Proceedings of the APAN – Research Workshop 2018
ISBN 978-4-9905448-8-1

A Convolutional Neural Network for Network Intrusion Detection System

Leila Mohammadpour, Teck Chaw Ling, Chee Sun Liew and Chun Yong Chong

***Abstract*— System administrators can benefit from deploying Network Intrusion Detection Systems (NIDS) to find potential security breaches. However, security attacks tend to be unpredictable. There are many challenges to develop a flexible and effective NIDS in order to prevent high false alarm rates and low detection accuracy against unknown attacks. In this paper, we propose a deep learning method to implement an effective and flexible NIDS. We used a convolutional neural network (CNN), an advanced deep learning technique, on NSL-KDD, a benchmark dataset for network intrusion. Our experimental results of a 99.79% detection rate when compared against the NSL-KDD test dataset show that CNNs can be applied as a learning method for Intrusion Detection Systems (IDSs).**

***Index Terms*— Convolutional neural network, CNN, intrusion detection system; IDS, network security, deep learning**

I. Introduction

THE Internet, together with enterprise networks, play a major role in global economic and business development. Yet, the variety of network attacks, and their continuously changing nature, can make difficult to achieve secure network. Flexible defense methods that can quickly investigate large quantities of network traffic and accurately detect different kinds of attacks is needed. In network security, anomaly-based IDSs are valuable methodologies to identify both known and unknown (new) attacks. Anomaly-based IDSs are trained to continuously observe normal patterns of behavior and recognize any deviations, or anomalies, from existing normal behaviors [1]. In anomaly-based IDSs, the occurrence of an anomaly can provide critical information. For example, an unusual network traffic pattern could mean that a server is under attack and that data is being moved to an unapproved destination. Anomalies in network traffic highlight not only previously-known attacks, but also identify brand-new attack patterns. However, in many scenarios, anomalies might be normal behaviors that have simply not yet been identified. Thus anomaly-based NIDSs need to be continuously updated with new behaviors and new network protocols. Many IDS methods still suffer from high false alarm rates and low detection accuracy against unknown attacks.

In the past few years, a class of machine learning algorithm, called deep learning, is increasingly being used in classification and pattern recognition. Deep learning applies several information processing layers into a hierarchical architecture to generate a deep model. Deep learning is different from conventional machine learning because of its ability to detect optimal features in raw data through consecutive nonlinear transformations, with each transformation reaching a higher level of abstraction and complexity [2]. Deep learning approaches had been effectively applied to diverse research fields such as medical image processing, natural language processing, speech recognition, and signal recognition [3], [4].

In the field of intrusion detection, a limited number of research studies have investigated deep learning, but none of these have efficiently exploited the full power of deep learning techniques [1]. Among different approaches in deep learning, convolutional neural network (CNN) obtained significant performance in computer vision, such as face and object recognition. CNNs are a variant of standard neural networks in that they use convolution and pooling layers instead of the fully connected hidden layers of traditional neural networks [5]. AlphaGo [6], a breakthrough computer program in deep learning, uses a CNN to sample a policy network. Furthermore, even though CNNs have been recognized as highly accurate, they have not been exploited in the field of IDS [6], [7]. This is where we believe that CNNs can make a difference. In this paper, we attempted to use CNN to address the low accuracy and high false alarm rates in conventional IDS.

The rest of this paper is organized as follows: Section II presents the background of study. Section III describes our proposed CNN-NIDS method in detail. Section IV shows

Leila Mohammadpou, Teck Chaw Ling and Chee Sun Liew (le.vesal@gmail.com; tchaw@um.edu.my and csliew@um.edu.my): Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia

Chun Yong Chong (chong.chunyong@monash.edu): School of Information Technology, Monash University Jalan Lagoon Selatan, Bandar Sunway, 47500 Subang Jaya, Selangor, Malaysia

experimental performances of the proposed algorithm on NSL-KDD. Finally, the conclusion is presented in Section V.

