A Comparative Study of Learning Techniques with Convolutional Neural Network Based on HPC-Workload Dataset

Anupong Banjongkan, Watthana Pongsena, Nittaya Kerdprasop, and Kittisak Kerdprasop

***Abstract*—High-Performance Computing or HPC is a computer system that has high computing power. The HPC supports various computational domains. A huge amount of jobs from a large group of users prefer to complete their jobs in this kind of system. Therefore, managing the jobs or job scheduling is very important since it involves the overall system efficiency. The analysis of an HPC-workload log file is a solution to improve system efficiency. Because some information may appear in the log file, this information can help the system scheduler to make an appropriate decision for job scheduling in the HPC system. This research proposed predictive models for predicting the job status at the finishing state in the HPC system. The model can be used as a tool for monitoring the jobs in the HPC system. We develop and build the three models including HPC-CNN, HPC-AlexNet, and HPC-VGG16 based on the two different learning techniques, which comprise Initial and Transfer Learning of Convolutional Neural Network based on the HPC-work load dataset. Moreover, the three state-of-the-art Machine Learning methods: Classification and Regression Tree (CART), Artificial Neural Network (ANN), and Support Vector Machine (SVM) are used as the baseline models for performance comparison. The results show that the model that performs the best predictive performance is the proposed HPC-CNN model. It archives 76.48% accuracy of the prediction followed with the CART model (75.60%), while the SVM model performs lowest the accuracy at 66.80%.**

***Index Terms*—Convolutional Neural network, machine learning, transfer learning, high-performance computing, HPC-workload log.**

I. Introduction

The HPC systems provide computing power in many computation domains [1]-[5]. The type of jobs, which are computed on the HPC system is diverse since it combines various computing domains from different users. Therefore, the job management or job scheduling as called job scheduler [6]-[8] is very important for the HPC system. The system efficiency of the HPC system can be evaluated from the job success rate in the system. In other words, the performance of the scheduler affects the overall system efficiency of the HPC system. The efficiency of the system can be evaluated as the power that the system consumes and the job success rate. This means that the HPC system that has a high job success rate indicating the high efficiency of the system. Whereas, the low job success rate indicates the poor efficiency of the system.

Job scheduler like a brain of the HPC system. It is a middle-ware for receiving the job from users. Then, it sends the job to appropriate computing resources with the best strategy. In the job scheduling process, the scheduler records all events that occur in the system with numeric or string to the file as called the HPC-workload log file. This file can be used as source information for a system administrator to investigates or tracks the problems when problems occur in the system. Moreover, the HPC-workload log file may contain some hidden information that the administrator can be used to improve the efficiency of the system. For the traditional HPC-workload log analysis, an administrator manually analyzes using basic statistical methods based on their knowledge. Analyzing that data in this manner may be inefficient since it takes a long time to process, event there is no flexibility to be used for the generic model.

In the last decade, data mining techniques have been widely applied in the log analysis domain [9]. This research proposed the classifier models using Deep Learning with different learning techniques of CNN based on the HPC-workload dataset. The proposed models for predicting the job status at the finishing state in the HPC system. Meanwhile, the three state-of-the-art models of Machine Learning including CART, ANN, and SVM are created based on the same dataset that can be used as the baseline models. This research uses the HPC-workload log file as a dataset. The raw dataset contains 421,459 records. Each record consists of 27 attributes. The dataset is collected from the production of the HPC system named "Atom Computer Cluster" of National Electronics and Computer Technology Center (NECTEC) in Thailand. This HPC is operated under the National e-Science Infrastructure Consortium project. It provides the computing resources to support various computational projects in Thailand since the middle of 2012.

The main objective of this research is (i) to propose the developing and modeling classifier models using Deep Learning with different learning techniques of the CNN based on the HPC-workload dataset, (ii) to propose the comparative study of the performance of the models based on Deep Learning techniques and the models based on Machine Learning techniques including CART, ANN and SVM, and (iii) to demonstrate the advantages as well as disadvantages of the proposed models based on the real-world dataset.

Manuscript received March 27, 2019; revised December 29, 2019.

Anupong Banjongkan, Nittaya Kerdprasop, and Kittisak Kerdprasop are with the School of Computer Engineering, Suranaree University of Technology (SUT), Thailand (e-mail: banjongkan@gmail.com, nittaya@sut.ac.th, kerdpras@sut.ac.th).

