Applications of Machine Learning to Resource Management in Cloud Computing

Chenn-Jung Huang, Yu-Wu Wang, Chih-Tai Guan, Heng-Ming Chen, and Jui-Jiun Jian

***Abstract*—There are various significant issues in resource allocation, such as maximum computing performance and the green computing, attract researchers' attentions recently. Therefore, how to accomplish tasks with the lowest cost has become an important issue when the resource on the earth is getting less. The goal of this research is to design a sub-optimal resource allocation system in cloud computing environment. A prediction mechanism is realized by using Support Vector Regressions (SVRs) to estimate the response time in the next measurement period, and the resources are redistributed based on the current status of all virtual machine installed in physical machines. Notably, a resource dispatch mechanism using genetic algorithms (GAs) is proposed in this study to determine the reallocation of resources. The experimental results show that the proposed scheme achieves an effective configuration via reaching the agreement between the utilization of resources within physical machine monitored by physical machine monitor and Service Level Agreements (SLA) between virtual machines operator and cloud services provider. In addition, our proposed mechanism can fully utilize hardware resources and maintain desirable performance in the cloud environment.**

***Index Term*—Cloud computing, support vector regression, genetic algorithms, resource allocation, prediction.**

I. Introduction

With the transformation of the architecture from mainframes to client-server models, information technology services have been rapidly developed and increased. The cloud computing concept addresses a new paradigm shift in Internet-based services that provides highly scalable distributed computing platforms where computational resources are offered as a service, and hence consumers do not need to understand how it works and can easily access various services via the Internet.

With advanced network technology, high speed bandwidth provision, and the popularity of the smart phones, people can upload their work on the Internet immediately at anytime and anywhere as long as we have network connection. It is also the most charming part of cloud computing, which has dramatically changed the procedure to process traditional information.

Originally, cloud computing is not Business to Business (B2B) model but Business to Consumer (B2C) model, which means users can access and operate the software and other services via the Internet. With the rapid development of cloud computing, more and more industries took cloud computing used in business model into account. That is because cloud computing has fully utilized the virtualization technology that can give users high flexibility and extension, and consumers can increase the storage space or computation ability only if they can consult with the Internet service providers (ISPs). Hence, it is an important issue of resource management for the cloud system to allocate the virtual machines (VMs) and make the cloud computing system automatically manage the entity resources through the strategies of creating or collecting VMs and migration among the physical machines (PHs). ISPs will formally sign a contract, called Service Level Agreement (SLA), with the consumers to determine the price of each level of service, the contracted content which sometimes refers to some performances metrics of resources, such as the performance of CPU, the capacity of memory, and the response time. ISPs can evaluate their applications to determine different prices according these performance metrics of resources, and they no longer need to find out the locations to store their application services. Instead, application services are now managed by the cloud computing system to achieve the efficiency through the distributed computation in VMs.

Resource management is always an important issue that is mostly applied in some arrangement of working tasks, which have its own different usages in problem solving and decision making. Different strategies of selections will bring different cost and efficiency. Therefore, how to find a sub-optimal resource allocation strategy, especially for the limited resource, aiming at each kind of goal is a demanding work. This research work focused on the application of Evolutionary Algorithms (EA) in the area of cloud computing. After considering the individual representation of some algorithms, the gene coding representation of GA is more suitable in this work. Besides, numerous researchers have proposed Genetic Algorithms (GAs) to deal with the optimization problems, and the Schema theorem for GA proposed by Holland [1] illustrated that GA is a robust searching approach. We thus chose GA as the resource allocation algorithm in this work, and we will choose other EAs, such as the hybrid meta-heuristic algorithm, called Evolution Strategy (ES), proposed by Nissen *et al.* [2], as compared methods in future work. Following the principles first presented by Charles Darwin of survival of the fittest, Genetic Algorithm (GA) is not only an adaptive heuristic search algorithm assumed on the evolutionary ideas of natural selection and genetic, but also represents an intelligent exploitation of a random search in a vase search space. The main advantage of GA compared with other heuristic methods is that it only needs a fitness function to evaluate the quality of different solutions and there is no

Manuscript received January 12, 2013; revised March 26, 2013.

Chenn-Jung Huang, Yu-Wu Wang are with the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan (e-mail: cjhuang@mail.ndhu.edu.tw).