II. Background

The solutions of the NIDS problem come from different discipline. Thus, we introduce the basic concepts of anomaly-based NIDS, deep learning techniques and NSL-KDD [8] dataset.

A. Anomaly-based NIDS

IDS can be categorized into host-based (HIDS) and network-based (NIDS) detection system. HIDS monitor hosts and analyses the host information such as system calls and log files. On the other hand, NIDS monitors whole networks by analyzing the network traffic [9]. NIDS analyses the network traffic, such as traffic volume, IP addresses, service ports, and protocol usage. Considering the growth of network and Internet technologies and new attack type, NIDS are required to process a huge amount of data, which might come from different resources with dynamic network environment [9].

When the system is not updated as frequent as it is supposed to be, some anomalies may be falsely treated as normal traffics. Consequently, with variations in network protocols and behaviors, the anomaly-based IDSs should continuously updated and adapted to dynamic network environments. Various approaches had been proposed in the field of IDS such as statistical-based, knowledge-based, and machine learning-based approaches. However, existing IDS techniques still face some challenges such as high false alarm, low detection accuracy against the unknown attacks, and insufficient analysis capability [1], [10], [11].

B. Deep Learning Techniques

Deep learning has recently gain much attention and popularity in many fields of research due to its effective way in search for an optimum solution given a finite amount of data. In our work, we applied one of the well-known deep learning network called Convolution Neural Network (CNN) in the IDS to use the advantage of feature learning and classification.

The essential part of the neural network is a neuron with an activation function (σ), a set of weights (W) and a set of biases (B) [2]. Regarding these parameters, transformation is defined by:

$$a = \sigma(w^T x + b) \quad (1)$$

Where x is the inputs of neurons, w is the weighs, b is bias, T is matrix transpose and σ is activation function. One of the most popular neural networks is the multi-layered perceptron (MLP). The MLPs has several layers of transformations. Each MLP has one input and one output layers, and it can have further layers between the input and output layers which are called hidden layers. A neural network with multiple hidden layers is usually called a deep neural network [12].

One of the main principles of deep learning is computing higher-level features from lower-level ones from observational data [13]. Recently, CNN has been successfully applied in many works such as image classification, scene text recognition, object tracking, speech recognition, posture estimation, natural language processing, visual saliency detection, and human action recognition [13].

C. Convolutional Neural Networks

The CNN is a variation of the neural network, where its goal is to learn suitable feature representations of the input data. A CNN has two main differences with MLPs, including weight sharing and pooling. Each layer of CNN can be composed of many convolution kernels which are used to generate different feature map. Each region of neighboring neurons is connected to a neuron of feature map of next layer. Furthermore, to generate the feature map, all the spatial locations of the input shares the kernel. After some convolution and pooling layers, one or multiple full connected layers are used for the classification [13]–[16].

Due to the using of shared weights in CNN the model can learn same pattern occurring at different position of inputs, without requiring learning separate detectors for each position. Therefore, the model can be robust to translation of inputs [12].

The pooling layers decreases the computational burden because it reduces the number of connections between convolutional layers [13]. Moreover, pooling layers increase the properties of translation invariance and improve the receptive field of subsequent convolutional layers. Generally, one or multiple fully connected layers are added at the end of the convolutional stream of the network, and a loss function is used to measure the errors for training purpose [12].

At each CNN layers, a set of n kernels $W=\{w_1,w_2,\ldots,w_n\}$ and their biases $B=\{b_1,b_2,\ldots,b_n\}$, is convolved with input data. The convolution between data and each kernel produce a new feature map x_k. For every convolutional layer l, transformation is defined by:

$$x_k^l=\sigma(w_k^{l-1} * x^{l-1} + b_k^{l-1}) \quad (2)$$

During the learning process of CNN, a small window is slidden over the inputs, and the values of bias and weights through this window can optimized from various features of the input data without of their position within the input data [12].

III. The Convolution Neural Network NIDS Method

The CNN-NIDS is a deep learning method that enables NIDS to detect the class of normal and abnormal data. In this paper, to present the possibility of the application of CNNs to

NIDS, the CNN-NIDS is implemented, and the evaluation results using standard dataset is presented.

