

Deep Autoencoder Networks Optimized with Genetic Algorithms for Efficient ECG Clustering

Tippaya Thinsungnoen, Kittisak Kerdprasop, and Nittaya Kerdprasop

Abstract—Deep analyses of electrocardiogram (ECG) signals can reveal hidden information that can be potentially useful for the accurate diagnosis of heart diseases. Time series data of ECGs are usually high dimensional and complex in their components. One of the key successes for this kind of learning is to learn from the representative data. In this research, we present Deep Autoencoder Networks (DANs) for efficient casting of time series representatives. To determine the appropriate DAN structure, we use genetic algorithms (GAs). ECG representatives are then clustered. The clustering results obtained from our proposed method are compared with those obtained using other time series representation techniques. This comparison is based on the grouping accuracy involving the correct data label and cluster purity. The experimental results show that we can cast for appropriate ECG representatives that yield better performance with regard to time series clustering with 30% improvement in grouping accuracy and 23% increase in the purity metric.

Index Terms—Time series representation, deep autoencoder networks, genetic algorithm.

I. INTRODUCTION

In the present era of digital devices, data in a time series format can be used in many applications in our daily lives, for example, stock prices, weather information, health measurements (e.g., blood pressure), and signals from electrocardiograms (ECGs) and electroencephalographs (EEGs) [1]. The main feature of a time series is its sequencing format from the data source that is continuously stored in a chronological manner over a period of time [1]-[3]. ECG signals are derived from the detection of electrical currents coming from the heart. The latent information in ECGs can be used to aid in the diagnosis of associated diseases or issuing an alarm before an oncoming disease. Therefore, if we can recognize abnormal ECGs early and accurately, it will be very useful in the medical field.

ECGs usually have large time series, high dimensions, and complex components. It is, therefore, a challenging area to implement machine learning. In order to deal with high dimensionality, researchers typically look for data

representatives. It is, however, difficult to find a good time series representation [3] because of the ordered characteristic inherent in such series. Many researchers have investigated time series representations. Representation methods such as Piecewise Aggregate Approximation [4], Adaptive Piecewise Constant Approximation [5], Symbolic Aggregate Approximation [6], Discrete Fourier Transform [7], and Wavelet Transform or Discrete Wavelet Transform [8], [9], have been proposed that can yield effective time series representations.

Recently, other techniques that are potentially effective in finding time series representations have been formulated. One such technique is Deep Autoencoder Networks (DANs), which apply deep learning using multiple connected network layers to transform and transmit signals between the layers [10], [11]. The aim of an autoencoder (AE) network is to model high-level data representation by automatically finding and integrating features to another level [10]-[12]. Restricted Boltzmann Machines (RBMs) are an example, which have been used in conjunction with an efficient technique called “AEs” for data representation. Such deep architecture is widely used in computer vision, speech recognition, natural language processing, bioinformatics, audio recognition, social networking filtering, and other time series applications [13], [18]-[21].

There are many proposals involving deep learning architecture, such as Deep Neural Networks, Deep Belief Networks [14], RBMs [15], Deep Boltzmann Machines [16], and DANs [17]. Here, DANs have outstanding features such as their capability of learning to represent high dimensional data. However, the design of these network models are limited by the fact that an inappropriate layout of the network may yield suboptimal results. Therefore, some researchers employ genetic algorithms (GAs) to find the best network structure [22]-[25].

In this paper, we propose a method for representing the time series data for higher efficiency in ECG clustering using DANs. This kind of network is based on RBMs and it is called an AE. In this research, we determine an appropriate network for DANs by using GAs to find the best network structure. The ECGs under consideration are then clustered using the Permutation Distribution Clustering (PDC) algorithm.

II. BACKGROUND

A. AEs

An AE is a feedforward neural network that is trained to transform the input data into another compressive format. General AE architecture comprises a hidden layer h that is

Manuscript received January 21, 2018; revised March 27, 2018. This work was supported in part by grants from Suranaree University of Technology and Nakhon Ratchasima Rajabhat University, Thailand.

