Dual-Phase Approach to Improve Prediction of Heart Disease in Mobile Environment

Yang Koo Lee, Thi Hong Nhan Vu, and Thanh Ha Le

In this paper, we propose a dual-phase approach to improve the process of heart disease prediction in a mobile environment. Firstly, only the *confident* frequent rules are extracted from a patient's clinical information. These are then used to foretell the possibility of the presence of heart disease. However, in some cases, subjects cannot describe exactly what has happened to them or they may have a silent disease — in which case it won't be possible to detect any symptoms at this stage. To address these problems, data records collected over a long period of time of a patient's heart rate variability (HRV) are used to predict whether the patient is suffering from heart disease. By analyzing HRV patterns, doctors can determine whether a patient is suffering from heart disease. The task of collecting HRV patterns is done by an online artificial neural network, which as well as learning knew knowledge, is able to store and preserve all previously learned knowledge. An experiment is conducted to evaluate the performance of the proposed heart disease prediction process under different settings. The results show that the process's performance outperforms existing techniques such as that of the self-organizing map and gas neural growing in terms of classification and diagnostic accuracy, and network structure.

Keywords: Healthcare service, heart disease, rule-based classification, neural network, prediction.

Manuscript received Aug. 9, 2014; revised Jan. 4, 2015; accepted Jan. 17, 2015.

This work was supported by the ICT R&D program of MSIP/IITP (10044844, Development of ODM-Interactive Software Technology supporting Live-Virtual Soldier Exercises).

Yang Koo Lee (yk_lee@etri.re.kr) is with the IT Convergence Technology Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Thi Hong Nhan Vu (vthnhan@vnu.edu.vn) and Thanh Ha Le (halt@vnu.edu.vn) are with the Faculty of Information Technology, UET, Vietnam National University, Hanoi, Vietnam.

I. Introduction

Wearable computing technology and wireless communications have been developed and used successfully in areas such as surveillance, human action recognition, virtual reality gaming, and training simulations [1]–[2]. Advances in these fields have helped pave the way for the advent of mobile healthcare services. A healthcare system can continually monitor a person's physical condition and detect abnormal activities using bio-signals acquired from body sensors [3]. The World Health Organization estimated that there would be about 23.6 million deaths caused by heart disease by 2030 [4].

Traditionally, heart disease is often predicted based on risk factors and symptoms. It can be diagnosed based on a number of tests; for instance, magnetic resonance imaging or electrocardiography (ECG). A point score prediction probability algorithm can be applied to estimate a 5- and 10-year risk of heart disease for individuals free of cardiovascular disease [5]. Currently, there is a lack of effective techniques that can efficiently interpret physiological signals recorded from sensors into some form of knowledge that is understandable to humans; this subsequently makes it very difficult when using raw data to try to correctly diagnose a person suffering from a cardiovascular disease. To address this problem, statistical analysis and data mining techniques have been developed to extract relationships from large clinical databases [6]–[9]. However, most of the related algorithms in the literature do not execute in real time [10].

Discriminant function analysis, which is based on logistic regression, can be used to estimate the probability of a disease; however, the results obtained from using such a technique are not easily interpretable [11]. Artificial neural network (ANN) models, such as multilayer perceptron [12], are well-known

tools for multivariate analysis and disease risk prediction in the field of data classification. Conventional ANNs only function when a whole dataset is known in advance; thus, they fail to predict an individual's risk of heart disease in a non-stationary environment. So far, some online learning methods in the field of data stream mining have been proposed cell structures [13], self-organizing map (SOM) [14], and growing neural gas (GNG) [15]. The biggest challenge for these machine learning techniques in a mobile environment is to preserve previously learned knowledge while learning new knowledge continuously and preventing overfitting.

In our novel predictive framework, a patient's clinical information, such as age, gender, serum cholesterol, glucose intolerance, and so on, is used to foretell the possibility of the presence of heart disease within the patient. To this end, patients are first classified into different heart disease risk levels. An association rule algorithm is then introduced to discover the relationships between the heart disease risk factors of patients. The *confident* frequent rules are extracted from the dataset of risk factors and are used to predict a patient's likelihood of contracting heart disease in the future. In practice, if physicians rely only on results that are a product of statistical analyses of static information, then this may lead them to incorrectly diagnose a patient or to fail to identify the presence of a disease altogether. Accordingly, for a doctor to improve the degree of certainty to which they can be sure of the presence of heart disease in a patient, the doctor must have a long-term record of the patient's ECG signals. In contrast to the discrete and static characteristics of clinical information, heart rate unceasingly alters over time. The abnormal state of a patient's heart can be recognized by examining the patient's heart rate variability (HRV) patterns, which are discovered by the online neural network PHIAN, introduced in [16], under different settings.

PHIAN is a classification model consisting of three layers; namely, input, middle, and output. The first layer is used to receive data from the input space. The middle layer is composed of neurons organized in a dynamic graph. The role of the neurons in the classification task is to separate the input dataset into classes. The output layer is responsible for separating the neurons into a number of decision regions in the output space. In a mobile environment, all of the data are not known prior to training the classification model; thus, new datasets accompanied with new classes may appear later. Hence, the classification model should be able to learn new classes continuously without forgetting the old ones. For this purpose, an adaptive and incremental learning strategy is applied in the training process of the PHIAN model. At each step of the training process, signals from ECG sensors and accelerometers are fed into the PHIAN model after being transformed into the form of a vector. Generally, input data cannot be linearly separated into classes, and there is some overlap between classes. To tackle the problem of non-linear classification, a Gaussian radial basis function (RBF) is used as an activation function. The PHIAN model starts with two neurons located randomly in the input space and is supplemented with new ones as training progresses. When the training process terminates, we obtain decision regions that are separated in the output space — each decision region corresponds to a class. To evaluate the proposed heart disease prediction approach in comparison with two previous online learning methods, SOM and GNG, we build a prototype system that firstly classifies the patients into three groups, each group corresponding to a risk level of heart disease.

A PHIAN model is constructed for each group of patients to categorize the patients into two classes, "Yes" or "No," of heart disease. To validate the performance of PHIAN, eleven scenarios of three different daily activities are set up to collect datasets for training and testing the classification model. The evaluation criteria include how well the input data distribution is represented by classes and PHIAN's ability to learn new patterns while preserving old ones (in a non-stationary environment). The experimental results show that PHIAN outperforms the existing techniques in terms of prediction accuracy and classification model complexity.

