A SEMI-SUPERVISED LEARNING APPROACH TO FORECAST CPU USAGES UNDER PEAK LOAD IN AN ENTERPRISE ENVIRONMENT

Nitin Khosla[1] and Dharmendra Sharma[2]

[1]Assistant Director – Performance Engineering, Dept. of Home Affairs, Austrlia
[2]Professor – Computer Science, University of Canberra, Austrlia

ABSTRACT

The aim of a semi-supervised neural net learning approach in this paper is to apply and improve the supervised classifiers and to develop a model to predict CPU usages under unpredictable peak load (under stress conditions) in a large enterprise applications environment with several hundred applications hosted and with large number of concurrent users. This method forecasts the likelihood of extreme use of CPU because of a burst in web traffic mainly due to web-traffic from large number of concurrent users. This model predicts the CPU utilization under extreme load (stress) conditions. Large number of applications run simultaneously in a real time system in an enterprise large IT system. This model extracts features by analysing the work-load patterns of the user demand which are mainly hidden in the data related to key transactions of core IT applications. This method creates synthetic workload profiles by simulating synthetic concurrent users, then executes the key scenarios in a test environment and use our model to predict the excessive CPU utilization under peak load (stress) conditions. We have used Expectation Maximization method with different dimensionality and regularization, attempting to extract and analyse the parameters that improves the likelihood of the model by maximizing and after marginalizing out the unknown labels. With the outcome of this research, risk mitigation strategies were implemented at very short duration of time (3 to 4 hours) compared to one week taken in the current practice. Workload demand prediction with semi-supervised learning has tremendous potential tin capacity planning to optimize and manage IT infrastructure at a lower risk.

KEYWORDS

Semi-supervised learning, Performance Engineering, Stress testing, Neural Nets, Machine learning applications.

1. INTRODUCTION

With the new emerging IT technologies, usages data centre applications in cloud have grown tremendously in the past decade to cater high user expectations. It is observed at many instances the web-traffic or number of hits increases exponentially to a particular IT applications within a very short span of time (called as internet traffic burst). As a results the CPU utilization of the system increases drastically and has adverse impact of the performance of the IT systems and it slows down the enterprise application system [6][14].

At many instances, the IT system crashes because the IT system cannot sustain the excessive load under the peak load (stress) conditions. Sometimes the critical applications, providing services to the public, e.g. air tickets booking, emergency hospital services, custom clearances at airports,

David C. Wyld et al. (Eds): ITCSE, NLCA, ICAIT, CAIML, ICDIPV, CRYPIS, WiMo - 2020
pp. 145-154, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.100513

etc. halt suddenly. These systems crash random and many times it happens due to unpredictable high load or high volume of internet traffic. This results in adversely impacting the productivity and the system performance degrades. In large enterprise organizations, it is found that many times the system alerts are not observed and practically it is not feasible to take any remedial actions e.g. load balancing, etc. Some key transactions become irresponsive and the IT systems are unable to process transactions requests because of extremely high transaction rate which peaks randomly. Managing the keys applications to run 24/7 at a high efficiency level is always constant challenge between productivity, functionality and resource management [8]. Sometimes it is observed that very little or no memory is available for the critical applications to run it leads to a system crash. When transactions are being generated through internet traffic in a wide area distributed network where the network latency and bandwidth are key factors impacting the performance of applications, the scenario become even more complex [4][6].

Main objective of this research paper is to develop and demonstrate the use of a semi-supervised neural net approach to predict the usages of CPU utilization under unpredictable high volume of internet traffic under peak load conditions. To achieve this the work load patterns of the system are observed and analysed for a long period of time (one / two years). Then critical work-load profiles are extracted, which are hidden in the data generated by the key transactions of the crucial applications. Profile data is collected to observe the CPU utilization under peak load conditions (extremely high volume of web traffic) using data mining techniques.

2. RESEARCH QUESTION

Patters of CPU utilization at different time periods (during last one year) were studied and analysed by collecting data from profile points which were configured at different instances in the system. Load profiles were plotted and analysed to identify patterns. The CPU utilization and work load variations were used to develop test scenarios for the validation tests. These tests were conducted in the test environment. This helped us to identify the issues related in estimating a peak load in a test environment. We used this information to forecast the likelihood of this peak load in real world (production) environment. Using semi-supervised neural nets model we have developed a forecasting model to predict the CPU performance under peak load (stress conditions) in an enterprise environment.