T. Thinsungnoen is with the Faculty of Science and Technology, Nakhon Ratchasima Rajabhat University (NRRU), 340 Suranarai Avenue, Muang, Nakhon Ratchasima 30000, Thailand (corresponding author; tel.: +66819671907; e-mail: tippayasot@hotmail.com).

Kittisak Kerdprasop, and Nittaya Kerdprasop are with the School of Computer Engineering, SUT, Thailand (e-mail: kerdpras@sut.ac.th, nittaya@sut.ac.th).

fully connected [20], [26], [27] in such a way that it can effectively encode the input data for unsupervised learning. Usually, AE is widely used in data representation, especially for reducing the dimensions of data [26], [28]. The AE network consists of two sections: an encoder function $h = f(x)$ and a decoder function $r = g(h)$ that recovers the data from the code, that is, input reconstruction. The general AE architecture, which involves mapping from an input x to an output r (code h), can be shown as that in Fig. 1 [26].

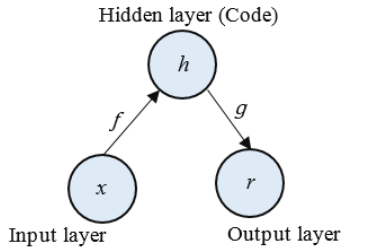


Fig. 1. The general autoencoder architecture.

B. Deep Learning

Deep learning, or deep structured learning or hierarchical learning, is a machine learning technique that attempts to model high-level abstractions in the data based on learning the representations of data. The concepts include processing among multiple layers such that each layer is derived from the previous layer [10], [26], [29].

C. RBMs

RBM's comprise a network of symmetrically coupled stochastic binary units with no hidden-to-hidden and no visible-to-visible connections [15], [16], [30], [31]. The RBMs are energy-based probability models that have a probability distribution defined from an energy function as follows [19]:

$$P(x, h) = \frac{e^{-\text{Energy}(x,h)}}{Z} \quad (1)$$

where x is a set of input units, and h corresponds to the hidden units introduced to increase the expressive power of the model. The partition function is defined for the normalization factor Z as follows [19]:

$$Z = \sum_{x,h} e^{-\text{Energy}(x,h)} \quad (2)$$

D. DANs

DANs are a powerful technique for data representation, especially for converting high dimensional data into low dimensional code. DANs work by training a multilayer neural network and fine-tuning the weights in the AE networks. This technique is an effective choice for assigning the beginning weights to adapt multilayer "encoder" networks to transform data into a low dimensional code, and a similar "decoder" network to recover the data from this code [17]. The DANs workflow has three main steps: Pretraining, Unrolling, and Fine-Tuning, as shown in Fig. 2.

The pretraining step is the weight initialization procedure for each layer of the AE network, where the processing is done by a RBM [17]. A connected setting (v, h) of the visible and hidden units has an energy [32] given by

$$E(v, h) = - \sum_{i \in \text{pixels}} b_i v_i - \sum_{j \in \text{features}} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \quad (3)$$

where v_i and h_i are the binary states of the pixel (target) i and feature j , respectively; b_i and b_j are the biases of i and j , respectively; and w_{ij} is the weight between i and j .

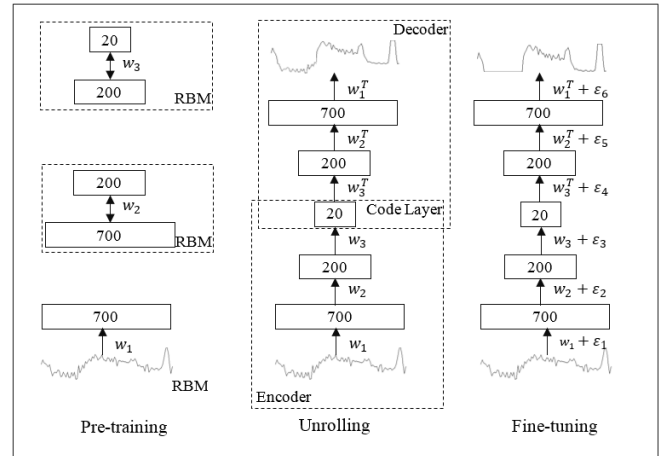


Fig. 2. The three main steps of Deep Autoencoder Networks.

