest pain, unable to breathe and even sudden death. However, a normal person may also experience symptoms similar but not including arrhythmia. For example, heart rate increases during exercise or palpitations when we feel scared or anxious. To detect this disease we need not only clinical diagnosis but also takes other investigations. Common examination of heart is done by checking heart signal which utilize electrocardiogram (ECG) tools. The results of the examination is the waves represent electrical activity

of the heart called ECG waves. These waves can describe the condition heart. But not everyone has the ability to translate the waves. In addition, there is a difference of interpretation between the experts. Furthermore, the number of cardiologist in Indonesia is much smaller than the number of population, approximately 1:665.730 [2]. This facts motivated us to develop an early detection mechanism that easily can be used not only by expert but also by wider society. Mechanisms of the system is implemented using artificial intelligence to detect heart diseases based on ECG waveforms.

There have been many studies related to this topic. Yeap et. al. proposed backpropagation algorithm for arrhythmia detection based on ECG data [3]. Yeap used ECG data derived from the AHA database. Ozbay et al did similar work, but used data from the MIT-BIH as training data and the data from its own institutions as testing data [4]. Ghongade et al focused on comparison of various feature extraction methods for ECG data [5]. Ghongade compared Discrete Fourier Transform (DFT), Principle Component Analysis (PCA), Discrete Wavelet Transform (DWT), and morphological features such as R-peak and QRS complex, then the result of features extraction was classified using Backpropagation algorithm.

The other previous studies was conducted by Made Agus et al., which develop new neural network algorithm named Fuzzy-Neuro Generalized Learning Vector Quantization (FNGLVQ) [6]. They modified previous algorithm named Generalized Learning Vector Quantization (GLVQ) with fuzzy membership function for vector references. GLVQ was proposed by Sato et al. in previous research [7]. The basic idea of GLVQ is applying gradient descent optimization to LVQ 2.1 in order to minimize misclassification error (MCE) during training process. LVQ 2.1 is variant of Learning Vector Quantization (LVQ) classifier [8]. LVQ is competitive based artificial neural network proposed by Kohonen [8]. The other variants of LVQ is LVQ 2 and LVQ 3 [8]. In other research, Benyamin et al applied fuzzy concept to LVQ algorithm resulting Fuzzy-Neuro Learning Vector Quantization (FNLVQ) [9]. They also made variant of FNLVQ by applying matrix similarity analysis (MSA) concept in training process. Later, Rochmatullah et al. proposed new improvement of FNLVQ by applying Particle Swarm Optimization (PSO) method resulting FNLVQ-PSO [10].

The aim of this research is to develop an automatic arrhythmias detection using ECG data (signal). ECG signal acquired from patient is processed in several steps to extract features. Then, the signal is classified to several types of heart condition. In this research, we use several LVQ

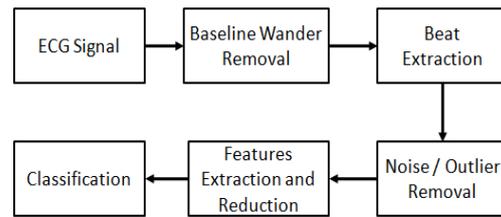


Figure 1. Proposed framework diagram

based classifiers. This framework is used for early detection and monitoring system of heart disease. Therefore, system can detect arrhythmias symptom automatically from heart beat signal. The application of this framework is expected to help people to maintain their heart health.

The rest of the paper is organized as follows. Section 2 discusses structure of the framework for automatic arrhythmias detection and classifiers used in this research. Section 3 discusses about experiment results and analysis, and we draw conclusion in Section 4.

2. Methods

This section explain structure of proposed framework. Process sequence of proposed framework is shown in Figure 1. ECG signal will be processed in five steps. The first step is base-line wander removal which used to straighten ECG signal. The second process is beat extraction. Next, heart beats from the same class was collected. Noises or outlier instances were removed from their class. Afterward, features were derived from each beat continued by features reduction. The last step is heart beat classification. Detail of each process will be explained in the following sub sections.

Baseline Wandering Removal

The first phase of the ECG signal processing is Baseline Wander Removal (BWR). Baseline is a condition in which the ECG signal is not being isoelectric line (the axes) but shifted up or down. This phenomenon is caused by appearance of low frequency in heartbeat recording which come from respiratory or body movements. Baseline of course will interfere analysis process of heartbeat signal and may lead to errors in the detection. Therefore it is necessary to remove the baseline.

A BWR process uses nonlinear cubic spline interpolation combined with reduction techniques (subtraction technique) so it can reduce noise without affecting ECG waveform significantly. The basic concept of BWR is shit estimation of main axis using representative points, one point for each beat, on the ECG waveform. Shit estimation of

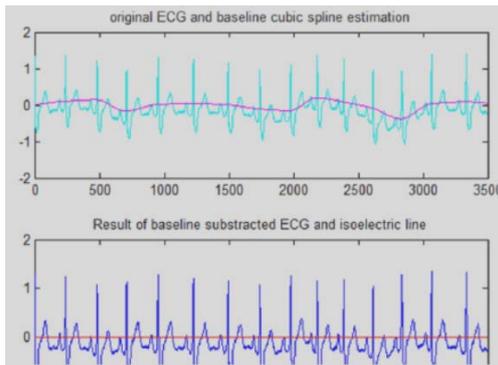


Figure 2. Baseline Wander Removal Process

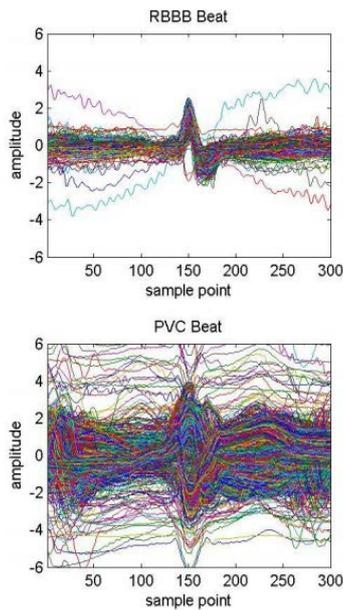


Figure 3. Beat segmentation result for class RBBB and class PVC

major axis is done by cubic spline interpolation.

