amcs

MULTIPLE NEURAL NETWORK INTEGRATION USING A BINARY DECISION TREE TO IMPROVE THE ECG SIGNAL RECOGNITION ACCURACY

Hoai Linh TRAN, Van Nam PHAM, Hoang Nam VUONG

School of Electrical Engineering
Hanoi University of Science and Technology, Dai Co Viet Str., No. 1, Hanoi, Vietnam
email: linh.tranhoai@hust.edu.vn

The paper presents a new system for ECG (*ElectroCardioGraphy*) signal recognition using different neural classifiers and a binary decision tree to provide one more processing stage to give the final recognition result. As the base classifiers, the three classical neural models, i.e., the MLP (*Multi Layer Perceptron*), modified TSK (*Takagi–Sugeno–Kang*) and the SVM (*Support Vector Machine*), will be applied. The coefficients in ECG signal decomposition using Hermite basis functions and the peak-to-peak periods of the ECG signals will be used as features for the classifiers. Numerical experiments will be performed for the recognition of different types of arrhythmia in the ECG signals taken from the MIT-BIH (Massachusetts Institute of Technology and Boston's Beth Israel Hospital) Arrhythmia Database. The results will be compared with individual base classifiers' performances and with other integration methods to show the high quality of the proposed solution.

Keywords: neural classifiers, integration of classifiers, decision tree, arrhythmia recognition, Hermite basis function decomposition.

1. Introduction

The paper proposes the application of a decision tree to integrate a number of neural classifiers to perform the task of the recognition of ECG signals. This task is based on the detection of irregularities (or the arrhythmia) of heart beats from the recorded signals, in which the QRS segments are the most important parts (de Chazal *et al.*, 2004; Lagerholm *et al.*, 2000). The main idea of the integration mechanism is the fact that, for the same problem and data samples, different classifiers may give different answers according to their data processing algorithms. Since no classifier is perfect, each still has some recognition errors, but among the classifiers these errors usually are not the same. In this situation, a combination of different classifiers will help to integrate the knowledge acquired by the component classifiers to improve the overall accuracy of the final classification.

The mechanism of combining multiple classifiers to improve the overall performance has been used not only in ECG beat classification but also in pattern recognition tasks (Troć and Unold, 2010). There are different ways to integrate the individual classifiers, which can be grouped using criteria like (Kuncheva, 2004; Woźniak and Krawczyk, 2012) integrating classifiers of the same types

vs. integrating classifiers of different types, integrating using the majority voting scheme vs. integrating using weighted votes or using performance-based measures, etc. For example, in the work of Ying *et al*. (2004), as recommended by Vapnik (1999), to get a better performance of ECG signal classification than in the case of each individual classifier, a number of SVMs are combined by one additional SVM. In the works of Huifang *et al*. (2010) and Zellmer *et al*. (2009), SVMs are integrated in a simpler way, using the majority vote strategy. Another example of integrating classifiers of the same type is that given by Sajedin *et al*. (2011), who used a number of diverse MLP networks acting as base classifiers to be combined using the so-called decision templates. Hu *et al*. (1997) used two types of classifiers: the global expert (trained with the same database of ECG beats of all possible types of arrhythmias) and the local expert (trained with data taken from a specific patient), and the final answer is weighted from individual classifiers' outputs.

The method of using different types of classifiers is more popular than the one of using classifiers of the same type (but with different parameters). In this way, we can enhance the advantages of each individual
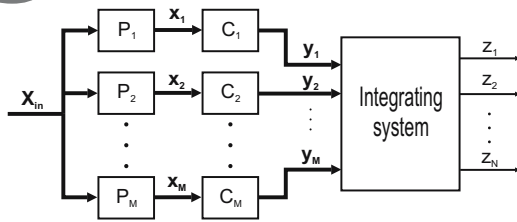
Fig. 1. General scheme of classification using many classifiers.

classifier to approach the goal. As in the work of Melgani and Bazi (2008), two types of classifiers (SVMs and a particle swarm optimization based system) are used in parallel to increase the accuracy by about 4% in ECG signal classification. In the work of Can Ye *et al.* (2012), two classifiers, one of the multi-class type and one of the two-class type, are combined to correct the final classification results by 1.4% and 2.7% for the class V and the class S beats, respectively. Ramirez *et al.* (2010) use fuzzy k-nearest neighbors and an MLP network combined by a Mamdani fuzzy inference system, which has an accuracy of 98% compared with 97.33% as the best individual result. Some articles already used the Decision Tree (DT) for ECG signal classification. But as a base classifier, the classical DT has quite limited performance due to its simplicity (the binary model usually has only simple splitting conditions like $x_i > c$), so direct application is not recommended. An example of DT modification is presented by Yujin *et al.* (2011), where in each decision node of the tree, instead of using simple decision rules like $x_i > c$, there can be an external classifier (of any type), which is used to divide the data into subgroups under the given node. This hybrid DT may boost up the accuracy in selected applications by 10%.