In summary, our predictive approach is able to determine to what extent a person is at risk from heart disease. With the support of location tracking techniques [17]–[19], it can be integrated in telemedicine systems to provide context-aware healthcare services anytime, anywhere.

II. Dual-Phase Heart Disease Prediction Framework

The framework shown in Fig. 1 is a new approach that enables doctors to monitor subjects even when they are out of hospital going about their daily routine. To estimate the degree of seriousness of heart disease in a patient and then make an effective decision about treatment, cardiac physicians first examine the patient's clinical information, such as age, gender, serum cholesterol, whether they smoke, systolic blood pressure, left ventricular hypertrophy, glucose intolerance, and so on. The patient is then asked about possible symptoms; for example, they may be asked about squeezing pains in the chest and shortness of breath. Such an examination is largely based on static clinical information and is not sufficient for a doctor to state with any great degree of certainty as to whether a patient is suffering from heart disease or not.

Since heart disease has a strong connection to HRV patterns, doctors need to analyze the patient's heart rate when the patient is undergoing some physical activities to be more certain as to whether or not they have heart disease.

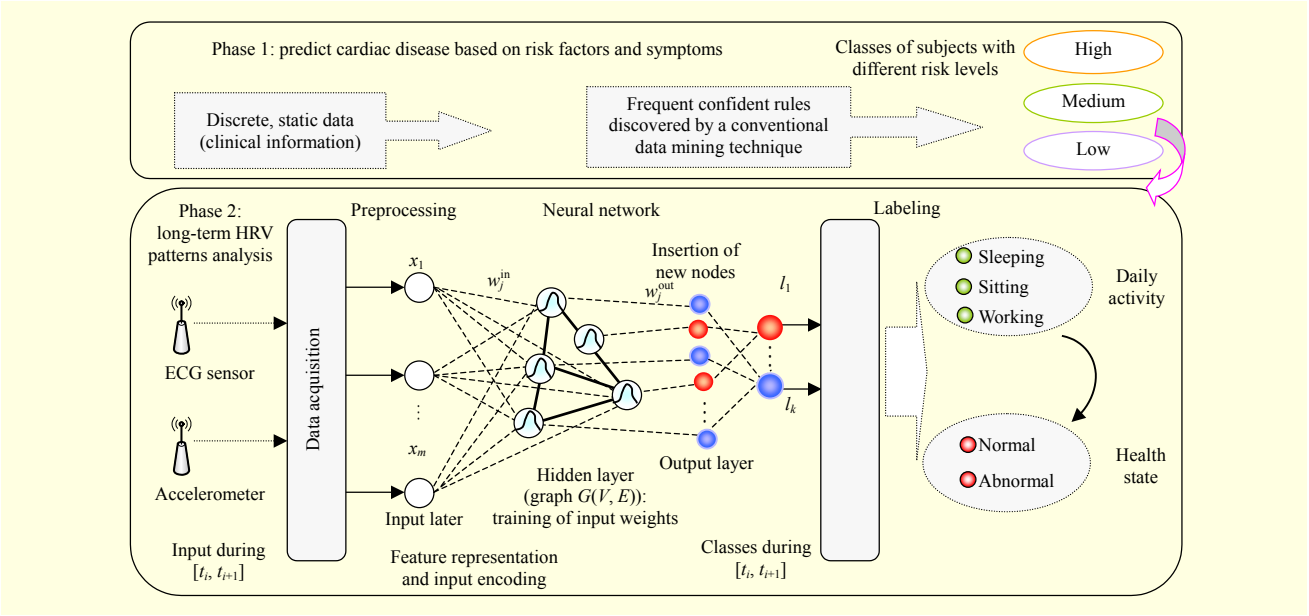

Fig. 1. Dual-phase framework for heart disease diagnosis.

Our proposed heart disease prediction process can be divided into two phases according to the properties of the risk factors used in a medical-decision support system for diagnosis of heart disease. Firstly, a rule-based classification technique uses patients' clinical information to categorize the patients into different classes. Secondly, patient HRV patterns are discovered from long-term ECG recordings. This task is accomplished by the online neural network model PHIAN [16]. Five main steps; namely, EGC signal collection, data pre-processing, classifier training, labeling, and performance validation, are included the second phase (see Fig. 1). A series of signals recorded from EGC sensors over an interval of time $[t_1, t_2]$ is converted into a vector $\mathbf{x} = (x_1, x_2, \ldots, x_m)$, in which each element represents a feature (extracted from the signals). Each vector is then assigned a class label. These labels represent the multiple heart states experienced by the patient during the interval $[t_1, t_2]$. These vectors are then used to train the neural network model shown in Fig. 1. This process is in fact a classification problem; thus, after a finite number of training steps, a number of distinct decision regions should begin to appear in the output space. The obtained model can then be used to assist doctors with cardiac disease diagnosis.

1. Rule Generation

To estimate an individual's level of risk of heart disease, we apply a rule-based classification technique. The technique makes use of the risk factors shown in Table 1. In a decision support system, a collection of IF-THEN rules is used. A classification rule is defined as *Condition* → *y* in which *Condition* is a combination of attributes and *y* is a single class label. An example of one such classification rule is "(*gender*=*Male*) ∧ (*fbs*=0) ∧ (*restecg*=0) ∧ (*oldpeak*∈ [0.3,*]) ∧ (*thal*=7) → (*num*=1)." Diagnosis is the output of the rule-based classification technique, which is given as a *decision* and represented by a class attribute. The class attribute indicates the level of risk of heart disease. It is the last risk factor, *num*, in Table 1.

Given a set *D* of records of risk factors and a set *Y* of class labels (*y*'s), each patient is associated with a class label *y*. Each record in *D* is called an instance. The problem is to find all of the possible rules from *D*. Each combination of attribute name and attribute value (*Risk factor* = *value*) is denoted as an item. A set $I = \{i_1, \ldots, i_n\}$ of distinct items is called an *itemset*. Prior to extracting the rules, we need to transform dataset *D* into a set of itemsets. For attributes that are of an ordinal data type, the attribute name is simply associated with its value. For those that are of a continuous data type, we need to first discretize the range of continuous-valued attributes into intervals. However, the intervals influence the resulting rules and thereby the classification accuracy. Thus, to reduce the resulting misclassification error, we utilize the Gini index, which is a measure of statistical dispersion, to determine the intervals. Assume that attribute values are split into *k* intervals. The quality of this discretization is then determined by

$$\mathrm{Gini}_{\mathrm{split}} = \sum_{i=1}^{k} \frac{r_i}{r} \mathrm{Gini}(i) , \qquad (1)$$

in which r_i is the number of instances belonging to the partition *i* and *r* is the total number of instances. The impurity of each

Table 1. Risk factors of heart disease.

