exeley INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS

Article | DOI: 10.21307/ijssis-2019-009 Issue 1 | Vol. 12 (2019)

Developing smart Tele-ECG system for early detection and monitoring heart diseases based on ECG signal: progress and challenges

Wisnu Jatmiko, M. Anwar Ma'sum,
Hanif Arief Wisesa and
Hadaiq Rolis Sanabila

Faculty of Computer Science,
Universitas Indonesia Kampus
Baru UI Depok, 16424,
Jawa Barat, Indonesia.

E-mails: wisnu@cs.ui.ac.id,
muhammad_anwar@cs.ui.ac.id,
hadaiq@cs.ui.ac.id

The paper was edited by
Ha-Duong Ngo.

Received for publication April 2, 2018.

Abstract

Technology is developed to benefit society. One of the applications of technology in the healthcare sector is telehealth monitoring system. The system proposes a new way of communication between the doctor and the patient, even in a very remote location. In this paper, we elaborate the progress and challenges regarding the development of Tele-ECG in Indonesia, which includes data acquisition, feature extraction, data compression, classification algorithm, mobile and web development system and small device implementation on an FPGA board. The classification is conducted by using LVQ, GLVQ, FNLVQ, FNLVQ-PSO, FNGLVQ, and AM-GLVQ. The compression is conducted by using SPIHT algorithm. Tele-ECG can assist in monitoring heartbeat anomalies and reduce the risk of heart attack. It could also be a solution for infrastructure discrepancy in healthcare.

Keywords
Telehealth, ECG, LVQ, GLVQ, FNLVQ, FNLVQ-PSO, FNGLVQ, AM-GLVQ, SPIHT, FPGA.

Mobile technologies provide a medium for long-distance communication systems, where the users interact with each other without having to physically meet. Long-distance communication systems, using mobile technologies, are useful for critical and urgent matter, such as healthcare. Many research works regarding the implementation of these systems in healthcare services have been conducted. This is shown by researchers who have developed these kinds of systems (Dabiri et al., 2009; Feng et al., 2015). In this field, the systems are commonly known as telehealth monitoring systems. These systems are primarily focused on the interaction between the doctor and the patient. The doctor could provide a consultation via the mobile device to monitor the patient's condition. In addition, the patient could input his/her own condition, so the doctor could review the progress of the patient's treatment. Another common feature included in these devices is a first response classification system. These telehealth monitoring systems could detect the anomalies in the data obtained from the patient. Several examples of these classified data are ECG (electocardiogram), USG (ultrasound images), EEG (electroencephalogram), and many other data. These data will be classified by the system according to the disease that could possibly cause the symptoms. The data will be sent to the appropriate practitioner in order to respond to the patient as fast as possible. The architecture of a telehealth monitoring system can be seen in Figure 1.

In the present day, there are a myriad of telehealth monitoring systems that have been researched and developed. An example of this telehealth system is the tele-USG system. This system is used to monitor fetus in a pregnant mother. The mother can consult the doctors to know the progress and condition of her fetus by using a mobile device. The system has been developed and implemented in our previous studies

---

© 2019 Authors. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 4.0 License https://creativecommons.org/licenses/by-nc-nd/4.0/

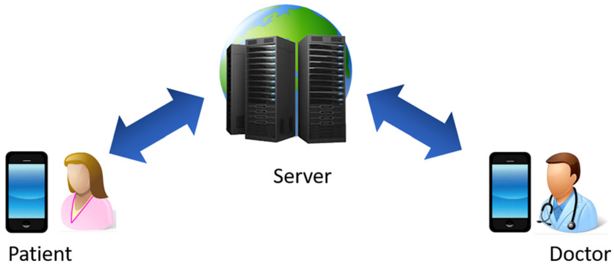


Figure 1: A telehealth system architecture.

(Jatmiko et al., 2015a). Our previous research in this field also includes an automatic detection for fetal organs, such as fetal head and fetal humerus in an USG image. The automatic detection of the organs was implemented in the Tele-USG system (Ma'sum et al., 2015a, 2015b; Jatmiko et al., 2015b).

One of the most interesting telehealth monitoring systems to be researched is Tele-ECG. This system is commonly researched due to the fact that the cardiovascular-type disease is one of the most common causes of death by a non-communicable disease (W. H. Organization, 2016). Therefore, many efforts have been done in order to try and solve this problem, which is divided into several fields. In the field of ECG data quality and compression, Tobón et al. proposed an electrocardiogram quality index to maintain the quality of the ECG signals that were obtained for use in an automatic classification system (Tobon et al., 2016). The quality is maintained by measuring the change of spectral image representation. In the field of ECG data recognition, an implementation of the telehealth monitoring system was conducted by Marcolino et al. (Soriano Marcolino et al., 2016). In several other studies, QRS algorithms was utilized to analyze the electocardiogram data in a telehealth monitoring system (Hayn et al., 2012; Khamis et al., 2016).

In this study, we examine the complete telehealth monitoring systems for cardiovascular disease patients. We focus on transmitting electrocardiogram (ECG) data by the patients in the telehealth monitoring system. The analysis includes an application for a consultation between the doctors and the patients. This study also analyzes several compression algorithms that are utilized to compress the data, before sending to the server, to minimize the size of the data. Furthermore, this study also analyzes and tracks the development of classification algorithms for classifying and analyzing the ECG data for a first response inside the system. In addition, the ECG pre-processing methods are also included in this study. To complete this paper, we have also included several implementations of telehealth systems that focus on monitoring cardiovascular disease.

Data preprocessing

In this paper, we will elaborate the progress and challenges of developing a smart tele-ECG device by examining several data pre-processing methods. Data pre-processing method is crucial in this study to extract the features of the heartbeat. These features are going to be used to process the data using various classification algorithms. This section will be divided into three different sub-sections. The first sub-section explains the anatomy of the ECG data. The next section will discuss about the normalization of the beats, which involves the process of removing the baseline wander, commonly known as baseline wander removal. The final sub-section will explain about the individual beat segmentation from the ECG data.

Electrocardiogram signal

Before we explain the pre-processing stage of ECG data, we need to know about the ECG data (Tawakal et al., 2012). A single beat in an ECG signal is comprised of several sections and points. The first point is the P point, which is the first small peak of the ECG signal. After the P point, there is a decline as the beat enters the Q point. It then peaks into the R peak, which is the tallest peak in the beat, and it declines again into the S point. Finally, the beat will reach the T point. The distance between the Q, R, and S is called a QRS complex. The distance after the P reaches its peak and the Q point is called a PR segment, whereas the distance before the P reaches its peak and the Q point is called the PR interval. The QT interval is the distance between the Q and the T point (after the peak), and the ST segment is the distance between the S and T point before the peak. The distance between two R peaks is called the RR interval. ECG beat is aptly illustrated in Figure 2.

Baseline wander removal

In the pre-processing stage of data, the data are usually normalized so that they are uniformed. This normalization process is key to having a correct classification system later on. In the case of the ECG data, the data are generated from the sensor continuously. During this continuous process, there are occasional noises that could affect the recording of the ECG beats. One example of the noise that

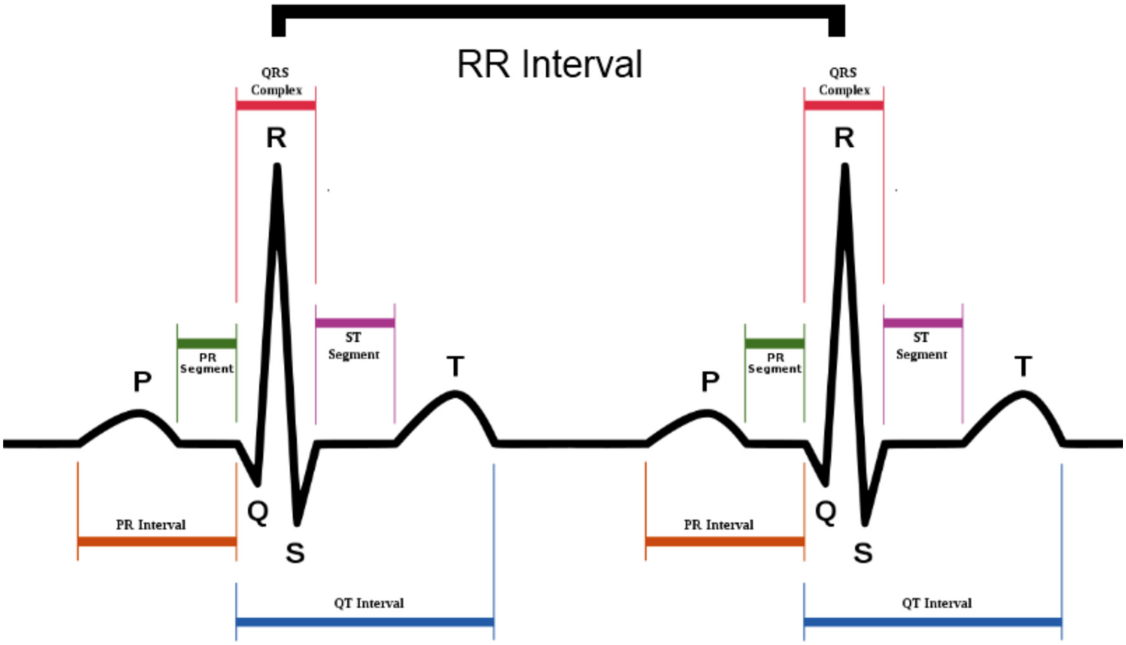
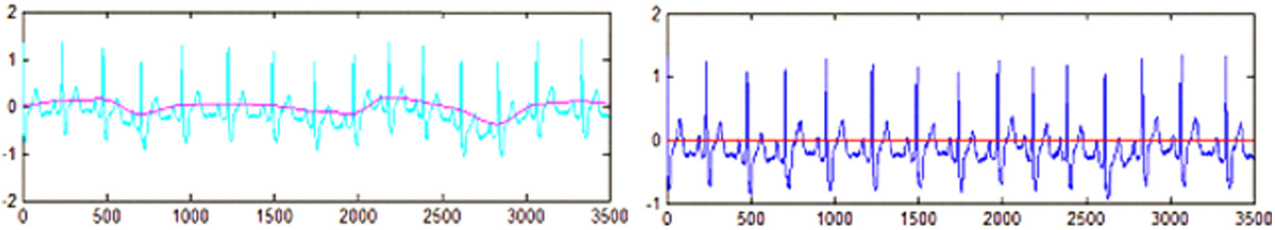


Figure 2: The illustration of QRS complex in an ECG signal.

could affect the ECG data is low frequency noise. This noise will affect the ECG data in terms of the up and down shifting of the ECG data. Therefore, the acquired data do not lie completely on a straight, iso-centric line. This is a challenge that needs to be addressed in order to have uniformed ECG data.

In a previous study conducted by Ima et al. (Setiawan et al., 2011), cubic spline interpolation is utilized in order to remove the baseline wander. The method will estimate the baseline from the ECG data so that the ECG data could be normalized and straightened on the iso-centric line. The PR segment was used in that study to generate the cubic spline, which, in turn, will estimate the baseline of the data. Therefore, it is essential to detect the QRS complex and PR segment from the data to conduct this process. The process of baseline wander removal can be seen in Figure 3.

In 2012, Isa et al. used discrete wavelet transform in order to remove the baseline wander from the ECG signal (Isa et al., 2012a). This wavelet transformation method provides a multiresolution analysis that helps to remove the baseline wander from the ECG signal. This is conducted according to a previous statement from the study of Sargolzaei et al., which mentioned that the baseline spectrum is below the spectrum of the ECG signal (Sargolzaei et al., 2009). They mentioned that the baseline wander could be detected from the approximation coefficients inverse wavelet transform during the time the energy coefficient in a level of the decomposition reaches the local minima.



Original ECG and baseline cubic spline estimation

Result of baseline substracted ECG and isoelektrik line

Figure 3: The ECG data before and after the cubic spline interpolation in order to remove the baseline wander.

Individual beat segmentation

The individual beat segmentation is a three-step process. The first process is the beat extraction process. This process will segment the continuous beats in an ECG data into individual beats. The next process involves removing the outliers on the beats. Finally, the features of the beat will be extracted and processed by the classification algorithm.

Beat extraction

As mentioned previously, the ECG data obtained from the ECG sensors will be continuous. The example of these continuous data can be seen in Figure 3, where the ECG data are comprised of many smaller individual beats. However, in order to extract the features of the beats, we will need to extract the continuous beats into individual beats. This process will utilize the cutoff technique, as previously conducted in the study of Jatmiko et al. (2016). The method approximates each individual beat to be 300 data points in length. In addition, the center of the individual beat will be on the R peak. This means that the individual beat starts on R-150 data point and ends on R + 149 data point. The continuous signal will then be cutoff at these points in order to extract the individual beats. The cutoff technique can be seen in Figure 4.

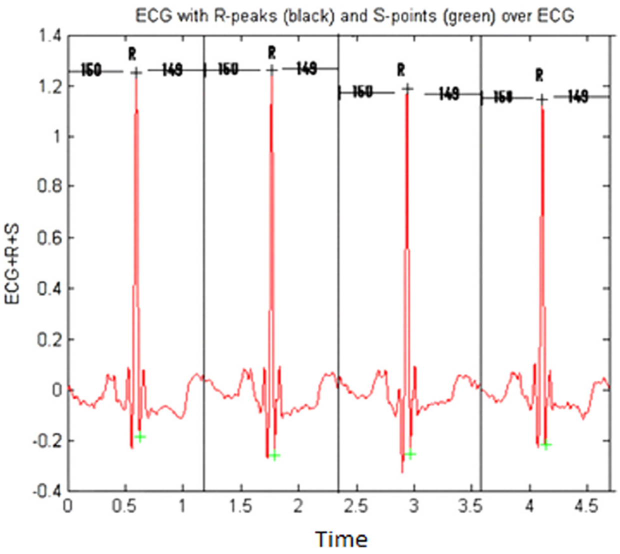


Figure 4: The individual beat segmentation.

Outlier removal

In the ECG data, there will be various unnecessary beats that are captured by the sensor while recording the data. These unnecessary beats tend to distract the classification process, which is conducted on the later stage. Therefore, these unnecessary beats need to be removed. These unnecessary beats are referred to as outliers in the data. These outliers are found outside of the distribution of the data; hence, Ima et al. and Jatmiko et al. used a simple outlier removing procedure known as the interquartile range (IQR) technique. The technique will determine the outlier by creating a boundary from the percentile of the data. In the study of Jatmiko et al. (2016), they chose the upper quartile (Q_1) in the 25th percentile and chose the lower quartile (Q_2) in the 75th percentile. The interquartile range (IQR) can be calculated by using the following equation:

$$\textit{Inter quartile range}\,(\mathrm{IQR}) = Q_3 - Q_1. \tag{1}$$

The lower extremity level and the higher extremity level for the boundary of the outliers can be calculated by using the following equation:

$$\begin{aligned} \textit{Lower level} &= Q_1 - 1.5 \times \mathrm{IQR}, \\ \textit{Higher level} &= Q_3 + 1.5 \times \mathrm{IQR}. \end{aligned} \tag{2}$$

After Equation (2) has been calculated, it is applied to every feature of the data. However, it should be noted that the correlation between each feature is not considered. Afterwards, the beat located outside the extremity levels (either from the lower and higher levels) will be removed from the whole data set. The process of removing these outliers using IQR can be seen in Figure 5, and the 'before and after' data of the outlier removal process can be seen in Figure 6.