Cubic interpolation is chosen because it can produce smoother signal. In addition, it can guarantee continuity of its first and second derivatives at the entire interval. Cubic spline interpolation is an estimation function that can be obtained using a third-degree polynomial in each sub-interval. Sub interval used in ECG signal is the PQ interval of each beat. After shift estimation of the main axis was obtained, then the real axis was determined using isoelectric pivot. In this research we use BWR proposed by Clifford et al [11]. BWR process is shown on Figure 2.

Beat Extraction

In this research we use beat classification for arrhythmias detection as used by previous researcher [12]. Using this approach, ECG signal is segmen-

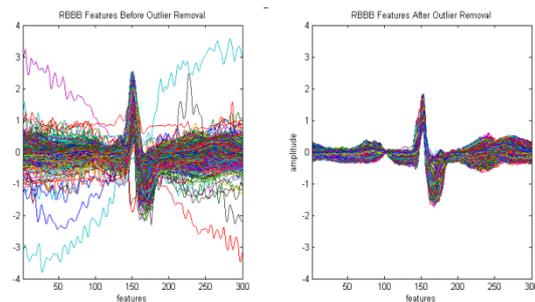


Figure 4. Noise removal for class RBBB

ted in to collection of single beat. This approach is chosen because the dataset has annotation for R-peak. Segmentation result shows that the beat has 300 points. The result of beat segmentation is shown in figure 3.

Noise/Outlier Removal

Beats segmented by previous steps are collected according their class label. Then, noises or outlier instances of each class were removed from dataset. In this research we used Inter Quarile Range method for noise/outlier removal which was used by previous researcher [6]. Sample of noise removal for class RBBB is shown in figure 4.

Feature Extraction and Reduction

A feature is information that used by classifier model to differentiate class of instances. Therefore, feature must be extracted from each instance of the dataset. In this research, we used Discrete Wavelet Transform (DWT) for features extraction. There are benefits can be obtained by using DWT. The first benefit is suitability of DWT for heart beat feature extraction. Arrhythmias can be detected and recognized by recognizing pattern on the heartbeat. In other hand, DWT can be used to acquire information that represent signal pattern so it is suitable for heart beat feature extraction. The second benefit is DWT can reduce the dimension of the signal without losing important information. So it helped to obtain simpler computational process. The third benefit is DWT use filter operation. Therefore signal produced by DWT is smoother than original signal. In this research, we used Daubechies 8 (db8) mother wavelet for feature extraction.

The last process in the proposed framework is classification. Input of this process is feature extracted by previous process. The output of this process is prediction of the system for given sample instances. We use LVQ based classifiers include LVQ1, LVQ2. LVQ2.1, GLVQ, FNLVQ,

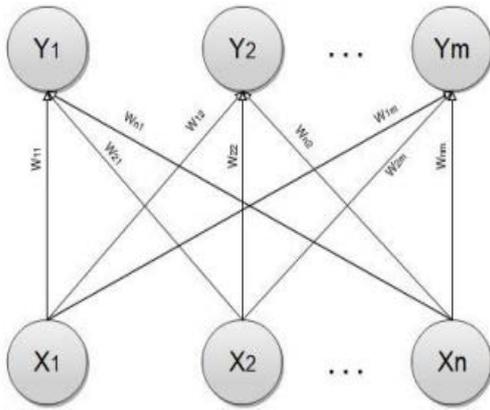


Figure 5. LVQ architecture

FNLVQ-MSA, FNLVQ-PSO, and FNG-LVQ.

Classifiers

These sub-section explain the classifiers used in this research.

Learning Vector Quantization (LVQ)

Learning Vector Quantization (LVQ) is competitive based neural network proposed by Kohonen [8]. LVQ has two layers, input layer and output layer. LVQ Architecture is shown in Figure 5.

Let denote x as vector input, Y as class label, and w_{ij} as reference vector for class j element i (feature i). During training process, w_{ij} is initiated by random value. Then iteratively, instance x was trained to the model. Next, distance of input vector x and reference vector class j was computed using equation(1).

$$d(j) = \sum_{i=1}^N (x_i - w_{ij})^2 \quad (1)$$

After computing distance for all class, then we search class with closest distance. Later, the class was called by winner class. Then reference vector of winner class w_p was updated. Using equation (2).

$$w_p \leftarrow w_p + \alpha(x - w_p) \text{ if } C_{w_p} = C_j \quad (2)$$

$$w_p \leftarrow w_p - \alpha(x - w_p) \text{ if } C_{w_p} \neq C_j \quad (3)$$

Where α is learning rate, C_{w_p} is class label of winner class and C_j is class label of sample x .