As mentioned earlier, combinations of classifiers can be found also in other types of pattern recognition tasks. Huan and Pan (2011) use this idea to get a better decision in SAR (*Synthetic Aperture Radar*) image target recognition, while Pagano *et al.* (2012) present an algorithm for a face recognition problem. Another example is the popular application of classifier integration in multi-sensor network problems (Chi-Hwa *et al.*, 2008).

As base ECG signal classifiers this paper will use the three classical neural models, such as the MLP, TSK and the SVM. The binary tree will take the outputs from these models to perform the final processing stage. All the four models (the three neural networks and the decision tree) are trained from the learning data sets and later tested on new data sets to check the generalization capability of the trained system. The data sets (learning and testing) are generated from QRS segments extracted from the ECG signals. These QRS segments are decomposed into Hermite basis functions (HER) in order to compose the feature vectors of the signals. The models also use

two time-based features such as the actual R-R peaks distance and the average of the last R-R peaks distances. The results of numerical experiments concerning the recognition of six types of arrhythmia and the normal sinus rhythm will be presented and discussed.

2. Integration methods

Figure 1 presents a general scheme of integration of many classifiers into one ensemble system (Lagerholm *et al.*, 2000), where M is the number of base classifiers, $\mathbf{X}_{in}$ is the input signal, $\mathbf{P}_i$ are preprocessing and feature generating blocks, $\mathbf{C}_i$ are classifying blocks, $\mathbf{z}_i$ are output signals indicating the group to which $\mathbf{X}_{in}$ is classified into. In general, $\mathbf{P}_i$ may be different for each classifiers, but in this paper we will use the same algorithm of signal preprocessing and feature extraction for all classifiers. The details of this block will be presented in Section 4.

Each classifier C_i has N outputs (corresponding to N classes), and the output signals of each classifier are arranged in the form of vectors $\mathbf{y}_i$ for $i = 1, 2, \ldots, M$. An ideal output vector $\mathbf{y}_i$ should have one position '1' and all other positions equal '0', but usually its values vary in $[0, 1]$. In this paper, the output vectors from base classifiers are concatenated to form one long input vector (of the length $N \times M$) to the integrating unit. The integrating unit will process the concatenated input vector and generate the final output vector $\mathbf{z}$ of the classifier ($\mathbf{z} \in \mathbb{R}^N$).

The integration of many classifiers into one ensemble of networks may be done using different methods. We will apply here the model of a decision tree. This choice was selected as a balance of of the model complexity. Since the base classifiers are already nonlinear and highly complex, the integrator can be a simpler model. On the other hand, popular methods of integration like majority voting, weighted voting, etc. usually calculate a single measure and base on in to make the final recognition. The decision tree is a classical model for data recognition and classification (Monson, 1997; Quinlan, 1993). It is simpler than neural networks but still performs multi-step comparison, and there are effective methods for training the parameters of the tree to adapt it to given data sets. Consequently, in our experiments we hope that the DT can have better performance than methods of integration like majority voting, weighted voting, etc. Among various models of decision trees, we will apply in this paper the linear model of a binary tree. This means the tree will use only simple single conditions such as "if x_i op A" at its nodes, where "op" includes comparing operators such as $=, >, <, \geq, \leq$.

A general structure of the binary decision tree is given in Fig. 2. This is the most popular type of decision tree, since any higher order tree can be transformed into an equivalent binary form, in which each node has maximally two subnodes. An example of node transformation is

shown in Fig. 3.

For the selected type of trees, our task of tree training (or tree building) is to find a tree structure and the splitting conditions of each tree non-leaf nodes to get the best performance on the given data samples. There are a number of algorithms to train this type of tree. In this paper we will use a popular algorithm, so-called ID3 (*Iterative Dichotomiser 3*) (Monson, 1997; Quinlan, 1993), which use nodes' entropy gain function to optimize the structure of the tree and the splitting conditions for each node of the tree. According to that, if at a node V we have N samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ belonging to M classes $C_1, C_2, \ldots, C_M$, then the entropy of the node is given as

$$E(V) = \sum_{i=1}^{M} -p_i \log_2(p_i), \tag{1}$$

where

$$p_i = \frac{|\{\mathbf{x}_j : \mathbf{x}_j \in C_i\}|}{N}$$

is the probability that a sample $\mathbf{x}_j$ of the node belongs to the class C_i. Now with a splitting condition S, the samples from node V are classified to K (for binary tree $K = 2$) subnodes SV_i ($i = 1, 2, \ldots, K$), with the appropriate numbers of samples being N_i ($\sum_{i=1}^{K} N_i = N$). Then the entropy gain for node V with splitting condition S is given as

$$\mathrm{Gain}(V, S) = E(V) - \sum_{i=1}^{K} \frac{N_i}{N} E(SV_i). \tag{2}$$

A good splitting condition is the one with a maximum value of entropy gain for a given node.