Chih-Tai Guan, Heng-Ming Chen, Jui-Jiun Jian are with the Department of Electrical Engineering, National Dong Hwa University, Hualien, Taiwan.

DOI: 10.7763/IJMO.2013.V3.256

necessary to offer a particular algorithm to solve a given problem. Feng *et al.* [3] presented a method to select members from different departments to resolve manpower distribution problem by using an improved non-dominated sorting genetic algorithm (INSGA). However, GAs often consumes much time to find the global optimum [4], [5], some studies utilized Hybrid Genetic Algorithm (HGA) [6] to improve the computation time of GAs.

Support Vector Regression (SVR) [7] is a kind of supervised machine learning method that recognizes patterns and analyze data, mostly used for classification and regression analysis. The major difference between the SVR and traditional regression techniques is that the SVR employs the structural risk minimization (SRM) approach, rather than the empirical risk minimization (ERM) approach typically adopted in statistical learning. The SRM attempts to minimize an upper threshold on the generalization rather than minimize the training error, and is expected to perform better than the traditional ERM approach. Furthermore, the SVR is a convex optimization, which guarantees that the local minimization is the unique minimization. In the recent literature, numerous researchers have adopted SVR to deal with the classification and regression problems. Wu *et al.* [8] used SVR to predict the time of driving according to the speed of vehicles, traffic flow, and weather conditions. Users can handle the overall schedule more efficiently with this method. In addition, Liu *et al.* [9] compared three regression approaches, including SVR, Back-propagation Neural Network, and Partial Least Squares, to predict the Cold Modulus of Silicon Ceramic, and the results showed that SVR obtained better performance in root mean square error than the other two methods.

This work aims to consider the efficiency and optimization of resource allocation in cloud computing environment. An application service prediction module built with SVR is used to estimate the response time in the next measurement period. When new VMs are demanded, the system will consider the individual loading to proceed the prediction with SVR. If the result needs to change the allocation strategies, the system will utilize GA at this stage to try its best to achieve the global deployment of the resources effectively, including creating or collecting VMs, to make sure that the proposed system can satisfy the SLA requested by the customers.

The remainder of this paper is organized as follows. Section II addresses the algorithm for the proposed resource allocation mechanism. The simulation results are given in Section III. Conclusion is made in Section IV.

II. Resource Management Algorithm

Fig. 1 illustrates the architecture of the resource management scheme for cloud computing proposed in this work. An application service resource pool is used to collect all applications provided by Internet service providers (ISPs), and an application monitor is used to record the overall utilization of system resources. Furthermore, a physical machine resource pool is used to provide resource, CPU or Memory, for the hosts; and two look-up tables, including the remaining resource table and the resource utilization rate table, are used to assist in determining the strategies of increasing or decreasing the number of virtual machines (VMs) requested by each application service. Notably, an application service prediction module built with Support Vector Regressions (SVRs) is used to estimate the response time in the next measurement period. Meanwhile, a global resource allocation module applied with Genetic Algorithm (GA) is utilized to redistribute the resources to the clients, including creating or collecting VMs, to make sure that the proposed system can satisfy the Service Level Agreement (SLA) requested by the customers.

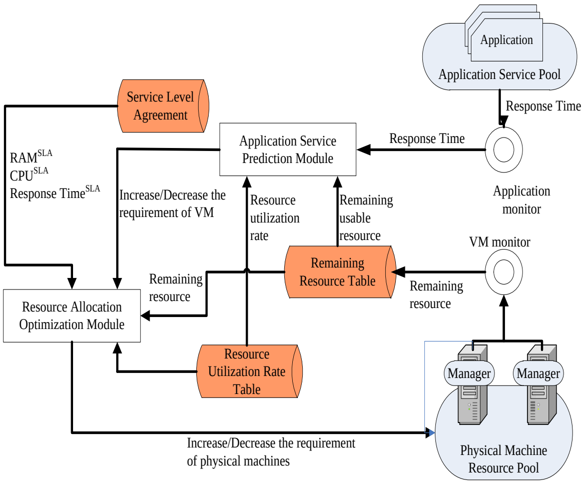
Application
Application Service Pool
Response Time
Service Level Agreement
Application Service Prediction Module
Response Time
RAM^SLA
CPU^SLA
Response Time^SLA
Increase/Decrease the requirement of VM
Resource utilization rate
Remaining usable resource
Application monitor
VM monitor
Remaining resource
Remaining Resource Table
Remaining resource
Resource Allocation Optimization Module
Resource Utilization Rate Table
Manager
Manager
Increase/Decrease the requirement of physical machines
Physical Machine Resource Pool

Fig. 1. The architecture of the proposed system.