Feature extraction

Feature extraction is one of the key aspects of pre-processing techniques, as this process will select the features that will be used by the classification process later on. Therefore, it is essential to extract the best features from the data set, as selecting better features will lead to better outcome during the classification process. To extract the features, wavelet transformation is proven effective, as shown in the studies of Setiawan et al. (2011) and Jatmiko et al. (2016). The wavelet transform definition can be seen in the following equation:

$$W_s f(x) = f(x) \times \psi_s(x) = \frac{1}{s}\int_{-\infty}^{+\infty} f(t)\,\psi\left(\frac{x-t}{s}\right)dt. \tag{3}$$

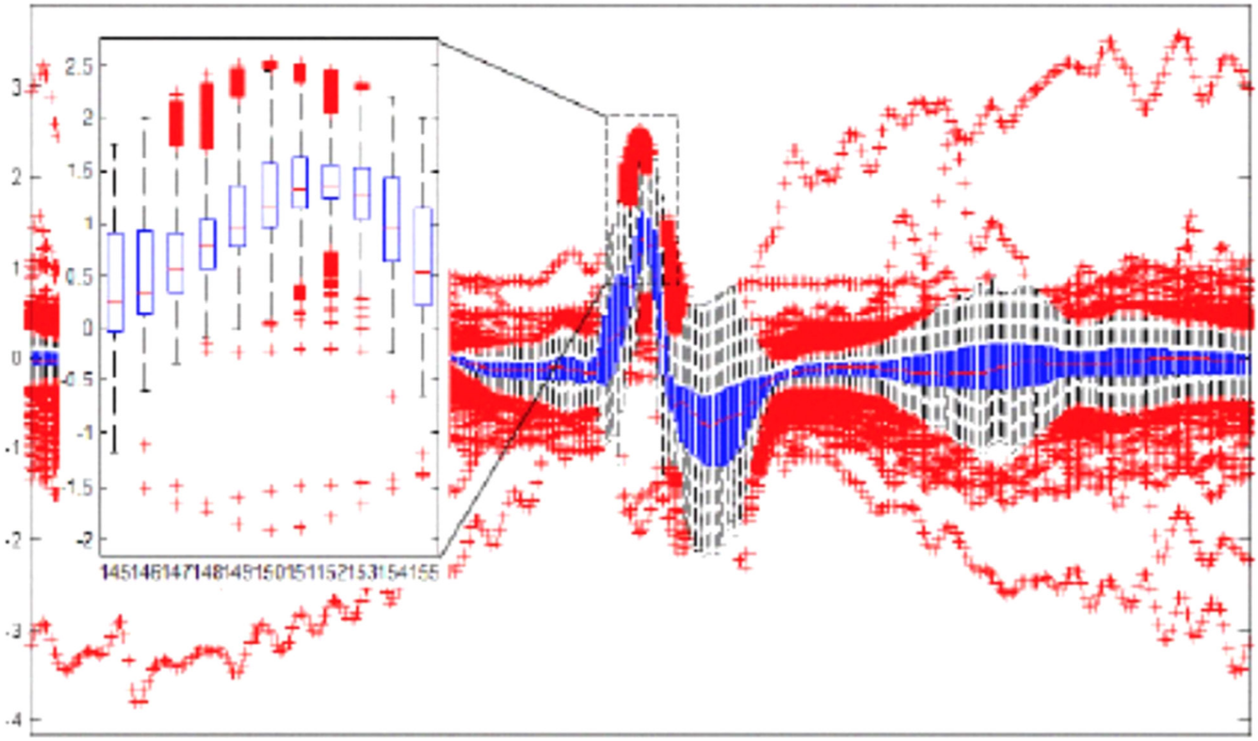
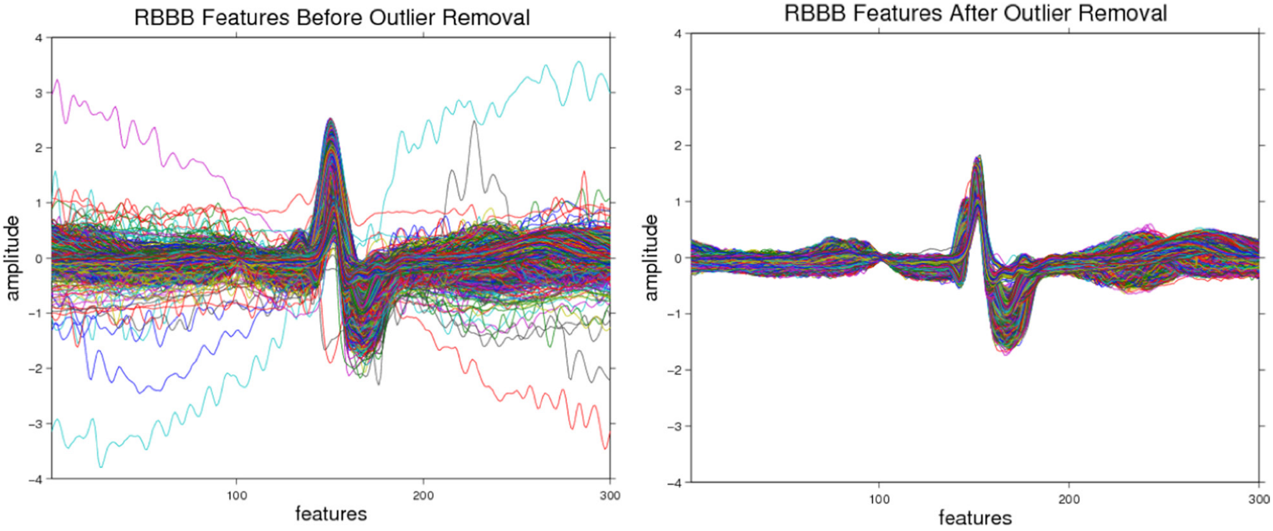


Figure 5: Outlier removal using IQR.

In Equation (3), the notation s denotes the scaling factor, the basic wavelet $\psi(x)$ with an s scaling factor dilation is $\psi_s(x)=\frac{1}{s}\psi\left(\frac{x}{s}\right)$. Let $s=2j$, where j is an element of the integral set. The wavelet transform is called a dyadic wavelet transform (Zhao and Zhang, 2005). Mallat algorithm (Mallat, 1989) can calculate the dyadic wavelet transform from a digital signal. The Mallat algorithm is shown in the following equations:

$$S_{2j}f(n)=\sum_{k\in Z}h_k S_{2j-1}f\left(n-2^{j-1}-k\right), \tag{4}$$

$$W_{2j}f(n)=\sum_{k\in Z}g_k S_{2j-i}f\left(n-2^{j-1}-k\right). \tag{5}$$


RBBB Features Before Outlier Removal

RBBB Features After Outlier Removal

Figure 6: Beat features before and after outlier removal using IQR.

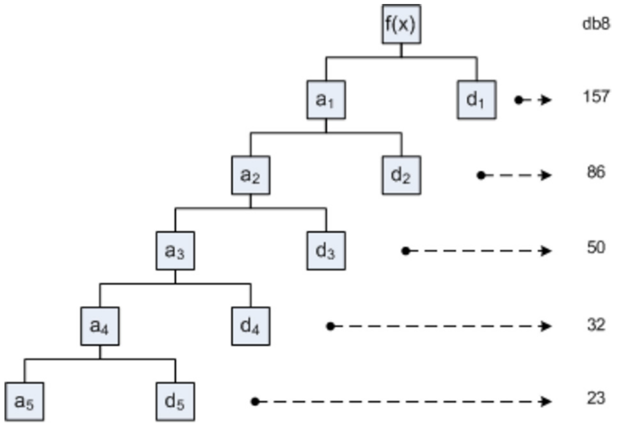


Figure 7: Daubechies 8 5-level decomposition (Setiawan et al., 2011).

In Equation (4), the notation S_{2j} denotes the smoothing operator, whereas the low-frequency coefficients are represented as $S_{2j}f(n)$. The coefficient is an approximation of the original signal $S_{2j}f(n)=a_j{\cdot}a_j$. In Equation (5), the $W_{2j}f(n)$ denotes the high-frequency coefficient. The high-frequency coefficient is the detail of the original signal $W_{2j}f(n)=d_j{\cdot}d_j$. In order to obtain the important information in the wavelet coefficients, a very crucial part in wavelet theory is to select the mother wavelet and also the decomposition level. In the case of the arrhythmia's usage of the ECG signals, Daubechies is selected as the mother wavelet since it achieves a better performance as, described by Senhadji et al. (1995). In the studies of Setiawan et al. (2011) and Jatmiko et al. (2016), Daubechies level 8 proved to have a better result than the other mother wavelet. The research then used the db8 as the mother wavelet and decomposed the individual beats from 1 to 5. The decomposition of Daubechies 8 can be seen in Figure 7.

The coefficient that will represent the signal has to be selected properly in order to select the best features that represent the signal. In the studies of Setiawan et al. (2011) and Jatmiko et al., (2016), d2, d3, d4, and d5 represent the high-frequency signal. a1, a2, a3, a4, and a5 represent the signal approximation; hence, it has the important features of the signal. Therefore, the study selects the coefficients for the beat. The decomposed ECG signal that was conducted by Jatmiko et al. (2016) can be seen in Figure 8. There are imbalanced data in ECG signal, which could degrade the performance on recognizing a minor, but significant, class. A study has been performed for examining the performance of oversampling on imbalanced data (Sanabila et al., 2016).

Classification: early algorithm

In this section, we will be discussing two basic and early classification algorithms that are commonly

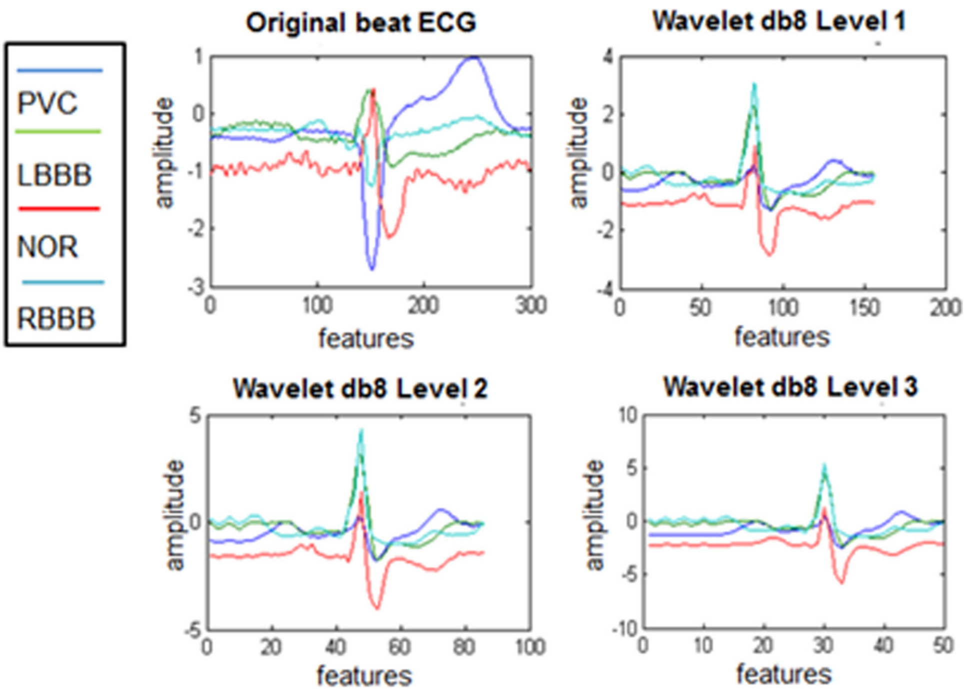


Figure 8: The ECG decomposed signal.

used in classifying ECG signal data. These algorithms are learning vector quantization (LVQ) algorithm and generalize learning vector quantization. The learning vector quantization is an earlier version of the generalize learning vector quantization algorithm. The generalize learning vector quantization algorithm provides several improvements in accuracy and cost function compared to the learning vector quantization algorithm.

Learning vector quantization (LVQ)

The learning vector quantization (LVQ) algorithm is an algorithm that is based on the self-organizing map (SOM) algorithm (Kohonen, 1990). This algorithm was first proposed by Kohonen. The algorithm modified the self-organizing map into a supervised method for classification by modifying the learning and update procedures. In addition, the class is also assigned to each codebook. The basic function of this algorithm is to measure the codebook and adjust it to the input vector. If the winning codebook and the input vector share the same class, then the winning codebook will be pulled closer to the input vector. However, if the winning codebook and the input vector do not share the same class, then the winning codebook is pushed away. This is shown in Equations (7) and (8). The closest codebook vector is defined in the following equation (Figure 9):

$$w_c = min\|x - w_i\|. \tag{6}$$

In Equation (6), notation w_c denotes the closest codebook vector, the input vector is defined as x and the notation w_i denotes the codebooks with a previously assigned class. These classes are denoted by i. The equation for pushing and pulling the codebook can be seen in Equations (7) and (8), respectively:

$$w_c(t+1) = w_c(t) + \alpha(t)\left[x(t) - w_c(t)\right]. \tag{7}$$

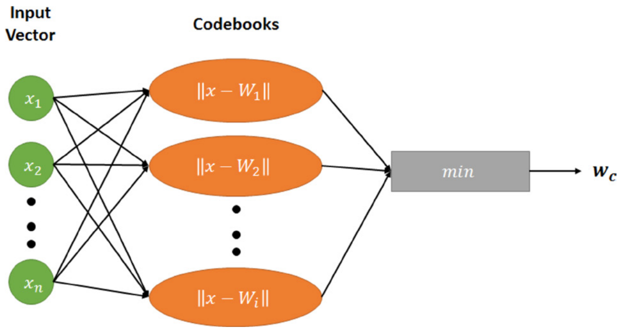


Figure 9: The architecture of the LVQ algorithm.

Equation (7) applies when x and the closest codebook are in the same class:

$$w_c(t+1) = w_c(t) - \alpha(t)\left[x(t) - w_c(t)\right]. \tag{8}$$

Equation (8) applies when x and the closest codebook are on different classes. The value of α decreases over time and $0 < \alpha < 1$.

According to Nascimento (2005), there are several known limitations of the LVQ algorithm. These limitations include the overtraining problems and non-well specified clustering goals. These drawbacks will be further addressed by the generalize learning vector quantization (GLVQ) algorithm, which is explained in the next section.