A. Data Preprocessing

The NSL-KDD dataset contains network connection attributes which is used for the evaluation of CNN-NIDS as the NIDS traffic data. In the preprocessing step, by using 1-to-n encoding, the nominal attributes are converted into discrete attributes. One of the attributes (num_out bound_cmds) column is always zero and this attribute does not have any effect on training and testing, therefore, it is removed from the attribute lists. Then, all the attributes are normalized in the range of [0, 1] by applying max-min normalization. After preprocessing steps, the number of attribute has been expanded from 41 to 121 [17].

B. Input Data Shape

CNN is mostly used in computer vision application, where images are used as the input. Each gray image can be represented as a 2D array, while the RGB images is presented by 3D images. Each array cell in the gray images presents the intensity of a pixel.

In our paper, 121 features of each record of NSL-KDD are translated to a 11×11 array. To translate from 1D array to a 2D array, we start from the first element of 1D array and each 11 elements are used as one row of the 2D array. Then 11 rows compose the 2D array.

Fig. 1 and Fig. 2 show 2 samples of normal and abnormal records, respectively which are randomly chosen from dataset. These examples are only to present that they can be processed in a visual way.

In both figures (Fig. 1 and Fig. 2), data are translated from a vector (1 × 121) to a 2D array (11× 11). It can be seen from Fig. 1 and Fig. 2; the feature shape of normal and abnormal data is different in appearance. Furthermore, in Fig. 1 and Fig. 2, the left side shows the value of features and right side shows the plot of the features.

For illustration of differences between translated 2D arrays of normal (Fig. 3, left) and abnormal data (Fig. 3, right), 100 randomly chosen samples of normal and abnormal data from NSL-KDD are presented in Fig. 3. We present all in one figure. Therefore, each figure consists of 10×10 records (2D arrays). It can be seen from Fig. 3 that the appearance of normal and abnormal data is different.

C. CNN-NIDS Architecture

Fig. 4 shows the architecture of layers in the proposed CNN-NIDS, where the convolution and pooling layers operate to generate the activations of the units in layers. CNN-NIDS transform the data layer by layer from the input layer to final class layer. Generally, after one or multiple convolution layers a pooling layer is used. The parameters in the layers are trained and tuned with an optimization algorithm.

	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	1	0.0588	1
3	1	0	0	0	0	0	0	0	0	0.0588	0
4	0	0	0	0	0	0	0	1	0	0	0
5	0	0	1	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	4.9514e-07	0	0	0
7	0	0	0	0	0	0	0	1.2074e-04	0	0	0
8	0	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	1	0

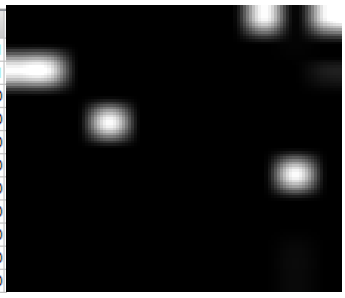

Figure 1. Sample of normal record presented as a 2D array

	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	1	0	1
3	1	0	0	0	0	0	0	0	0	0.0020	0
4	0	0	0	0	0	0	0	1	0	0	0.0800
5	0	0	1	0	0	0	0	0	0	0	0.0200
6	0	0	0	0	0	0	0	2.8419e-05	0	0	0
7	0	0	0	0	0	0	0	0.0022	0	0	0
8	0	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0
11	0	0	0	0	0	0	0	0	0	0.0472	0