The unrolling step involves the procedure for creating DANs. After pretraining multiple layers of the feature detectors, the model is unfolded to the encoder and decoder networks that have the same weights, as shown in Fig. 2.

The fine-tuning step involves global weight fine-tuning for optimal reconstructions using backpropagation through the whole AE. This is followed by replacing the stochastic activities by deterministic, real-valued probabilities [17]. The change in a weight depends on the learning rate (ϵ).

E. GAs

GAs are search algorithms designed from the biological evolution in a natural genetic system based on Charles Darwin's theory of evolution [24], [33]. GAs are widely known as stochastic sampling methods that can be used to determine optimum solutions in terms of special objective functions, such as multimodal, discontinuous, non-differentiable, etc. These algorithms initialize a population of potential solutions and implement their search for better solutions based on the "survival of the fittest" strategy [24]. The process of GAs consists of six fundamental steps [34], namely, (1) chromosome representation; (2) fitness evaluation; (3) genetic operators comprising selection, crossover, and mutation; (4) creation of the initial population; (5) termination criteria; and (6) replacement.

F. PDC

PDC is an agglomerative hierarchical approach to cluster time series data [35], [36]. A dissimilarity in time series is formalized as the squared Hellinger distance between the permutation distributions of the embedded time series. This distance is robust to outliers. This approach involves two main concepts regarding clustering time series [35], which can be explained as follows.

The first concept is identifying a time series representative using permutation distribution patterns. Given a time series $X = \{x(i)\}_{i=0}^T$, the time series is embedded into m -dimensions with a time delay t , ranging from 0 to T . The permutation

distribution of X' is obtained by counting the frequency of distinct observed ordinal patterns for the sub-sequence $x' \in X'$ with a total of $T' = T - (m - 1)t$. Let $\Pi(x)$ be the permutation such that the sub-sequence $x \in \mathbb{R}^m$; then, the permutation distribution of X' can be defined as follows:

$$p_\pi = \frac{\#\{x' \in X' \mid \Pi(x') = \pi\}}{T'} \quad (4)$$

The second concept is the dissimilarity measurement of the time series using the squared Hellinger distance between the permutation distributions of the embedded time series. Let $P = (p_1, p_2, \dots, p_n)$ and $Q = (q_1, q_2, \dots, q_n)$ be two permutation distributions; then, the squared Hellinger distance is

$$D(P, Q) = \frac{1}{\sqrt{2}} \|\sqrt{P} - \sqrt{Q}\|_2^2 \quad (5)$$

III. MATERIALS AND METHODS

A. Dataset Description

In this study, we used ECG signals taken from [37], [38]. These datasets consist of RR intervals that were recorded by one electrode during one heartbeat obtained from the Holter ECG tapes sampled at 128 Hz. The data were analyzed by domain experts as either normal or abnormal. Each ECG signal is a long sequence that contains 96 ventricular events of equal length. Chosen at random [39], from 200 data records, 133 records were identified as normal and 67 were identified as abnormal. Previews of normal and abnormal ECG time series are shown in Fig. 3. In Fig. 3(a), series S2, S5, S6, S9, and S10 are normal cases. The other five series (S1, S3, S7, S8 and S15) shown in Fig. 3(b) are examples of abnormal cases.