Generalize learning vector quantization (GLVQ)

In 1996, Sato and Yamada proposed an improvement of the learning vector quantization algorithm, which is called the generalize learning vector quantization (GLVQ) algorithm (Sato and Yamada, 1995). The advantage of this algorithm is that it minimizes the cost function from the LVQ algorithm. In addition, it also minimizes the rate of the misclassification error. The reference vector for the input vector class $C_x = C_{w_1}$ is defined by w_1, and the closest reference class from a different class $C_x \neq C_{w_2}$ is defined as w_2. The error of misclassification could be defined as Equation (9):

$$\varphi(x) = \frac{d_1 - d_2}{d_1 + d_2}. \tag{9}$$

In Equation (9), the distance between the input vector x and the reference vector w_1 is denoted by d_1, whereas the distance between the input vector x and the reference vector distance between w_2 and input vector x is denoted by d_2. Equation (10) will minimize the cost function during the learning process:

$$S = \sum_{i=1}^{N} f(\varphi(x)). \tag{10}$$

In Equation (10), $f(\varphi(x))$ is a rising monotonic function. To minimize S, we used the steepest descent method. The update rules for w_1 and w_2 are defined in the following equation:

$$w_i \leftarrow w_i - \alpha \frac{\delta S}{\delta w_i}, \quad i = 1,2. \tag{11}$$

We could assume that the discriminant function is a Euclidian; therefore, Equation (11) can be formulated as Equation (12) and Equation (13):

$$\frac{\delta S}{\delta w_1} = \frac{\delta S}{\delta \varphi}\frac{\delta \varphi}{\delta d_1}\frac{\delta d_1}{\delta w_1} = -\frac{\delta f}{\delta \varphi}\frac{4d_2}{(d_1+d_2)^2}(x-w_1), \tag{12}$$

$$\frac{\delta S}{\delta w_2} = \frac{\delta S}{\delta \varphi}\frac{\delta \varphi}{\delta d_2}\frac{\delta d_2}{\delta w_1} = -\frac{\delta f}{\delta \varphi}\frac{4d_1}{(d_1+d_2)^2}(x-w_2). \tag{13}$$

The update rules can be derived as Equation (14) and Equation (15):

$$w_1 \leftarrow w_1 + \alpha\frac{\delta f}{\delta \varphi}\frac{4d_2}{(d_1+d_2)^2}(x-w_1), \tag{14}$$

$$w_2 \leftarrow w_2 + \alpha\frac{\delta f}{\delta \varphi}\frac{4d_2}{(d_1+d_2)^2}(x-w_2). \tag{15}$$

In GLVQ algorithm, the sigmoid function that is used as the monotonic function can be defined as Equation (16) and Equation (17):

$$f(\varphi,t) = \frac{1}{1+e^{-\varphi t}}, \tag{16}$$

$$\frac{\delta f}{\delta \varphi} = f(\varphi,t)\left(1-f(\varphi,t)\right). \tag{17}$$

Classification: advanced algorithm

In this section, we will examine several advanced algorithms that are used for classifying ECG signal data. The algorithms that will be discussed are fuzzy-neuro learning vector quantization (FNLVQ), fuzzy-neuro learning vector quantization particle swarm optimization (FNLVQ-PSO), fuzzy-neuro generalize learning vector quantization (FNGLVQ), and the adaptive multilayer generalize learning vector quantization (AM-GLVQ). These algorithms are developed from early algorithms that were explained in the third section by modifying the structure, membership function, update rule, or adding optimization function.

Fuzzy-neuro learning vector quantization

In the previous research, Benyamin et al. improved LVQ classifier by applying fuzzy membership function (Kusumoputro et al., 2002). The new algorithm is called fuzzy-neuro learning vector quantization (FNLVQ). In the research, FNLVQ is deployed in odor discrimination system. The FNLVQ architecture is shown in Figure 10. FNLVQ has three layers, that is input layer, output layer, and hidden layer. The input layer is used for representing the feature of the data, the hidden layer is used for representing the reference vectors (codebook) of the classifier, and the output layer is used for representing the output class. As the reference vector uses fuzzy membership function, each vector has three components/values, that is $w_{ij}(l)$, w_{ij}, $w_{ij}(r)$, which represent min, mean, and max of the fuzzy triangle, respectively.

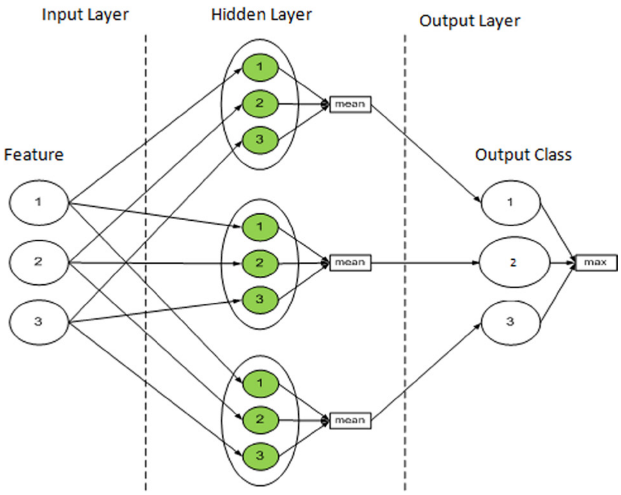


Figure 10: FNLVQ architecture (Kusumoputro et al., 2002).

The FNLVQ uses winner-take-all approach in training phase as used in LVQ method. The reference vector is updated on the basis of the winner vector. Same as in LVQ, the winner vector is the closest vector to the input, which is the vector with highest similarity value. The similarity value for FNLVQ is the intersection of input fuzzy triangle and reference vector fuzzy triangle. The method to compute the similarity value for two fuzzy triangles is shown in Figure 11. In Figure 11, *x* is the input vector, h_x is the fuzzy membership function of vector *x*, w_i is the reference vector for class *i*, and h_{wi} is the fuzzy membership function of reference vector w_i.

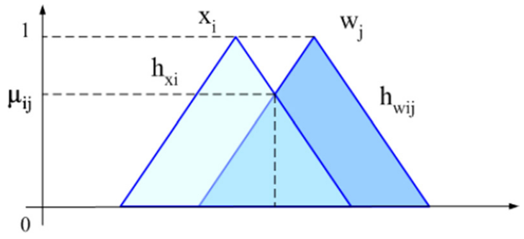


Figure 11: Computing similarity in FNLVQ (Kusumoputro et al., 2002).

Mathematically, the similarity value for feature *j* in class *i*, μ_{ij}, is calculated by using the following equation:

$$\mu_{ij} = \max\left(h_{xj} \Lambda h_{wij}\right). \tag{18}$$

Then, the similarity value for class *i* is calculated by using the following equation:

$$\mu_{j} = \min\left(\mu_{ij}\right). \tag{19}$$

The update for reference vector is conducted by following three conditions. The first condition is that if the winner class label is the same as the input vector class label, then the reference vector will be updated by using the following equations:

$$d_{ij} \leftarrow \alpha \times \left(\left(\{1-\mu_i\}\right) \times \left(\{x_j - w_{ij}(t)\}\right)\right), \tag{20}$$

$$w_{ij} \leftarrow w_{ij} + d_{ij}, \tag{21}$$

$$w(l)_{ij} \leftarrow w(l)_{ij} + d_{ij}, \tag{22}$$

$$w(r)_{ij} \leftarrow w(r)_{ij} + d_{ij}. \tag{23}$$

Then, Equations (20)–(23) are followed by widening or narrowing of the fuzzy triangle using the following equations:

$$w(l)_{ij} \leftarrow w(l)_{ij} + \left(\beta \times \left(w_{ij} - w(l)_{ij}\right)\right), \tag{24}$$

$$w(r)_{ij} \leftarrow w(r)_{ij} - \left(\beta \times \left(w(r)_{ij} - w_{ij}\right)\right). \tag{25}$$

The second condition is that if the vector class is not the same as the input label class, then the reference vector will be updated using the following equations:

$$d_{ij} \leftarrow \alpha \times \left(\left(\{1-\mu_i\}\right) \times \left(\{x_j - w_{ij}(t)\}\right)\right), \tag{26}$$

$$w_{ij} \leftarrow w_{ij} - d_{ij}, \tag{27}$$

$$w(l)_{ij} \leftarrow w(l)_{ij} - d_{ij}, \tag{28}$$

$$w(r)_{ij} \leftarrow w(r)_{ij} - d_{ij}. \tag{29}$$

Then, Equations (26)–(29) are continued by fuzzy triangle adjustment (widening or narrowing) that is conducted by using the following equations:

$$w(l)_{ij} \leftarrow w(l)_{ij} + \left(\beta \times \left(w_{ij} - w(l)_{ij}\right)\right), \tag{30}$$

$$w(r)_{ij} \leftarrow w(r)_{ij} - \left(\beta \times \left(w(r)_{ij} - w_{ij}\right)\right). \tag{31}$$

The third and final condition is that if the winner class label is the same as the label of input class, but the similarity value is 0, then the reference vector will be updated by using Equations (20)–(23). Then, the fuzzy triangle is widened and narrowed by using the following equations:

$$w(l)_{ij} \leftarrow w(l)_{ij} - \left(\beta \times \left(w_{ij} - w(l)_{ij}\right)\right), \tag{32}$$

$$w(r)_{ij} \leftarrow w(r)_{ij} + \left(\beta \times \left(w(r)_{ij} + w_{ij}\right)\right). \tag{33}$$

In the equations above, α is the learning rate value and β is a constant value between 0 and 1. Concerning testing in FNLVQ, the process is conducted by computing the closest vector (winner vector) to the input vector. The predicted class from the classifier is this winner vector class label.

Fuzzy-neuro learning vector quantization particle swarm optimization (FNLVQ-PSO)

In a previous research, Jatmiko et al. added optimization process for FNLVQ training. The authors applied particle swarm optimization (PSO) algorithm to train FNLVQ, which resulted in FNLVQ-PSO (Jatmiko et al., 2009). PSO is an optimization algorithm that uses colony of agents to find the optimum value (Kennedy, 2011). In FNLVQ-PSO, optimization is used to find the best reference vector. The fitness value to measure the reference vector is the training error or matrix similarity analysis (MSA). The structure of FNLVQ-PSO is shown in Figure 12.

PSO is an optimization method that is inspired from colony of animal. The PSO is used to find an optimum value on a search space. Each particle will move and look for the optimum point (position) using the following equations:

$$\begin{aligned} V_i(t) \leftarrow \gamma \cdot \big(&V_i(t-1) + c1 \cdot rand() \cdot \left(P_i(t-1) - X_i(t-1)\right) \\ &+ c2 \cdot rand() \cdot \left(P_g(t-1) - X_i(t-1)\right)\big), \end{aligned} \tag{34}$$

$$X_i(t) \leftarrow X_i(t-1) + V_i(t). \tag{35}$$

In the above equations, $V_i(t)$ is the current velocity of the particle, $V_i(t-1)$ is the previous velocity of the particle, $X_i(t)$ is the current position of the particle, $X_i(t-1)$ is the previous position of the particle, *c*1 and *c*2 are constants, γ is a construction factor that has a value between 0 and 1, $P_i(t-1)$ is the previous local base of the particle, and $P_g(t-1)$ is the previous global base of all particles. The local best is the

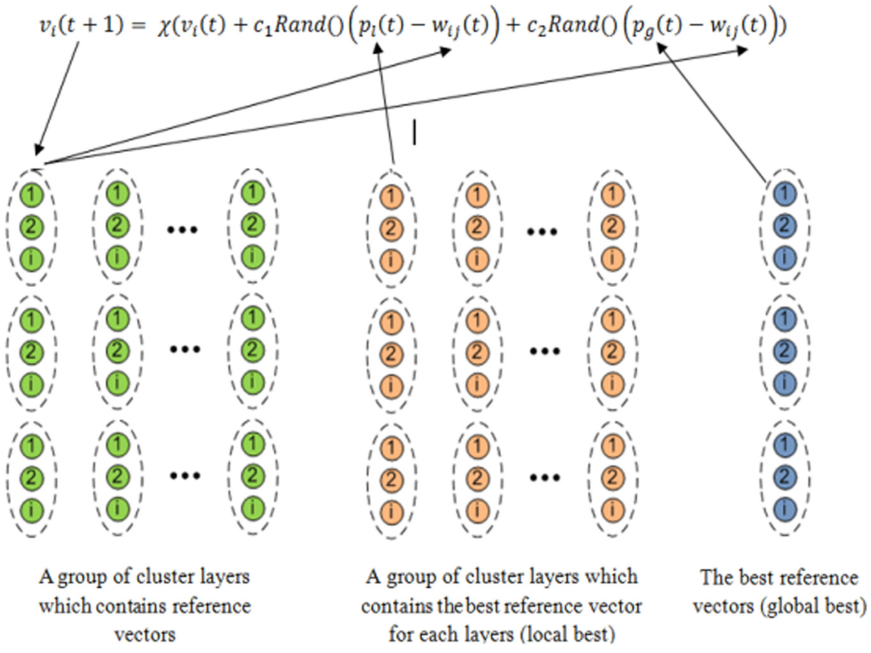

$v_i(t+1) = \chi(v_i(t) + c_1 Rand()\left(p_i(t) - w_{ij}(t)\right) + c_2 Rand()\left(p_g(t) - w_{ij}(t)\right))$
1
2
i
A group of cluster layers which contains reference vectors
A group of cluster layers which contains the best reference vector for each layers (local best)
The best reference vectors (global best)

Figure 12: FNLVQ-PSO architecture (Jatmiko et al., 2009).

point with optimum fitness value for a particle from start to current time (iteration). The global best is the point with optimum fitness value for all particles from start to current time (iteration). In the FNLVQ-PSO, each particle is represented as a reference vector (codebook), same as reference vector (codebook) in FNLVQ. During the training process, the classifier is optimized to find the optimum reference among all the candidates (particles). The training process in FNLVQ-PSO is conducted by using the following steps:

1. Initializing the reference vector: the number of particle is given by user input. In FNLVQ-PSO, the particles are reference vectors. Therefore, by doing this step, we have several candidates of reference vector to be chosen later.
2. Training the FNLVQ algorithm for each particle by computing similarity value between reference vector and input vector.
3. Computing matric similarity analysis (MSA) for each particle: the MSA is an $n \times n$ size matrix where n is the number of output class.
4. Computing MSA and fitness using the following equation:

$$f_k = \sum_{i=1}^{n} m_{ii} - \left| \sum_{i=1}^{n} \sum_{j=1}^{n} m_{ij} \ \ if\ i \neq j \right|. \qquad (36)$$

5. Computing local best for each and global best for all particles.
6. Updating the reference vector using the following equation:

$$w_{ij}(t+1) = w_{ij}(t) + v_i(t+1), \qquad (37)$$

$$v_i(t+1) = \gamma \left| \begin{array}{c} v_i(t) + \\ c1 \cdot rand \cdot \left(p_l(t) - w_{ij}(t)\right) + \\ c2 \cdot rand \cdot \left(p_g(t) - w_{ij}(t)\right) \end{array} \right|. \qquad (38)$$

7. Updating the fuzzy triangle of each particle using the following equation:

$$w(l)_{ij}(t+1) = w(l)_{ij}(t) + d_{ij}, \qquad (39)$$

$$w(r)_{ij}(t+1) = w(r)_{ij}(t) + d_{ij}, \qquad (40)$$

   where d_{ij} is the distance between the particle and input vector.
8. Repeating Steps 2–7 until given number of iteration or until the reference vector converges.

Fuzzy-neuro generalize learning vector quantization (FNGLVQ)

FNGLVQ is an enhancement of GLVQ by adding the fuzzy membership function. In the previous work, FNGLVQ was proposed to classify heart diseases (Setiawan et al., 2011). The architecture of FNGLVQ is illustrated in Figure 13 (Setiawan et al., 2011).

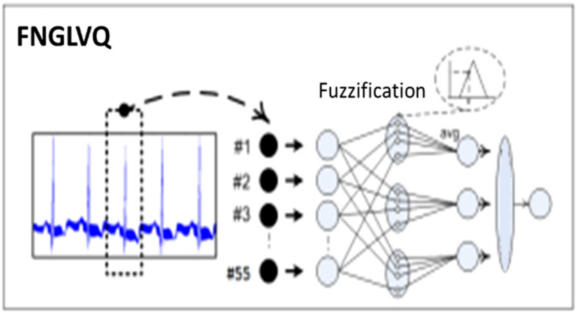


Figure 13: FNGLVQ architecture (Setiawan et al., 2011).