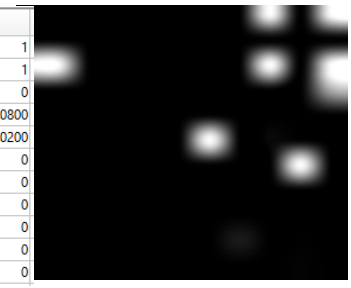

Figure 2. Sample of abnormal record presented as a 2D array

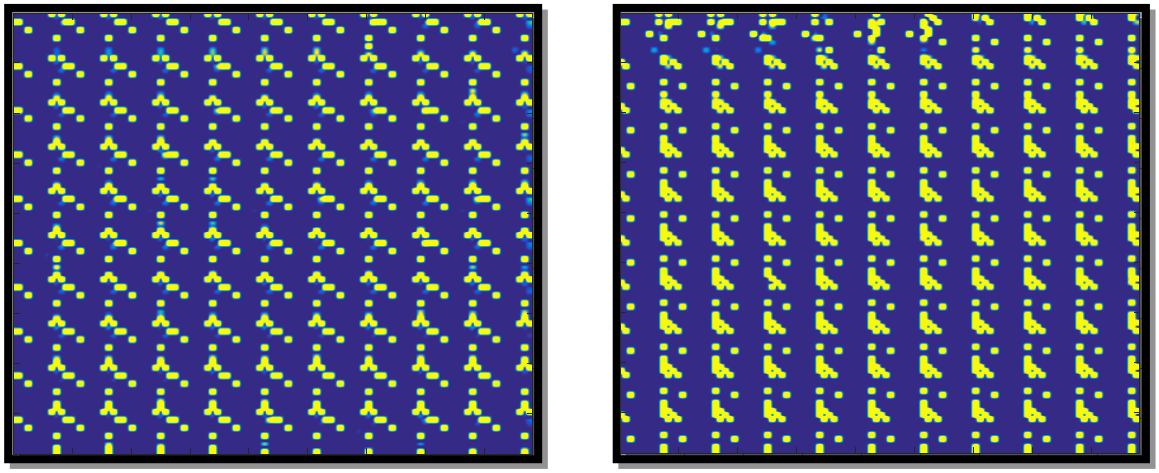

Figure 3. Sample of 100 randomly chosen of translated normal (left) and abnormal (right) records.

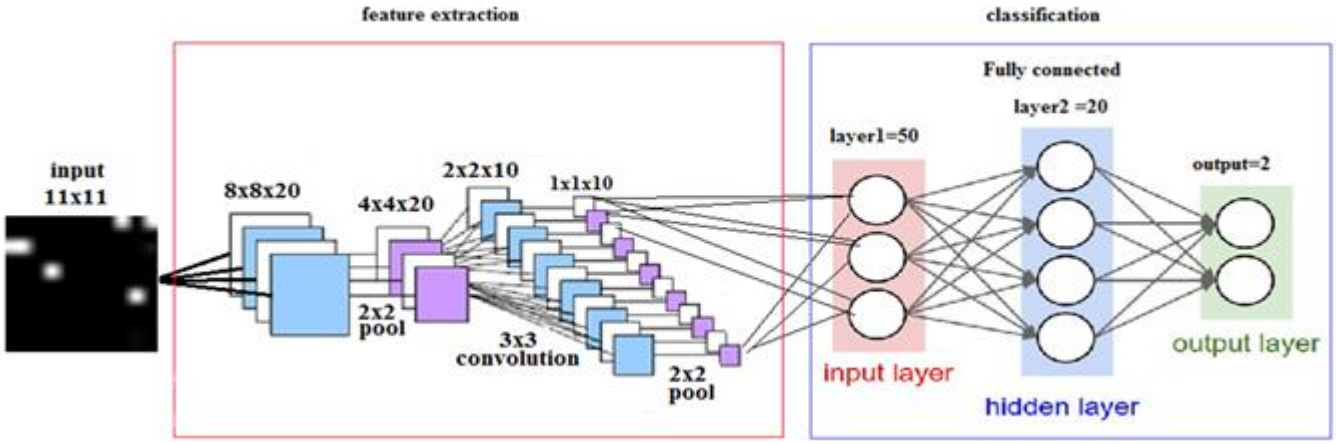

Figure 4. Proposed structure include convolution, pooling and fully connected layers

IV. EXPERIMENTS ENVIRONMENT

In this experiment, the CNN-NIDS used for 2-class classification. The model has been implemented using Python and Keras (the Python Deep Learning library) on a machine equipped with 16GB RAM and CPU Intel Core i7, installed with Windows 10.