Factor	Meaning	Data type
Age	Age	Numerical
Oldpeak	ST depression induced by exercise relative to rest	Numerical
Threstbps	Resting systolic blood pressure on admission to the hospital (mmHg)	Numerical
Thalach	Maximum heart rate achieved	Numerical
Relaca	Number of major vessels colored by fluoroscopy	Numerical
Chol	Serum cholesterol (mg/dl)	Numerical
Gender	Gender	0 if female 1 if male
Cp	Chest pain type	1 typical angina 2 atypical angina 3 non-anginal pain 4 asymptomatic
Fbs	Fasting blood sugar over 120 mg/dl?	1 if yes 0 if no
Restecg	Resting electrocardiographic results	0 normal 1 having ST-T wave abnormality 2 LV hypertrophy
Exang	Exercise induced angina?	1 if yes 0 if no
Slope	Slope of the peak exercise ST segment	1 upsloping 2 flat 3 downsloping
Thal	Exercise thallium scintigraphic defects	3 normal 6 fixed defect 7 reversible defect
Num	Class label giving diagnosis of heart disease	0, 1. Low 2. Medium 3, 4. High

partition after discretization is determined by the following formula:

$$\mathrm{Gini}(i) = 1 - \sum_{y} p(y)^2 , \quad (2)$$

in which $p(y)$ is the number of instances belonging to a class y. If the Gini index is zero, then all instances belong to one class, which means there would be no misclassification error.

For efficient computation, the values of the attributes are firstly sorted and linearly scanned. Candidate split positions are then computed by taking the midpoint between two adjacent sorted values. Finally, the split point is determined by that that gives the minimum Gini index. Figure 2 illustrates an example of Gini index computations used to determine the split point for

Class	No		No		No		Yes		Yes		Yes		No		No		No		No	
	Cholesterol																			
	60		70		75		85		90		95		100		120		125		220	
	55		65		72		80		87		92		97		110		122		172	230
	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		0.300		0.343		0.375		0.400		0.420	

Fig. 2. Example of split point determined by minimum Gini index value.

the attribute *cholesterol* with the assumption that there are two classes, "Yes" and "No." After computing the Gini indexes for the attribute cholesterol, the selected split point is 97, which corresponds to the smallest Gini index, 0.300.

Assume that all the itemsets are lexicologically sorted. If an itemset $C \subseteq I$, then we say that I satisfies C. *Support* of an itemset C is the number of instances in D containing it. The itemset C is said to be *frequent* if its support is greater than or equal to a predefined threshold, minsup. A rule, r, *covers* an instance I if the instance I satisfies the condition of the rule (or r is triggered by I). The coverage of a rule is defined as the number of instances that satisfy the condition of a rule. The accuracy of a rule is defined as the number of instances that are able to trigger the rule, where the labels of such instances must be equal to the label y belonging to the rule. A set $X \subseteq I$ with $k = |I|$ is called a k-itemset. The discovery process has two main tasks; namely, the discovery of all frequent itemsets and class-label assignment. To find all of the frequent itemsets, multiple passes have to be made.

Concretely, dataset D is first scanned to find the frequent 1-itemsets. For $k > 2$, candidate k-itemsets are generated as follows: given a set of frequent $(k-1)$-itemsets, I_{k-1}, the candidates for the next pass are created by making a join with I_{k-1}. An itemset $C_1 = \langle i_1, i_2, \ldots, i_{k-1} \rangle$ joins with another one $C_2 = \langle i'_2, i'_3, \ldots, i'_k \rangle$ and the candidate *cand* is produced if after dropping the first item of C_1 and the last item of C_2 the rest of the two itemsets are equal; that is, $i_2 = i'_2, \ldots, i_{i-1} = i'_{i-1}$. The candidate will be an extension of C_1; that is, the last item of C_2 is added to it ($cand = \langle i_1, i_2, \ldots, i_{k-1}, i'_k \rangle$). Its support is identified by scanning the transformed dataset D. If this itemset is frequent (that is, $support(cand) \geq$ minsup), then we proceed to the next stage for labeling. In principle, for each label y in Y, a candidate rule of the form $cand \rightarrow y$ is created. The accuracy of all the candidate rules would then be determined and *cand* would then be assigned with the label that gave the highest accuracy. However, candidate rules with an accuracy value that is less than a predefined threshold, minconf, would be eliminated.

With the final set of discovered rules, set R, we can diagnose

the risk level of a subject as follows: given the risk factors of a subject in the form of "itemset x," for every $r \in R$, we check whether x satisfies the condition of r. There might be more than one rule being triggered by x; hence, we sum the support and accuracy values and choose the highest total value. Based on the output of the rule-based prediction, doctors are then able to make a more informed decision as to whether a patient is likely to be suffering from heart disease. If necessary, a patient can undergo a second examination of HRV patterns under different daily activities. The heart state of the patient is recognized by HRV patterns, which are discovered by PHIAN in the second phase of the model.

2. Incremental Neural Network for Recognizing Heart Disease Based on Long-Term ECG Signals

This part is devoted to data preprocessing, which involves both feature representation and input encoding. First, the HRV patterns contained within a specified interval of time are analyzed to extract feature vectors. These feature vectors are then used to train the PHIAN model.

A. HRV Analysis

HRV is defined as the alteration of beat-to-beat RR intervals. Heart rate has a great influence on the activity of two branches of the automatic nervous system; namely, the sympathetic and parasympathetic systems. The balance between these systems is reflected through the spectral analysis of RR intervals. Two bands, a low-frequency (LF) band (0.04 Hz to 0.15 Hz) and a high-frequency (HF) band (0.15 Hz to 0.4 Hz), are found. It is believed that the sympathetic–parasympathetic balance is reflected by the ratio LF/HF. A Poincaré plot is proposed to analyze the changes in a patient's HRV and suggested as an efficient method for detecting patients at risk of heart disease with short-term ECG measurements [7]. In principle, for a certain time interval, a Poincaré plot is plotted using a sequence of RR intervals.