3. Neural classifiers

Various classifier solutions can be applied in practice. In this paper we will consider only neural classifiers
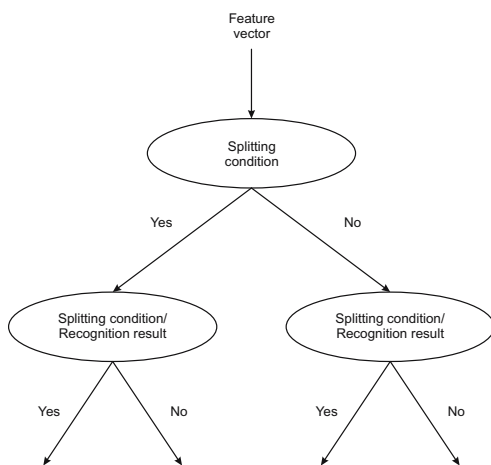
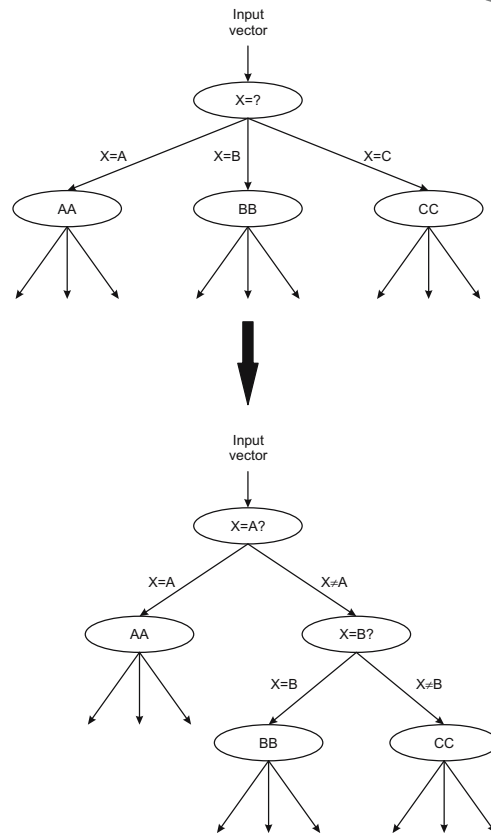

Fig. 2. Example of a binary decision tree.

Fig. 3. Example of transformation of a third-order node into an equivalent binary subtree.

of different types. They include the Multi-Layer Perceptron (MLP), neuro-fuzzy networks of a modified Takagi–Sugeno–Kang (TSK) structure and the Support Vector Machine (SVM). In the next subsections we will just briefly discuss these models, though they are already very well known and thoroughly presented in many other articles.

3.1. MLP. The multilayer perceptron (Haykin, 1999) is the most popular neural network, which consists of layers of many simple neuron-like processing units. An example of an MLP network with one hidden layer is presented in Fig. 4. With a fixed structure (the number of layers, the number of neurons in each layer, the activation function of each neuron) the weights between neurons of the MLP can be adapted during the learning process to give the desired output signals. The learning algorithm used for the MLP in this paper was the one proposed by Levenberg and Marquardt (Haykin, 1999).

3.2. TSK neuro-fuzzy network. The second neural classifier used in this paper is a modified Takagi–Sugeno–Kang network (Jang *et al.*, 1997; Łęski, 2003; Osowski and Linh, 2003).

It is implemented in the neuro-like structure realizing the fuzzy inference rules with the crisp TSK conclusion, described by the linear function. The structure of this model is shown in Fig. 5.

The TSK network can be associated with the approximation function $y(\mathbf{x})$:

$$y(\mathbf{x}) = \sum_{i=1}^{K} W(\mathbf{x} \approx \mathbf{A}_i) f_i(\mathbf{x}), \qquad (3)$$

where $W(\mathbf{x} \approx \mathbf{A}_i)$ is the activation level of the inference rule given as the product of fuzzifiers' Gaussian membership functions (Osowski and Linh, 2003),

$$W(\mathbf{x} \approx \mathbf{A}_i) = \prod_{j=1}^{N} W_j(\mathbf{x}_j \approx A_{ij}), \qquad (4)$$

and $f_i(\cdot)$ are linear TSK functions (Jang *et al.*, 1997; Osowski and Linh, 2003),