Watthana Pongsena is with the School of Computer Engineering, SUT, and also with the Sisaket Rajabhat University, Thailand (e-mail: watthana.p@sskru.ac.th).

doi: 10.18178/ijmlc.2020.10.1.891

In the next section, we illustrate the existing works that relate to our research. Section III, the methodology and the dataset are explained in this section. The experimentation of this research is described in section IV. Section V illustrates the results. Section VI and VII illustrate the discussion and conclusion of this research, respectively.

II. Literature Review

The log file is a time series event-based record of the systems or applications while the process is online. The contents in a log file consist of many types of messages, such as only text, only numeral, or the combination of text and numeral. Analysis of the log file to extract useful information that investigates the root cause of the problem in order to find the suitable configuration of the system, the characteristic of the user's behavior, and etc. Currently, machine learning techniques play an important role in the log analysis domain. In this section, we describe the existing works that related to the use of a machine learning technique in the log analysis.

A. Log Analysis using Machine Learning

The existing works in network log analysis, D. J. Arndt and Zincir-Heywood [10] conduct a comparative study of the three classifier models to classify binary-class problems of encrypted network traffic (SSH encrypted or Non-SSH encrypted). The models are built based on machine learning methods, which include C4.5, K-means, and K-mean with Multi-Objective Genetic Algorithm (MOGA). This research shows C4.5 classifier archiving in overall performance. Meanwhile, the K-mean with MOGA gives the highest accuracy in some test cases as well as reduces time complexity of K-mean. Bujlow *et al*. [11] propose a classifier model with a decision tree method. The C5.0 method is applied to create the model for classifying the seven types of network traffic (Skype, FTP, P2P, Web, Web radio, Game, and SSH). The dataset in this research is a real-world dataset that is collected by their Volunteer-Based System. The result shows that their classifier has a better performance than the previous work. The performance in terms of accuracy of their model is approximately 99.3 - 99.9%.

Cao and Qiao [12] develop an Abnormal Detection System (ADS) for predicting the cyber-attack of the web (normal access or abnormal access) through the two levels of machine learning techniques. For the first level, they create three classifiers: logistic regression, decision tree, and support vector machine to label the data. For the second level, they choose the dataset, which is labeled from the best model according to the first level. Then, the classifier model is built with the Hidden Markov Model (HMM) based on the chosen dataset. The results in terms of performance comparison show that the proposed model archives the highest accuracy at 93.54%. The dataset is collected from the industrial.

Ertam and Kaya [13] conduct a comparative study of the classifiers for classifying the package permission, which composes of Allow, Deny, Drop and Reset-Both. The SVM algorithm is applied to build the model with different kernel functions including Linear, Polynomial, Radial Basis Function (RBF) and Sigmoid function. The dataset is a firewall log, which is taken from the firewall device of the Firat University. The result shows that the SVM classifier model using an RBF function overcomes other kernel functions with the best F_1 score at 76.4%.

B. High-Performance Computing Log Analysis

Hsu and Feng [14] propose a prototype of power awareness in the HPC system. The main objective of the research is to help the HPC system to reduce power consumption. This research uses the β-adaption Algorithm with Dynamic Voltage and Frequency Scaling technology to control the CPU workload in the system. Computer profiling (Real-time log) is used as a dataset. For experimentation, the model runs using benchmark applications. The result demonstrates that the proposed method reduces the power consumption of the HPC system around 20% for sequential Benchmark test cases and 25% for the parallel benchmark test case.

Taerat, *et al.* [15] conduct research using descriptive analysis to explain the characteristics of the HPC system based on system failures. The HPC log file of the IBM Blue Gene/L system of Louisiana Tech University is used as a dataset. The result shows in terms of the enumerated information, such as the severity level of failures, time to repair (TTR) or mean time to failures (MTTF). The conclusion of the analysis assumes a time to repair (TTR) as 10 minutes. Then, the results suggest that the system has a mean time to failure (MTTF) at 5.89 hours, or around 4 times a day.

Pelaez *et al.* [16] develop a system failure detection through the improvement of the clustering algorithm. The proposed method so-called Decentralized Online Clustering (DOC). The system is built based on a case study of the Ranger supercomputer of the Texas Advanced Computing Center. The result illustrates that the performance of the system failure detection is not different compared to the baseline. Meanwhile, the proposed model reduces approximately 2% of the overall overhead (CPU, memory and network bandwidth).

Klinkenberg *et al*. [17] propose a monitoring system for predicting system failures for the HPC system of the RWTH Aachen University. The first phase of the research uses a descriptive statistic to identify the events through the characteristics of the event. In the second phase, a comparative study of classification methods: logistic regression, decision tree, random forest, SVM, and multilayers perceptron based on preprocessing data in the first phase. The performance evaluation using 10-fold cross-variation demonstrates that the proposed model archives 98% precision and 91% recall.