LVQ 2

The basic concept of LVQ 2 is similar with LVQ 1. LVQ 2 update winner reference vector (w_p) and also runner up reference vector (w_r). LVQ 2 also

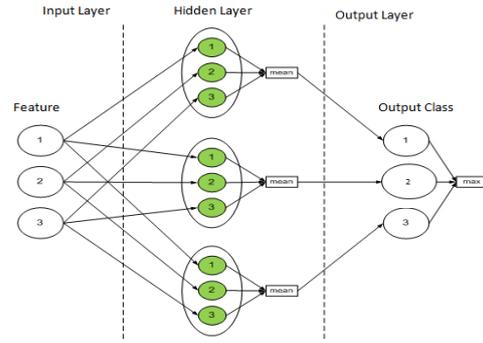


Figure 6. FNLVQ architecture

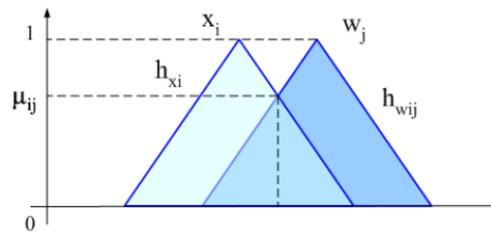


Figure 7. Computing similarity in FNLVQ

define new constant (ω) that used for limitation between $d(p)$ and $d(r)$. $D(p)$ is distance between winner vector and input vector, whereas $d(r)$ is distance between runner up vector and input vector. Update process will be conducted according to the given equation(4).

$$\frac{d_p}{d_r} > (1 - \omega) \text{ and } \frac{d_r}{d_p} < (1 - \omega) \quad (4)$$

If condition as mentioned in equation(4) was accepted, then those two reference vectors would be updated using equation(5) and equation(6).

$$w_p \leftarrow w_p - \alpha(x - w_p) \quad (5)$$

$$w_r \leftarrow w_r + \alpha(x - w_r) \quad (6)$$

LVQ 2.1

LVQ 2.1 is variant of is LVQ 2 that uses assumption that the winner label class is same as input label class $C_x = C_{w_p}$. LVQ 2.1 uses condition written in equation(7).

$$C_{w_r} \neq C_{w_p} \text{ and } \min\left(\frac{d_r}{d_p}, \frac{d_p}{d_r}\right) > \frac{(1 - \omega)}{(1 + \omega)} \quad (7)$$

Fuzzy-Neuro Learning Vector Quantization (FNLVQ)

FNLVQ was developed from LVQ by applying fuzzy concept to reference vector. In previous re-

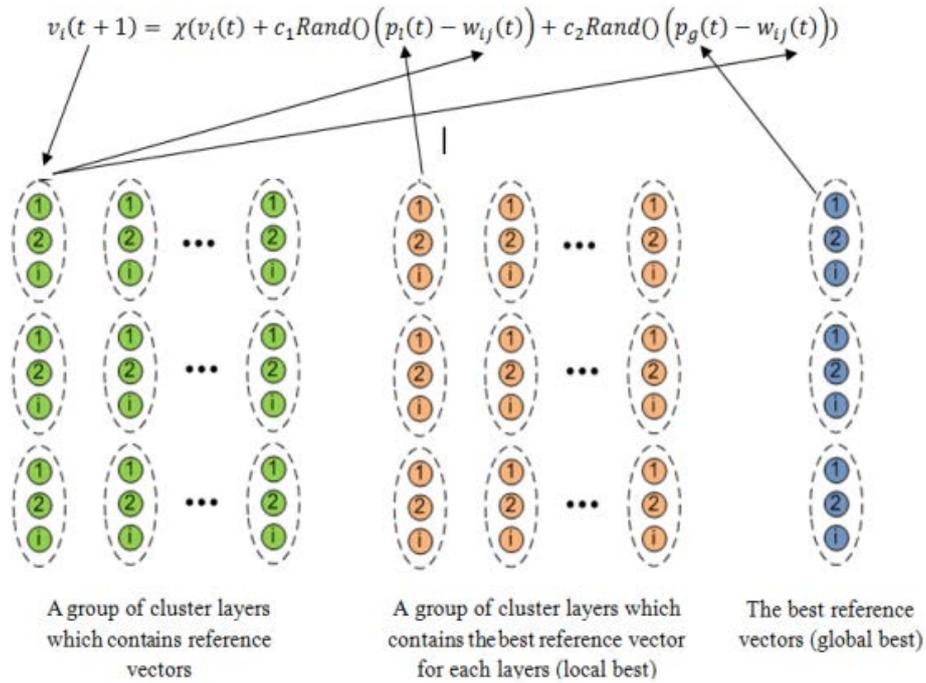


Figure 9. FNLVQ-PSO architecture

search, FNLVQ was proposed by Benyamin et al for artificial odor discrimination system [9]. FNLVQ Architecture is shown in figure 6. There are three layers in FNLVQ, input layer, hidden layer and output layer. Input layer consists of input vector, hidden layer consists of class reference vectors, and output layer consists of output class of the input layer. Applying fuzzy concept, reference vector is represented by three values, i.e. $w_{ij}(l)$, w_{ij} , $w_{ij}(r)$ that represent min, mean, and max of the triangle.

Similar to other LVQ based neural network, FNLVQ use winner take all approach in training process. Reference vector will be updated based on winner vector, which is closest to the input. In FNLVQ, closest vector is obtained by computing similarity value from its fuzzy membership function. Suppose that x is input vector, h_x is fuzzy membership function of input vector x , w_i is reference vector for class i , h_{wi} is fuzzy membership function of reference vector w_i , then similarity value between input vector and reference vector can be computed using fuzzy triangle as shown in Figure 7 [9].