2.1. Complex Integrated Environment

Public service departments of big size incorporates different types of system architectures which includes some old applications (legacy) and some developed recently e.g. smart mobile applications, video and face recognition in a cloud computing set-up, etc. We collected the experimental data from a large and complex integrated environment with more than 300 servers where many of them were distributed across multiple geographical locations (countries). Validation were performed in a test environment (called as pre-production environment), which represents a subset of the whole enterprise set-up containing and contains all applications with the most recent releases (builds) but with limited data set. This test environment was also used to represent a set-up with all applications of the department which are distributed in more than 52 overseas posts across the world.

2.2. IT Performance Issues

Computer applications are generally developed upon business specifications and are demand driven. The business specifications are mainly dependent upon the user requirements which keep changing over a period of time. There are some critical limitations when we evaluate and measure the performance of IT applications or performance of key transactions in the current practices, such as -

- Reliability Issues: System behaviour predictions e.g. response time, performance, etc., under high volume of traffic are not reliable and consistent

- Robustness Issues: Lack of a robust practical approach which can provide useful results in short time frames. It is mainly due to the unpredictable and dynamic web traffic

- Risk Based Approach: IT performance testing (Load and Stress) are mainly done on the critical (high priority) transactions or on the high-risk areas only because testing each and every scenario or their combinations is extremely time consuming and costly. So, the performance tests are designed and performed on -

    - Key transactions (high risk) which has critical impact
    - Important functions which could impact people, important services or have financial implications

3. FEATURE EXTRACTION

We collected raw data of key transactions from their data logs / files which were created at fixed periodic intervals thought a day over one year period. Profile points stored data continuously on pre-defined time intervals. These profile points were configured at different layers in the IT infrastructure. Different types of transactional data representing key transactions was captured e.g. transaction time responses, CPU utilization, memory used, bandwidth utilization, etc. and this was used for analysis, training and validation purposes.

Performance testing experiments (load and stress) for validation were performed in an IT test environment which represented a production like environment (representing a real work scenario). This test environment was configured and integrated with other systems in such a way that it simulated the real-world transaction behaviour. Monitoring of the identified transactions were done using the profile points which gathered the response-time data during the server-response paths (server to client and client to server). Analysis of data, identification of work-load patterns helped to improve our predictive model to forecast critical peaks considering the dynamic nature and variability of the load patterns [13].

3.1. Identifying Work Load Patterns

Workload patterns are dynamic and last for very short time span. Some patterns are different from the normal behaviour of a CPU. Many workload patterns are repeated at periodic instances due to some internal processes. We created a virtual traffic in a test environment to generate these type of workload patterns. We have captured transactions and relevant data for last one year with the help of profile data points. These profile (data capturing) points were configured at different threads, nodes and layers of the applications in the integrated test environment. We studied these patterns and analysed the CPU behaviour and patterns.

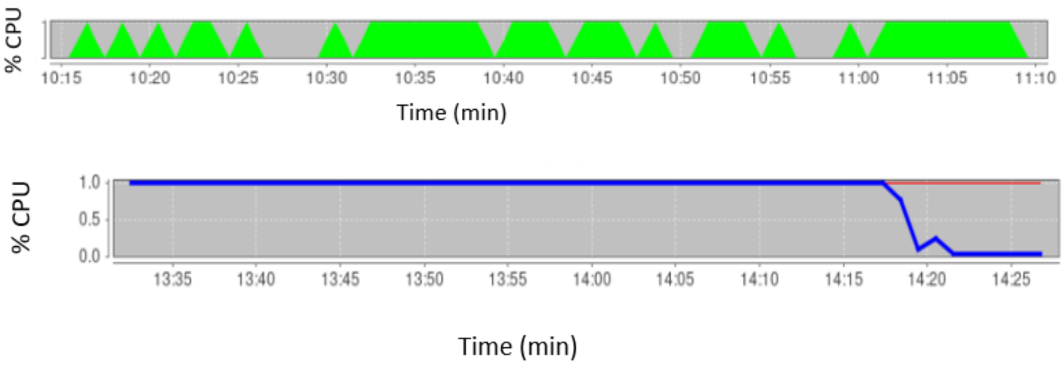

Figure 1. CPU workload pattern (% CPU Utilization), approx. 2500 hits per minute.