Figure 3 shows an example of HRV patterns belonging to patients having a low-level risk of heart disease and average heart rate of 53 Hz. The results in the upper-right corner represent cases where patients had a breathing frequency of 0.1 Hz, and the results in the lower-left corner represent cases where patients had a breathing frequency of 0.2 Hz.

The patterns of points are then converted into the form of an HRV encoding vector. This task is tackled by decomposing the space into a number of regular cells. All cells have the same size. Each cell corresponds to an element of the HRV encoding (input) vector. It is assigned a value of "0" or "1" depending on whether it contains a data point. This vector is then extended with some elements of the features extracted from accelerometer recordings.

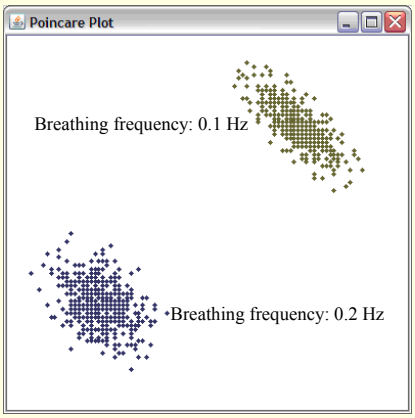

Fig. 3. Two patterns of points represented in a Poincaré plot for two cases of breathing frequency.

B. Network Learning Mechanism

The classification model used in our approach is named Pointcaré coding-based HRV patterns discovering incremental artificial neural network (PHIAN), which is trained to recognize the heart states along with the physical activities of the patients.

a. Network Structure

The neural network model is composed of three layers; namely, input, middle, and output (see Fig. 1). Incremental learning takes place in the second layer and is represented by a dynamic graph, G. This graph consists of a number of vertices (neurons) that are connected by edges. Therefore, the middle layer is denoted by $G(V, E)$.

The input layer connects to a neuron through an n-dimensional input weight vector, $\mathbf{w}_j^{\text{in}}$. Associated with each neuron is an activation function — here, a Gaussian RBF is selected in the hope that the training process results in fast convergence. The input weight vector $\mathbf{w}_j^{\text{in}}$ represents the center of a cluster of data (class center) in the input space and is the center of RBF as well. For each neuron, j, in the set V, the standard deviation, σ_j, of the Gaussian RBF is computed by (it is the mean distance of the edges that emanate from j)

$$\sigma_j = \frac{1}{N_j} \sum_{c \in N_j} \| \mathbf{w}_j^{\text{in}} - \mathbf{w}_c^{\text{in}} \| , \tag{3}$$

where N_j denotes the number of neighboring neurons of j and $\mathbf{w}_c^{\text{in}}$ is the input weight vector of a neighbor c. After training, classes are represented by decision regions in the output space whose positions are indicated by an m-dimensional weight vector, $\mathbf{w}_j^{\text{out}}$. Each neuron is also associated with a variable, Err_j.

This variable stores the local error caused by the neuron in classification.

b. Training Strategy

In principle, the training of the PHIAN model is the process of finding a topology for graph *G*. Graph *G* starts with two neurons connected by an edge. Their positions in the input space are represented by two random vectors, $\mathbf{w}_1$ and $\mathbf{w}_2$. Given a dataset *D* of samples, each sample is represented by a pair <**x**, **z**> in which $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ is the input vector and $\mathbf{z} = \{z_1, z_2, \ldots, z_m\}$ denotes the desired output vector. At each learning step, an input vector **x** is fed into the model. The neuron that is closest to the input vector **x** (best matching neuron), *b*, is found by the Euclidean similarity measure. The weight vector $\mathbf{w}_b^{\text{in}}$ of neuron *b* and its neighbors are then rewarded with some value so that they become closer to the sample input vector **x** in the input space.

Rules for updating errors and centers of neurons are defined as follows**:**

As the environment is not stationary, input data have a high temporary probability density. We train a model that is able to give a uniform distribution of local error. To this end, an error-modulated Kohonen rule along with a monotonically decreasing function *g*: $R_{\geq 0} \rightarrow [0, 1]$ is used. The error variable of *b* is updated by $\text{Err}_b = \gamma \times \text{Err}_b + (1-\gamma) \times \text{Err}(\mathbf{x})$, where the error Err(**x**) is caused by the input **x** and γ is a constant in the range [0, 1]. Let l_b be the learning rate of *b* and l_c be the learning rate of its neighbors. Neuron *b* and its neighboring nodes are rewarded in the sense that they are allowed to be closer to the input vector **x** by a distance of $\Delta\mathbf{w}$, which is computed by (4) and (5) below, respectively.

$$\Delta\mathbf{w}_b^{\text{in}} = l_{\text{b}} \times g\left(\frac{\overline{\text{Err}}}{\text{Err}_b}\right) \times \| \mathbf{x} - \mathbf{w}_b^{\text{in}} \|, \tag{4}$$

$$\forall c \in N_b : \Delta\mathbf{w}_c^{\text{in}} = l_c \times g\left(\frac{\text{Err}_c}{\text{Err}_b}\right) \times \| \mathbf{x} - \mathbf{w}_c^{\text{in}} \|, \tag{5}$$

where $\overline{\text{Err}} = \frac{1}{N_b} \sum_{c \in N_b} \text{Err}_c$ with $N_b = \{v \mid (b, v) \in E\}$.

Adapting the center vector $\mathbf{w}_b^{\text{in}}$ in this way implies that neuron *b* wins in the competition for the best matching node to an input vector **x** only when its error accumulation Err_b is higher than the average value of its neighboring neurons *c*, $\overline{\text{Err}}$.

To achieve separated classes in the output space, we need to adapt their positions as learning progresses. This procedure is performed as follows. Let $\mathbf{o} = \{o_1, o_2, \ldots, o_m\}$ be the actual output for input vector **x**. When the input vector **x** is presented to the network, it activates every Gaussian neuron *j* in *V* to some degree, computed by $f_j = \exp\left(-\frac{\| \mathbf{x} - \mathbf{w}_j^{\text{in}} \|}{2 \times \sigma_j^2}\right)$. These activations are then spread forward to node *k* in the output layer. We take the sum of the products of the activation values and connection weights $\left(w_{j,k}^{\text{out}}\right)$ coming from neuron *j* in the middle layer; that is, $I_k = \sum_j w_{j,k}^{\text{out}} \times f_j$. Thereby, the weight of the connection between neuron *j* and node *k* is updated as $w_{j,k}^{\text{out}} = w_{j,k}^{\text{out}} + l_{\text{o}} \times (z_k - o_k) \times I_k$ where l_{o} is the output learning rate.