Mathematically, similarity value for feature j in class i μ_{ij} can be computed using equation(8).

$$\mu_{ij} = \max(h_{xj} \wedge h_{wij}). \quad (8)$$

Then, similarity value for class i can be computed using equation(9).

$$\mu_j = \min(\mu_{ij}). \quad (9)$$

Update process of reference vector is conducted according three conditions. First, if winner class is same as input label class, then reference vector will be updated using equation(10) to equation(13).

$$d_{ij} \leftarrow \alpha * ((\{1 - \mu_i\}) * (\{x_j - w_{ij}(t)\})) \quad (10)$$

$$w_{ij} \leftarrow w_{ij} + d_{ij} \quad (11)$$

$$w(l)_{ij} \leftarrow w(l)_{ij} + d_{ij} \quad (12)$$

$$w(r)_{ij} \leftarrow w(r)_{ij} + d_{ij} \quad (13)$$

Then, equation(12) to equation(13) are followed by widening or narrowing of the fuzzy triangle using equation(14) and equation(15).

$$w(l)_{ij} \leftarrow w(l)_{ij} + (\beta * (w_{ij} - w(l)_{ij})) \quad (14)$$

$$w(r)_{ij} \leftarrow w(r)_{ij} - (\beta * (w(r)_{ij} - w_{ij})) \quad (15)$$

Second, if vector class is not same as input label class, then reference vector will be updated using equation(16) to equation(19).

$$d_{ij} \leftarrow \alpha * ((\{1 - \mu_i\}) * (\{x_j - w_{ij}(t)\})) \quad (16)$$

$$w_{ij} \leftarrow w_{ij} - d_{ij} \quad (17)$$

$$w(l)_{ij} \leftarrow w(l)_{ij} - d_{ij} \quad (18)$$

$$w(r)_{ij} \leftarrow w(r)_{ij} - d_{ij} \quad (19)$$

Then, equations(18) and equation(19) are followed by widening or narrowing of the fuzzy triangle using equation(20) and equation(21).

$$w(l)_{ij} \leftarrow w(l)_{ij} + (\beta * (w_{ij} - w(l)_{ij})) \quad (20)$$

$$w(r)_{ij} \leftarrow w(r)_{ij} - (\beta * (w(r)_{ij} - w_{ij})) \quad (21)$$

Third, if winner class is same as input label class, but similarity value is 0, then reference vector will be updated using equation(12) to equation (15), followed by widening and narrowing fuzzy triangle using equation(22) and equation(23).

$$w(l)_{ij} \leftarrow w(l)_{ij} - (\beta * (w_{ij} - w(l)_{ij})) \quad (22)$$

$$w(r)_{ij} \leftarrow w(r)_{ij} + (\beta * (w(r)_{ij} - w_{ij})) \quad (23)$$

where α is learning rate, β is constant between 0 and 1. Testing process in FNLVQ is conducted to determine winner (closest) reference vector to the input. This winner vector is the result of system prediction.

Fuzzy-Neuro Learning Vector Quantization–Matrix Similarity Analysis (FNLVQ-MSA)

During training process, FNLVQ will stop after reach the number of epoch given as parameter. Matrix similarity analysis (MSA) can be used to determine when it will stop the iteration. MSA describes reference vector at an iteration. It can give similarity value of the reference vector at the iteration. Therefore, we can set threshold of similarity value in which system should stop training process. FLVQ-MSA was proposed by Benyamin et al [8] for artificial odor discrimination system. Structure of MSA is shown in Figure 8.

$$M = \begin{bmatrix} m_{11} & \dots & m_{n1} \\ \vdots & \ddots & \vdots \\ m_{1n} & \dots & m_{nn} \end{bmatrix}$$

Each element of M, m_{ij} can be computed using equation(24).

$$m_{mj} = \frac{1}{N} \sum_{k=1}^N \max(\min(\mu_{ij})(k)) \quad (24)$$

where M is $n \times n$ dimension MSA, n is the number of output class, μ_{ij} is similarity value. Training

process of FNLVQ-MSA can be executed using steps: 1) Initiate reference vector; 2) Compute matrix similarity analysis; 3) Train the model; 4) Compute current MSA. If current MSA is better than previous MSA, update reference vector; 5) If similarity values are less than threshold and the number of iteration is less than epoch then go to step 3. Testing process in FNLVQ-MSA is same as in FNLVQ.

Fuzzy-Neuro Learning Vector Quantization–Particle Swarm Optimization (FNLVQ-PSO)

In previous research, Rochmatullah et al optimized Fuzzy-Neuro Learning Vector Quantization with Particle Swarm Optimization algorithm resulting FNLVQ-PSO [10]. Optimization is used during training process. PSO is optimization algorithm that uses colony of agents to find optimum value [13]. Architecture of FNLVQ-PSO is shown in Figure 9 [10].