Yoo, Sim, and Wu [18] conduct a comparative study using six methods of machine learning including decision tree, random forest, logistic regression, and naïve bayes to build the classifier models for predicting the job unsuccess at running state in the HPC system. The dataset is an HPC-workload log file of Genepool Scientific Cluster Computer of the NERSC. The result shows that the model based on the random forest method outperforms other models. The performance of the classifier archives 99.8% accuracy, 83.6% recall and 94.8% precision.

In conclusion, the literature review we mentioned above

demonstrates that machine learning techniques are widely applied in many log analysis domains, especially for HPC log analysis. However, in this research, we proposed classifier models using deep learning techniques with different learning techniques of the convolutional neural network.

III. Research Methodology

The classification technique is a technique in machine learning. It is a supervised learning technique to classify or predict binary or multi-label classification problems. Currently, deep learning is a subset of machine learning that becomes a popular technique in the artificial intelligent domain. This research applies deep learning techniques with the CNN method to develops classifier models for predicting the job status at the finishing state in the HPC system.

A. Convolutional Neural Network

The Convolutional Neural Network (CNN) also known as ConvNet is an algorithm, which uses the process of the neural network. The architecture and process of neural networks mimic the process of the human brain. Therefore, the ConvNet is a popular algorithm in deep learning technique that has been applied in many domains, such as the self-driving car system [19], medical science [20], [21], and environmental science [22].

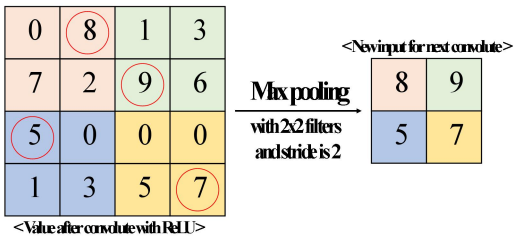
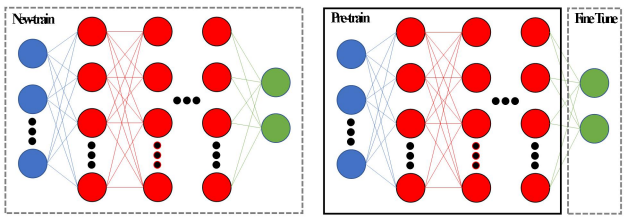


Fig. 1. The max pooling processes.

The architecture of the ConvNet composes of the input layer (receive input data), the hidden layer (computational process), and the output layer (classify or predict the result). In a part of the hidden layer of the ConvNet, it is different from the normal neural network. Therefore, it can be separated into two main procedures. The first procedure is the process of convolution to maintain the value with Rectified Linear Unit (ReLU). The second procedure reduced features using pooling techniques (select one feature in the region). Then, the network re-processes the two procedures again until finish convolution loop. Fig. 1 shows the example of the max pooling technique.



(a) (b)

Fig. 2. Initial learning (a) and transfer learning (b).

There are two techniques to build a classifier model through the ConvNet algorithm. The first technique is the initial learning technique [23], [24]. This technique creates all-new architecture as well as initial learning the data from zero at the model learning state. The second technique is the transfer learning technique [25], [26]. This technique uses the pre-train network with fine-tune technique and modifies the pre-train network according to the dataset. The transfer learning technique reduces the model learning time in the learning state. Generally, this technique suits for using with the general image. Fig. 2 shows a comparison of the initial learning and transfer learning techniques. In this research, we use AlexNet and VGG16 as the pre-train network. Table I shows the characteristics of the pre-train network.

TABLE I: Property of the Pre-train Network

Pre-train	Layers	Hyper Parameter	Input Size
AlexNet	8	61M	227×227
VGG16	16	138M	224×224

B. The State-of-The-Art Machine Learning

Machine learning algorithms are divided into two groups according to the learning process including supervised and unsupervised learning. The supervised learning means the target variable must be defined at the learning state while unsupervised learning the target variable has not to be defined at the learning state. Mostly, the algorithms that propose classification and regression tasks are grouped as a supervised learning technique. Meanwhile, the algorithms that propose a clustering task are grouped as an unsupervised learning technique. In this paper, we use the three state-of-the-art machine learning algorithms including Classification and Regression Tree (CART), Artificial Neural Network (ANN), and Support Vector Machine (SVM) building as the baseline models.

The CART is a tree based-algorithm [27]. The CART algorithm supports the model for classification as well as a regression task. In other words, this algorithm can be used with the dataset that is a categorical and continuous type of target variable. Therefore, the learning data to create the tree structure rule of the CHART algorithm are the Gini index value and variance reduction criterion for classification and regression task, respectively.