Practically, there always exists some overlap between decision regions, yet the probability density of the overlapping region is often low compared to the probability density of the class centers. The removed nodes are those that do not have any neighbors. This operation is done when the number of learning steps is equal to an integer multiple (λ) of input vectors presented to the network. From this moment onwards, new neurons might be added to the network. To determine where to insert a new node into the network, firstly we have to find the neuron with the highest error. If the error of this neuron, named *q*, is greater than some insertion criteria, insTh, then a new neuron, *p*, located between *q* and its neighbor *f* with maximum error would be added. This insertion operation leads to 50% decline in the error accumulation for *q* and *f*. This is because the new neuron gets that error reduction as its initial error variable value. This reduction helps avoid another insertion at the same place as neuron *q*. At each adaptation step, all local error accumulations are multiplied by a constant, β, where $\beta \in [0, 1]$, to stress the importance of recently occurred errors.

In fact, the edges of the graph in the middle layer are used to determine the diameter of the Gaussian RBF. However, the locations of neurons are slightly moved at each adaptation step. Furthermore, the node insertion operation causes changes to the network topology. Therefore, neighborhood information in the network needs to be continuously updated. To address this, each edge in the graph is associated with an age variable. For an input vector **x**, the second-best matching neuron, *s*, is also identified beside the best matching neuron *b*. If there exists an edge between *b* and *s*, then its age variable is set to zero; otherwise a connection is created and initialized with zero. When a new edge is created, age variables of all of the edges that start with node *b* are increased by one. After updating the neuron centers, some edges may become invalid. They would then be deleted. A threshold, a_{max}, is used to determine the obsolete edges in this case. The training process is repeatedly performed until the model converges, which is determined by observing the mean squared error (MSE) of the neural network model.

III. Results and Analysis

In this section, we conduct experiments to evaluate the performance of the proposed framework.

1. Assessment of Rule-Based Heart Disease Diagnosis

A system is constructed to predict the risk levels of patients based on the rules extracted from their clinical information. The Cleveland dataset from the UCI repository is used in the prototype system [20]. It is divided into sets; namely, training and testing. Rules are extracted from the former, and the latter is used to test the prediction accuracy. Three levels of risk; namely, low, medium, and high are distinguished.

As explained in the previous section, the number of rules is influenced by two parameters — minimum support and minimum confidence. It thereby influences the efficiency of the system; for example, the amount of time spent matching rules when predicting and diagnostic accuracy. Therefore, the number of rules to be used must be decided before the rules are integrated into the knowledge base of the system. Two experiments were conducted. The first experiment is to find the most suitable parameter values to set as the default values of minimum support and minimum confidence. The second experiment is to test the accuracy of the rule-based prediction. In the first experiment, we run two types of tests by fixing minimum support and varying the minimum confidence; and vice versa. For each set of rules obtained from a pair of minsup and minconf, classification accuracy is assessed. We finally select the ones that give the highest accuracy. The following illustrates the results we obtained for the most suitable pair of minsup and minconf. In the first test, minconf is fixed, and we observe that the number of rules sharply decreases as the value of minsup increases (see Fig. 4).

In the second test, minsup is fixed, and we can observe that the number of rules decreases as the value of minconf increases (see Fig. 5).The rules that are discovered with the parameter values of minsup and minconf, 15 and 30, respectively, are integrated into the prototype system. The testing dataset is used to assess the prediction accuracy. We divided the training dataset into two groups of people (that is, a group of people at low risk of heart disease and a group of people at medium or high risk of heart disease) and evaluated the prediction accuracy for each group. The rule-based prediction accuracy is measured by the percentage of correctly classified people in each group. The results showed that for the group of people at low risk of heart disease, the prediction accuracy is 95%. However, the prediction accuracy is only about 75% for the other group. According to experts in cardiovascular disease, the inaccuracy in prediction may have occurred because the same symptoms can be shown in many other diseases; for example, irregular heart rhythms can be related to thyroid problems. To be certain of whether a subject has heart disease, it is crucial to monitor and examine the subject's ECG signals as they perform their daily activities.

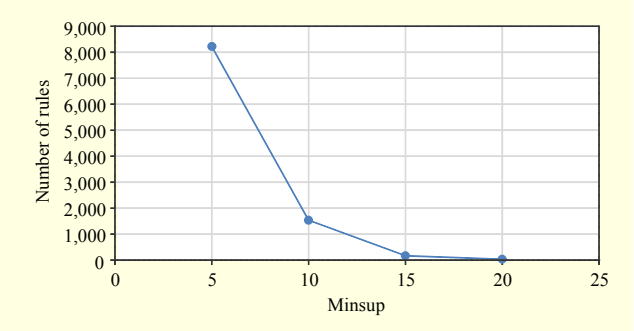

Fig. 4. Number of rules as a function of minsup (minconf = 30).

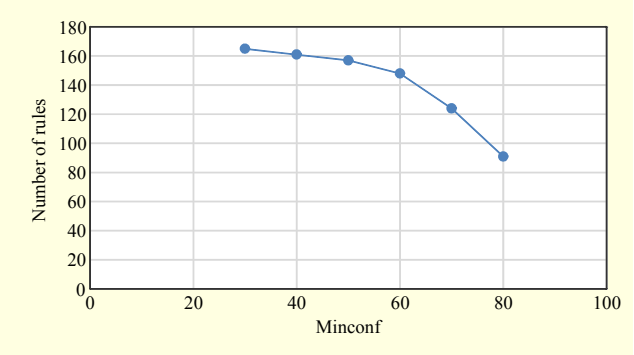

Fig. 5. Number of rules as a function of minconf (minsup = 15).

2. Assessment of Predicting Heart Disease Based on HRV Patterns

In the second phase, prediction evaluation is done for each group of individuals discovered in the first phase.