PSO is optimization algorithm that uses colony of agents to find optimum value [13]. Each particle will search optimum point using equation (25) and equation(26).

$$Vi(t) \leftarrow \gamma \cdot (Vi(t-1) + c1 \cdot rand() \cdot (Pi(t-1) - Xi(t-1)) + c2 \cdot rand() \cdot (Pg(t-1) - Xi(t-1))) \quad (25)$$

$$Xi(t) \leftarrow Xi(t-1) + Vi(t) \quad (26)$$

where $V_i(t)$ is current particle velocity, $V_i(t-1)$ is previous particle velocity, $X_i(t)$ is current particle position, $X_i(t-1)$ is previous particle position, c_1 and c_2 are constant, γ is construction factor which has value between 0 and 1, $P_i(t-1)$ is previous local base of the agent, and $P_g(t-1)$ is previous global base. Local best is optimum point for a particle. So each particle has their own local best. Global best is optimum point for all particle. In other words, global best is the most optimum of local best points. In FNLVQ-PSO, particle is represented as reference vector. Target of the optimization is to find the optimum reference vector during training. Training process In FNLVQ-PSO can be executed using following steps: 1) Initialize reference vector as much as the number of particle. In FNLVQ-PSO, reference vector is generated in several number called particle. So there are several candidates of reference vector; 2) Train FNLVQ algorithm in each particle. For each particle, compute similarity value between input vector and reference vector; 3) Compute MSA in each particle. The MSA will have $n \times n$ size matrix where n is the numbers of output class; 4) Compute MSA and fitness using equation(27); 5) Compute local best

and global best; 6) Update reference vector using equation(28) and equation(29); 7) Update fuzzy membership function of each particle using equation(30) and equation(31) where d_{ij} is distance between particle and input vector.

$$f_k = \sum_{i=1}^n m_{ki} - \left[\sum_{i=1}^n \sum_{j=1}^n m_{ij} \text{ if } i \neq j \right] \quad (27)$$

$$w_{ij}(t+1) = w_{ij}(t) + v_i(t+1) \quad (28)$$

$$v_i(t+1) = \gamma \left(\begin{array}{c} v_i(t) + \\ c1.rand.(p_l(t) - w_{ij}(t)) + \\ c2.rand.(p_g(t) - w_{ij}(t)) \end{array} \right) \quad (29)$$

$$w(l)_{ij}(t+1) = w(l)_{ij}(t) + d_{ij} \quad (30)$$

$$w(r)_{ij}(t+1) = w(r)_{ij}(t) + d_{ij} \quad (31)$$

8) Repeat step 2 to step 7 until reference vector reach convergence. Convergence also can be defined when the number of epoch is reached.

Generalize Learning Vector Quantization (GLVQ)
Generalize Learning Vector Quantization (GLVQ) was proposed by Sato and Yamada [7]. Sato and Yamada modified LVQ 2.1 with minimization of cost function and misclassification error. They used gradient descent for minimization process. In GLVQ, w_1 is defined as the reference vector of input vector class $C_x = C_{w_1}$, and w_2 is the closest reference class from different class $C_x \neq C_{w_2}$. Miss classification error is defined as equation(32).

$$\varphi(x) = \frac{d_1 - d_2}{d_1 + d_2} \quad (32)$$

Where d_1 is distance between w_1 and input vector x , d_2 is distance between w_2 and input vector x . The value of $\varphi(x)$ will be positive if input vector x is correctly classified by system and negative if input vector x is classified in to wrong class. Target of the learning process is to minimize cost function as defined in equation(33).

$$S = \sum_{i=1}^N f(\varphi(x)) \quad (33)$$

where $f()$ is rising monotonic function. To minimize S , w_1 and w_2 will be updated using steepest descent method as equation(34).

$$w_i \leftarrow w_i - \alpha \frac{\delta S}{\delta w_i}, i = 1, 2 \quad (34)$$

If discriminant function used is Euclidean, then equation(27) can be written as equation(35) and equation(36).

$$\begin{aligned} \frac{\delta S}{\delta w_1} &= \frac{\delta S}{\delta \varphi} \frac{\delta \varphi}{\delta d_1} \frac{\delta d_1}{\delta w_1} \\ &= - \frac{\delta f}{\delta \varphi} \frac{4d_2}{(d_1 + d_2)^2} (x - w_1) \end{aligned} \quad (35)$$

$$\begin{aligned} \frac{\delta S}{\delta w_2} &= \frac{\delta S}{\delta \varphi} \frac{\delta \varphi}{\delta d_2} \frac{\delta d_2}{\delta w_2} \\ &= - \frac{\delta f}{\delta \varphi} \frac{4d_1}{(d_1 + d_2)^2} (x - w_2) \end{aligned} \quad (36)$$

Therefore, the rules of learning process can be written as equation(37).

$$w_1 \leftarrow w_1 + \alpha \frac{\delta f}{\delta \varphi} \frac{4d_2}{(d_1 + d_2)^2} (x - w_1) \quad (37)$$

$$w_2 \leftarrow w_2 + \alpha \frac{\delta f}{\delta \varphi} \frac{4d_1}{(d_1 + d_2)^2} (x - w_2) \quad (38)$$

In GLVQ algorithm, the monotonic function used is sigmoid function as written in equation below.

$$f(\varphi, t) = \frac{1}{1 + e^{-\varphi t}} \quad (39)$$

$$\frac{\delta f}{\delta \varphi} = f(\varphi, t)(1 - f(\varphi, t)) \quad (40)$$

Fuzzy-Neuro Generalize Learning Vector Quantization (FNGLVQ)

Fuzzy-Neuro Generalize Learning Vector Quantization (FNGLVQ) was proposed by Made Agus et al. They combine fuzzy concept to GLVQ algorithm [6]. FNGLVQ architecture is illustrated in Figure 10 [6].

First, they complement distant value into $d = 1 - \mu$, then substitute to equation 26 resulting equation(41).

$$\varphi(x) = \frac{\mu_2 - \mu_1}{2 - \mu_2 - \mu_1} \quad (41)$$

where μ_1 is similarity value between input vector and reference vector from same class $C_x = C_{w_1}$, and μ_2 is similarity value between input vector and closest reference vector from different class $C_x \neq C_{w_2}$.