First, the distance value in Equation (9) is redefined into $d = 1 - \mu$, resulting in the following equation:

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

where μ_1 is the similarity between the input vector and the reference vector from the same class $C_x = C_{w1}$, whereas μ_2 is the similarity between the input vector and the closest reference vector from a different class $C_x \neq C_{w2}$. Then, reference vector update is defined by the following equation:

$$\frac{\delta s}{\delta w_i} = \frac{\delta s}{\delta \varphi} \cdot \frac{\delta \varphi}{\delta \mu} \cdot \frac{\delta \mu}{\delta w_i}. \tag{42}$$

By applying fuzzy membership function, the reference vector is defined with a triangular function, $w_{ij}(w_{min,ij}, w_{mean,ij}, w_{max,ij})$, which is defined as follows:

$$\mu = h(x, w_{min}, w_{mean}, w_{max}) = \begin{cases} 0, & x \leqslant w_{min} \\ \dfrac{x - w_{min}}{w_{mean} - w_{min}}, & w_{min} \leqslant x \leqslant w_{mean} \\ \dfrac{w_{max} - x}{w_{max} - w_{mean}}, & w_{mean} \leqslant x \leqslant w_{max} \\ 0, & x \geqslant w_{max} \end{cases}. \tag{43}$$

Derivation of the membership function to (w_{mean}) lead is divided into three conditions, resulting in the learning formula below:

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

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

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

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

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

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

- If $x \leq w_{min}$ then $x \geq w_{max}$:

$$w_i(t+1) \leftarrow w_i(t), \quad i = 1,2. \tag{48}$$

In these equations, w_1 is the reference vector from the same class as the input vector $C_x = C_{w1}$, and w_2 is the reference vector from a different class $C_x \neq C_{w2}$. Update rules for w_{min} and w_{max} follow the equation below:

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

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

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

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

To achieve a better performance, additional rules need to be defined to adjust w_{min} and w_{max}, as written in the conditions below:

- If ($\mu_1 > 0$ or $\mu_2 > 0$), and $\varphi < 0$, then increase the fuzzy triangular width by using the following equations:

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

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

- If the input class is recognized into the wrong class ($\varphi \geq 0$), then decrease the triangular width using the following equations:

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

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

- If $\mu_1 = 0$ and $\mu_2 = 0$, then the fuzzy triangular vectors must be increased using the following equations (γ is a constant, with a value of 0.1 in this research):

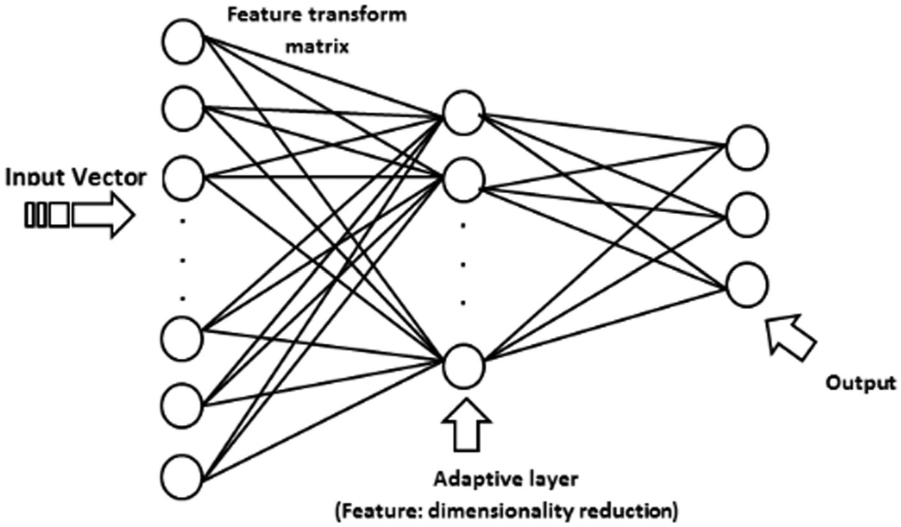


Figure 14: AM-GLVQ architecture (Imah et al., 2012).

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

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

Adaptive multilayer generalize learning vector quantization (AM-GLVQ)

Adaptive multilayer generalized learning vector quantization (AM-GLVQ) is an enhancement of GLVQ by using a multi-layer approach proposed by Imah et al. (2012). The idea is to modify GLVQ into a multi-layer structure. The architecture of the AM-GLVQ is shown in Figure 14.

In AM-GLVQ, the input vector is denoted as *x*. The input data in the Eigenspace are denoted as *x'*, as written in the following equation:

$$x' = T^T x \quad (58)$$

Therefore, we need to find the best values of transformation matrix *T* during the training process. Update rule for matrix *T* is defined in the following equations:

$$T_t^T \leftarrow T_{t-1}^T + \alpha \frac{\delta f}{\delta \varphi} \frac{4d_2}{\left(d_1 + d_2\right)^2} \left(T^T x - T^T w_1\right)\left(x - w_1\right) \quad (59)$$

$$w_1 \leftarrow w_1 + \alpha \frac{\delta f}{\delta \varphi} \frac{4d_2}{\left(d_1 + d_2\right)^2} \left(T^T x - T^T w_1\right) \quad (60)$$

$$w_2 \leftarrow w_2 - \alpha \frac{\delta f}{\delta \varphi} \frac{4d_1}{\left(d_1 + d_2\right)^2} \left(T^T x - T^T w_2\right) \quad (61)$$

where w_1 is the nearest reference vector that belongs to the same class of *x*. Similarly, w_2 is the nearest reference vector that belongs to a different class from *x*.

Signal compression methods

With the large amount of ECG data that will be processed, there has to be an efficient method to transmit and store the data, so that there are no interruptions in the classification process. In this section, we will discuss several methods to compress the ECG data in order to efficiently store and transmit the ECG data. These methods were proposed by the previous researchers. This paper primarily discusses compression methods by using set partitioning in hierarchical trees (SPIHT). There will be two main sub-sections in this section. The first section will explain two-dimensional SPIHT compression, while the second sub-section will discuss about three-dimensional SPIHT compression.

Two-dimensional SPIHT

In the present day, several researchers have proposed using two-dimensional SPIHT method to compress ECG data. In 2012, Isa et al. used SPIHT in order to compress the ECG data (Sato and Yamada, 1995; Isa et al., 2012a, 2012b). This method is a state-of-the-art

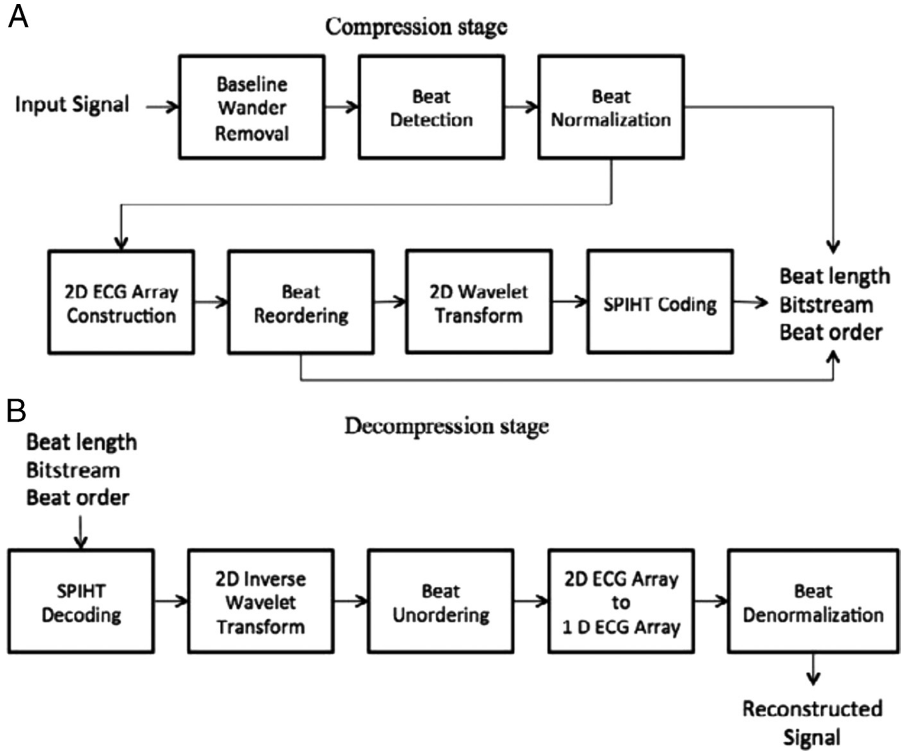


Figure 15: The process of two-dimensional SPIHT as proposed by Isa et al.

lossless compression method. In this sub-section, we will discuss the compression stage of the algorithm. The stages that were proposed by Isa et al. in SPIHT coding can be seen in Figure 15. We will not discuss the 2D wavelet transform phase, as it has been discussed previously in this paper (please refer to the pre-processing section).

Beat reordering

During the compression stage of the SPIHT, the initial method that is conducted is beat reordering method. The beat reordering method is conducted in order to arrange the ECG beat to optimize the SPIHT coding. The arrangement of beats will be based on their similarities, as Isa et al. pointed out that a higher compression rate is more achievable in highly predictable data. This method is chosen due to the fact that the ECG beat data do not have a highly predictable nature because they repeat according to the heart rate (Zhao and Chen, 2006).

In the beat reordering method, the initial step is to use fuzzy c-means clustering techniques in order to cluster the beats that have a high similarity. After we have clustered the similar beats, we will arrange the order of the beats of each cluster on the basis of the distance to the center of the cluster. In Figure 16, we can see the before and after result of the beat reordering technique. As shown in the figure, it can be inferred that after the beat reordering method has been applied to the ECG data, there are less noticeable peaks and granulation of the data. This results in a smoother ECG data, and this is because the beat reordering method will reduce the high-frequency components and reduce the variance among beats that are adjacent to each other. This method will optimize the process of SPIHT compression in ECG data.

Overview of SPIHT

Several works on ECG signal compression have been conducted previously (Lu et al., 2000; Moazami-Goudarzi et al., 2006; Linnenbank et al., 1992). In this section, we will be discussing about SPIHT. The set partitioning in hierarchical trees (SPIHT) is a wavelet-based coding method that uses a set partitioning algorithm in order to transform coefficients based on the sub-band pyramid. In the SPIHT method, the most important information coefficient will be sent first. The hierarchical quad-tree data structure

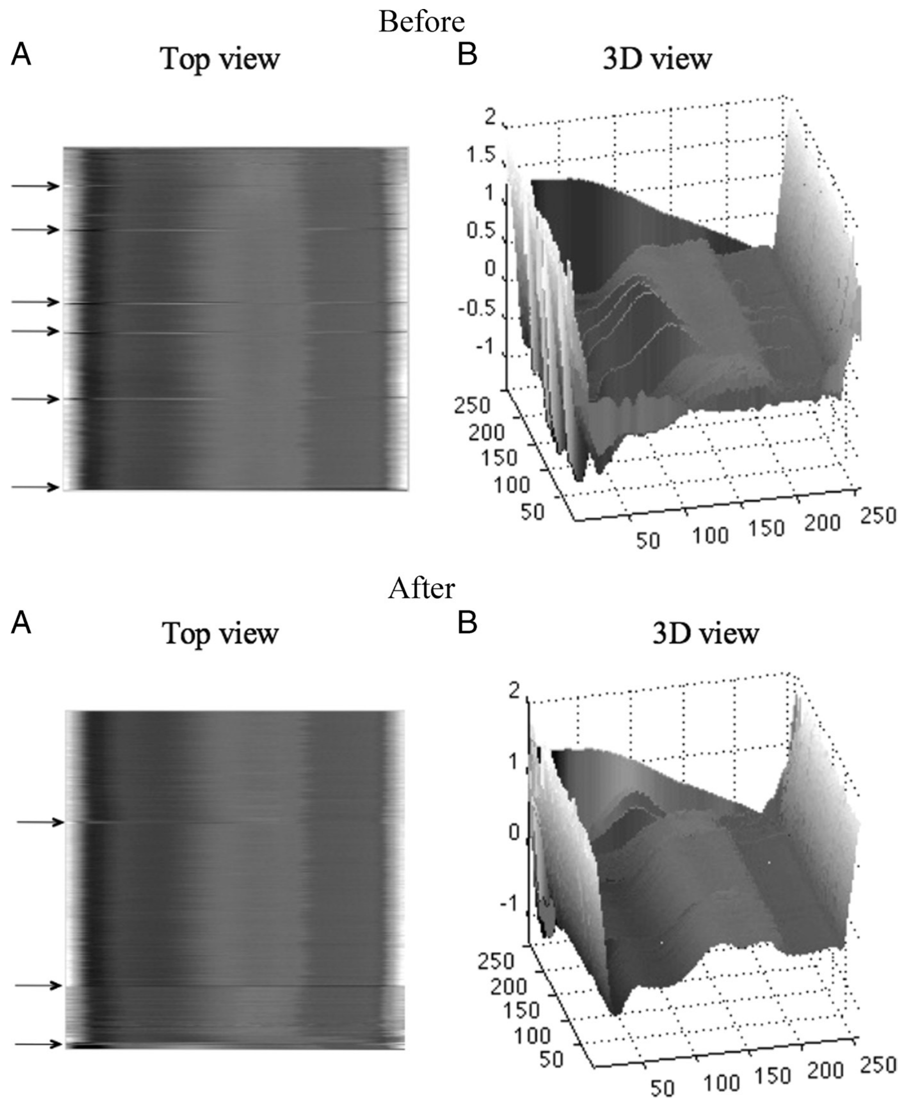


Figure 16: The ECG data before and after the beat reordering method.

is adopted on a wavelet signal. The low-frequency coefficient is the center of the wavelet transformed signal, where the coefficients are ordered in a hierarchy that includes a sub-band parent–child relationship.

There are four main steps in the SPIHT method, with the first step being the initialization step. The initialization step will empty the List of Significant (LIS). Afterwards, the list of insignificant points (LIP) and the list of the insignificant sets (LIS) will also be set. The threshold is also set at $T_0=2^n$, with n log (max $c(i, j)$). In the threshold, $c(i, j)$ denotes the coefficient at position (i, j). The next step in the SPIHT method is the sorting pass in LIP, where the coefficients in the LIP are examined. The significant and important coefficients are transferred to the LSP. Then, we encode sign bits of the important coefficient. The next phase is the sorting pass in the LIS. If the entry in the LIS is important, 1 will be sent. Also, two of the offsprings will be examined. However, if it is not important, a 0 will be sent instead. The final phase will be the refinement pass in which the previous entries of the LSO are examined. If it is important with the current threshold, 1 is sent and the magnitude is reduced by the current threshold. However, if it is not important, a 0 is sent.

Improvement with predictive coding

In 2015, Jati et al. proposed an improvement over Isa et al.'s SPIHT coding by adding predictive coding (Linnenbank et al., 1992). The main objective of this

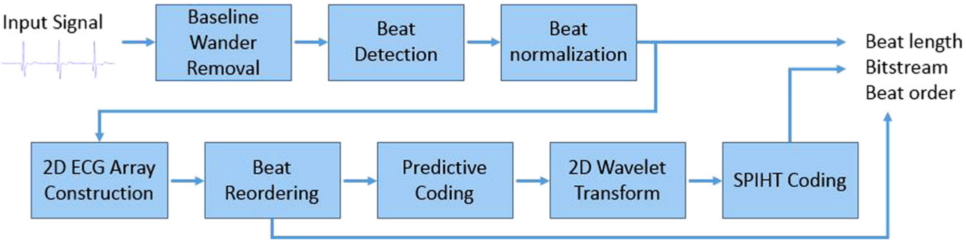


Figure 17: The process of two-dimensional SPIHT as proposed by Jati et al.

predictive coding method is to reduce the variation of amplitudes in the 2D ECG signals. By applying the predictive coding method, the signal value will be relatively smaller than before. The flowchart of the SPIHT process that was conducted by Jati et al. can be seen in Figure 17.