A. *Application Service Prediction Module*

This module mainly estimates the requirements of the resource in the applications to remind the system of creating or collecting the VMs in accordance with the real Internet situation. In addition, a well-known time series predictor, namely SVR, is embedded in this module to predict the response time in the next measurement period with the assistance of two lookup tables, including the remaining usable resource table, which records all related usable resources in the applications, and the resource utilization rate table,which stores the utilization rate of each VM in the AP. With the above-mentioned methods, this module will determine whether the VMs should be increased or decreased for the request from the application services. More descriptions about SVR approach will be addressed in the next sub-section.

1) *Support Vector Regression (SVR)*

To solve a nonlinear regression or functional approximation problem, the SVR nonlinearly maps the input space into a high-dimensional feature space using an appropriate kernel representation, such as polynomials and radial basis functions with Gaussian kernels. This approach is utilized to build a linear regression hyperplane in the feature space, which is nonlinear in the original input space. The parameters can then be derived by solving a quadratic programming problem with linear equality and inequality constraints [10].

A training data set $D=\{(\mathbf{x}_i, y_i)\in\Re^n\times\Re, i=1,\ldots,l\}$ comprising l pair training data $(x_i, y_i), i=1,\ldots l$, is given. The input x_i terms are n-dimensional vectors, and the system response y_i terms are continuous values. The SVR attempts to approximate the following function using data set D:

$$f(\mathbf{x},\mathbf{w}) = \sum_{i=1}^{N} w_i \cdot \varphi_i(\mathbf{x}) + b, \tag{1}$$

where b denotes the bias term, and the w_i terms represent the subjects of learning. Furthermore, a mapping $z=\Phi(x)$ is selected in advance to map input vectors x into a higher-dimensional feature space F, which is spanned by a set of fixed functions $\varphi_i(x)$.

By defining a linear loss function with the following ε-insensitivity zone:

$$|y_i - f(\mathbf{x}_i,\mathbf{w})|_\varepsilon = \begin{cases} 0 & \text{if } |y_i - f(\mathbf{x}_i,\mathbf{w})| \le \varepsilon \\ |y_i - f(\mathbf{x}_i,\mathbf{w})| - \varepsilon & \text{otherwise} \end{cases} \tag{2}$$

Schölkopf *et al.* [11] developed a modification of original Vapnik's SVR algorithm, called ν-SVR, and claimed that it can automatically minimize the radius ε. Lagrange multiplier methods can be employed to demonstrate that the constrained optimization problem.

The best nonlinear regression hyperfunction is then represented as:

$$f(\mathbf{x}) = \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) \cdot k(\mathbf{x}_i,\mathbf{x}) + b, \tag{3}$$

where b denotes the optimal bias.

B. Resource Reallocation Module

As shown in Fig. 1, this module collects the information from the Application Service Prediction module and two lookup tables, along with SLAs requested by the customer, to determine the number of VMs requested by each application service. GA is applied in this work to deal with the optimization problem, and the fitness function is designed in accordance with the actual cloud computing environment.

1) Genetic algorithm (GA)

We first transfer the VMs which are established in the physical machine into a binary code as the initial population, called chromosomes. Each element in the chromosome is either 0 or 1, and the higher fitness value will be kept to generate the next generation during the procedure of recombination and mutation. The new generation will run the same steps as their parents did until the stop criteria are satisfied. In addition, we do not need to set fixed time interval to activate GA because the system will proceed the adjustment strategies according to the real-time demand of VMs. Once a new request of VM arrives, the system will run GA to adjust the overall allocation of the resources.

2) Initial population

Before finding the best solution with GA, a set of initial population is generated in random, and the chromosomes in matrix format, with n VMs, and m physical machines, is shown as Fig. 2. A horizontal row stands for the location that each VM is allocated in which physical machines, and a vertical column represents the distribution of each VM belonging to the specific physical machine. Notably, the sum of each element on the column should be less than or equal to the maximum resource which a physical machine can provide for the customers.