Then adjustment of reference vector is made based on similarity value, as defined in equation (42).

$$\frac{\delta S}{\delta w_i} = \frac{\delta S}{\delta \varphi} \cdot \frac{\delta \varphi}{\delta \mu} \cdot \frac{\delta \mu}{\delta w_i} \quad (42)$$

In case of reference vector is defined with triangu-

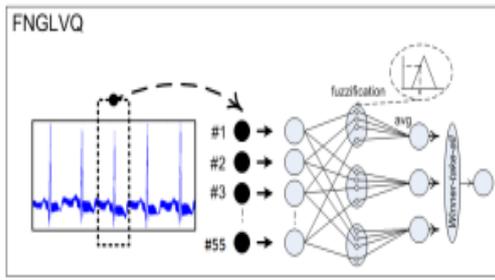


Figure 10. FNGLVQ architecture

lar function $w_{ij}(w_{min,ij}, w_{mean,ij}, w_{max,ij})$, membership function can be defined as the following equation(43).

$$\mu = h(x, w_{min}, w_{mean}, w_{max}) \quad (43)$$

$$= \begin{cases} 0, & x \leq w_{min} \\ \frac{x - w_{min}}{w_{mean} - w_{min}}, & w_{min} \leq x \leq w_{mean} \\ \frac{w_{max} - x}{w_{max} - w_{mean}}, & w_{mean} \leq x \leq w_{max} \\ 0, & x \geq w_{max} \end{cases}$$

Derivation of the membership function to (w_{mean}) lead is divided into three conditions and lead to learning formula as the following equation (44).

If $w_{mean} < x \leq w_{mean}$

$$w_1(t+1) \leftarrow w_1(t) - \alpha x \frac{\delta f}{\delta \varphi} x \frac{2 \cdot (1 - \mu_2)}{(2 - \mu_1 - \mu_2)^2} \cdot x \left(\frac{x - w_{min}}{(w_{mean} - w_{min})^2} \right) \quad (44)$$

$$w_2(t+1) \leftarrow w_2(t) \alpha x \frac{\delta f}{\delta \varphi} x \frac{2 \cdot (1 - \mu_1)}{(2 - \mu_1 - \mu_2)^2} x \left(\frac{x - w_{min}}{(w_{mean} - w_{min})^2} \right) \quad (45)$$

If $w_{mean} < x < w_{max}$

$$w_1(t+1) \leftarrow w_1 + \alpha x \frac{\delta f}{\delta \varphi} x \frac{2 \cdot (1 - \mu_2)}{(2 - \mu_1 - \mu_2)^2} x \left(\frac{w_{max} - x}{(w_{max} - w_{mean})^2} \right) \quad (46)$$

$$w_2(t+1) \leftarrow w_2(t) - \alpha x \frac{\delta f}{\delta \varphi} x \frac{2 \cdot (1 - \mu_1)}{(2 - \mu_1 - \mu_2)^2} x \left(\frac{w_{max} - x}{(w_{max} - w_{mean})^2} \right) \quad (47)$$

If $x \leq w_{min}$ dan $x \geq w_{max}$

$$w_i(t+1) \leftarrow w_i(t), i = 1, 2 \quad (48)$$

where w_1 is reference vector from same class as input vector $C_x = C_{w_1}$, and w_2 reference vector from different class $C_x \neq C_{w_2}$. Update rules for w_{min} and w_{max} follow equation(49) and equation (50).

$$w_{min} \leftarrow w_{mean}(t+1) - (w_{mean}(t) - w_{min}(t)) \quad (49)$$

$$w_{max} \leftarrow w_{mean}(t+1) - (w_{mean}(t) - w_{min}(t)) \quad (50)$$

The value of α is between 0 and 1, and its value will decrease along with the value of number iteration (t), as defined in equation(51).

$$\alpha(t+1) = \alpha_0 \times \left(1 - \frac{t}{t_{max}} \right) \quad (51)$$

To gain better performance, we use additional rule to adjust w_{min} and w_{max} as defined in conditions below:

If $\mu_1 > 0$ or $\mu_2 > 0$, then at least one of two reference vectors recognize the input, therefore if $\varphi < 0$ then increase fuzzy triangular width using equation(52) and equation(53).

$$w_{min} \leftarrow w_{mean} - (w_{mean} - w_{min}) x (1 + (\beta \cdot \alpha)) \quad (52)$$

$$w_{max} \leftarrow w_{mean} + (w_{max} - w_{mean}) x (1 + (\beta \cdot \alpha)) \quad (53)$$

If input class is recognized into wrong class ($\varphi \geq 0$), then decrease triangular width using equation (54) and equation(55).

$$w_{min} \leftarrow w_{mean} - (w_{mean} - w_{min}) x (1 - (\beta \cdot \alpha)) \quad (54)$$

$$w_{max} \leftarrow w_{mean} + (w_{max} - w_{mean}) x (1 - (\beta \cdot \alpha)) \quad (55)$$

If $\mu_1 = 0$ and $\mu_2 = 0$, then it means both reference vector cannot recognize input vector. Therefore, fuzzy triangular vectors must be increased using equation(56) and equation(57) where γ is a constant value of 0.1.

$$w_{min} \leftarrow w_{mean} - (w_{mean} - w_{min}) \times (1 - (\gamma \cdot \alpha)) \quad (56)$$

$$w_{max} \leftarrow w_{mean} + (w_{max} - w_{mean}) \times (1 + (\gamma \cdot \alpha)) \quad (57)$$

3. Results and Analysis

In this research we used MIT-BIH *Arrhythmia* database [14]. The dataset has 48 records from research subjects in BIH *Arrhythmia Laboratory*. The dataset have been taken from 1975 to 1979. Each record has ± 3 minutes duration. Sample of the dataset is visualized in figure 11. The dataset used in this research consists of six class as, they are: 1) Normal beat (NOR); 2) Left Bundle Branch Block (LBBB); 3) Right Bundle Branch Block (RBBB); 4) Premature Ventricular Contraction (PVC); 5) Paced beat (PACE); 6) Atrial Premature beat (AP).