The above Figure 1 shows a CPU work-load pattern which follows a cyclic sequence. While in the second graph, the % CPU utilization drops suddenly about 12% from 95%. The key transactions response times were very high when the CPU usages were 95% and the system responded slowly during the peak load spikes. We captured the data during the peak intervals where we observed a typical pattern e.g. a higher CPU utilization for a longer duration of time, clearly shows an abnormal behaviour of CPU utilization. We have also collected some data related to memory, disk usages, database hits, network bandwidth, etc. during the peak CPU utilization periods and did some analytics to find insights from these patterns for predictive modelling. Hierarchical dependence and the impact of secondary transactions are out-of-scope and will be investigated as an extension to current work.

It was noticed that the cumulative CPU usages generally follows a cyclic behaviour for some transactions. These patterns can be represented by a time series consisting of a cyclic component.

4. Semi Supervised Learning Model

We used a labelled based semi-supervised learning approach to train our model and used labelled data initially along with some amount of unlabelled data [12]. There are some advantages associated with this research work such as -

a) We can optimise efficiency in terms of time and accuracy by predicting results which could provide alerts to avoid failures

b) A scalable predictive approach

c) A model simulating analogies of work-load patterns based upon data sets captured from different profile points

Assumptions: To develop a practical implementation of the semi-supervised learning approach to work, we have assumed some assumptions e.g. when two distinct points d_1, d_2 are close enough, then there might be respective outputs y_1, y_2. These assumptions helped to develop a practical model for a known number of training data sets to predict a set of infinitely number of test-cases which are mainly unseen or unpredictable [11].

We have also used some labelled data points such as - effort, time, tools and resources. In view of the potential implementation of the outcome of this research work, the semi-supervised learning along with forced-training method [3][7] has provided some useful outcomes because it is based upon -

i) Assumptions of forced regularization can reduce the training time

ii) Learning of data set with both labelled and unlabelled data

We can take the feature vector x (1 D) as represented by $d \times 1$ and it represents the % CPU utilization of the system under test simulating the production environment. Let x is represented by j×d matrix which is a feature matrix of the labelled data samples. Let x_u be the matrix of Dimension $U \times d$ of unlabelled data samples. Let w denotes the weight vector of our classifier and y be the L×1 vector with labels encoded between the range of {0, 1} showing the features of % CPU usages. The loss function in our classifier is defined as –

$$L_s(w) = \sum_{i=i}^{j} \left(x_i^T w - y_i\right)^2 + \mu \, \|w\|^2 \qquad (1)$$

Where µ is the weight decay L2 regularization. This is used to improve the performance of the model for unseen data.

When we minimize of the above objective function, the weight is given by -

$$w = (X^T X + \mu I)^{-1} X^T Y \qquad (2)$$

An updated object is labelled with a threshold of ½-

$$c_w(w) = \begin{cases} 1, & x^T w > \frac{1}{2} \\ 0, & otherwise \end{cases} \qquad (3)$$

A variable u is introduced in the objective function (eq. 1) and this includes the unlabelled objects. Therefore, the updated objective function can be defined as –

$$L_u(w,u) = \left\| X_c w - \binom{y}{u} \right\|^2 + \mu \|w\|^2 \qquad (4)$$

Where X_c is the concatenation of X and X_u

Once we take the gradient (slope) of the function (eq. 4) and minimize labelling, each of the label representing the % CPU utilization can be projected within {0, 1}.

5. Experimental Set up and Validation Methodology

We designed and implemented the following experiment set-up to execute of experiments in the test environment –

i) Virtual User Generator: to simulate critical end-user business processes or transactions

ii) Controller: to manage, control and monitor the execution of tests with specific ramping up and ramping down slopes

iii) Load Generators: configured on servers to generate virtual user load. It simulates work-load patterns with large number of virtual users generating web-traffic hits simulating work-load patterns like web-traffic bursts