A. Settings for Validating Neural Network

The neural network model is assessed using the dataset built from the group of individuals aged between 46 years old and 50 years old. With regards to the daily activities of the subjects, three physical activities — resting, working, and exercising — are distinguished. Figure 6 shows an excerpt from a time-series signal streaming from an accelerometer sensor. One of two heart states, normal (N) and abnormal (A), is recognized for each of the subjects. Each measurement for an activity takes place for about four minutes. RR intervals are captured at every 3 ms during this period. The visual space of the scatter plot was partitioned into 784 regular cells.

To acquire data samples for constructing the classification model, several scenarios were set up. In a scenario, more than

ETRI Journal, Volume 37, Number 2, April 2015
http://dx.doi.org/10.4218/etrij.15.2314.0103

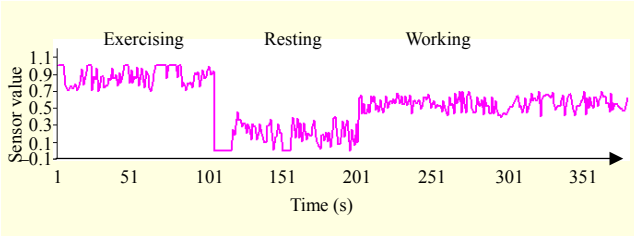

Fig. 6. Excerpt from a time-series signal from accelerometer sensor.

Table 2. Parameters for HRV data generation.

	Resting		Working		Exercising
Class label	N	A	N	A	N
Range of heart rate	50–56	60–65	65–73	126–135	141–142
Average heart rate	53	62.5	69	130	141–142
Heart rate standard deviation	1.6475	1.3693	2.211	2.494	141.5
Breathing frequency (Hz)	0.1–0.2		0.25		0.4
LF/HF ratio	0.5		1		4

one activity was performed and an activity could be repeated many times under the control of heart rate and breathing frequency. Two types of training sets were generated. The first one is denoted as $D(e)$, where e indicates the number of samples belonging to more than one class. Parameter e occupies about 2% of the total samples. The second one is denoted as D(rand), in which samples were generated randomly without controlling the degrees of overlap between classes. Data set $D(e)$ is collected from seven scenarios, whereas D(rand) is collected from eleven. Each scenario corresponding to an environment gives a subset D_i of data examples. Table 2 shows the values of the parameters corresponding to the three aforementioned activities.

To validate whether our model can continuously learn new knowledge, we tracked the percentage of classification error as the experiment progressed. Classification error is defined by (5) below. The test dataset should be built so that it contains data samples belonging to all of the classes.

$$\text{Generalization error} = \frac{\#\,\text{mistakenly classified examples}}{\#\,\text{total examples of test set}}. \quad (5)$$

To know how well decision regions represent the input probability distribution, we apply the MSE as a quantity measure. This measure indicates the classification quality obtained after training the model. MSE is computed by (6) below.

$$\text{MSE} = \frac{1}{M}\sum_{i}^{M}(o_i - z_i)^2, \quad (6)$$

in which M is the number of data samples in the training set, o_i is the output given by the model for the example i and z_i is the target value of the model for example i. The smaller the value of MSE, the better the classification quality. We exploit this measure as the termination condition of the training process; that is, when MSE reaches a threshold value of about 0.01, the training stops. Some parameters with default values used in the training process include best learning rate, l_b = 0, 1; learning rate of neighbor, l_c = 0.001, output learning rate, l_o = 0.1, constants β = 0.995 and γ = 0.8; λ = 30, age threshold, a_{max} = 50; and insertion threshold, insTh = 0.5. The experiments in [16] and [21] manifested that with these values the final model would result in the best result; therefore, we used them too.

B. Performance Assessment of PHIAN

Figure 7 displays the generalization error of PHIAN trained on D(rand) as learning progresses. It is observed that only a few classes appear in the environment (points 1 to 4), so the classification error is relatively high. However, as the environment changes, new classes may appear and some old classes still remain, so the generalization error sharply decreases. Then, the classification error becomes stable in the environment between points 4 and 5 — this is because some classes in the previous environments appear again. Learning continues by feeding the new samples and classes into the models until all classes are presented to the model. We observe that the classification error reaches zero at the end of the environment (point 6). However, after this, new samples belonging to more than one class begin to show up and the resulting confusion leads to an increase in classification error. As explained in the learning strategy, new nodes were inserted with the hope of minimizing the classification error (see Fig. 8).

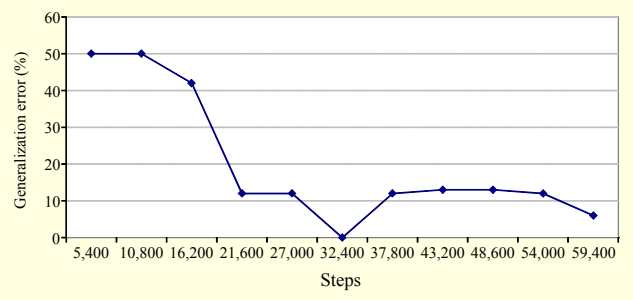

Fig. 7. Generalization error of PHIAN trained on D(rand) as environment changes.

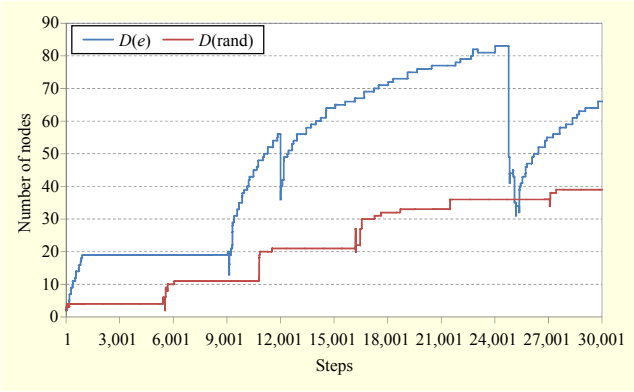

Fig. 8. Number of nodes for PHIAN trained on $D(e)$ and D(rand), respectively.

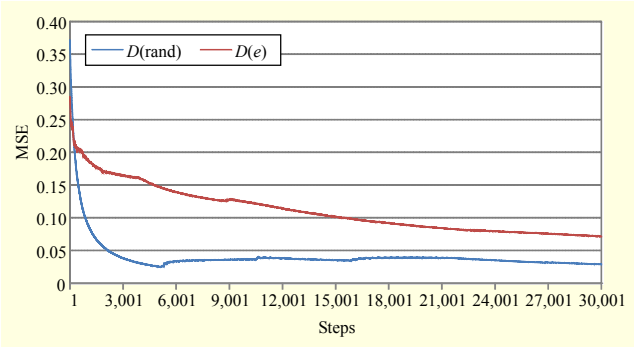

Fig. 9. MSE as a function of learning step.