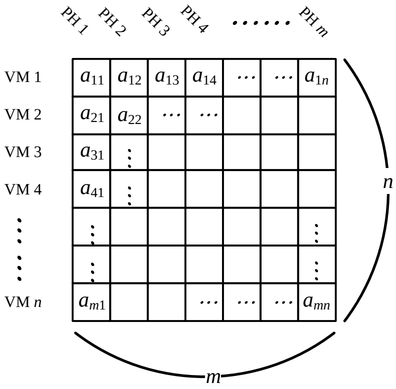

Fig. 2. The chromosomesin matrix format.

Notably, If VM i was not established in PH j, the entry of the matrix, a_{ij} is set to 0. Meanwhile, before GA is utilized in the initialization, the system will run SVR to evaluate the number of resource utilization according to the SLA of each process. The system has proceeded the allocation of VMs for each individual demand in the initialization step, and the operation of GA is to finely adjust the overall allocation to achieve better performance.

3) Fitness function

We first evaluate the resources of the physical machines that initially can provide the resource distribution for all virtual machines. Two related evaluation metrics are defined as follows,

$$\hat{u} = \sqrt{\frac{1}{m}\sum_{j=1}^{m} u_j^2 - \left(\frac{1}{m}\sum_{j=1}^{m} u_j\right)^2} \tag{4}$$

$$\hat{v} = \sqrt{\frac{1}{m}\sum_{j=1}^{m} v_j^2 - \left(\frac{1}{m}\sum_{j=1}^{m} v_j\right)^2} \tag{5}$$

where $\hat{u}$ represents the standard deviation of each u_j for the supply side of the processor, and $\hat{v}$ stands for the standard deviation of each v_j for the supply side of the memory.

We then define a matching score c as follows:

$$c = -f\left((a_p - u_q)/\hat{u}\right) - f\left((b_p - v_q)/\hat{v}\right) \tag{6}$$

where $f(x)$ represents a penalty function, a_p and b_p respectively represents the demand of processors and memories that VM p requests, and u_q and v_q stands for the processors and memories that PH p supplies, respectively.

For a better understanding how GA works in this work, we present an illustrative example. We assume that there are four physical machines, including PH 1 (CPU 4GHz, Memory 8G), PH 2 (CPU 6GHz, Memory 10G), PH 3 (CPU 6GHz, Memory 8G), and PH 4 (CPU 4GHz, Memory 8G), and five virtual machines, including VM 1 (CPU 2GHz, Memory 4G), VM 2 (CPU 1.5GHz, Memory 2G), VM 3 (CPU 1GHz、Memory 2G), VM 4 (CPU 3GHz、Memory 8G), and VM 5 (CPU 2GHz、Memory 2G). According to this setting, we can initialize the population as shown in Fig. 3. In Parent 1, the first column means that VM 1 and VM 3 are established in PH 1, the second column means that VM 2 are established in PH 2, and so on.

	PH 1	PH 2	PH 3	PH 4
VM 1	1	0	0	0
VM 2	0	1	0	0
VM 3	1	0	0	0
VM 4	0	0	1	0
VM 5	0	0	0	1

Parent 1

	PH 1	PH 2	PH 3	PH 4
VM 1	0	0	0	1
VM 2	1	0	0	0
VM 3	0	0	0	1
VM 4	0	0	1	0
VM 5	1	0	0	0

Parent 2

Fig. 3. An example of the population initialization.

Next, we encode the chromosomes of the two parents into binary strings. From up to down and left to right, the Parent 1 and Parent 2 can be represented as 10100010000001000001 and 01001000000001010100. While $\beta = 10$ and $\lambda = 0.15$, the fitness value can thus be 76.59 and 89.84.