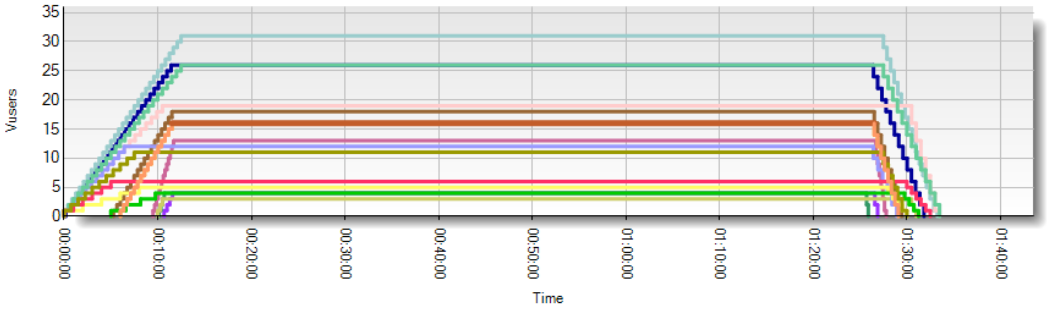

Figure 2. Load profile (ramp up and ramp down slopes) with virtual users

The above Figure 2 shows a work load profile of a group of virtual users (under peak load conditions) with different slopes of ramp up and ramp down times. This simulates real work user's type scenario. This set up was used for validation of our results in the test environment.
Limitations: Different virtual users have different ramping up slopes. It is assumed that these virtual users represent real time users but the actual ramp-up could have slightly different gradient and randomness.

Our validation methodology and experiments incorporated simulated work-load patterns showing burst in traffic at pre-defined intervals in the complex enterprise test environment. Then we collected respective transactional data. The test environment contained a sub-set of full production data which represents large data associated with the integrated applications in real word environment. Over 215 real applications, fully functional, were installed and configured in the test environment representing the real applications environment. This process included –

i) Data collection, features extraction, analysis of workload demand patterns

ii) Generate synthetic workloads patterns in the test environment

iii) Execute stress tests in the test environment with large number of virtual users just as a real

iv) world scenario

v) Confirmation of results by gathering data from different profile points configured at application threads, nodes and layers

vi) Train the model using labelled based semi-supervised learning approach (deep learning paradigm with Expected Maximization) [7],

vii) Forecast the likelihood of excessive CPU usages due to burst in the internet traffic [4][6].

6. Predicting Trends

To forecast a trend in the identified load patterns we have worked out the aggregate demand difference of each occurrence of the pattern from the original workload and compared them. We have used the modified exponential smoothing (ETS) algorithm with ETS point approximation where point-predicts are equal to the medians of the predict distributions [12].

Figure 3(a) shows the results of a semi-supervised neural network model (using EM. This is used to predict the % CPU usages under burst of internet traffic (web based) [9][10]. This model is now part of the monitoring process to continuous evaluate the demand patterns, as shown in Figure 3(b). This model provides information to system architects to set up alarms to take remedial actions e.g. re-allocation of IT resources for efficiency and to avoid a system crash or failure.

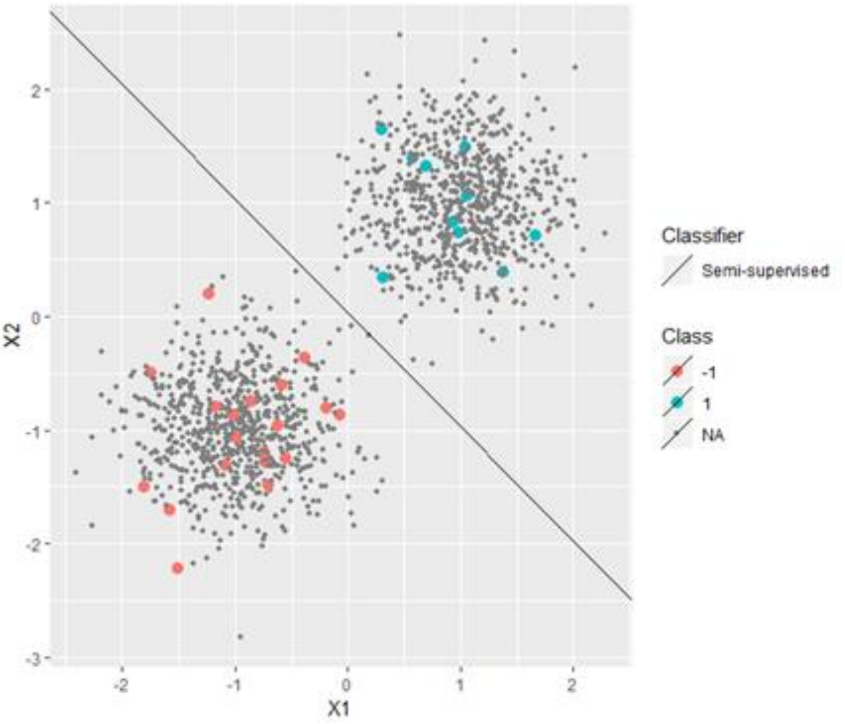

Figure 3 (a) Semi-supervised learning classifier (1 year data set)