As it is presented in Fig. 4, two convolution layers, two pooling layers, and three fully connect layer is used. The kernel size of the convolution layers is [4 *4] and [3 *3] respectively, and the pooling size for both pooling layers are [2 * 2]. Furthermore, three fully connected layers include 50, 20 and 2 neurons are used. To prevent overflow, a dropout by 0.2 is considered. The Rectified Linear Unit (ReLU) activation function is used in all layer except the last layer, which uses the 'Softmax' activation function. For optimization, Adaptive moment estimation (Adam) method is used, and the number of epoch and size of each batch is set to 100 and 500 respectively. We then evaluate the performance of our proposed method by measuring its accuracy

A. NSL-KDD Dataset

We use NSL-KDD dataset because it is standard benchmark dataset in IDS researches, which is derived from KDD Cup 99 [8], [18]. All records in NSL-KDD consists of 41 features and 1 label. The label is set either normal or a specific type of attack. Furthermore, out of the 41 features, there are 3 kind features where the number of continuous, nominal, and binary features are 34, 3, and 4 respectively. The training data consist of normal class and 22 attack types. The testing data consist of normal data and 37 attack type, consist of 21 attack type similar attacks in training data and 16 new attacks. In the evaluation of learning algorithms, the novel attacks in testing data is considered as the unknown attacks.

B. Performance Evaluation

In NSL-KDD, training and test data are originally separated. Therefore, the training and testing are performed by using the original training and testing data. We perform anomaly detection (2-class), which detect the input connection (features) is normal or abnormal.

As shown in Fig. 4, the input size 11×11 is convoluted to 20 channels with size 8×8 after the first convolution. Sample of 4 channels of this convolution output is shown in Fig. 5.

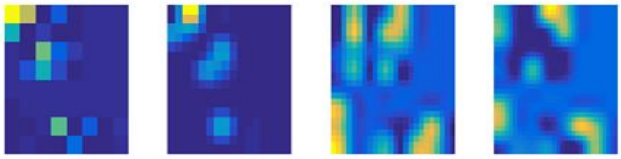

Figure 5. Sample of 4 output channel (8 * 8) after first convolution layer

It can be seen from Fig.5 different features are mapped after first convolution layer.

We implemented a Deep Neural Network (DNN) model to compare the CNN-NIDS to DNN-NIDS. The evaluation result of test data of the NSL-KDD dataset shows 99.79% accuracy of correct detection (percentage of normal cases, detected as normal and abnormal cases, detected as abnormal). However, the result using DNN is 98.90% accuracy of correct detection.

C. Discussion

The CNN has some properties such as locality and pooling, which can improve anomaly recognition performance. Locality property in the convolution layers protect model against noise effects on data. Therefore, the extracted features in convolution layers are robust to noises. Because the extracted features from normal and abnormal data are similar in lower level, traditional machine learning methods may not be able to accurately classify them. However, CNN handles these similarities by creating different higher-level features from lower level features. Pooling layers pools together the similar feature values computed at different locations and exemplified by one value. The pooling layer can control the detection of anomaly with different distribution.

V. CONCLUSION

We have conducted an experiment to show to the feasibility of adopting CNNs in the field of NIDS in order to detect network anomaly. We examined that it is possible to use CNN for anomaly detection. Although the structure, format, and nature of data for NIDS is different from conventional CNN used in image recognition, we proposed a way to overcome the issue. We translate a1D array from network flow to a 2D array to feed into CNN.

Evaluation using the NSL-KDD using (on) our proposed approach shows that the proposed CNN-NIDS method, leveraging the power of deep learning, is feasible to be used as for identifying and detecting potential network intrusions. As part of the future work, we plan to evaluate the proposed model on real-time dataset to dynamically detect various form of potential network intrusions.

REFERENCES

[1] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," in Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), 2016, pp. 21–26.

[2] S. Vieira, W. H. L. Pinaya, and A. Mechelli, "Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications," Neurosci. Biobehav. Rev., 2017.