$$f_i(\mathbf{x}) = p_{i0} + \sum_{k=1}^{N} p_{ik} x_k. \qquad (5)$$

The parameters of the premise part of the inference rules (the membership values $W_j(x_j \approx A_{ij})$ and the
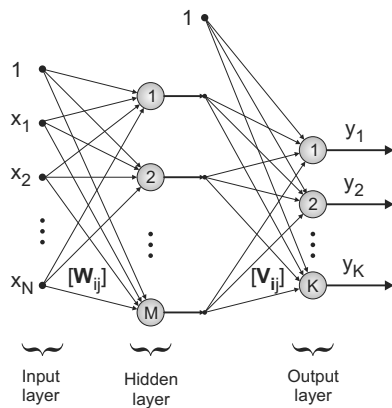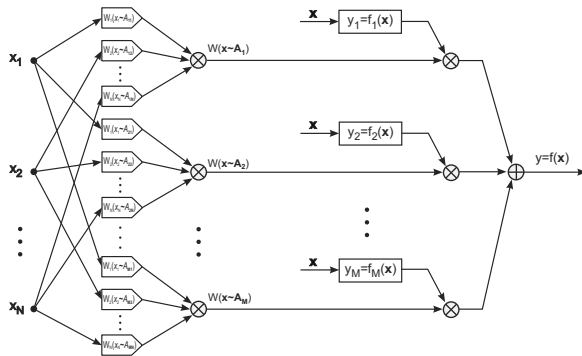

Fig. 4. MLP network with one hidden layer.

Fig. 5. Structure of a modified TSK network.

parameters p_{ij} of the linear TSK functions are trained by using a hybrid algorithm containing two repeating stages: learning linear parameters and learning nonlinear parameters (Osowski and Linh, 2003).

3.3. SVM classifier. The last base classifier used in this paper is the support vector machine network (Scholkopf and Smola, 2002; Vapnik, 1998; 1999). It is known as an efficient tool for classification problems, of a very good generalization ability. The SVM is a linear machine working in a high-dimensional feature space formed by the nonlinear mapping of the N-dimensional input vector $\mathbf{x}$ into a K-dimensional feature space ($K > N$) through the use of a nonlinear function $\varphi(\mathbf{x})$. The equation of the hyperplane separating two classes is defined in terms of these functions,

$$y(\mathbf{x}) = \sum_{j=1}^{K} w_j \varphi_j(\mathbf{x}) + b = 0,$$

where b is the bias and w_j the synaptic weight of the network. The parameters of this separating hyperplane are adjusted in such a way as to maximize the distance between the closest representatives of both classes. In practice, the learning problem of the SVM can be solved in two stages involving the solution of the primary and dual problems (Scholkopf and Smola, 2002; Vapnik, 1998).

The most distinctive fact about the SVM is that the learning task is simplified to quadratic programming by introducing the Lagrange multipliers α_i. All operations in learning and testing modes are done in the SVM using kernel functions $K(\mathbf{x}, \mathbf{x}_i)$, satisfying the Mercer conditions (Osowski and Linh, 2003; Osowski *et al.*, 2004). The best known kernels are Gaussian, polynomial, linear or spline functions. The output signal y(x) of the SVM network is finally determined as

$$y(\mathbf{x}) = \sum_{i=1}^{p} \alpha_i d_i K(\mathbf{x}_i, \mathbf{x}) + b, \qquad (6)$$

where $d_i = \pm 1$ is the binary destination value associated with the input vector $\mathbf{x}_i$. The positive value of the output signal means membership of the vector $\mathbf{x}$ to the particular class, while the negative one—to the opposite one. Although the SVM separates the data into two classes only, the recognition of more classes is straightforward by applying either one-against-one or one-against-all methods (Hsu and Lin, 2002). The more powerful is one against-one approach, in which many SVM networks are trained to recognize between all combinations of two classes of data. For N classes, we have to train $N(N-1)/2$ individual SVM networks. In the retrieval mode, the vector $\mathbf{x}$ belongs to the class of the highest number of winnings in all combinations of classes.
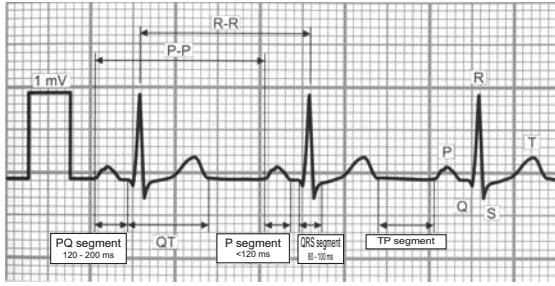
Fig. 6. Typical ECG signal and its characteristic peaks P-Q-R-S-T.

Fig. 7. Hermite basis functions of different orders.