B. The Proposed Work

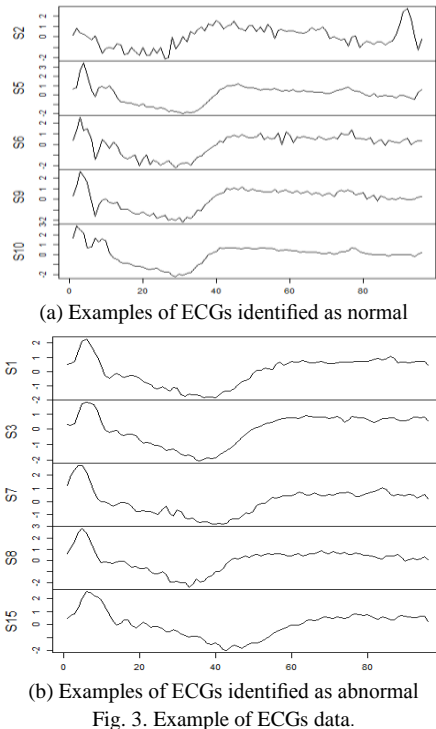


Fig. 3. Example of ECGs data.

The objective of this research is to use DANs optimized

with GAs searching for optimal networks that can transform raw time series into optimal time series, which can be eventually used for efficient ECG clustering. In this work, a DAN model comprised three hidden layers; according to the literature, such a structure can yield the best representation. The flowchart of the proposed work is given in Fig. 4. The overall process can be explained as follows.

An input process is reading the data and manually setting the parameters. The dataset X is the ECG signal with population size $P=20$, iteration $M=3$, and epoch $E=50$ for the learning DAN model. The initial population is generated at random with the chromosome encoding defined as *chromosome* (*hidden1*(500-1000), *hidden2*(100-500), *code layer*(1-30)).

The fitness evaluation is a purity value based on the DAN model that can produce the best representation.

Then, selection, crossover, and mutation are used to reproduce the next generation. The process is repeated until meeting the termination condition, followed by the exit loop.

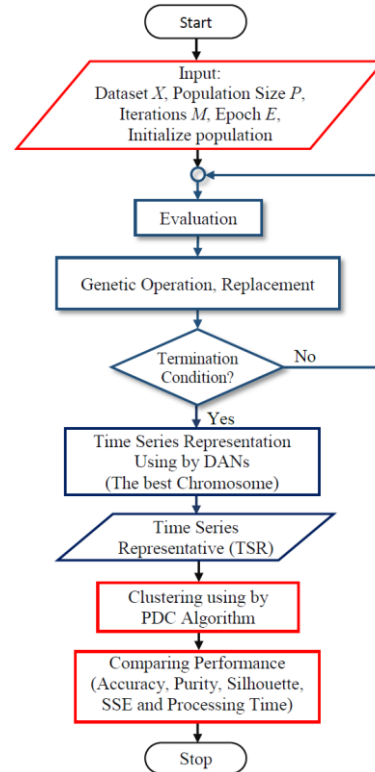


Fig. 4. The flowchart of our proposed method.

After genetic operation optimization, the most optimal DAN (the best chromosome) is used to produce a time series representation. Then, the time series representatives are clustered using the PDC algorithm. To evaluate the performance of the proposed method for time series representation (whether it is suitable for clustering or not), we compare the clustering results with those obtained from other representation techniques. Based on this proposed algorithm, we implement the program using the R language.

IV. EXPERIMENTAL RESULTS

This study focuses on DAN optimization using GAs for effectiveness of time series clustering. We, thus, present the experimental results based on our main concern as follows.

The best chromosome, which is the best solution for the 3-layer AE network with the structure 700-200-20, has a fitness value (purity) of 82.5%. The details of the network structures optimized by GAs are summarized in Table I.