When the environment changes from points 7 through 11, only the data samples from existing regions are fed into the model. New neurons are still inserted together with the operation of center adaptation. The learning process tries to adapt to the new environment, and this is repeated until no further learning is needed. Finally, the model becomes stable and gives the minimum classification error.

Figure 8 illustrates the variation in the number of nodes for PHIAN trained on $D(e)$ and D(rand), respectively. As there is a big overlap between classes in $D(e)$, the number of nodes given for PHIAN trained on $D(e)$ is greater than that for PHIAN trained on D(rand), though the size of D(rand) is much larger than that of $D(e)$. This is because the learning strategy is based on the idea that new neurons are added when there are signals coming from new regions. In the same environment, neuron insertion has to be stopped if it does not lead to a decrease in classification error.

Figure 9 displays the results obtained after the model is trained on the data sets $D(e)$ and D(rand) for two epochs. It is observed that MSE gradually declines in both cases. In other words, classes are well separated in the output space at the end of the training process. However, the result given by the data set D(rand) is better than that of $D(e)$ because the degree of overlap among classes in $D(e)$ is quite high. This also explains why the number of nodes for $D(e)$ is greater than that of D(rand) (see Fig. 8).

C. Comparing Efficiency of PHIAN with Existing Techniques

The effectiveness of our approach is evaluated in comparison with two well-known online learning techniques, SOM and GNG. Figure 10 compares generalization error as a function of training steps. To evaluate the effectiveness of the algorithms PHIAN, GNG, and SOM, we trained three neural network models on D(rand). Technically, GNG and PHIAN work similarly, so their classification accuracy is almost the same, except in some places where there is overlap between regions PHIAN works more effectively than GNG. Since SOM is incapable of preserving old patterns in a non-stationary environment, it cannot predict examples of old classes, which makes the classification error higher compared to when using the other two techniques.

To affirm the effectiveness and efficiency of the proposed model, we conducted a test to compare the network structure of PHIAN and GNG. Figure 11 shows that there is a big gap between the number of nodes given by PHIAN and GNG. The result of PHIAN indicates that the network structure learned under data set $D(e)$ with serious overlap is still simpler than that learned by GNG under data set D(rand). In brief, the classification accuracy of PHIAN is the same or even better in

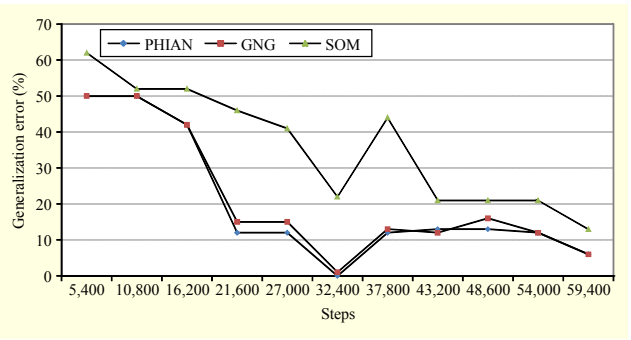

Fig. 10. Generalization errors of three methods.

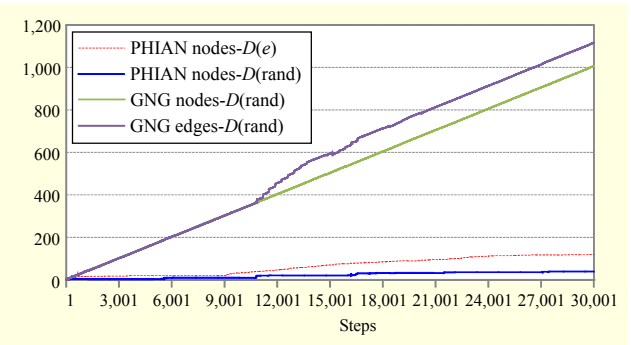

Fig. 11. Network structure of PHIAN compared with GNG.

ETRI Journal, Volume 37, Number 2, April 2015
http://dx.doi.org/10.4218/etrij.15.2314.0103

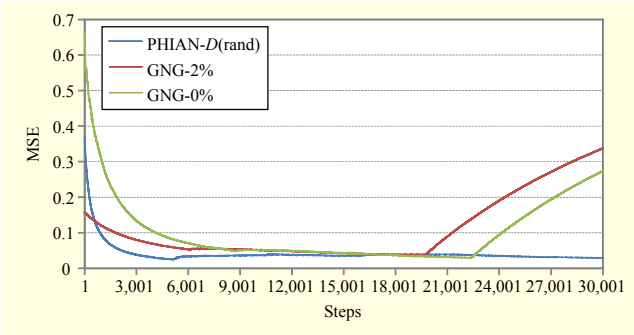

Fig. 12. Variations in MSE for PHIAN and GNG variants.

some cases than that of GNG, while its number of nodes is far fewer than that for GNG.

Figure 12 displays the variations in MSE for the three models. We observe that GNG works as well as PHIAN in the case of non-overlap only; however, owing to unlimited new-node allocation during the training process, overfitting occurred. On the contrary, our method inserted a new node only when the local error was truly high; otherwise the data sample was assigned to the closet neuron. In conclusion, the classes learned by PHIAN represent the input distribution better than those learned by GNG in all cases.

Our dual-phase framework helps improve the accuracy of heart disease diagnosis. Consequently, with the support location prediction technique in [24], this framework can be integrated in telemedicine systems to provide patients with cardiac care services anytime, anywhere.

IV. Conclusion

We proposed a dual-phase heart disease diagnostic framework. The risk level of a subject is firstly predicted by using *confident* frequent rules, which are extracted from risk factors. From our experimental results, we could see that such a rule-based method may lead to incorrect conclusions regarding a patient's heart disease status. This is because sometimes subjects cannot describe precisely what has happened to them and medical researchers cannot accurately characterize how disease modifies the normal functioning of the body. To be certain about the presence of heart disease, doctors need to examine the beat-to-beat temporal variations in a patient's heart by asking them to undertake various daily activities. To continuously discover HRV patterns, we applied the online artificial network PHIAN. With a dynamic network structure and incremental learning rule, new patterns can be learned while old ones are still preserved even though the environment changes.