4. ECG signals and the feature extraction method

4.1. Hermite representation of ECG. An important step in building the efficient classifier system is diagnostic features extraction. In our approach to the problem, we applied the QRS complex decomposition into Hermite basis functions and used the decomposition coefficients as the features of the ECG signals. These coefficients, together with two classical time-based features: the instantaneous R-R interval of the beat (the time span between two consecutive R peaks) and the average R-R interval of 10 preceding beats, form the feature vector $\mathbf{x}$ applied to the input of the classifier (Osowski and Linh, 2003; Osowski *et al.*, 2004; 2006).

In the Hermite basis function expansion method, we have Hermite polynomials defined by a recurrent formula:

$$H_{n+1}(t) = 2t \cdot H_n(t) - 2n \cdot H_{n-1}(t), \qquad (7)$$

for $n \geq 1$, where $H_0(x) \equiv 1$, $H_1(x) = 2x$. The Hermite basis functions are

$$\phi_n(t) = \left(\sqrt{\pi} \cdot 2^n \cdot n!\right)^{-\frac{1}{2}} e^{-\frac{t^2}{2}} H_n(t). \qquad (8)$$

Denote by $s(t)$ the QRS complex of the ECG curve. Its expansion for given time points $t_0, t_1, \ldots$, spread on the signals duration into a Hermite series may be written as

$$s(t_i) \approx \sum_{n=0}^{N-1} c_n \phi_n(t_i), \qquad (9)$$

where c_n are the expansion coefficients, $\phi_n(t_i)$ are the Hermite basis functions of the n-th order (Lagerholm *et al.*, 2000; Osowski *et al.*, 2006), N is the number of Hermite basis functions used in the decomposition. As can be seen in Fig. 7, the Hermite basis function of higher order contains more higher frequency components.

In this paper, $N = 16$ was selected because the 15-th order Hermite basis function contains frequency components high enough to approximate sudden changes in the ECG signals (Osowski and Linh, 2003; Osowski *et al.*, 2006). The coefficients c_n of a Hermite basis
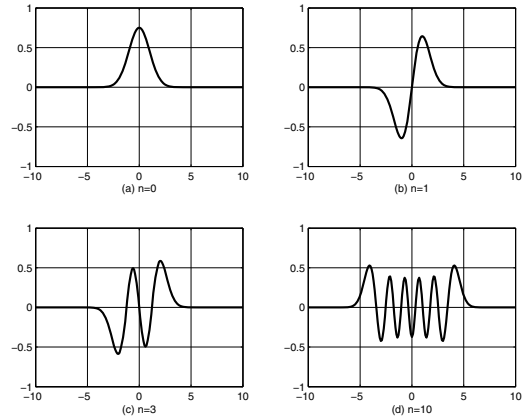
function expansion may be treated as features used in the recognition process. They may be obtained by minimizing the sum squared error

$$E = \sum_i \left[s(t_i) - \sum_{n=0}^{N-1} c_n \phi_n(t_i, \sigma) \right]^2 \qquad (10)$$

by using singular value decomposition (Lagerholm *et al.*, 2000).

In numerical computations, we used the QRS segments of the ECG signals, whose length was 91 data points around the R peak (45 points before and 45 ones after) (Osowski and Linh, 2003). At the data sample rate of 360 Hz, this gives a window of 250 ms, which is long enough to cover most of QRS signals. The data were also additionally zero-padded by adding 45 zeros to each end of the QRS segment in order to make the signal look more similar to the Hermite basis function (Osowski and Linh, 2003).

4.2. ECG signal database. For comparison purposes, we use similar signals taken from the popular MIT-BIH Arrhythmia Database (Moody and Mark, 2001) like in the work of Osowski *et al.* (2006). The recognition of arrhythmia proceeds on the basis of the QRS segments of the registered ECG waveforms of 19 patients (records number 100, 105, 106, 109, 111, 114, 116, 118, 119, 124, 200, 202, 207, 208, 209, 212, 214, 221 and 222). From all the selected records, only the waveforms from the lead number of 1 were used. Some patients have more than one arrhythmias in their records. The worst case is the patient number 207, which has all seven types of arrhythmia. An important difficulty in accurate recognition of the arrhythmia type is the large variability of the morphology of the ECG rhythms belonging to the same class (Moody and Mark, 2001). Moreover, the beats belonging to different classes are also morphologically alike. Hence confusing different classes is very likely.

Table 1. Number of learning and testing samples of each beat type.

Beat type	Learn samples	Test samples
N	1065	935
L	639	561
R	515	485
A	504	398
V	549	451
I	271	201
E	68	37

In our numerical experiments, like in the work of Osowski *et al.* (2006), we considered six types of arrhythmia: left bundle branch block (L), right bundle branch block (R), atrial premature beat (A), ventricular premature beat (V), ventricular flutter wave (I), ventricular escape beat (E), and the waveforms corresponding to the normal sinus rhythm (N). 3611 data pairs were generated for the purpose of learning and 3068 were used for testing purposes.