In order to reduce the signal value, we first need to select the beat that will represent the group of clustered beats during the beat reordering method. This selected beat will be chosen as the benchmark beat. To select the benchmark beat, we can apply several statistics values of all samples of 2D ECG array (mean, min, or max) or we can use the most important beat in the ordered cluster. After we have obtained the benchmark beat, we have to recreate the new 2D ECG array by computing the difference between the real signal value with the obtained benchmark value. Figure 18 shows the process of the proposed predictive coding by Jati et al.

Three-dimensional SPIHT

To accommodate the recent improvements in the biomedical imaging, Isa et al. proposed a three-dimensional SPIHT compression (Isa et al., 2014). The three-dimensional SPIHT compression is a modified two-dimensional SPIHT compression by modifying the spatial orientation tree concept. The original 2D SPIHT compression has four offsprings on each node of a wavelet coefficient, whereas the proposed 3D SPIHT has eight offsprings. Figure 19 shows the concept of the 3D SPIHT compression.

In Isa et al. (2014), the multi-lead ECG signal was constructed with a combination of two-dimensional residual arrays. This combination will produce a three-dimensional residual array. The samples, ECG leads, and the heartbeat are defined by the axis of the volumetric structures. The intra-beat, the inter-beat, and the inter-lead redundancies are de-correlated by the 3D SPIHT method. The 3D residual array of the 3D SPIHT method can be seen in Figure 20.

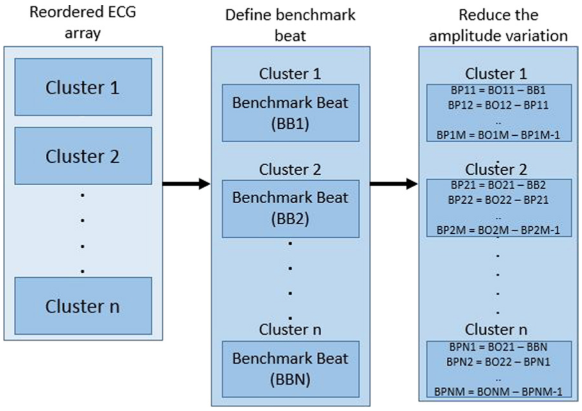


Figure 18: The process of predictive coding by Jati et al.

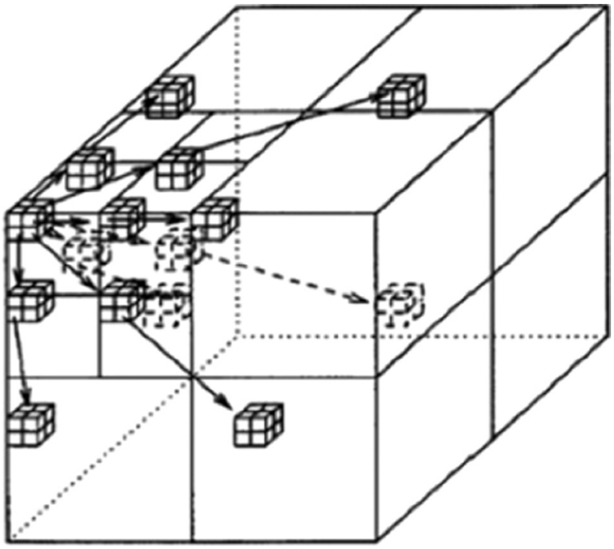


Figure 19: The spatial orientation tree of 3D SPIHT.

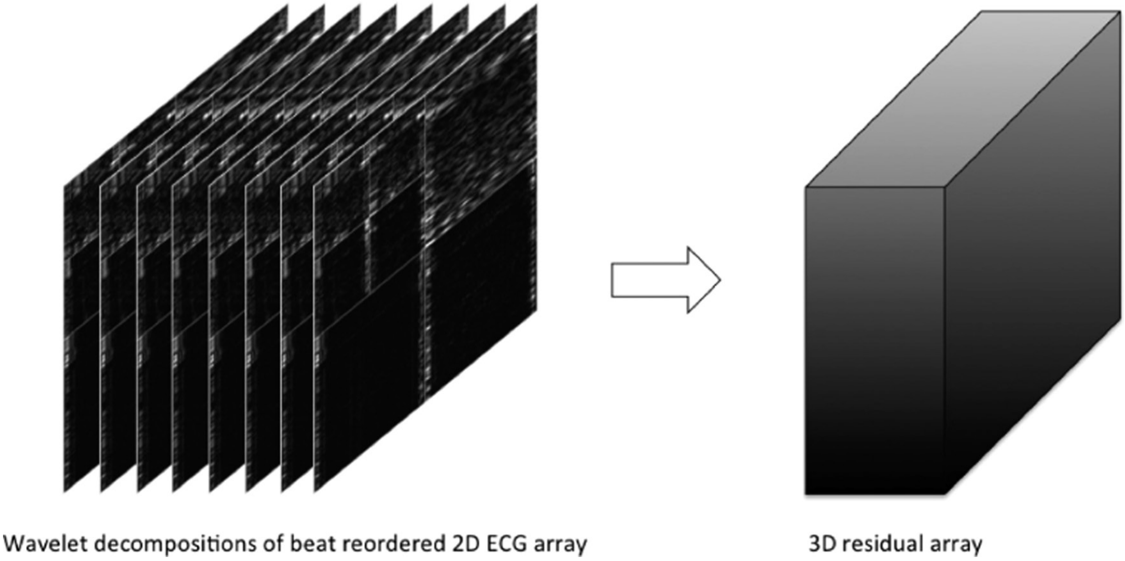


Figure 20: 3D Residual array of the 3D SPIHT method.

System implementation

In the study before, a smart Tele-ECG system has been developed. The main purposes of the system are early detection of heart diseases and heart monitoring. Several works have been conducted on telehealth systems (Sudhamony et al., 2008; Hababeh et al., 2015). The architecture of the system is shown in Figure 21. The system is comprised of four components: the ECG sensor, smartphones, FPGA, and the server. The first component, the sensor, is used to acquire heartbeat signal from patient's body. The transducer, a component of the sensor, that is used in this system is commonly used in Indonesian hospitals. The second component is the smartphone. The smartphone is used to visualize and analyze the heartbeat signal sent by ECG sensor. The third component, the FPGA, was developed for ECG signal processing and classification. The FPGA implementation is intended for designing a chip that is dedicated to process ECG data. The ECG sensor is connected to the smartphone or FPGA through a bluetooth network. The last component is the server that is used for communication between the patient and the cardiologist. The mobile application of the smart Tele-ECG accommodates both patient and the

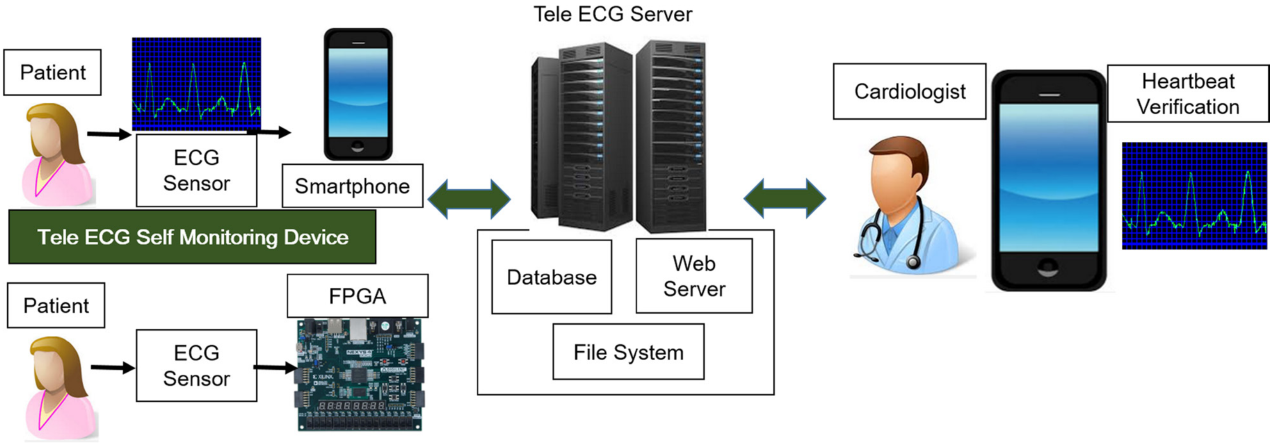


Figure 21: The architecture of Tele-ECG.

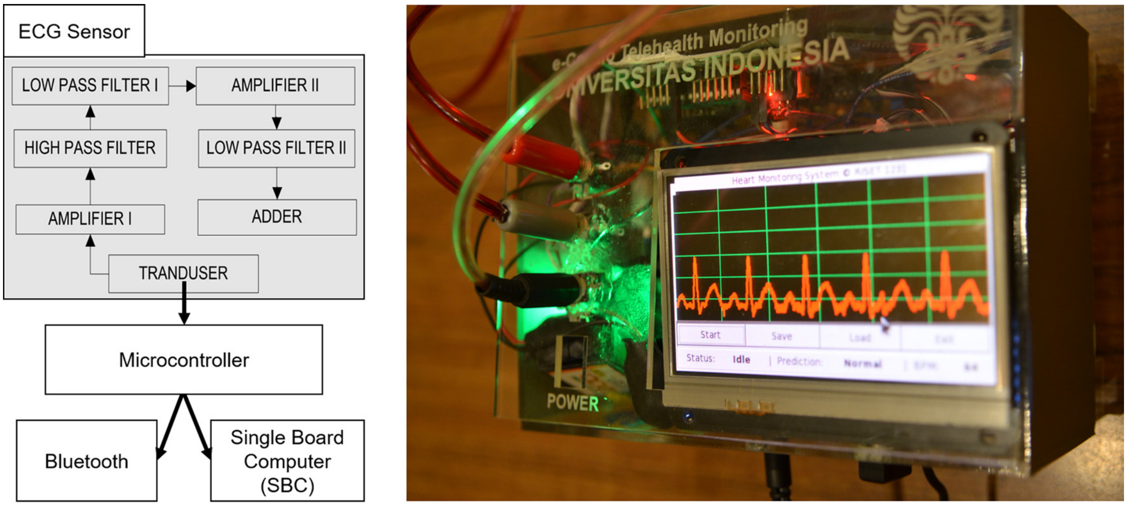


Figure 22: The ECG sensor.

doctor. However, the doctor has a unique privilege to verify the heartbeat signal sent by the patient.

Single-lead ECG sensor

In the previous research, we built an ECG sensor using electrical components. The ECG sensor used in this system has several components, as shown in Figure 22. The first component is the transducer that is attached to patient's body. The amplifier component is functioned to amplify the acquired raw signal. The main amplifier chip in the ECG sensor is INA118. This amplifier can generate output signals up to 10 times the original input signal's strength. In this first amplifier, we used electronic circuit called the right-leg-driven loop. The circuit is used as inverted version of the interference at the right foot of the human body. Therefore, the circuit can decrease interference generated by human body. This electronic circuit is also used for human safety system. Detailed characteristics and functionality of the circuit is explained in the INA118 datasheet.

The second amplifier is used to amplify the signal resulted from previous components. This amplifier strengthens the input signal that is in range 5 mV to 40 mV. The OP07 circuit is used for the second amplifier. The Op07 is a non-inverted amplifier. In the sensor we built, the reinforcement value for the second amplifier is 100 times. The following component is a filter, a component that is used to separate the wanted signal from unwanted signal. The ECG signal has two types of filters that is low pass filter (LPF) and high pass filter (HPF). LPF is used to pass signal that has lower frequency than the threshold, whereas HPF is functioned to pass signal from higher frequency. In the sensor, The HPF is used to remove human body noise, whereas LPF is used to remove contained noise that is already in ECG signal.

Besides the single-lead ECG sensor, the system also used multi-lead ECG sensor to acquire heartbeat signals from the human body. The main engine of the multi-lead ECG machines is similar to the engine in the single-lead sensor. The main modules in multi-lead sensor are amplifier, filter, and adder. Figure 23 shows the electronics of ECG single-lead device.

ECG telehealth mobile application

In the previous research, smartphone is the component for visualizing, classifying, and saving/storing the user's heartbeat data. The application is deployed as mobile apps in Android operating system. The view of the Tele-ECG mobile application is shown in Figure 24. The application has several menus to accommodate the users for heartbeat monitoring and early detection of heart diseases. The menus are retrieve heartbeat data from ECG, heartbeat history, user management, cardiologist information, hospital information, heart disease information, doctor's special menu, settings, and help.

The main processes in the smartphone device are heartbeat signal acquisition, signal preprocessing, signal classification and compression. The classification is used to predict the patient's heart condition. It predicts if the patient heartbeat shows symptoms of heart disease. To classify the heartbeat signal, the system uses neural network classifier that has been mentioned in the previous section: generalized learning

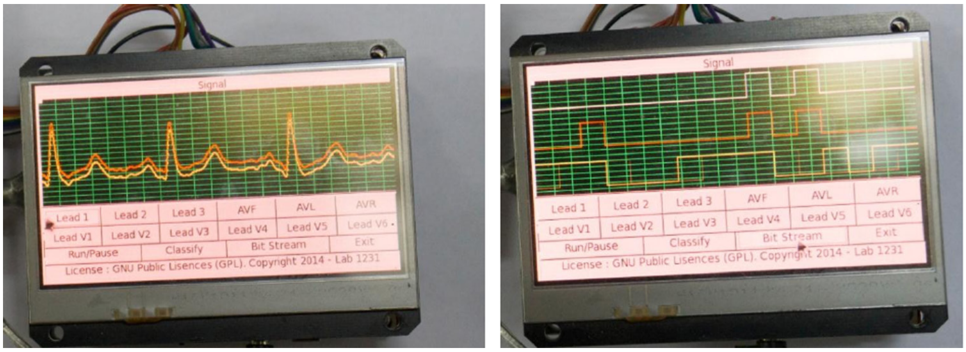


Figure 23: Multi-lead and single-lead ECG machines.

vector quantization (GLVQ) and fuzzy-neuro generalized learning vector quantization (FNGLVQ). We select these classifiers due to the simple structure of the methods and fast process, and the good performance of the method. The computational time of the methods is relatively faster than other neural network methods, such as multi-layer perceptron (MLP) or deep learning.

ECG telehealth server

The server has two modules: web services and database. The tele-ECG server was implemented using java platform and Java EE framework. The web services provide many functionalities for create, read, update, and delete (CRUD) and other services, that is save heartbeat, verify heartbeat, register user data, etc. There are 11 services built in the server that provide the users server to carry out the telehehalth function of the system. The detailed information about the services provided by Tele-ECG server is shown in Table 1.