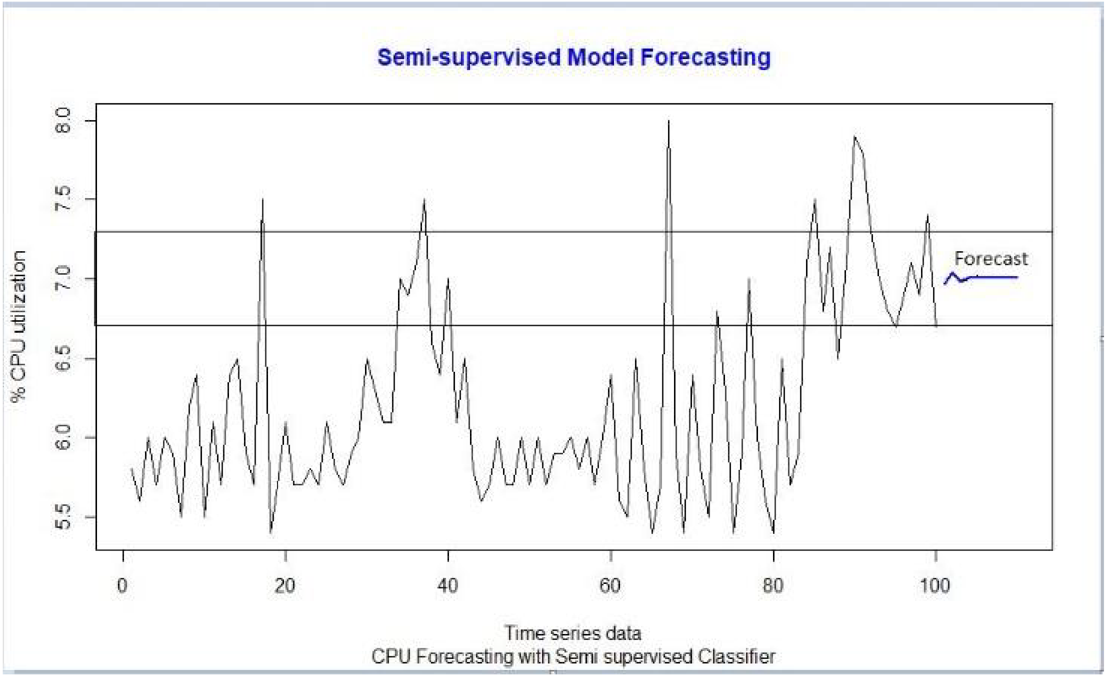

Figure 3(b) % CPU usages with peak load of internet traffic

Table 1 shows a comparison of % error loss with some relevant classifiers where our modified semi supervised model (EM based) has shown minimum loss.

Table 1: Comparison of % Mean Squared Loss with two data sets of peak load

Learning Algorithms	**Data set730**	**Dataset1460**
Least Square Classifier	13.05464	12.65698
EM Least Square Classifier	12.19138	11.79293
EM Semi supervised	11.93027	11.74969

7. CONCLUSION

We have designed and implemented a novel practical approach to predict % CPU utilization under the circumstances of unpredictable burst in web based in a complex and highly integrated environment (test or pre-production) where over 230 IT applications were live. Thousands of virtual users were used to generate a dynamic user-load under stress conditions. Our integrated enterprise environment had a distributed system with more than 300 servers serving more than 500 clients concurrently. Using our updated semi-supervised neural network approach (EM), the proposed methodology predicts and identifies the sharp increase in % CPU utilization in a complex enterprise IT infrastructure. Data analytics enabled the system architects and IT system capacity planners to distribute the load appropriately at different servers. The outcome of this research has mitigated the risk of potential failure and improved the system performance and outcomes. The mitigation strategies were implemented at very short duration of time (3 to 4 hours) compared to about 1 - 2 weeks taken in the current practice. Validation of our results were done in an integrated test environment and alerts generated as soon as the CPU utilization of the combined server’s crosses 75% threshold critical limit. This validated that our proposed methodology to predict excessive % CPU utilization worked effectively. In addition, we have found that this research is beneficial for our department in planning future IT capacity, optimizing IT resources in the complex IT enterprise IT environment. As a result of this research, the load balancing was appropriately balanced and database server capacities were shared to handle unexpected web traffic.