Artificial Neural Network (ANN) [28] is an algorithm developed from the motivation of the human brain works. Typically, the ANN architecture composes of three parts including the input layer, hidden layer, and an output layer. The multilayer perceptron is a basis of ANN architecture (one input layer, one hidden layer, and one output layer). The process of the ANN algorithm sends the data into the input layer, and then, propagates the data into the hidden layer. At the same time, the input values are computed by multiplying the weight including the bias values. The result is called “net input”. Then, the activate function is taken to the net input. Finally, the result is processed in the output layer for classifying the data.

	Positive Predict	Negative Predict
Positive Actual	True Positive (TP)	False Negative (FN)
Negative Actual	False Positive (FP)	True Negative (TN)

Fig. 3. The confusion matrix of a two-class problem.

Support Vector Machine (SVM) [29] is an algorithm that

finds the appropriate line to separate the data in a hyperplane. The line is defined from a mathematical function called kernel function. The popular SVM kernel function is Linear, Radial Basis Function, Polynomial, and Sigmoid. Previously, the native SVM supports only binary-label classification problems. Presently, modern SVM can be used to handle the multi-label classification problem as well as increasing the robustness to the outliner. However, finding the appropriate kernel function of the SVM algorithm is a difficult task.

C. Assessments

To evaluate the performance of the models, we select the four evaluators including accuracy, recall, precision, and F-measure. All evaluators are computed from the confusion matrix table. Fig. 3 illustrates the example of the confusion matrix of a two-class problem.

The true positive (TP) is the number of the predicted value is "True", and the actual value is "True". The false negative (FN) is the number of the predicted value is "False", while the actual value is "True". The false positive (FP) is the number of the predicted value is "True", while the actual value is "False". The true negative (TN) is the number of the predicted value is "False", and the actual value is "False".

The accuracy (1) is an evaluator that assesses the overall performance of the model. The recall (2) regards the model performance based on the actual value point view. Meanwhile, the precision (3) observe the model performance base on the predicted value point of view. The F-measure or F_1 score (4) is a harmonic mean of precision and recall.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

```
1 01/01/2017 00:37:22;E;92995            .nectec.or.th;user=c1280hd
   group=p128 jobname=Zr-S-Ads queue=long ctime=1483099421 qtime=148
  3099421 etime=1483099421 start=1483099422 owner=c1280
    .nectec.or.th exec_host=sodium-0-0.ib/11+sodium-0-0.ib/10+sodium
  -0-0.ib/9+sodium-0-0.ib/8 Resource_List.ncpus=1 Resource_List.need
  nodes=1:ppn=4 Resource_List.nodect=1 Resource_List.nodes=1:ppn=4 R
  esource_List.walltime=336:00:00 session=12466 end=1483205842 Exit_
  status=0 resources_used.cput=116:10:54 resources_used.mem=1747740k
  b resources_used.vmem=5130256kb resources_used.walltime=29:33:40
2 01/01/2017 11:13:47;Q;93015            .nectec.or.th;queue=long
3 01/01/2017 11:13:48;S;93015            .nectec.or.th;user=c1290hg
   group=p129 jobname=Ge-Defect-TS2 queue=long ctime=1483244027 qtim
  e=1483244027 etime=1483244027 start=1483244028 owner=c1290hg
         .nectec.or.th exec_host=sodium-0-0.ib/11+sodium-0-0.ib/10+s
  odium-0-0.ib/9+sodium-0-0.ib/8 Resource_List.ncpus=1 Resource_List
  .neednodes=sodium-0-0.ib:ppn=4 Resource_List.nodect=1 Resource_Lis
  t.nodes=1:ppn=4 Resource_List.walltime=336:00:00
```

Fig. 4. The example of some records in the raw HPC-workload log file.

D. Dataset and Tools

This research uses the dataset, which is collected from the National Electronics and Computer Technology Center, Thailand or NECTEC. The dataset is an HPC-workload log from the PBS/Torque scheduler in a production computer cluster called "Atom computer cluster". Atom computer cluster is a medium size HPC system in Thailand that composes of 580 computing elements, 2.7 TBytes of memory, and 50 TBytes of the storage capacity. This system provides free-computing resources for the research in Thailand since mid-2012. The raw HPC-workload log file contains 421,659 records. Each record composes of 27 attributes. Fig. 4 illustrates the example of the raw data of the HPC-workload log file.