In this research, we use two experimental scenarios. The first scenario is conducted to measure classifier performance according the number of features. The second scenario is used to measure impact of round robin data to performance classifier. In this experiment, learning rate was set to 0.05, and maximum epoch was set to 250. The result of first scenario is shown in Figure 12.

We use six number of feature variation. Figure 12 shows that FLGLVQ has the best performance among them for any number of features. The best accuracy reached by FNGLVQ is 94.07%, when 300 features are used. FNLVQ-PSO is consistent in the second rank.

FNLVQ-PSO has best accuracy in 300 features dataset with 92.54 % accuracy. GLVQ and LVQ 2.1 compete in the third and fourth ranks. GLVQ reaches its best performance in 86 features dataset with 84.58 % accuracy, whereas LVQ2.1

has its best performance in 32 features dataset with 86.55% accuracy. FNLVQ-MSA is consistent in fifth rank. Its best accuracy is 83.66% which is in 300 features case. FNLVQ and LVQ2 compete for sixth and seventh rank. FNLVQ has best performance in 300 features dataset with 82.25% accuracy. LVQ2 reach its best performance in 32 features dataset with 82.29% accuracy. The worst classifier in this scenario is LVQ1. Its highest accuracy is 74.62% which is in 86 features data.

For the second scenario, we used 300 features dataset, and six variation of round robin value. In round robin scenario, training data is formatted as table below. Cm is data with label class m, and n is number of round robin value.

The result of the second scenario is shown in Figure 13. Figure 13 shows that GLVQ surprisingly has the best performance for any kind of round robin value. It reach best performance in 30 round robin data with 98.12% accuracy. FNGLVQ, LVQ 2.1, and FNLVQ-PSO compete for second, third and fourth ranks. FNGLVQ reached its best performance in 30 round robin case with 94.31% accuracy. LVQ2.1 also reached its best performance in 30 round robin case with 98.04% accuracy. Whereas FNLVQ-PSO reached its best performance in 50 round robin case with 90.43% accuracy. FNLVQ, FNLVQ-MSA, and LVQ2 compete for fifth, sixth, and seventh position. LVQ2 best accuracy is 86.75% which is in 50 round robin case. FNLVQ-MSA has best accuracy in 30 round robin case with 86.12, and so does the eighth position of FNLVQ 84.50% accuracy. The worst classifier for round robin data is LVQ1 with 74.78% best accuracy in 50 round robin case. Overall, Figure 3 shows that classifiers accuracy have been enhanced when round robin value was less than or equal to 50.

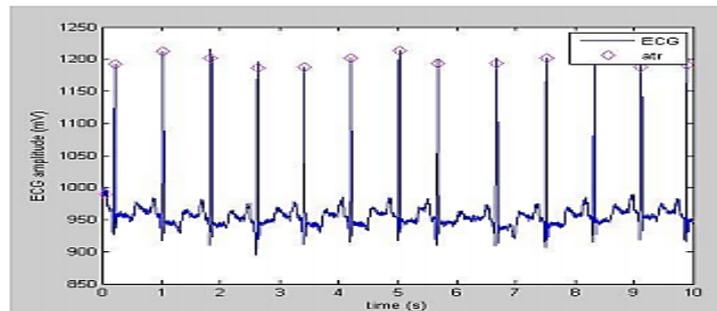


Figure 11. Sample of MIT-BIH dataset

TABLE 1.
ROUND ROBIN FORMAT

Cycle-1 (n x m data)	Cycle-1 (n x m data)
C1,C2..Cm,...,C1,C2....,Cm	C1,C2..Cm,...,C1,C2....,Cm

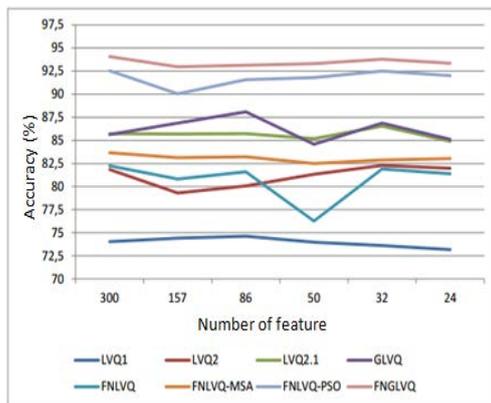


Figure 12. Classifier performance according to the number of feature (scenario-1)

From those scenarios we can say that round robin dataset has no significant improvement for classifiers with fuzzy membership function such as FNLVQ, FNLVQ-PSO, FNLVQ-MSA, and FNLVQ. While it gives significant improvement for non-classifiers based on LVQ. When dataset is formatted as round robin, the sequence is always sorted as C1, C2, ..., Cm. After update reference vector for input instance with C1 label, the next instance is always not C1. It causes classifiers guess it as C1 which is wrong class. Therefore classifiers will widen its fuzzy triangle regarding the misclassification. Too wide fuzzy triangle is not good. It can cause misclassification. Therefore round robin dataset give no significant improvement for classifiers with fuzzy