Table 1 presents the number of representatives of the beat types used in learning and testing. The limited number of representatives of some beat types (for example, E or I) is a result of the limitation of the MIT-BIH database (Moody and Mark, 2001). The original number of normal beats is very high, but as we consider the normal beats to be the "easy" ones to classify and in order to make the overall results more independent of the number of samples of each groups, we limited the number of normal beats used in the experiments to a reasonable level of about 1000 beats (Osowski *et al.*, 2006).

The set of all samples was split into learning and testing sets using the following method. First, we randomly select about 50% of samples of each arrhythmia's type into the learning set, the rest belongs to the testing set. Then we iteratively update the two sets with the following method:

- Train the nets (using the learning set) and test them (with the testing set).

- After that, randomly select a small subset (from 3 to 5) of learning samples, which were correctly learned to move to the testing set. Then randomly select a small subset (from 3 to 5) of testing samples which failed the test to move back to the learning set.

This iterative process is done when we cannot achieve better testing performance for 20 consecutive trials. After that, the two sets corresponding to the best testing performance are selected for further application.

5. Numerical experiments and results

With three base classifiers, the MLP, the SVM and modified TSK, their parameters were trained separately on the same sets of features of the beats from the same learning data set. The MLP needed only one hidden layer with 20 neurons, 7 output neurons (corresponding to 7 types of arrhythmia). The structure of the MLP was selected by the trial-and-test method, in which we started with a network with a small number of hidden neurons and we increased the hidden neurons until a good testing error was achieved. The SVM learning parameters were selected as in the work of Osowski *et al.* (2004). With 7 classes and the one-against-one method to find the winner class in the SVM, a total number of 21 SVMs were trained. The TSK network's structure was selected as in the work of Osowski and Linh (2003); the final model had 21 rules and 7 outputs. After that their results formed the input (by concatenation) for the DT, and there will be one more training process of the parameters of the DT to map the outputs of base classifiers to the desired result of classification. In this way the integration of 3 base classifiers was created. The trained model is later verified using new testing data sets, not containing the samples used in the training process. From the practical point of view, the testing results are more important than those of learning because they present the capability of generalization of the trained model. We will present the results of the testing mode and compare them with those from previous works.

Table 2 presents the results of testing all individual classifiers and the integrated system. All classifier networks were first learned on the same learning data set and then tested on another testing data set, the same in all cases. The best results of single classifiers were achieved by the SVM network. The MLP network generated the worst results. The relative difference between the accuracy of the best and the worse classifier is very large (almost 250%). It can be seen that the integration mechanism (by using the trained decision tree) gave better results, which were improved to the level of 1.24%.

These results once again proved that integration of many classifiers significantly improves the recognition results. The improvement rate depends on the integration scheme applied and the quality of the individual classifiers. The quality of results can be assessed in detail on the basis of the error distribution within different beat types. Table 3 presents the distribution of classification errors for the testing data in the form of a confusion matrix

Table 2. Testing errors of the individual base classifiers and integration of classifiers.

Classifier system	No. of errors	% of errors
MLP	148	4.82%
TSK	100	3.26%
SVM	60	1.96%
Integrated with the DT	38	1.24%

Table 3. Detailed classifying results for 7 types of rhythms of testing data.

	N	L	R	A	V	I	E
N	928	1	1	10	0	0	0
L	1	557	0	2	3	1	0
R	1	0	481	1	0	2	0
A	5	0	3	383	1	0	0
V	0	2	0	1	446	0	0
I	0	0	0	1	1	198	0
E	0	1	0	0	0	0	37
Total	935	561	485	398	451	201	37

divided into beat types. The diagonal entries of this matrix represent right recognition of the beat type and the off diagonal—the misclassifications. The column presents the original type. The row presents the results given by the integrated classifiers.

The analysis of the error distribution shows that some classes are confused more frequently than others. It is evident that most misclassifications were committed between two classes: N and A (5 N-rhythms were classified as A-rhythms and 10 A-rhythms were recognized as N-rhythms). This confusion is a result of a high similarity of ECG waveforms for these two rhythms. The other confusions are minor (≤ 3 errors for the whole datasets).

In addition, we will present a comparison with other results done on the same data set. As presented by Osowski *et al.* (2006), four popular integration methods used to compare are majority voting, weighted voting, Kullback–Leibler and modified Bayes. Each of these methods integrate not only results from the same three base classifiers and feature sets used in this work, but also the same three classifiers with features generated based on Higher-Order Statistics (HOS) measures (Moody and Mark, 2001; Osowski and Linh, 2001; Osowski *et al.*, 2006). The misclassification rates are shown as performances in Table 4.

As can be seen, all the integrated systems have better performance than single classifiers (the worst integrated system has a testing error of 1.63% vs. the best single classifier the SVM with a testing error 1.96%), and the proposed method using the decision tree has the lowest error rate for testing data.