4) Selection, reproduction, crossover, and mutation

According to Darwin's theory of evolution, better individuals get higher chance to survive and create offspring. In this work, we adopt roulette wheel selection and one-point recombination as the recombination method, respectively, and the recombination rate is 100%. To make sure that the sum of each column corresponds to the total VMs that can be allocated by a physical machine, we restrict each recombination occurring only in the same column. Notably, we chose one-point recombination, which has lower overhead, as our recombination type because of the real-time demand of VMs in this work. We would take other recombination types into consideration without scarifying the overall performance in future work. The mutation rate is set as the reciprocal of the size of the initial population in this work. A bit from the element in matrix is chosen in random to perform the NOT operation on it.If we randomly choose a_{21} as the mutation element, the bit will be transferred from 1 to 0. Besides, to make sure the sum in the same row is equal to 1, if the chosen bit is 0 initially, then it will be changed to 1, and the other bits in the same row will be transferred to 0 after the mutation operation.

5) Termination conditions

The population size is set to 20 in our experiment. The purpose of setting the termination conditions is to reduce the computing time and to avoid unnecessary evolution. In general, the process is repeated until one of the given termination conditions has been reached. First, a fixed number, saying one hundred, of generations reached. Second, the penalty function of 80% parents reaches zero.

III. Simulation Results

We first observed the difference between the non-optimized approach and the optimized approach proposed in this work. Six physical machines, which were equipped with 8GB RAM and 2 TB hard disk, were used, and a simulation software, installed in Windows XP platform with Intel Core 2 Quad 3.2GHz and 4GB RAM, CloudSim, was adopted in this work as the simulator of cloud infrastructures. The maximum of the number of VMs was set up to 100. To reflect the characteristics of the real world, different complexities of service applications were designed to verify the feasibility and effectiveness of our proposed work.

Fig. 4 shows the accumulated counts of completed service applications in comparison with optimized and non-optimized approaches. Notably, our proposed SVR-GA mechanism adopted SVR technique as a predictor that can moderately assign VMs according to current network utilizations. Additionally, GA is applied to find out an optimized set of resource allocation strategies. The system that utilized GA without executing the prediction via SVR is denoted by NO SVR-GA. NO SVR-GA used random allocation in the initialization step of GA without allocating VMs in advance. The difference between SVR-GA and NO SVR-GA is that NO SVR-GA cannot determine if the distribution strategies of VMs need adjusting, and also cannot proceed local pre-allocation for individual demands in the initialization. Therefore, our approach can complete more service applications while compared with the non-optimized method.

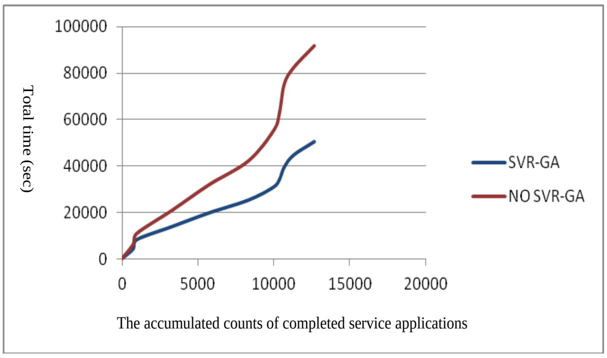

Fig. 4. The comparison of the number of completed service applications.

The statistical analysis of the response time comparison for the three strategies is illustrated in Table I. The data collected from Fig. 4 were analyzed by one-way repeated measure ANOVA method. At the 95% confidence level, it can be observed that there is significant difference among three strategies, and the proposed SVR-GA method has the best response time performance.

IV. Conclusion

In this work, a resource allocation optimization system in cloud computing environment is proposed. According to our experimental findings, the proposed approach, which considers application service prediction module built with SVR can estimate and decrease the response time in the next measurement period more accurately than other two representative cloud resource allocation strategies in the recent literature.

TABLE I: PAIRWISE COMPARISONS

(I) Algorithm	(J) Algorithm	Mean Difference (I-J)	Std. Error	Sig.(a)
SVR-GA	DAIaS	-1.616(*)	.121	.000
	ERM	-2.709(*)	.246	.000
DAIaS	SVR-GA	1.616(*)	.121	.000
	ERM	-1.094(*)	.164	.000
ERM	SVR-GA	2.709(*)	.246	.000
	DAIaS	1.094(*)	.164	.000

Based on estimated marginal means

* The mean difference is significant at the .05 level.

a. Adjustment for multiple conparisons: Least Significant Difference (equivalent to no adjustments)

This work mainly examines the performance of EA, such as GA, applied in cloud computing. In this present stage, we only evaluated two resources, including CPU and RAM, and the experimental results showed the effectiveness of applying GA in the resource allocation of cloud computing. In future work, we will proceed more work focused on more resources, such as Internet and the access of Auxiliary Memory to make our work more practical in real life applications.