In this research, we use MATLAB software version R2018b to build the models through different learning techniques of the CNN network. In addition, we use the IBM SPSS Modeler 18.0 for creating the baseline classifier models through the machine learning methods. Moreover, we use Python 3.4 to handle the raw HPC-workload log in the data preprocess state. All experimentation is run on the working station computer (Intel Xeon Silver 4116 CPU, 2.10 GHz, 24 GB of memory without GPU).

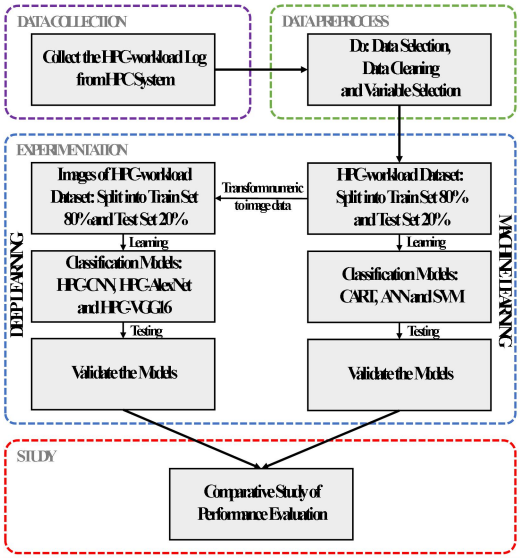


Fig. 5. The research workflow.

IV. Experimentation

In this research, we divide the experimental process into four main parts including data collection, data preprocessing, experimentation, and analyzing the results. The data collection process has already been described in section III. The experimental results and discussions are presented in sections V and VI, respectively. In this section, we describe the data preprocessing and experimentation parts.

TABLE II: Details of All Attributes in the Dataset

Attribute	Description	Type
Queue Type	Queue type in HPC	Categorical
Execute Host	Compute node at job running	Categorical
Limit Wall Time	Time limit which depends on queue type	Continuous
CPU Usage	Number of CPU at the job requires	Continuous
Memory Usage	Memory space at job requires	Continuous
VMemory Usage	Memory space while job running	Continuous
Queueing Time	Time of job waiting in the queue	Continuous
Wall Time	Total time of job stay in HPC	Continuous
Execute Time	The computation time of the job	Continuous
CPU Time	Computation time × CPU Usage	Continuous
Finish Status	Exit code at job ending	Categorical

A. Data Preprocess

After the raw data is collected from the system, we prepare the dataset through the data preprocessing process. This

process makes a suitable dataset for the experimental process. This dataset is a good quality one since, in the year 2016, the HPC system has a little downtime (around 6%). Then, we clean the data by eliminating outliers and missing values. Next, we select 11 out of 27 attributes using expert knowledge. There are 10 predictor variables including "Queue Type", "Execute Host", "CPU Usage", "Memory Usage", "VMemory Usage", "Queueing Time", "Execute Time", "CPU Time", "Limit Wall Time", and "Wall Time". The "Finish Status" is a target variable in this research. The target variable is a binary-class problem that composes of "success" and "error" state. Table II shows details of all attributes in the dataset.

B. Classifiers Modelling

In this process, we use the HPC-workload dataset that is already prepared according to the previous process. We separate the experimentation into two phases. The first phase according to the main objective (i) of this research that is to model the classifier models for predicting the job status at the finishing state of the HPC system. The models are built through the different learning techniques of the CNN network. The second phase according to the objective (ii) of this research that builds the three baseline models through the machine learning methods, which include CART, ANN, and SVM.

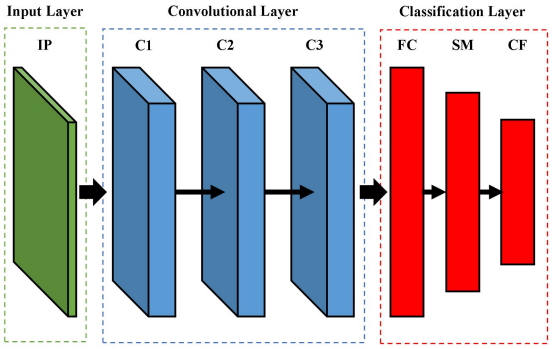


Fig. 6. The HPC-CNN architecture.