Classification on field-programmable gate array (FPGA)

In this section, we will discuss about the implementation of ECG signal in a microprocessor. Field-programmable gate array (FPGA) is an integrated circuit that is designed to be configured after the manufacturing process. This device enables the programmer to add new features by changing the configuration even after the program has been installed on the board. In our previous research, we used Spartan 3AN board from Xilinx (Xilinx, 2011). Several research studies on FPGA that were focused on analyzing ECG signal have been conducted (Figueiredo and Michael, 2013; Risman et al., 2014). In this section, we will elaborate

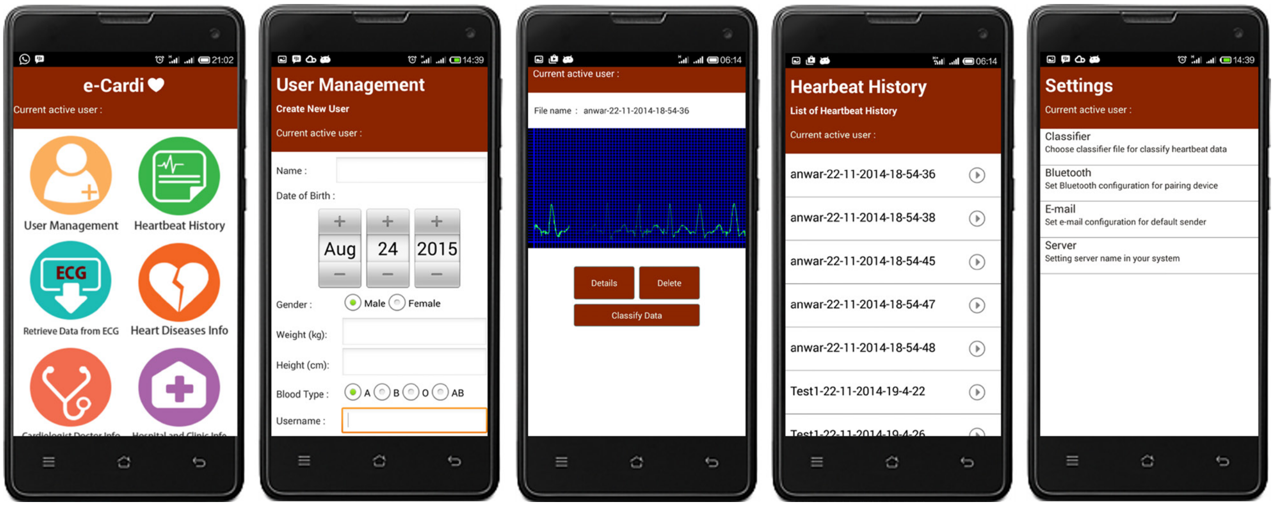


Figure 24: The mobile system on Tele-EG.

Table 1. Tele-ECG services.

No.	URL	Service
1	/RegsiterPatien	Registers user data as patient, then the data are saved in patient table of the database
2	/RegsiterDoctor	Registers user data as doctor, then the data are saved in doctor table of the database
3	/UploadHistory	Uploads patient heartbeat history, then the heartbeat data are saved in history table of the
4	/LookHistory	Checks status of patient heartbeat history if it is verified by a doctor (cardiologist)
5	/GetDoctorData	Downloads doctor (cardiologists) information
6	/GetHospitalData	Downloads hospital information
7	/GetUnverifiedHistory	Downloads unverified patient heartbeat history to be verified by doctor (cardiologist)
8	/VerifyHistory	Verifies patient heartbeat history by doctor (cardiologist)
9	/RegisterHospital	Registers hospital data, then the data are saved in hospital table of the database
10	/RegisterAffiliation	Affiliates the doctor (cardiologist) into hospital
11	/GetDoctorAffiliation	Downloads doctor's (cardiologists') affiliations.

the implementation of FLVQ, FNLVQ, and AFNGLVQ algorithm in FPGA (Setiawan et al., 2011; Suryana et al., 2012; Afif et al., 2015).

The design of ECG classifier in FPGA consists of three modules. These modules are ROM module, preprocess module, and classification module. The ROM module serves as the initial data storage. The preprocess module will decompose the input data with the fourth-order wavelet Daubechies and prepare the data that will be used for in the next stage, which is the classification module. This module consists of three sub modules, namely the wavelet Daubechies 4 module, float to fixed converter module, and data normalization unit module. Wavelet Daubechies 4 module is employed for extracting 300 features per beat into 40 features per beat. The float to fixed converter module is used to convert float-point (wavelet Daubechies module result) into fixed-point number, which becomes the input type for the classification module. The data normalization unit module normalizes the data using *Z*-score. The last module is classification module, which is responsible for doing classification using FLVQ on ECG data. The result of the classification stages will be displayed on the LCD module. The illustration of the ECG classifier on FPGA can be seen in Figure 25.

During the preprocessing stages, the original signal is decomposed into half-length for each iteration. This decomposition process uses 16-bit floating points, consisting of 5-bit exponent and 10-bit

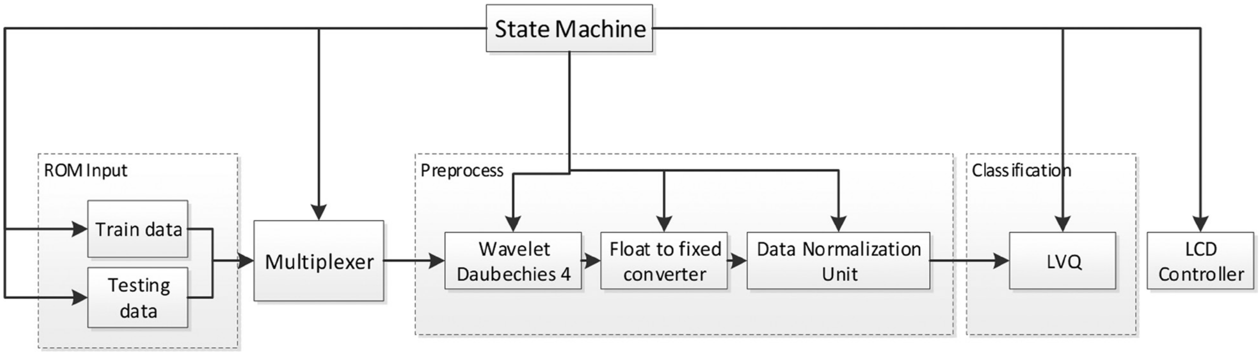


Figure 25: The ECG classifier on FPGA.

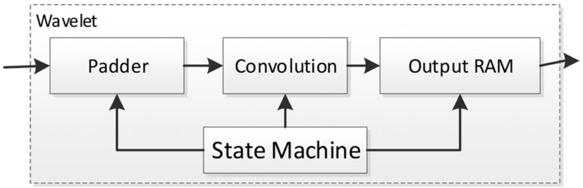


Figure 26: The Daubechies Wavelet architecture.

mantissa. The input signals have 300 features long scalar and will be run for three iterations that will produce 40 features of long scalar. The wavelet Daubechies architecture is shown in Figure 26. The features should be padded before they are passed to the convolution stages. The signal will be padded with mirroring at the head and tail of the input signal. As the main part of the Wavelet Daubechies submodule, the convolution component has a role to convolve the padded input signal. Furthermore, the convoluted signal is downsampled into half-length data and stored in the output RAM. Values kept in the output RAM would serve as the input for the next decomposition operation.

The convolution unit consists of four components: Shift Register (4 temporary registers), Daubechies Constant (4 hard-coded constants), 16-bit FPMultiplier that handles multiplication operation, and 16-bit FP Adder that handles addition operations. The convolution process initially separates 306 features into 19 chunks of data. Furthermore, the FPMultiplier performs multiplication operation between each data chunk and Daubechies Constant. The results of the multiplication process are summed up to obtain the convolution result. Figure 27 shows the convolution architecture.

The classification module is the implementation of FLVQ algorithm for classifying the data. The component of this module consists of input data RAM, vector reference RAM, expected value RAM, neuron component, similarity divider component, updater component, and state machine. The input data RAM is the location for storing the output of the preprocessing process from the wavelet decomposition. Meanwhile, the vector references RAM is used to store the vector references that come from the output of the fuzzification process. The neuron component is responsible for measuring the similarities of each feature; meanwhile, the similarity divider component has a role to determine the winner vector. Moreover, the winner vector is updated by moving or changing the shape of the corresponding fuzzy triangle based on the update case by the updater component. The output of this classification process is stored in expected value RAM. The design of the FLVQ architecture in FPGA is shown in Figure 28.

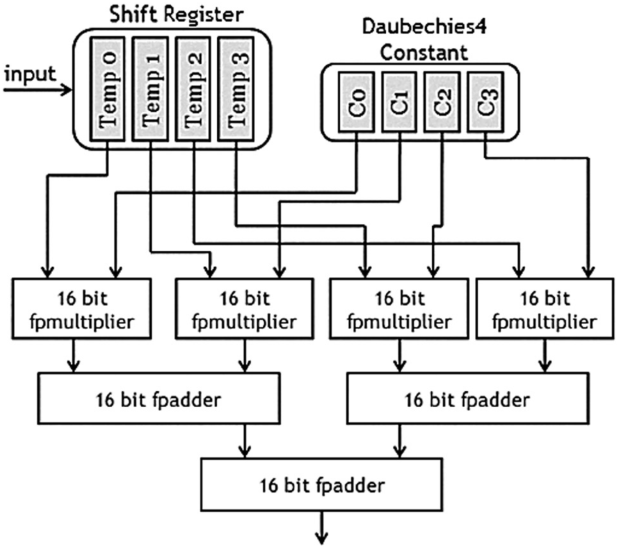


Figure 27: The Convolution Unit Architecture.

We have also implemented FNGLVQ in the FPGA. Some additional modifications have been done, including the addition of a sigmoid derivative core unit, which expands the number. The arithmetic operations in this design uses 32-bit fixed-point format, which consists of 16-bit integer and 16-bit fraction. The main reason of using fixed-point rather than floating-point format is the expressiveness and the resource usage. Moreover, using the fixed-point format can increase the computation speed and increase the efficiency. As a comparison, an addition operation of two 32-bit number fixed-point unit spends 32 slices, whereas a floating-point unit spends 500 slices when run on 1 clock. The design of FNGLVQ in FPGA is shown in Figure 29.

The FNGLVQ algorithm is implemented in a state machine. The state machine defines the connectivity of the component in the design. The designed state machine is illustrated in Figure 30. Based on the experiment (Sudhamony et al., 2008), the performance of FNGLVQ in the FPGA is slightly degraded compared to the top-level implementation of FNGLVQ. The performance of FNLGLVQ in the FPGA degrades around 2%. The main reason of this degradation in performance is the number format that is used in the FPGA, which is a fixed-point number format. Floating-point number format has a more detailed and precise value compared to a fixed-point number. Thus, it produces better performance. However, floating point number in

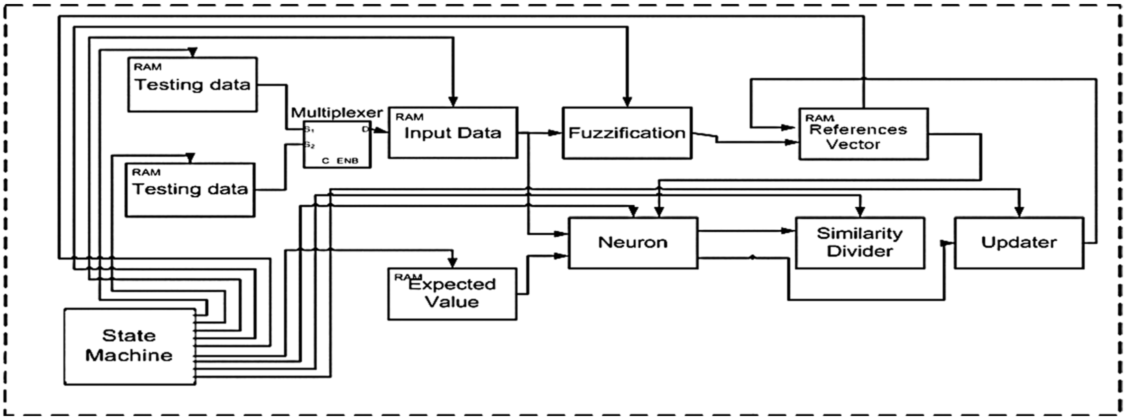


Figure 28: The design of FLVQ architecture.

FPGA requires substantial resource allocation and extra computational time. Fixed-point number has a slightly lower performance; however, it is faster and requires smaller resource than a floating-point number.

In addition to the FNGLVQ, we had also implemented the AFNGLVQ algorithm in the FPGA board. The design of the AFNGLVQ algorithm is relatively similar to the FNGLVQ. The main difference is in the state machine. The state machine will receive the input vectors and store them in the w_{min}, w_{mean}, and w_{max}. Furthermore, the state machine will update the value of reference vector weight and store it in three components, that is w_{min}, w_{mean}, and w_{max}. The designs of AFNGLVQ and state machine during the training phase are depicted in Figures 31 and 32.

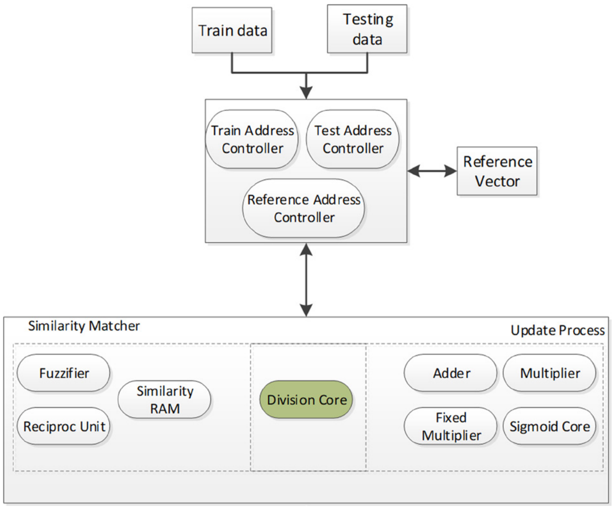


Figure 29: The design of FNGLVQ in FPGA.

In the FPGA board, AFNGLVQ achieves a better performance compared to the FNGLVQ (Afif et al., 2015). The adaptive feature on AFNGLVQ has a beneficial impact on performance. Hence, it still has

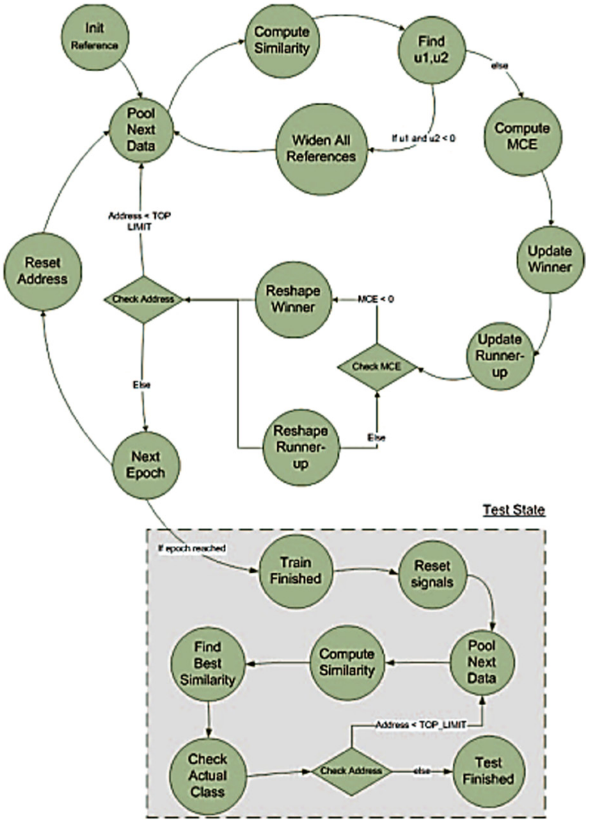


Figure 30: The State machine of FNGLVQ.