4. Conclusion

From the research performed, it can be concluded that automatic arrhythmias detection system has been successfully implemented. In this research, *Arrhythmia* dataset from MIT-BIH is used to measure system performance. Result of first experiment said that for non-round robin dataset, the system can reach 94.07% accuracy with FNLVQ algorithm. It is followed by FNLVQ-PSO with 92.54% of accuracy, GLVQ with 88.09%, LVQ2.1 with 86.55%, FNLVQ-MSA with 83.66%, LVQ2 with 82.29%, FNLVQ with 82.25%, and LVQ1 with 74.62% of accuracy. The result of the first experiment said that round robin data with less than or equal 50 can enhance accuracy. For round robin dataset, the system can reach 98.12% of accuracy with GLVQ algorithm. It is followed by LVQ2.1 with 98.04%, FNLVQ with 94.31%, FNLVQ-PSO with 90.43%, LVQ2 with 86.75%, FNLVQ-MSA with 86.12%, FNLVQ with 84.50%, and LVQ1 with 74.78%.

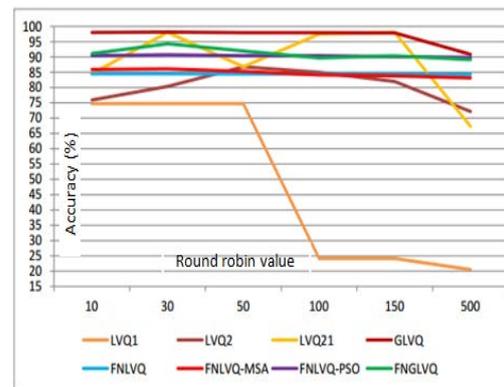


Figure 13. Classifier performance according to the number of round robin value (scenario-2)

References

- [1] Bryg, R. M. "Heart Disease and Abnormal Heart Rhythm". 2009. <http://www.webmd.com/heart-disease/guide/heart-disease-abnormal-heart-rhythm?page=3>
- [2] Decrease mortality caused by coronary heart disease. <http://www.ugm.ac.id/en/berita/1593-penguakuan.prof.bambang.irawan%3A.upaya.menurunkan.angka.kematian.akibat.jantung.koroneron> March 2, 2014
- [3] T. Yeap, F. Johnson, and M. Rachniowski, "ECG beat classification by a neural network," in *Engineering in Medicine and Biology Society*, 1990., Proceedings of the Twelfth Annual International Conference of the IEEE, pp. 1457–1458, nov 1990.
- [4] Y. Ozbay and B. Karlik, "A Recognition of ECG Arrhythmias using artificial neural networks," in *Engineering in Medicine and Biology Society*, 2001. Proceedings of the 23rd Annual International Conference of the IEEE, vol. 2, pp. 1680–1683 vol.2, 2001.
- [5] R. Ghongade and A. Ghatol, "A brief performance evaluation of ECG feature extraction techniques for artificial neural network based classification," in *TENCON 2007 - 2007 IEEE Region 10 Conference*, pp. 1–4, 30 2007-nov. 2 2007.
- [6] Setiawan, I.M.A.; Imah, E.M.; Jatmiko, W., "Arrhythmia classification using Fuzzy-Neuro Generalized Learning Vector Quantization," *Advanced Computer Science and Information System (ICACSIS), 2011 International Conference on*, vol., no., pp.385,390, 17-18 Dec. 2011.
- [7] A. Sato and K. Yamada, "A formulation of learning vector quantization using a new misclassification measure," in *Proceedings of the 14th International Conference on Pattern*

- Recognition-Volume 1 - Volume 1, ICPR '98, (Washington, DC, USA), pp. 322–, IEEE Computer Society, 1998
- [8] T. Kohonen, "Learning Vector Quantization for Pattern Recognition," Report TKK-F-A 601, Helsinki University of Technology, Espoo, Finland, 1986.
- [9] B. Kusumoputro, H. Budiarto, and W. Jatmiko, "Fuzzy-neuro LVQ and its comparison with fuzzy algorithm LVQ in artificial odor discrimination system," *ISA Transactions*, vol. 41, no. 4, pp. 395 – 407, 2002.
- [10] Jatmiko, W.; Rochmatullah; Kusumoputro, B.; Sanabila, H. R.; Sekiyama, K.; Fukuda, T., "Visualization and statistical analysis of fuzzy-neuro learning vector quantization based on particle swarm optimization for recognizing mixture odors," *Micro-NanoMechatronics and Human Science*, 2009. *MHS 2009. International Symposium on*, vol., no., pp.420,425, 9-11 Nov. 2009.
- [11] G. D. Clifford, "Baseline wander removal open source ecg package," 2005
- [12] Ghongade and A. Ghatol, "A brief performance evaluation of ecg feature extraction techniques for artificial neural network based classification," in *TENCON 2007 - 2007 IEEE Region 10 Conference*, pp. 1 –4, 30 2007-nov. 2 2007.
- [13] Kennedy, J.; Eberhart, R. "Particle Swarm Optimization". *Proceedings of IEEE International Conference on Neural Networks IV*. pp. 1942–1948. (1995)
- [14] G. B. Moody, "Mit-bih arrhythmia database." <http://physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm>. May 1997.