TABLE I: PERFORMANCE OF THE DEEP AUTOENCODER NETWORKS OPTIMIZED WITH GENETIC ALGORITHM

#Model	Hidden1	Hidden2	Code Layer	Fitness
1	500	100	3	78.5
2	500	100	10	81.5
3	500	200	1	77.0
4	500	200	10	76.5
5	600	100	10	76.5
6	600	200	1	76.5
7	600	200	10	82.0
8	600	200	20	76.5
9	600	500	10	75.5
10	600	500	20	79.0
11	700	100	30	78.0
12	700	200	20	82.5
13	800	200	10	79.0
14	800	300	20	75.0
15	800	400	10	75.0
16	900	100	10	76.0
17	900	300	20	78.0
18	900	500	3	76.5
19	1000	200	10	80.5
20	1000	300	30	77.0

The time series representatives, which are generated from optimal DANs (called TSR-DANs), are shown in Fig. 5. Examples of ECG representatives for normal cases are shown in Fig. 5(a) and abnormal cases are shown in Fig. 5(b).

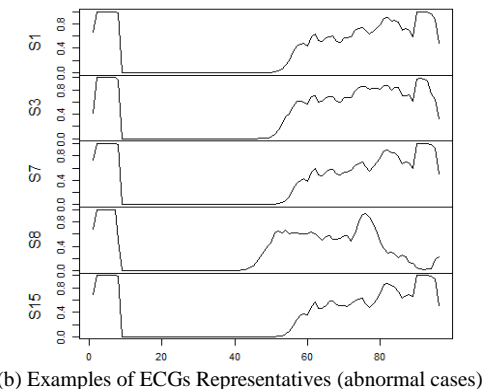
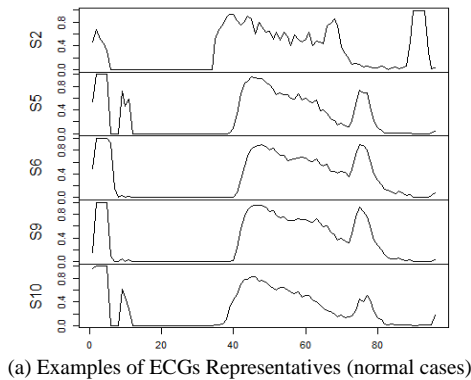


Fig. 5. An example shape of ECGs representative.

The clustering and comparing performance is the last step

in our experiment. We use the PDC algorithm in the R package for clustering the raw data, PAA representation, SAX representation, and TSR-DAN representation (our proposed method for time series representation). The results show that our method provides the best representation of ECGs for clustering using the PDC algorithm. The clustering performance evaluation based on the comparison of the accuracy, purity, and silhouette coefficient criteria is shown in Fig. 6. Based on the accuracy and purity metrics, our TSR-DANs reveal the best results. When considering the silhouette coefficient metric, our method comes in the second place. For comparing the increase in performance (Fig. 7), our TSR-DANs are the best. However, the processing time of our method is higher than the others.

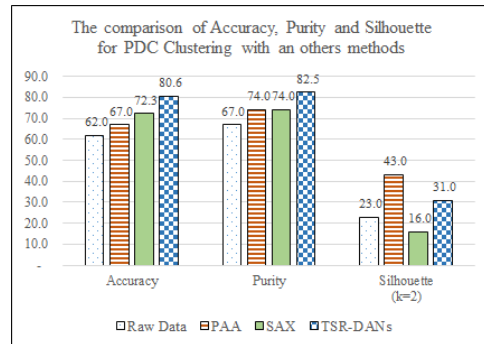


Fig. 6. The comparison of accuracy, purity, and silhouette coefficient.

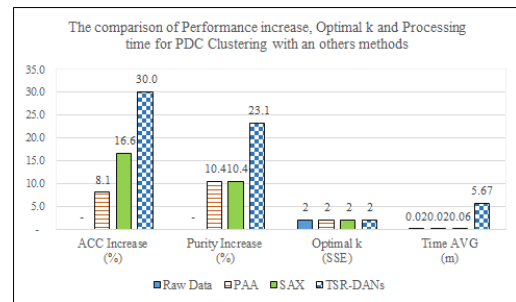


Fig. 7. The comparison of performance increase, optimal k (based on SSE), and processing time.