In addition, the resource optimization module applied with GA can effectively adjust the resource allocation strategies and accomplish more applications in limited time compared with the non-optimized approach. In the future work, we plan to modify the algorithms to decrease the calculation time in terms of the prediction process to fasten the GA’s convergence speed.

ACKNOWLEDGMENT

The authors would like to thank the National Science Council of the Republic of China, Taiwan (Contract numbers NSC 99-2511-S-259-008-MY3 and 101-2622-E-259-001-CC3)for financially supporting this research.

REFERENCES

[1] J. H. Holland, *Adaptation in natural and artificial systems*, Ann Arbor, MI: Univ. Michigan Press, pp. 89-97, 1975.

[2] V. Nissen, M. G ünther, and R. Schumann, “Integrated generation of working time models and staff schedules in workforce management,” in *Proc. the 2011 international conference on Applications of evolutionary computation*, vol. 2, pp. 491-500, 2011.

[3] B. Feng, Z. Z. Jiang, Z. P. Fan, and N. Fu, “A method for member selection of cross-functional teams using the individual and collaborative performances,” *European Journal of Operational Research*, vol. 203, no. 3, pp. 652–661, June 2010.

[4] K. A. D. Jong, “Genetic algorithms: A 10 year perspective,” in *Perspectives on Adaptation in Natural and Artificial Systems*, L. Booker, S. Forrest, M. Mitchell, and R. Riolo, Eds.: Oxford University Press, 2005.

[5] P. Preux and E. G. Talbi, “Towards hybrid evolutionary algorithms,” *International* Transactions *in Operational Research*, vol. 6, pp. 557–570, 1999.

[6] A. Quintero and S. Pierre, “On the design of large-scale UMTS mobile networks using hybrid genetic algorithms,” *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2498-2508, July 2008.

[7] H. Drucker, J. C. Chris, Burges, L. Kaufman, A. Smola, and V. Vapnik, “Support vector regression machines,” *Advances in Neural Information Processing Systems 9*, NIPS 1996, pp. 155–161, 1997.

[8] C. H. Wu, J. M. Ho, and D. T. Lee., “Travel-Time prediction with support vector regression,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276-281, 2004.

[9] X. Liu, W. C. Lu, S. G. Jin, Y. W. Li, and N. Y. Chen, “Support vector regression applied to materials optimization of sialon ceramics,” *Chemometrics and Intelligent Laboratory Systems*, vol. 82, pp. 8–14, 2006.

[10] C. Cortes and V. Vapnik, *Support-Vector networks, machine learning*, 1995.

[11] B. Schölkopf, A. Smola, R. Williamson, and P. L. Bartlett, “New support vector algorithms,” *Neural Computation*, vol. 12, pp. 1207-1245, 2000.

Chenn-Jung Huang received the B. S. degree in electrical engineering from National Taiwan University, Taiwan and the M. S. degree in computer science from University of Southern California, Los Angeles, in 1984 and 1987. He received the Ph. D degree in electrical engineering from National Sun Yat-Sen University, Taiwan, in 2000. He is currently a Professor in the Department of Computer Science& Information Engineering, National Dong Hwa University, Taiwan. His research interests include computer communication networks, data mining, and diagnosis agent for e-learning.

Yu-Wu Wang is pursuing adoctoral degree at the Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan. Hisresearch interests include computer communication networks, data mining and applications of machine learning techniques.

Chih-Tai Guan is pursuing a doctoral degree at the Department of Electrical Engineering, National Dong Hwa University, Taiwan. His research interests include computer communication networks, data mining and applications of machine learning techniques.

Heng-Ming Chen is pursuing adoctoral degree at the Department of Electrical Engineering, National Dong Hwa University, Taiwan. Hisresearch interests include computer communicationnetworks, data mining and applications of machine learning techniques.

Jui-Jiun Jian is pursuing a Master’s degree at the Institute of Electronics Engineering, National Taiwan University, Taiwan. His research interests include computer communication networks, data mining and applications of machine learning techniques.