8. FUTURE WORK

As further work, we are working on developing a hierarchical semi-supervised learning model to extract patterns while considering the impact of different parameters e.g. memory, hard -disk failures, network latency, etc. and are trying to design an efficient semi-supervised learning approach for predictive modelling.

REFERENCES

[1] Avital Oliver, Augustus Odena, Colin Raffel, Ekin D Cubuk, et, 2018. 6th International Conference on Learning Representations, ICLR, “Realistic Evaluation of Semi-supervised Learning Algorithms”, Vancouver, BC, Canada.

[2] Chao Yu, Dongxu Wang, Tianpei Yang, 2018. PRICAI - Proceedings Part-1, “Adaptive Shaping Reinforcement Learning Agents vis Human Reward”, Springer.

[3] Yuzong Liu, Katrin Krichhoff, 2013. Interspeech , “Graph Based Semi-supervised Learning for Phone and Segment Classification”, France.

[4] Xishun Wang, Minjie Zhang, Fenghui Ren, 2018 PRICAI Proceedings Part-1, “DeepRSD: A Deep Regression Method for Sequential Data”, Springer.

[5] Danilo J Rezende, Shakir Mohamed, Daan Wierstra, 2014. Proceedings of the 31st International Conference on Machine Learning, “Stochastic Backpropagation and Approximate Inference in Deep Generative Models”, Beijing, China.

[6] Daniel Gmach, Jerry Rolia, Ludmila Cherkasova, Alfons Kemper, 2007. IEEE 10th International Symposium on Workload Characterization, “Workload Analysis and Demand Prediction of Enterprise Data Center Applications”, Boston, USA.

[7] Diederik P. Kingma, Danilo J Rezende, Shakir Mohamad, Max Welling, 2014. Proceedings of Neural Information Processing Systems (NIPS), “Semi-supervised Learning with Deep Generative Models”, Cornell University, USA.

[8] Kenndy John, Satran Michael, 2018. Microsoft Windows Documents, “Preventing memory leaks in Windows Applications”.

[9] Kingma Diederik, Rezende Danilo, Mohamed Shakir, Welling M, 2014. Proceedings of Neural Information Processing Systems (NIPS), “Semi-supervised Learning with Deep Generative Models”.

[10] H. Zhao, N. Ansari, 2012. Journal of Computing and Information Technology, 20(1). “Wavelet Transform Based Network Traffic Prediction: A Fast Online Approach”.

[11] L. Nie, D. Jiang, S. Yu, H. Song; 2017. IEEE Wireless Communication and Networking Conference, “Network Traffic Prediction Based on Deep Belief Network in Wireless Mesh Backbone Networks”, USA.

[12] M.F. Iqbal, M.Z. Zahid, D. Habib, K. John, 2019. Journal of Computer Networks and Communications Volume, “Efficient Prediction of Network Traffic for Real Time Applications”

[13] A. Sankar, X. Zhang, K. Chen-Chuan Chang; 2019; Proceeding of the IEEE 2019/ACM International Conference on Advances in Social Network Analysis and Mining ASONAM; “Meta-GNN: metagraph neural network for semi-supervised learning in attributed heterogenous information networks”.

[14] T. Miyato, S. Maeda, M. Koyama, S. Ishii; 2019; IEEE Transactions on Pattern Analysis and Machine Intelligence (Vol 41, Issue 8, Aug 2019); “Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning”.

AUTHOR

Nitin Khosla Mr Khosla has worked about 15 years as Asst. Professor at MNIT in the Department of Electronics and Communication Engineering before moving to Australia. He acquired Master of Philosophy (Artificial Intelligence) from Australia, Master of Engineering (Computer Technology) from AIT Bangkok and Bachelor of Engineering (Electronics) from MNIT. His expertise is in Artificial Intelligence (neural nets), Software Quality Assurance and IT Performance Engineering. Also, he is a Certified Quality Test Engineer, Certified Project Manager and a Quality Lead Assessor. During last 14 years, he worked in private and public services in New Zealand and Australia as a Senior Consultant in Software Quality. Currently he is Asst. Director in Australian Federal Government in Performance and Capacity Management and leading multiple IT projects.