[3] G. LeCun, Yann and Bengio, Yoshua and Hinton, Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[4] Y. Yoo et al., "Deep Learning of Joint Myelin-T1w MRI Features on Normal-Appearing Brain Tissues Distinguishes Multiple Sclerosis from Healthy Controls," Mult. Scler. J., vol. 23, no. 2, p. 315, 2017.

[5] M. E. Aminanto and K. Kim, "Deep Learning in Intrusion Detection System: An Overview," Proc. Int. Res. Conf. Eng. Technol., pp. 1–12, 2016.

[6] D. Silver et al., "2016 - Mastering the game of Go with deep neural networks and tree search - DeepMind nature16961," Nature, vol. 529, no. 7587, pp. 484–489, 2016.

[7] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey," arXiv Prepr. arXiv1701.02145, pp. 1–43, 2017.

[8] M. Tavallaee and E. B. and W. a. G. A. A. Lu, "'A detailed analysis of the KDD CUP 99 data set,' in Computational Intelligence for Security and Defense Applications," in Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on, 2009, pp. 1–6.

[9] J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: A review," Comput. Secur., vol. 30, no. 6–7, pp. 353–375, 2011.

[10] L. Mohammadpour, M. Hussain, A. Aryanfar, V. M. Raee, and F. Sattar, "Evaluating performance of intrusion detection system using support vector machines: Review," Int. J. Secur. its Appl., vol. 9, no. 9, 2015.

[11] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," Cluster Comput., pp. 1–13, 2017.

[12] G. Litjens et al., "A Survey on Deep Learning in Medical Image Analysis," CoRR, vol. 1702.05747, 2017.

[13] R. C. O'Reilly, D. Wyatte, S. Herd, B. Mingus, and D. J. Jilk, "Recurrent processing during object recognition," Front. Psychol., vol. 4, no. APR, pp. 1–38, 2013.

[14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv Prepr. arXiv1207.0580, 2012.

[15] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks arXiv:1311.2901v3 [cs.CV] 28 Nov 2013," in Computer Vision–ECCV 2014, 2014, vol. 8689, pp. 818–833.

[16] Y. Shi et al., "Early endothelial progenitor cells as a source of myeloid cells to improve the pre-vascularisation of bone constructs," Eur. Cells Mater., vol. 27, pp. 64–80, 2014.

[17] R. Singh, H. Kumar, and R. K. Singla, "An intrusion detection system using network traffic profiling and online sequential extreme learning machine," Expert Syst. Appl., vol. 42, no. 22, pp. 8609–8624, 2015.

[18] H. S. and B. S. D., "KDD Cup '99 Dataset," 1999. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Leila Mohammadpour received her BS degree in Computer Engineering from Islamic Azad University of Shiraz 2010 IRAN. She obtained her MSc. degree in Computer Science from the University Putra Malaysia (UPM) in 2014. Presently she is PhD candidate at University of Malaya, Malaysia. Her

research interests are network security, IDS and Deep Learning.

Teck Chaw Ling is an Associate Professor at the Faculty of Computer Science and Information Technology, University of Malaya and also the chairperson of Malaysia Research and Education Network-MYREN Network and Distributed Systems Working Group. His research areas include Software Defined Networking, Green computing, Core network research, inter-domain Quality of Service (QoS), Voice over IP (VoIP), cloud computing, and network security.

Chee Sun Liew has 15 years of experience in distributed computing. His research is focusing on scientific workflow systems. Dr. Liew has on-going collaborations in the area of workflow technologies with many research groups in and out of Malaysia including those from University of Edinburgh and Manchester Metropolitan University. Dr. Liew is the founder and the head of UM Data-intensive Computing Centre. DICC facilitates the use of data-intensive computing and high-performance computing technologies in accelerating the advancement in scientific discovery.

Chun Yong Chong is a lecturer at the School of Information Technology, Monash University, Malaysia. His research interests include Software Engineering, Software Maintenance, Software Clustering, Software Remodularization, Software Fault Prediction and Cloud Computing.