In the first phase of the experiment, we propose the three classifier models to predict the job status at the finishing state of the HPC system based on the HPC-workload dataset. A deep learning technique is used to build models. The HPC-CCN is the proposed model, which is modeled through the initial learning technique of the CNN network. The network architecture and configuration of the HPC-CNN are defined using expert knowledge as showed in Fig. 6. The other two proposed models are HPC-AlexNet and HPC-VGG16. These models are built using the transfer learning technique of the CNN network. The AlexNet and VGG16 are used as a pre-train network for HPC-AlexNet and HPC-VGG16, respectively. In the transfer learning process, we fine-tune the three layers of the output port of the pre-train networks. As the input of the proposed models must be an image, we perform an extra-process for transforming the HPC-workload dataset into an image dataset. In this process, the categorical value of the predictor variable is changed to be a nominal value. Then, the numeric value in a dataset is normalized to 0 - 255. At the end of this process, the HPC-workload dataset is ready transformed into the image data. The image data is created one by one from rows in the HPC-workload dataset. We duplicate nine times of row to be 10×10 pixels image data (Fig. 7). Fig. 8 shows an example of the image data after the transformation process is done. The color channel of the image data for the HPC-CNN model is 1 channel (grayscale), while the proposed models, which are modeled from the pre-train network (HPC-AlexNet and HPC-VGG16) are 3 channels (RGB). After the HPC-workload image dataset is created, we randomly split the dataset into 80% train-set and 20% test-set. The three proposed models (HPC-CNN, HPC-AlexNet, and HPC-VGG16) are built from the train-set with the same configuration as shown in Table III. The accuracy, recall, precision, and F-measure score are used to evaluate the performance evaluation of the proposed models. Then, the model, which perform the best performance is selected in order to compare its performance with the baseline models.

	1	2	3	4	5	6	7	8	9	10
1	36	125	113	81	0	44	127	11	255	76
2	36	125	113	81	0	44	127	11	255	76
3	36	125	113	81	0	44	127	11	255	76
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
10	36	125	113	81	0	44	127	11	255	76

Fig. 7. The example of a 2D array 10×10 for creating the image data.

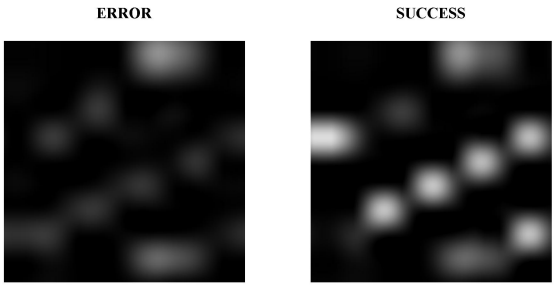


Fig. 8. The example of grayscale image data with a label (x50).

TABLE III: THE CONFIGURATION OF THE LEARNING PROCESS

Parameter	Value
Optimizer	sgdm
Mini Batch Size	100
Max Epochs	3
Initial Learning	0.1
Validation Frequency	10
Validation Patience	Inf

In the second phase, we create the baseline models for our comparative study with the proposed model. The three machine learning methods including CART, ANN, and SVM are used to build the baseline classifier models. We also randomly split the dataset into 80% train-set and 20% test-set. For the ANN configuration, it composes of one input layer with 10 neural nodes, one hidden layer with 7 neural nodes, and the output layer has 1 neural node. For SVM, the RBF kernel function is applied with the Gamma = 0.1 and $C = 3$.

V. RESULTS

From the experimentation, the three CNN network models HPC-CNN, HPC-AlexNet and HPC-VGG16 that are the classifier model, which are used to predict the job status at the finishing state in the HPC system. The performance

evaluation of the models illustrates that the HPC-CNN model archives the highest accuracy at 73.55%. In a part of the two models, which are built using the transfer learning technique, HPC-AlexNet and HPC-VGG16 provide the accuracy of the prediction at 57.35% and 42.65%, respectively. For the performance in terms of recall, precision, and F-measure, only HPC-CNN returns the results. The results are 59.69% recall, 73.35% precision, and 65.79% F-measure score as shown in Table IV. The model building time of the three models shows that the HPC-CNN model takes less time of 5 minutes and 13 seconds while the HPC-VGG16 takes the longest time at 11 hours (Table V). Fig. 9 shows the confusion matrix of the HPC-CNN model.

TABLE IV: THE PERFORMANCE OF THE THREE PROPOSED MODELS

Model	Learning Techniques	Evaluators			
		Accuracy	Recall	Precision	F_1
HPC-CNN	Initial Learning	**0.735**	**0.596**	**0.733**	**0.658**
HPC-AlexNet	Transfer Learning	0.573	0	n/a	n/a
HPC-VGG16	Transfer Learning	0.426	0	n/a	n/a

TABLE V: TIME CONSUMPTION AT THE LEARNING PROCESS

Model	Time Consumption
HPC-CNN	**5 min 13 sec**
HPC-AlexNet	93 min 3 sec
HPC-VGG16	698 min 50 sec

We increase the epoch at the learning state of HPC-CNN up to 18 epochs (Fig. 10). As a result, the accuracy of the model increases up to 76.49%. Table VI illustrates the performance of the HPC-CNN model compared with the baseline models. The results show that the HPC-CNN model (76.5% accuracy) outperforms the others as shown in Table VI.