We checked also the quality of the combined

Table 4. Errors of different classifier integration methods.

No	Integration method	Testing error
1	Majority voting	1.63%
2	Weighted voting	1.37%
3	Kullback–Leibler	1.47%
4	Modified Bayes	1.56%
5	Decision tree	1.24%

Table 5. Results of classifying ECG beats into normal and abnormal classes.

	Normal	Abnormal
Normal (classified)	928	12
Abnormal (classified)	7	2121

classifiers with the statistics: sensitivity and specificity. To do that, we let all 6 types of arrhythmia be "abnormal type". We also let the abnormal beats be the positive cases and the normal beats the negative ones of Table 5 presents the results of classifying beats into normal vs. abnormal.

From Table 5, we have

$$\text{Sens} = \frac{TP}{TP + FN} = \frac{2121}{2121 + 21} = 99.02\%, \quad (11)$$

$$\text{Spec} = \frac{TN}{TN + FP} = \frac{928}{928 + 7} = 99.25\%. \quad (12)$$

The high values of the two statistics help to confirm the quality of the solution proposed in this paper.

6. Conclusions

An application of the decision tree to integrate the results of a set of individual neural classifiers working in parallel into one classification system was presented and discussed in the paper. The base classifiers applied include the multi layer perceptron, the modified neuro-fuzzy TSK network and support vector machine classifiers. The classifiers use features consisting of the coefficient of ECG signal decompositions using Hermite basis functions and R-R peak-to-peak periods. The experiments performed for seven heart beat types taken from MIT-BIH AD showed that, in general, integration of the results of many classifiers improves the quality of the final classification system. In particular, for the presented datasets, the method proposed in this paper achieved the best performance with the testing error equals to 1.24%. This error is about 9.5% better than in the case of the runner-up method of integration using the weighted voting mechanism (with the testing error equal to 1.37%).

References

Can Ye, Vijaya Kumar, B.V.K. and Coimbra, M.T. (2012). Combining general multi-class and specific two-class classifiers for improved customized ECG heartbeat

classification, *Proceedings of the 21st International Conference on Pattern Recognition (ICPR 2012), Arlington, VA, USA*, pp. 2428–2431.

de Chazal, P., O'Dwyer, M. and Reilly, R.B. (2004). Automatic classification of heartbeats using ECG morphology and heartbeat interval features, *IEEE Transactions on Biomedical Engineering* **51**(7): 1196–1206.

Chi-Hwa, S., Jun, W., Dong-Hun, S. and Won-Don, L. (2008). Solving multi-sensor problem with a new approach, *Proceedings of the First International Conference on the Applications of Digital Information and Web Technologies (ICA-DIWT), Ostrava, Czech Republic*, pp. 348–353.

Haykin, S. (1999). *Neural Networks. A Comprehensive Foundation, 2nd Edition*, Prentice-Hall, Englewood Cliffs, NJ.

Hu, Y.H., Palreddy, S. and Tompkins, W. (1997). A patient adaptable ECG beat classifier using a mixture of experts approach, *IEEE Transactions on Biomedical Engineering* **44**(9): 891–900.

Huan, R. and Pan, Y. (2011). Decision fusion strategies for SAR image target recognition, *Radar, Sonar & Navigation, IET* **5**(7): 747–755.

Huifang, H., Guangshu, H. and Li, Z. (2010). Ensemble of support vector machines for heartbeat classification, *Proceedings of the 10th IEEE International Conference on Signal Processing (ICSP), Beijing, China*, pp. 1327–1330.

Hsu, C.W. and Lin, C.J. (2002). A comparison methods for multi class support vector machines, *IEEE Transactions on Neural Networks* **13**(2): 415–425.

Jang, L., Sun, C.T. and Mizutani, E. (1997). *Neuro-fuzzy and Soft Computing*, Prentice-Hall, Englewood Cliffs, NJ.

Kuncheva, L. (2004). *Combining Pattern Classifiers: Methods and Algorithms*, Wiley, Hoboken, NJ.

Lagerholm, M., Peterson, C., Braccini, G., Edenbrandt, L. and Sornmo, L. (1997). Clustering ECG complexes using Hermite functions and self-organizing maps, *IEEE Transactions on Biomedical Engineering* **47**(7): 838–847.

Łęski, J. (2003). A fuzzy *if-then* rule-based nonlinear classifier, *International Journal of Applied Mathematics and Computer Science* **13**(2): 215–223.

Moody, G. and Mark, R. (2001). The impact of the MIT-BIH Arrhythmia Database, *IEEE Engineering in Medicine and Biology* **20**(3): 45–50.