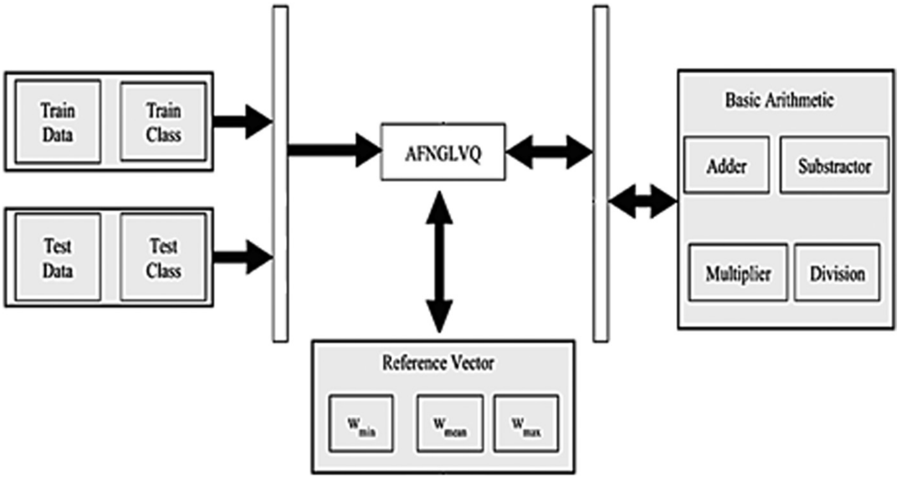


Figure 31: The top-level design of AFNGLVQ in FPGA.

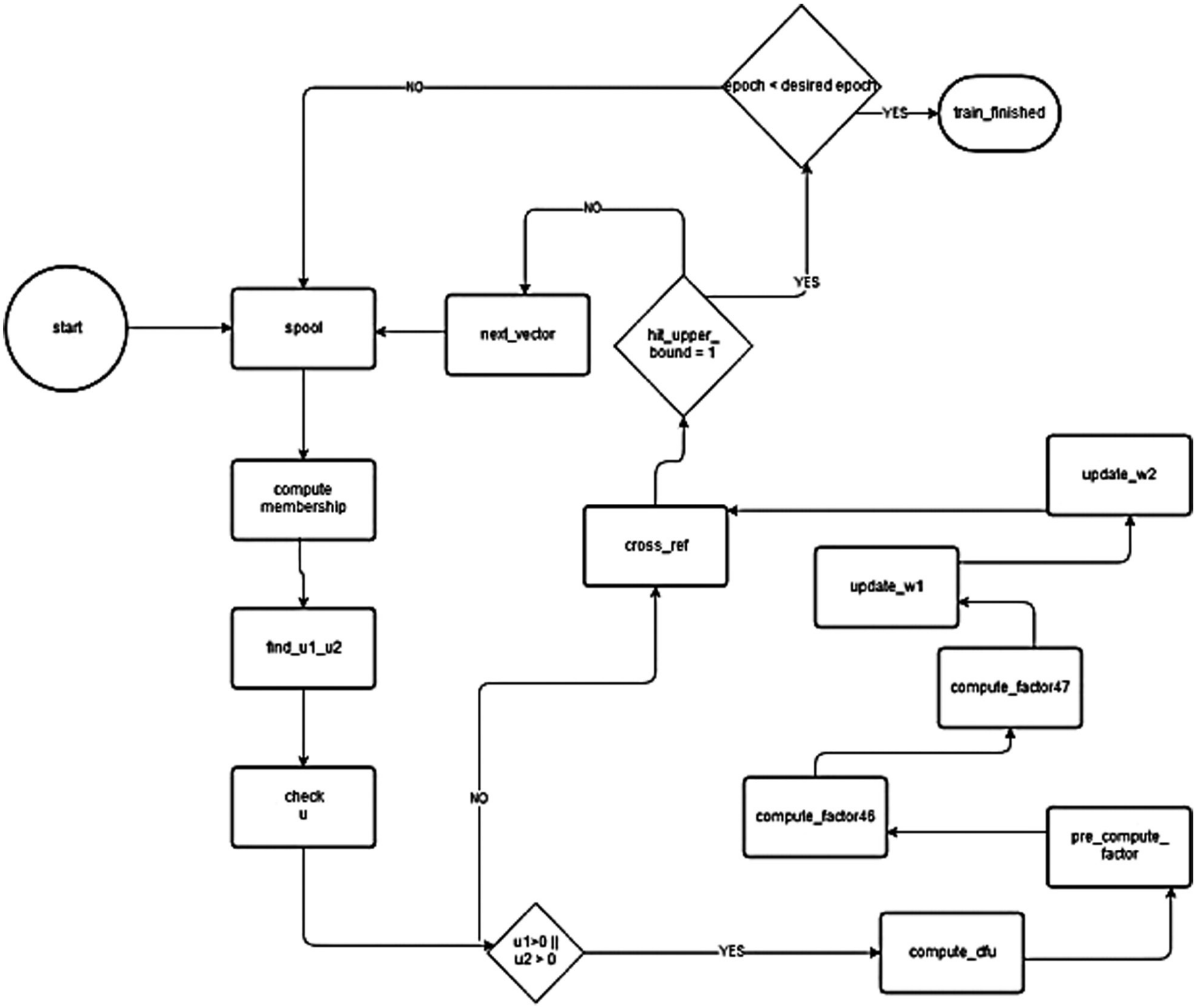


Figure 32: The top-level design of AFNGLVQ in FPGA.

a slightly lower performance compared to the high-level implementation of AFNGLVQ due to the number format. Floating-point number has a more detailed and precise value compared to the fixed-point number. Thus, the FPGA implementation has had competitive result, and it is worth to be implemented in a micro device.

Performance evaluation

This section discusses the performance of the ECG telehealth system. The measurement consists of the accuracy of the arrhythmia classification, compression error and arrhythmias classification in FPGA. This section summarizes the experiments that were conducted in the previous study.

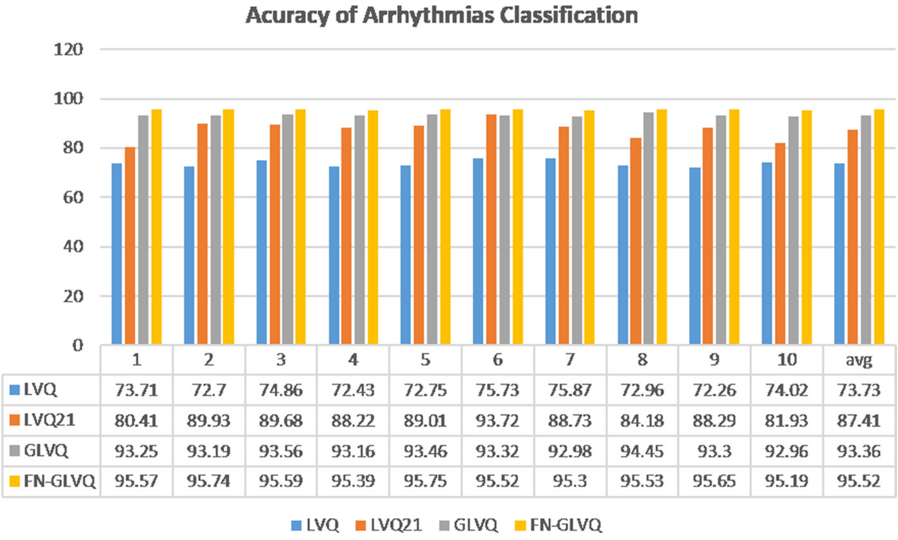
Arrhythmias classification using LVQ-based classifier

As mentioned previously, FNGLVQ is a GLVQ that has a fuzzy membership function. A study by Setiawan et al. utilized FNGLVQ to classify arrhythmias based on the ECG sensor data (Setiawan et al., 2011). Setiawan et al. also used MIT–BIH arrhythmias dataset, which consists of 12 classes. These twelve classes are normal beat (NOR), right bundle branch block beat (RBBB), left bundle branch block beat (LBBB), paced beat (P), premature ventricular contraction beat (PVC), atrial premature beat (AP), fusion of paced and normal beat (fPN), fusion of ventricular and normal beat (fVN), nodal (junctional) escape beat (NE), nodal (junctional) premature beat (NP) aberrated atrial premature beat (aAP), and ventricular escape (VE). Before classifying, there are several preprocessing steps that were applied to the dataset. These preprocessing steps are baseline wandering removal (BWR), outlier removal, and wavelet transform.

In the experiment, FNGLVQ was compared to LVQ, LVQ21, and GLVQ. The experiment was conducted using 150 epochs. The dataset was divided into 10 parts. Each part has training and testing set with 50:50 ratio. The experiment results are shown in Figure 33. Figure 33 shows that parts 1 to 10 have a similar trend. FNGLVQ has the highest accuracy, followed by GLVQ, LVQ21, and LVQ. Over all, FNGLVQ achieved 95.52% accuracy, GLVQ achieved 93.36%, LVQ21 reached 87.41% accuracy, and LVQ reached 73.73% accuracy. This means that in the MIT–BIH arrhythmias dataset, FNGLVQ outperformed GLVQ, LVQ21, and LVQ, with a margin of 2.16, 8.07, and 21.79%, respectively. This infers that adding fuzzification in arrhythmia classification improves the accuracy.

Round robin impact in arrhythmias classification

Diane et all continued the research by Setiawan by investigating the impact of round robin approach on training process (Fitria et al., 2014). The investigation was applied in various LVQ-based classifiers. The classifiers are LVQ1, LVQ2, LVQ2.1, FNLVQ, FNLVQ MSA, FNLVQ-PSO, GLVQ, and FNGLVQ. In the study,



	1	2	3	4	5	6	7	8	9	10	avg
■ LVQ	73.71	72.7	74.86	72.43	72.75	75.73	75.87	72.96	72.26	74.02	73.73
■ LVQ21	80.41	89.93	89.68	88.22	89.01	93.72	88.73	84.18	88.29	81.93	87.41
■ GLVQ	93.25	93.19	93.56	93.16	93.46	93.32	92.98	94.45	93.3	92.96	93.36
■ FN-GLVQ	95.57	95.74	95.59	95.39	95.75	95.52	95.3	95.53	95.65	95.19	95.52

Figure 33: Accuracy of Arrhythmias classification using LVQ-based classifier.

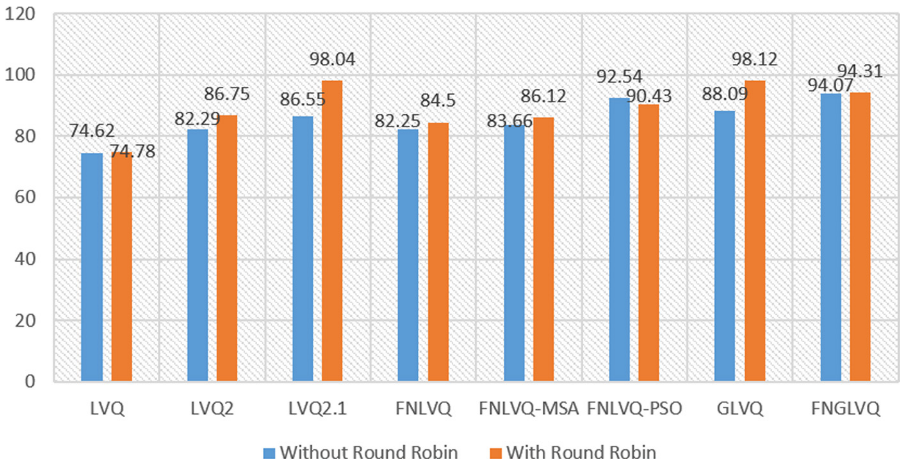


Figure 34: Impact of Round Robin in Arrhythmias classification.

the authors only used 5 classes from the MIT–BIH dataset. The classes are normal beat (NOR), right bundle branch block (RBBB), left bundle branch block (LBBB), paced beat (PACE), and premature ventricular contraction (PVC).

The experiment result is shown in Figure 34. The figure shows that without round robin approach, the classifiers achieved 74.62, 82.29, 86.55, 82.25, 83.66, 92.54, 88.09, and 94.07% for LVQ1, LVQ2, LVQ2.1, FNLVQ, FNLVQ MSA, FNLVQ-PSO, GLVQ, and FNGLVQ, respectively. However, when using round robin approach, the classifiers achieved 74.78, 86.75, 98.04, 84.5, 86.12, 90.43, 98.12, and 94.31% for LVQ1, LVQ2, LVQ2.1, FNLVQ, FNLVQ MSA, FNLVQ-PSO, GLVQ, and FNGLVQ, respectively. Figure 34 shows that round robin approach improves the accuracy for almost all classifiers: LVQ2, LVQ21, FNLVQ, FNLVQ-MSA, and GLVQ. The round robin approach increases the classification accuracy from 2 to 11%. For LVQ and FNGLVQ, the round robin approach has a slightly higher accuracy than without round robin. But for FNLVQ-PSO, the round robin approach actually decreased the accuracy.

Arrhythmias classification with unknown class

As mentioned in the previous section, Imah et al. developed an adaptive multilayer generalized learning vector quantization (AM-GLVQ), which integrates feature extraction and classification for Arrhythmia heartbeat classification (Imah et al., 2012). In the study, Imah et al. also investigated the performance of the classifiers in the dataset with unknown class and compared them with normal case (without unknown class). The authors used MIT–BIH Arrhythmias dataset, as used by Setiawan et all. There were 14 classes, where 12 classes were explained in subsection A, with an addition of escape beat (AE), and supra ventricular premature beat (SP).

There are two scenarios, without unknown class and with unknown class. In this evaluation, AM-GLVQ is compared to backpropagation, GLVQ, LVQ, and SVM. The experiment results are shown in Figure 35. For both scenarios, AM-GLVQ achieves the highest accuracy among all the algorithms. AM-GLVQ reached 95.16 and 95.04% accuracy for the data set without unknown class and with unknown class, respectively. As shown in Figure 35, in the dataset with the unknown class scenario, the accuracy of many classifiers decreased when compared to the scenario without unknown class in the dataset. The reduction of accuracy varies from 3 to 4%. However, AM-GLVQ can still maintain its performance in scenario with unknown class.

Compression error

This sub-section summarizes performance of ECG compression from previous studies. To evaluate the performance of ECG compression, we used percentage root-mean-square difference (PRD). This sub-section will summarize the performance of 2D SPIHT proposed by Sani et al., 3D SPIHT proposed by Sani et al., and 2D SPIHT on embedded device proposed by Jati et al. (Isa et al., 2012b, 2014; Grafika Jati et al., 2014). 2D SPIHT was tested in MIT–BIH dataset, whereas 3D SPIHT was tested in INCART data set.

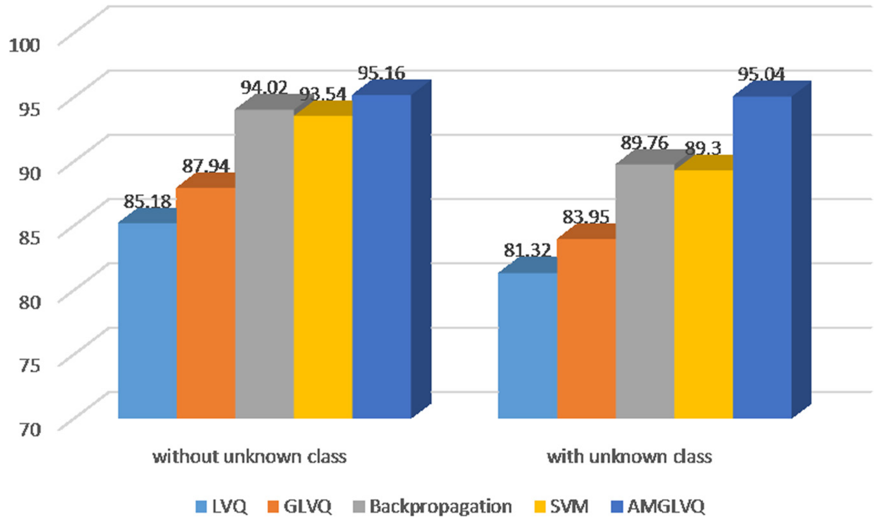


Figure 35: Arrhythmias classification with unknown class.