TABLE VI: THE PERFORMANCE COMPARISON OF PROPOSED VS BASELINE

Machine Learning	Accuracy
CART	0.754
ANN	0.729
SVM	0.668
Deep Learning	**Accuracy**
HPC-CNN	**0.765**

VI. DISCUSSION

The result shows that the performance of the HPC-AlexNet and HPC-VGG16 models are very poor as they cannot return the results of recall, precision, and F-measure. Based on this result, it could be concluded that the models, which are built using the transfer learning technique of CNN from the pre-train networks (AlexNet and VGG16) are unsuitable for the HPC-workload dataset. This could be because the network architecture of the pre-train networks is inconsistent with the input data. In other words, there are some unnecessary of the hidden layers (convolutional part) since the HPC-workload is a low dimensional dataset. This conclusion seems to be supported by the result that the HPC-CNN network archives higher accuracy than the HPC-AlexNet and HPC-VGG16. This is possibly because it has only three hidden layers.

Fig. 9. The confusion matrix of the HPC-CNN model.

VII. CONCLUSION

This research proposed classifier models to predict the job status at the finishing state of the HPC system based on the HPC-workload dataset. The two learning techniques including initial and transfer learning of the CNN network is utilized to model the proposed models. The HPC-CNN network uses the initial learning technique of the CNN network. Meanwhile, HPC-AlexNet and HPC-VGG16 use the transfer learning technique. The AlexNet and VGG16 network is used as the pre-train network. The performance comparison of three proposed models demonstrates that the HPC-CNN model archives the highest accuracy at 76.5%. Moreover, this research is a comparative study of the proposed model with the three state-of-the-art machine learning methods including CART, ANN, and SVM. The results show that the proposed HPC-CNN network outperforms the others with 76.49% accuracy, while the baseline models CART, ANN, and SVM provide the accuracy of the prediction at 75.4%, 72.9%, and 66.8%, respectively.

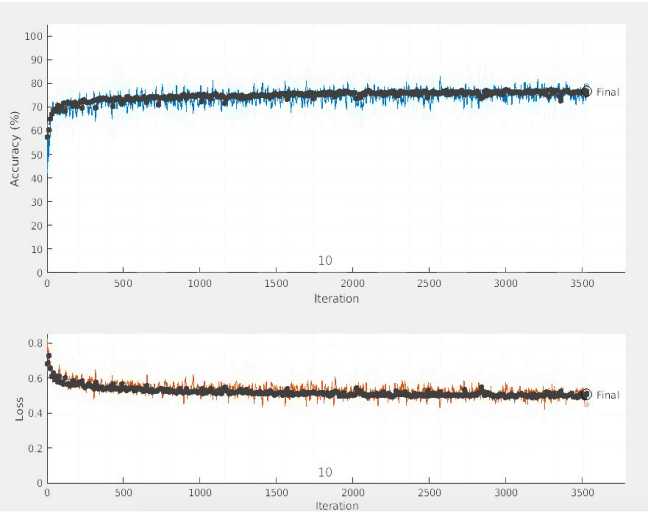


Fig. 10. The progress status of the HPC-CNN model at the learning state with 18 epochs.

For the future work, we will apply a grid search or random search to find the best CNN configurations based on the HPC-workload dataset and increase the scale of the dataset in

order to enhance the performance of the model.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

The first author is designing the research framework, organizing the experimentation steps and preparing the manuscript. The second author helps to validate the manuscript. The third author had approved the final version. The last author takes part in the experimentation design.

Acknowledgment

The authors would like to acknowledge the "National e-Science Infrastructure Consortium" of NECTEC for providing the HPC-workload as a dataset that we use in this research (URL: http://www.escience.in.th). The first author has been supported by a scholarship from the Suranaree University of Technology (SUT). The second author has been supported by a scholarship from the Ministry of Science and Technology, Thailand. The third, fourth, and fifth authors are researchers of the Data and Knowledge Engineering Research Unit, which has been fully supported by a research grant from SUT.

References

[1] V. Sipkova, L. Hluchy, M. Dobrucky, J. Bartok, and B. M. Nguyen, "Manufacturing of weather forecasting simulations on high-performance infrastructures," in *Proc. 2016 IEEE 12th International Conference on e-Science (e-Science)*, Baltimore, MD, USA, 2016, pp. 432–439.