Melgani, F. and Bazi, Y. (2008). Classification of electrocardiogram signals with support vector machines and particle swarm optimization, *IEEE Transactions on Information Technology in Biomedicine* **12**(5): 667–677.

Monson, L. (1997). *Algorithm Alley Column: C4.5*, Dr. Dobbs, http://www.drdobbs.com.

Nikias, C. and Petropulu, A. (1993). *Higher-Order Spectra Analysis: A Nonlinear Signal Processing Framework*, Prentice-Hall, Englewood Cliffs, NJ.

Osowski, S. and Tran, H.L. (2001). ECG beat recognition using fuzzy hybrid neural network, *IEEE Transactions on Biomedical Engineering* **48**(11): 1265–1271.

Osowski, S. and Tran, H.L. (2003). On-line heart beat recognition using Hermite polynomials and neuro-fuzzy network, *IEEE Transactions on Instrumentation and Measurement* **52**(4): 1224–1230.

Osowski, S., Tran, H.L. and Markiewicz, T. (2004). Support vector machine based expert system for reliable heart beat recognition, *IEEE Transactions on Biomedical Engineering* **51**(4): 582–589.

Osowski, S., Markiewicz, T. and Tran, H.L. (2006). Ensemble of neural networks for improved recognition and classification of arrhythmia, *Proceedings of the XVIII International Measurement Confederation World Congress, Rio de Janeiro, Brazil*, pp. 201–206.

Pagano, C., Granger, E., Sabourin, R. and Gorodnichy, D.O. (2012). Detector ensembles for face recognition in video surveillance, *Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia*, pp. 1–8.

Quinlan, J.R (1993). *C4.5 Programs for Machine Learning*, Morgan Kaufmann Publishers, San Francisco, CA.

Ramirez, E., Castillo, O. and Soria, J. (2010). Hybrid system for cardiac arrhythmia classification with fuzzy k-nearest neighbors and multi layer perceptrons combined by a fuzzy inference system, *Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain*, pp. 1–6.

Sajedin, A., Ebrahimpour, R. and Garousi, T.Y. (2011). Electrocardiogram beat classification using classifier fusion based on decision templates, *Proceedings of the 5th IEEE International Conference on Cybernetic Intelligent Systems (CIS), Quindao, China*, pp. 7–12.

Scholkopf, B. and Smola, A. (2002). *Learning with Kernels*, MIT Press, Cambridge, MA.

Troć, M. and Unold, O. (2010). Self-adaptation of parameters in a learning classifier system ensemble machine, *International Journal of Applied Mathematics and Computer Science* **20**(1): 157–174, DOI: 10.2478/v10006-010-0012-8.

Vapnik, V. (1998). *Statistical Learning Theory*, Wiley, New York, NY.

Vapnik, V. (1999). *The Nature of Statistical Learning Theory*, Springer, New York.

Woźniak, M. and Krawczyk, B. (2012). Combined classifier based on feature space partitioning, *International Journal of Applied Mathematics and Computer Science* **22**(4): 855–866, DOI: 10.2478/v10006-012-0063-0.

Ying, Y., Xiao-Long, W. and Bing-Quan, L. (2004). A gradual combining method for multi-SVM classifiers based on distance estimation, *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics, Shanghai, China*, pp. 3434–3438.

Yujin, Z., Yihua, T., Haitao L. and Haiyan, G. (2011). A multi-classifier combined decision tree hierarchical classification method, *Proceedings of the 2011 International Symposium on Image and Data Fusion (ISIDF), Yunnan, China*, pp. 1–3.

Zellmer, E., Fei, S. and Hao, Z. (2009). Highly accurate ECG beat classification based on continuous wavelet transformation and multiple support vector machine classifiers, *Proceedings of the 2nd International Conference on Biomedical Engineering and Informatics, Tianjin, China*, pp. 1–5.

Hoai Linh Tran is an associate professor in the School of Electrical Engineering of the Hanoi University of Science and Technology. He received the M.Sc. degree in applied informatics in 1997, the Ph.D. and the habilitation degree in electrical engineering in 2000 and 2005, respectively, from the Warsaw University of Technology. His current research concentrates on application of artificial intelligence in modern instrumentation devices and in power system control.

Van Nam Pham is a Ph.D. student in the School of Electrical Engineering of the Hanoi University of Science and Technology. He received the M.Sc. degree in instrumentation and industrial informatics in 2008 from the same university. His current research interest is in implementation of a smart mobile ECG recorder with automatic arrhythmia detection and classification.

Hoang Nam Vuong was born in 1980 in Hanoi, Vietnam. He received the M.Sc. and Ph.D. degrees in 2005 and 2012, respectively, from the Hanoi University of Science and Technology. He is currently a lecturer in the School of Electronics and Telecommunications, Hanoi University of Science and Technology. His professional research interests are biomedical signal processing, multimedia applications.