PRD values of the compression algorithm are shown in Table 2. Table 2 shows that 2D SPIHT and 3D SPIHT achieve a good performance, their PRD values are less than 5, with a compression ratio of 16 and 14 for 2D SPIHT, and 3D SPIHT, respectively. 2D SPIHT that was implemented in the embedded device has an acceptable compression ratio up to 32. Above this ratio, the PRD value is more than 10.

ECG telehealth services responsiveness

In addition to the classification and compression, in the previous study, the telehealth system responsiveness was also evaluated. There are URLs available in the ECG telehealth, which are as follows: /RegisterDoctor, /RegisterPatient, /LookHistory, /UploadHistory, /GetHospitalData, /GetDoctorData, /VerifyHistory, /GetUnverifiedHistory, /RegisterAffiliation, /RegisterHospital, and /GetDoctorAffiliation. Each URL has a different function and service itself. In the previous study, we measured the responsiveness of the service.

The response time for the ECG telehealth is shown in Table 3. In the table, the mean response time varies from 129.6 ms (register patient) to 556.4 ms (upload history). Upload history takes longest service time because this service is used to upload the heartbeat

Table 2. ECG compression performance.

Compression algorithm	Data set	Proposed by	CR	PRD
2D SPIHT	MIT–BIH	Isa et al. (2012b)	13	2.49
			16	2.82
3D SPIHT	INCART Arrhythmia	Isa et al. (2014)	14	4.58
2D SPIHT on embedded device	MIT–BIH	Grafika Jati et al. (2014)	8	1.48
			16	2.17
			24	3.17
			32	4.06
			64	11.00
			128	12.32

Table 3. ECG compression performance.

No	URL	Mean response time (ms)
1	/RegisterPatient	129.6
2	/RegisterDoctor	218.9
3	/LookHistory	206.6
4	/UploadHistory	556.4
5	/GetHospitalData	231.7
6	/GetDoctorData	192.3
7	/VerifyHistory	160.7
8	/GetUnverifiedHistory	227.8
9	/RegisterAffiliation	201.8
10	/RegisterHospital	215.0
11	/GetDoctorAffiliation	423.9
Mean		251.3

data by the patient. The size of the data is relatively larger than the patient's identity data. Therefore, this service takes the longest response time. The average service response time is 251.3 ms.

Arrhythmias classification in FPGA

Jatmiko et al. developed the architecture of FNGLVQ and GLVQ in FPGA. The architecture of the classifiers and the state diagram is explained in the previous section. The classifiers that were implemented in the FPGA were evaluated using MIT–BIH data set. However, only five sub-sets of the MIT–BIH data set were used in this experiment. The goal of the experiment is to find the performance of the classifiers in the FPGA and compare it to high language (Matlab) version.

The result of the experiment is shown in Figure 36. Figure 36 shows that there are similar trends in all sets. FPGA implementation of FNGLVQ has a higher accuracy compared to FPGA implementation of GLVQ. However, the accuracy is lower compared to Matlab version of FNGLVQ. Overall, the GLVQ implementation in FPGA achieved 68.3% accuracy, whereas FNGLVQ implementation in FPGA achieved 75.16% accuracy. Matlab implementation of FNLGVQ achieved 81.41% accuracy.

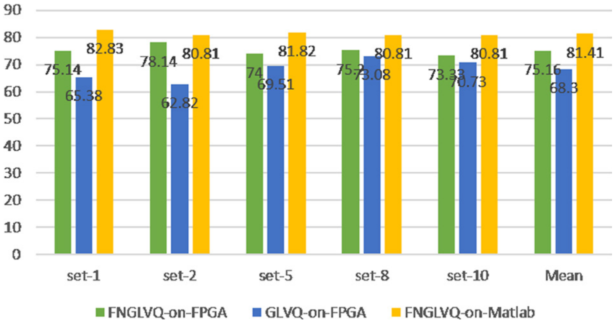


Figure 36: Arrhythmias classification in FPGA.

Conclusion

We have elaborated the progress and challenges in developing Tele-ECG system for early detection and monitoring heart disease. This includes the data acquisition, data and feature extraction, data compression, classification algorithm, web and mobile system implementation, and micro device implementation on the FPGA board. Tele-ECG system provides new fundamental interaction between doctor and patients by exploiting technology. Tele-ECG system helps to monitor the anomalies of heartbeat and reduces the risk of heart attack. Early detection of heart beat should be conducted; thus, the risk can be eliminated, which, in turn, can reduce the mortality of heart attack.

Several issues regarding Tele-ECG should be considered and explored. Issues in various areas such as security, privacy, and the effect of embedding the new device in a patient should be explored. Nevertheless, researchers should also consider about the adaptation of new technology in the medical field. Thus, the Tele-ECG is a promising technique to diminish the risk of heart failure by giving an early warning, speed up the diagnosis, improve patient treatment and long-term outcomes, and alleviate patient discomfort and travel time.

Future works

Nowadays, the data growth in any area is exceptional, including the medical field. Based on specific medical data, anyone could learn patterns and draw conclusions regarding someone's life. Therefore, the concern of security and privacy in telehealth system should be considered. It involves several security risks, such as medical entity breach, appointment data privacy, and patient privacy. Therefore, we will conduct deep exploration in ECG data encryption and employ differential privacy methods in a medical record.

Acknowledgments

This study is supported by Insentif Riset Sistem Inovasi Nasiona (INSINAS) Research Grant No. 5459/UN2.R3.1/HKP05.00/2018, entitled 'Pengembangan Sistem Telehealth Cerdas Terintegrasi Berbasiskan Perangkat Portable dan Big Data Platform untuk Meningkatkan Pelayanan Kesehatan di Indonesia' from Ministry of Research and Higher Education, Republic of Indonesia.

Literature Cited

Afif, I. N., Wardhana, Y. and Jatmiko, W. 2015. Implementation of adaptive fuzzy neuro generalized learning vector quantization (AFNGLVQ) on field programmable gate array (FPGA) for real world application. 2015 International Conference on Advanced Computer Science and Information Systems, Depok, pp. 65–71.

Dabiri, F., Massey, T., Noshadi, H., Hagopian, H., Lin, C. K., Tan, R., Schmidt, J. and Sarrafzadeh, M. 2009. A telehealth architecture for networked embedded systems: a case study in in vivo health monitoring. *IEEE Transactions on Information Technology in Biomedicine* 13(3): 351–359.

Feng, D., Kim, J., Khadra, M., Hudson, D. L. and Roux, C. 2015. Guest editorial: telehealth systems and applications. *IEEE Journal of Biomedical and Health Informatics* 19(1): 81.

Figueiredo, C. G. and Michael, T. 2013. A VLSI architecture for arrhythmia detection and its implementation on FPGA. 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Tiruchengode, pp. 1–5.

Fitria, D., Ma'sum, M. A., Imah, E. M. and Gunawan, A. A. 2014. Automatic Arrhythmias detection using various types of artificial neural network based learning vector quantization (LVQ). *Jurnal Ilmu Komputer dan Informasi* 7(2): 90–100.

Grafika Jati, M., Ma'sum, A. and Jatmiko, W. 2014. 2D set partitioning in hierarchical tree (SPIHT) on embedded devices for multilead ECG signal compression. *International Conference on Physics and Its Applications,* Solo, Indonesia.

Hababeh, I., Khalil, I. and Khreishah, A. 2015. Designing high performance web-based computing services to promote telemedicine database management system. *IEEE Transactions on Services Computing* 8(1): 47–64.

Hayn, D., Jammerbund, B. and Schreier, G. 2012. QRS detection based ECG quality assessment. *Physiological Measurement.* 33(9): 1449–1461.

Imah, E. M., Jatmiko, W. and Basaruddin, T. 2012. Adaptive multilayer generalized learning vector quantization (AMGLVQ) as new algorithm with integrating feature extraction and classification for Arrhythmia heartbeats classification. 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 150–155.

Isa, S. M., Noviyanto, A., Jatmiko, W. and Arymurthy, A. M. 2012a. The effect of electrocardiogram signal compression using beat reordering and SPIHT on automatic sleep stage classification. *Procedia Engineering* 41: 888–896.

Isa, S. M., Jatmiko, W. and Arymurthy, A. M. 2012b. Beat reordering for optimal electrocardiogram signal compression using SPIHT. 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 226–231.

Isa, S. M., Jatmiko, W. and Arymurthy, A. M. 2014. 3D SPIHT for multi-lead ECG compression. 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 488–493.

Jatmiko, W., Ma'sum, M. A., Isa, S. M., Imah, E. M., Rahmatullah, R. and Wiweko, B. 2015a. Developing smart telehealth system in Indonesia: progress and challenge. 2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 29–36.

Jatmiko, W., Habibie, I., Ma'sum, M. A., Rahmatullah, R. and Satwika, I. P. 2015b. Automated telehealth system for fetal growth detection and approximation of ultrasound images. *International Journal on Smart Sensing and Intelligent Systems* 8(1): 697–719.

Jatmiko, W., Rochmatullah, R., Kusumoputro, B., Sekiyama, K. and Fukuda, T. 2009. Fuzzy learning vector quantization based on particle swarm optimization for artificial odor dicrimination system. *WSEAS Transactions on Systems* 8(12): 1239–1252.

Jatmiko, W., Setiawan, I. M. A., Ali Akbar, M., Eka Suryana, M., Wardhana, Y. and Febrian Rachmadi, M. 2016. Automatic arrhythmia beat detection: algorithm, system, and implementation. *Makara Journal of Technology* 20(2): 82–92.

Kennedy, J. 2011. Particle swarm optimization, in Sammut, C. and Webb, G. I. (Eds), *Encyclopedia of machine learning*, Springer, Boston, MA, pp. 760–766, available at: https://link.springer.com/referencework/10.1007/978-0-387-30164-8

Khamis, H., Weiss, R., Xie, Y., Chang, C.-W., Lovell, N. H. and Redmond, S. J. 2016. QRS detection algorithm for telehealth electrocardiogram recordings. *IEEE Transactions on Biomedical Engineering* 63(7): 1377–1388.

Kohonen, T. 1990. The self-organizing map. *Proceedings of the IEEE* 78(9): 1464–1480.

Kusumoputro, B., Budiarto, H. and Jatmiko, W. 2002. Fuzzy-neuro LVQ and its comparison with fuzzy algorithm LVQ in artificial odor discrimination system. *ISA Transactions* 41(4): 395–407.

Linnenbank, A., Kemmeiings, J., Muilwijk, S., Peper, A. and Grimbergen, C. 1992. An effective compression scheme for multilead ECG signals containing ectopie ventricular activation. 1992 14th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society, Vol. 2, October, pp. 527–528.

Lu, Z., Kim, D. Y. and Pearlman, W. 2000. Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm. *IEEE Transactions on Biomedical Engineering* 47(7): 849–856.

Mallat, S. G. 1989. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(7): 674–693.

Ma'sum, M. A., Jatmiko, W., Wiweko, B. and Bowolaksono, A. 2015a. Automatic fetal organs detection and approximation in ultrasound image. *International Journal on Smart Sensing and Intelligent Systems* 8(1): 720–748.

Ma'sum, M. A., Rahmah, N., Sanabila, H. R., Wisesa, H. A. and Jatmiko, W. 2015b. Automatic fetal head approximation using particle swarm optimization based Gaussian elliptical path. 2015 International Symposium on Micro-NanoMechatronics and Human Science (MHS), pp. 1–6.

Moazami-Goudarzi, M., Moradi, M. H. and Abbasabadi, S. 2006. High performance method for electrocardiogram compression using two dimensional multiwavelet transform. 2005 IEEE 7th Workshop on Multimedia Signal Processing, 30 October–2 November, Shanghai, China.

Nascimento, S. 2005. *Fuzzy clustering via proportional membership model*, Vol. 119, IOS Press, Amsterdam.

Risman, N. S., Yassin, S. N. M., Sia, C. W., Hau, Y. W. and Nayan, N. A. 2014. FPGA design and implementation of Electrocardiogram biomedical embedded system. 2014 IEEE Conference on Biomedical Engineering and Sciences (IECBES), Kuala Lumpur, pp. 566–570.

Sanabila, H. R., Kusuma, I. and Jatmiko, W. 2016. Generative oversampling method (GenOMe) for imbalanced data on Apnea detection using ECG data. 2016 International Conference on Advanced Computer Science and Information Systems, Malang, pp. 572–578.

Sargolzaei, A., Faez, K. and Sargolzaei, S. 2009. A new robust wavelet based algorithm for baseline wandering cancellation in ECG signals. 2009 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), pp. 33–38.

Sato, A. and Yamada, K. 1995. Generalized learning vector quantization. *NIPS* 8: 423–429.

Senhadji, L., Carrault, G., Bellanger, J. J. and Passariello, G. 1995. Comparing wavelet transforms for recognizing cardiac patterns. *IEEE Engineering in Medicine and Biology Magazine* 14(2): 167–173.

Setiawan, I. M. A., Imah, E. M. and Jatmiko, W. 2011. Arrhythmia classification using fuzzy-neuro generalized learning vector quantization. 2011 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Jakarta, pp. 385–390.

Soriano Marcolino, M., Minelli Figueira, R., dos Santos, J., Silva Cardoso, C., Luiz Ribeiro, A. and Alkmim, M. B. 2016. The experience of a sustainable large scale Brazilian telehealth network. *Telemedicine and e-Health* 22(11): 899–908.

Sudhamony, S., Nandakumar, K., Binu, P. J. and Issac Niwas, S. 2008. Telemedicine and telehealth services for cancer-care delivery in India. *IET Communications* 2(2): 231–236.

Suryana, M. E., Fajar, M., Tawakal, M. I., Jatmiko, W. and Setiawan, I. M. A. 2012. FNGLVQ FPGA design for sleep stages classification based on electrocardiogram signal. 2012 IEEE International Conference on Systems Man and Cybernetics (SMC), pp. 2711–2716.

Tawakal, M. I., Suryana, M. E., Noviyanto, A., and Satwika, I. P. 2012. Analysis of multi codebook GLVQ versus standard GLVQ in discriminating sleep stages. 2012 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 197–202.

Tobon, V. D. P., Falk, T. H. and Maier, M. 2016. MS-QI: a modulation spectrum-based ECG quality index for telehealth applications. *IEEE Transactions on Biomedical Engineering* 63(8): 1613–1622.

W. H. Organization. 2016. Cardiovascular Diseases (CVDs). W. H. Organization.

Xilinx. 2011. Extended Spartan-3A family overview. Xilinx inc. February, available at: www.xilinx.com/support/documentation/data_sheets/ds706.pdf.

Zhao, Q. and Zhang, L. 2005. ECG feature extraction and classification using wavelet transform and support vector machines. International Conference on Neural Networks and Brain, 2005, ICNN&B'05, 2: 1089–1092.

Zhao, Z.-D. and Chen, Y.-Q. 2006. A new method for removal of baseline wander and power line interference in ECG signals. 2006 International Conference on Machine Learning and Cybernetics, pp. 4342–4347.