The performance of the proposed approach was assessed in terms of classification error for both rule-based and HRV pattern–based classification. Compared to the predictive approach, using the learning algorithms of SOM and GNG, our framework is better with regard to classification accuracy and neural network structure complexity. It is a new effective approach that can be applied to a telemedicine system to help predict the likelihood of heart disease within a patient.

References

[1] J.Y. Chang and S.W. Nam, "Fast Random-Forest-Based Human Pose Estimation Using a Multi-scale and Cascade Approach," *ETRI J.*, vol. 35, no. 6, Dec. 2013, pp. 949–959.

[2] D. Jo et al., "Tracking and Interaction Based on Hybrid Sensing for Virtual Environments," *ETRI J.*, vol. 35, no. 2, Apr. 2013, pp. 356–359.

[3] S. Jeong, Y. Kim, and C. Youn, "Personalized Healthcare System for Chronic Disease Care in Cloud Environment," *ETRI J.*, vol. 36, no. 5, Oct. 2014, pp. 730–740.

[4] S.I. McFarlane et al. "Hypertension in the High-Cardiovascular-Risk Populations," *Int. J. Hypertension*, 2011.

[5] K.M Anderson et al., "An Updated Coronary Risk Profile: A Statement for Health Professionals," *Circulation J.*, Jan. 1991, pp. 356–361.

[6] N.A. Sundar, P.P. Latha, and M.R. Chandra, "Performance Analysis of Classification Data Mining Techniques over Heart Disease Database," *Int. J. Eng. Sci. Adv. Technol.*, vol. 2, no. 3, 2012, pp. 470–478.

[7] F. Azuaje et al., "A Neural Network Approach to Coronary Heart Disease Risk Assessment Based on Short-Term Measurement of RR Intervals," *Comput. Cardiology*, Lund, Sweden, Sept. 7–10, 1997, pp. 53–56.

[8] B. Mirkin, "*Clustering For Data Mining: A Data Recovery Approach*," New York, USA: Chapman and Hall/CRC, 2005.

[9] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," *Int. Conf. Very Large Databases*, 1994, pp. 487–499.

[10] S.-W. Lee and K. Mase, "Activity and Location Recognition Using Wearable Sensors," *IEEE Pervasive Comput.*, vol. 1, no. 3, 2002, pp. 24–32.

[11] R. Detran et al., "International Application of a New Probability Algorithm for the Diagnosis of Coronary Artery Disease," *American J. Cardiology*, vol. 64, no. 5, Aug. 1989, pp. 304–310.

[12] H. Yan et al., "A Multilayer Perceptron-Based Medical Decision Support System for Heart Disease Diagnosis," *Expert Syst. Appl.*, vol. 30, no. 2, Feb. 2006, pp. 272–281.

[13] A. Ingo, B. Jorg, and S. Gerald, "On-line Learning with Dynamic Cell Structures," *Int. Conf. Artif. Neural Netw.*, 1995, pp. 141–146.

[14] T. Kohonen, "*Self-Organizing Maps*," Berlin, Germany: Springer-Verlarg, 2001.

[15] B. Fritzke, "A Growing Neural Gas Network Learns Topologies," in *Adv. Neural Inf. Process. Syst. 7*, Cambridge, MA, USA: MIT Press, 1995, pp. 625–632.
[16] T.H.N. Vu and N. Park, "Heart Rate Variability Pattern Recognition in Ambulatory Environments," *IEEE Int. Conf. Comput. Ind. Eng.*, Awaji, Japan, July 25–28, 2010, pp. 1–6.
[17] S. Teerakanok et al., "Preserving User Anonymity in Context-Aware Location-Based Services: A Proposed Framework," *ETRI J.*, vol. 35, no. 3, June 2013, pp. 501–511.
[18] Shian-Ru Ke et al., "Human Action Recognition Based on 3D Human Modeling and Cyclic HMMs," *ETRI J.*, vol. 36, no. 4, Aug. 2014, pp. 662–672.
[19] T.H.N. Vu, J.W. Lee, and K.H. Ryu, "Spatiotemporal Pattern Mining Technique for Location-Based Service System," *ETRI J.*, vol. 30, no. 3, June 2008, pp. 421–431.
[20] Heart Disease Dataset, Machine Learning Repository. Accessed June 10, 2014. https://archive.ics.uci.edu/ml/datasets/
[21] T.H.N. Vu et al., "Online Discovery of Heart Rate Variability Patterns in Mobile Healthcare Services," *J. Syst. Softw.*, vol. 83, no. 10, Oct. 2010, pp. 1930–1940.

Yang Koo Lee received his BS degree in computer and information engineering from Cheongju University, Rep. of Korea, in 2002 and his MS and PhD degrees in computer science from Chungbuk National University, Cheongju, Rep. of Korea, in 2004 and 2010, respectively. He was a research student at the University of Aizu, Fukushima, Japan, in 2009. From 2010 to 2011, he was a post-doctoral fellow at Chungbuk National University, Rep. of Korea. Since 2011, he has been with the Electronics & Telecommunications Research Institute, Daejeon, Rep. of Korea, where he is now a senior researcher. His main research interests include location-based services, sensor networks, data mining, and human–computer interaction technology for virtual-reality applications.

Thi Hong Nhan Vu received her BS degree in information technology from the College of Technology, Vietnam National University, Hanoi, in 2001. She received her MS and PhD degrees in computer science from Chungbuk National University, Cheongju, Rep. of Korea, in 2004 and 2007, respectively. She worked for the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea in 2007. From 2009 to 2010, she was a postdoctoral researcher at Ohio University in Athens, USA. Since 2011, she has been with the Faculty of Information Technology, University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam, where she is now a lecturer. Her major research interests are applications of pervasive and ubiquitous computing technology for healthcare and wellness; data mining; artificial intelligence; and human–computer interaction.

Thanh Ha Le received his BS and MS degrees in information technology from the College of Technology, Vietnam National University, Hanoi, in 2005. He received his PhD degree in computer science at the Department of Electronics Engineering, Korea University, Seoul, Rep. of Korea. In 2010, he joined the Faculty of Information Technology, University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam, as a lecturer and researcher. His research interests are multimedia processing, coding satellite image processing, and computer vision. He is now undertaking research on forest fires using remote sensing approaches and highly efficient multi-view coding.