V. CONCLUSION AND FUTURE WORK

In this research, we study DANs that optimize their network structure using GAs for ECG clustering. The findings of this research are as follows. DANs are the most effective technique to represent ECG data, especially when their structures have been properly configured. Further, we show that GA is appropriate for obtaining the global optimization solution for DANs. In this study, the methodology combining DANs and GAs yield DANs that are optimized beyond expectations and can represent ECGs suitable for clustering. Moreover, the proposed modeling and optimization methods show great potential in complicated applications.

The high processing time of TSR-DANs is still a challenging problem. Reducing time complexity while maintaining its power to produce effective time series representation is, therefore, our future research direction.

REFERENCES

[1] S. Aghabozorgi, A. S. Shirkhorshidi and T. Y. Wah, "Time-series clustering – a decade review," *Information Systems*, vol. 53, pp. 16-38, April 2015.

- [2] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*, Springer, 2016.
- [3] M. Långkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling" *Pattern Recognition Letters*, vol. 42, pp. 11-24, 2014.
- [4] E. Keogh, M. Pazzani, K. Chakrabarti, and S. Mehrotra, "A simple dimensionality reduction technique for fast similarity search in large time series databases," *Knowledge Information Systems*, vol. 1805, no. 1, pp. 122-133, 2000.
- [5] E. Keogh, K. Chakrabarti, and M. Pazzani, "Locally adaptive dimensionality reduction for indexing large time series databases," in *Proc. ACM SIGMOD Conference on Management of Data*, Santa Barbara, pp. 151-162, May 21-24, 2001.
- [6] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: A novel symbolic representation of time series," *Data Mining. Knowledge Discovery.*, vol. 15, issue 2, pp. 107-144, 2007.
- [7] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient similarity search in sequence databases," *Found. Data Organ. Algorithms*, vol. 46, pp. 69-84, 1993.
- [8] A. Graps, "An introduction to wavelets," *IEEE Computational Science and Engineering*, vol. 2, no. 2, pp. 50-61, 1995.
- [9] C. S. Burrus, R. A. Gopinath, and H. Guo, "Introduction to wavelets and wavelet transforms: A primer," Prentice Hall, 1997.
- [10] Y. Bengio, Y. LeCun, and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436-444, 2015.
- [11] L. Deng and D. Yu, "Deep learning: Methods and applications. Foundations and trends," *Signal Processing*, vol. 7, issues 3-4, pp. 197-387, 2014.
- [12] B. A. Olshausen, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607-609, 1996.
- [13] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. 2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3642-3649.
- [14] G. E. Hinton, S. Osindero and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, issues 7, pp. 1527-1554, 2006.
- [15] G. E. Hinton, "Boltzmann machine," *Scholarpedia*, vol. 2, no. 5, p. 1668, 2007.
- [16] R. Salakhutdinov and G. E. Hinton, "Deep boltzmann machines," *AISTATS*, vol. 1, p. 3, 2009.
- [17] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, issues 5786, pp. 504-507, 2006.
- [18] T. Thinsungnoen, K. Kerdprasop, and N. Kerdprasop, "A deep learning of time series for efficient analysis," *International Journal of Future Computer and Communication*, vol. 6, no. 3, p. 123, 2017.
- [19] V. J. R. Ripoll, A. Wojdel, E. Romero, P. Ramos, and J. Brugada, "ECG assessment based on neural networks with pretraining," *Applied Soft Computing*, vol. 49, pp. 399-406, 2016.
- [20] N. Gianniotis, S. D. Kügler, P. Tiño, and K. L. Polsterer, "Model-coupled autoencoder for time series visualization," *Neurocomputing*, vol. 192, pp. 139-146, 2016.
- [21] E. Busseti, I. Osband, and S. Wong, "Deep learning for time series modeling," Technical Report, Stanford University, 2012.
- [22] K. Suksut, K. Kerdprasop, and N. Kerdprasop, "Support vector machine with restarting genetic algorithm for classifying imbalanced data," *International Journal of Future Computer and Communication*, vol. 6, no. 3, p. 92, 2017.
- [23] R. Chuentawat, N. Kerdprasop, and K. Kerdprasop, "The forecast of PM10 pollutant by using a hybrid model," *International Journal of Future Computer and Communication*, vol. 6, no. 3, p. 128, 2017.
- [24] C. Shen, L. Wang, and Q. Li, "Optimization of injection molding process parameters using combination of artificial neural network and genetic algorithm method," *Journal of Materials Processing Technology*, vol. 183, no. 2, pp. 412-418, 2007.
- [25] M. J. Demongeot, M. J. Mazoyer, M. P. Peretto, and M. D. Whitley, *Neural Network Synthesis using Cellular Encoding and the Genetic Algorithm*, 1994.
- [26] L. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
- [27] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1-127, 2009.
- [28] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," arXiv preprint arXiv:1312.6114, 2013.
- [29] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436-444, 2015.
- [30] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McClelland, Eds., vol. 1, pp. 194-281, MIT Press, 1986.
- [31] J. Z. Mohammed and M. Wagner, "Data mining and analysis: fundamental concepts and algorithms," Cambridge University Press, USA, 2014.
- [32] J. J. Hopfield. "Neural networks and physical systems with emergent collective computational abilities," in *Proc. the National Academy of Sciences*, vol. 79, no. 8, pp. 2554-2558, 1982.
- [33] J. H. Holland, "Adaptation in natural and artificial system," Michigan: University of Michigan Press, 1975.
- [34] R. H. Christopher, A. J. Jeffery, and G. K. Michael. *A Genetic Algorithm for Function Optimization: A MATLAB Implementation*. [Online]. Available: www.ie.ncsu.edu/mirage
- [35] M. A. Brandmaier. "PDC: An R package for complexity-based clustering of time series," *Journal of Statistical Software*, vol. 67, issue 5, pp. 1-23, October 2015.
- [36] A. Brandmaier, "Permutation distribution clustering and structural equation model trees," Dissertation, Saarland University, Saarbrücken, 2012.
- [37] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. (2015). The UCR time series classification archive. [Online]. Available: URL www.cs.ucr.edu/~eamonn/time_series_data/
- [38] R. T. Olszewski, "Generalized feature extraction for structural pattern recognition in time-series data," Report No. CMU-CS-01-108, School of Computer Science, Carnegie-Mellon University, Pittsburgh, U.S.A., 2001.
- [39] E. Keogh and M. Pazzani. "An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback," in *Proc. the 4th International Conference of Knowledge Discovery and Data Mining*, 1998, pp. 239-241.



T. Thinsungnoen is a lecturer with the Faculty of Science and Technology, Nakhon Ratchasima Rajabhat University. Currently, she is a doctoral student in the School of Computer Engineering, Suranaree University of Technology (SUT), Thailand. She received her bachelor degree in computer science from Nakhon Ratchasima Rajabhat Institute, Thailand, in 1999, master degree in computer engineering from SUT in 2007. Her current research of interest includes data mining, deep boltzmann machines, deep learning of time series data.



K. Kerdprasop is an associate professor and chair of the School of Computer Engineering, SUT. He received his bachelor degree in Mathematics from Srinakarinwirot University, Thailand, in 1986, master degree in computer science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in computer science from Nova Southeastern University, U.S.A., in 1999. His current research includes Data mining, artificial intelligence, computational statistics.



N. Kerdprasop is an associate professor in Computer Engineering School, SUT. She received her bachelor degree in radiation techniques from Mahidol University, Thailand, in 1985, master degree in computer science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in computer science from Nova Southeastern University, U.S.A, in 1999. Her research of interest includes data mining, artificial intelligence, and intelligent databases.