[2] S. Sehrish, J. Kowalkowski, M. Paterno, and C. Green, "Python and HPC for high energy physics data analyses," in *Proc. the 7th Workshop on Python for High-Performance and Scientific Computing*, Denver, CO, USA, 2017, pp. 1–8.

[3] R. Dolezal, T. C. Ramalho, T. C. C. França, and K. Kuca, "Parallel flexible molecular docking in computational chemistry on high-performance computing clusters," in *Computational Collective Intelligence*, M. Núñez, N. T. Nguyen, D. Camacho, and B. Trawiński, Eds. Cham: Springer International Publishing, 2015, vol. 9330, pp. 418–427.

[4] A. Kawalia *et al.*, "Leveraging the power of high-performance computing for next generation sequencing data analysis: Tricks and twists from a high throughput exome workflow," *PLOS ONE*, vol. 10, no. 5, p. e0126321, May 2015.

[5] E. J. Nielsen and B. Diskin, "High-performance aerodynamic computations for aerospace applications," *Parallel Computing*, vol. 64, pp. 20–32, May 2017.

[6] A. Reuther *et al.*, "Scheduler technologies in support of high-performance data analysis," in *Proc. 2016 IEEE High-Performance Extreme Computing Conference (HPEC)*, Waltham, MA, USA, 2016, pp. 1–6.

[7] M. Etinski, J. Corbalan, J. Labarta, and M. Valero, "Parallel job scheduling for power constrained HPC systems," *Parallel Computing*, vol. 38, no. 12, pp. 615–630, Dec. 2012.

[8] Z. R. M. Azmi, K. A. Bakar, M. S. Shamsir, N. W. Manan, and A. H. Abdullah, "Scheduling grid jobs using priority rule algorithms and gap filling techniques," *International Journal of Advanced Science and Technology*, vol. 37, p. 16, 2011.

[9] A. Oliner, A. Ganapathi, and W. Xu, "Advances and challenges in log analysis," *Communications of the ACM*, vol. 55, no. 2, p. 55, Feb. 2012.

[10] D. J. Arndt and A. N. Zincir-Heywood, "A comparison of three machine learning techniques for encrypted network traffic analysis," in *Proc. 2011 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, Paris, France, 2011, pp. 107–114.

[11] T. Bujlow, T. Riaz, and J. M. Pedersen, "A method for classification of network traffic based on C5.0 machine learning algorithm," in *Proc. 2012 International Conference on Computing, Networking and Communications (ICNC)*, Maui, HI, USA, 2012, pp. 237–241.

[12] Q. Cao, Y. Qiao, and Z. Lyu, "Machine learning to detect anomalies in web log analysis," in *Proc. 2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, Chengdu, 2017, pp. 519–523.

[13] F. Ertam and M. Kaya, "Classification of firewall log files with multiclass support vector machine," in *Proc. 2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, Antalya, 2018, pp. 1–4.

[14] C.-H. Hsu and W.-C. Feng, "A power-aware run-time system for high-performance computing," in *Proc. ACM/IEEE SC 2005 Conference (SC'05)*, Seattle, WA, USA, 2005, pp. 1–1.

[15] N. Taerat, N. Naksinehaboon, C. Chandler, J. Elliott, G. Ostrouchov, and S. L. Scott, "Using Log Information to Perform Statistical Analysis on Failures Encountered by Large-Scale HPC Deployments," in *Proc. In High Availability and Performance Computing Workshop*, 2008, p. 6.

[16] A. Pelaez, A. Quiroz, J. C. Browne, E. Chuah, and M. Parashar, "Online failure prediction for HPC resources using decentralized clustering," in *Proc. 2014 21st International Conference on High-Performance Computing (HiPC)*, Goa, India, 2014, pp. 1–9.

[17] J. Klinkenberg, C. Terboven, S. Lankes, and M. S. Muller, "Data mining-based analysis of HPC center operations," in *Proc. 2017 IEEE International Conference on Cluster Computing (CLUSTER)*, Honolulu, HI, USA, 2017, pp. 766–773.

[18] W. Yoo, A. Sim, and K. Wu, "Machine learning based job status prediction in scientific clusters," in *Proc. 2016 SAI Computing Conference (SAI)*, London, United Kingdom, 2016, pp. 44–53.

[1] A. Shustanov and P. Yakimov, "CNN design for real-time traffic sign recognition," *Procedia Engineering*, vol. 201, pp. 718–725, 2017.

[19] A. Pal, U. Garain, A. Chandra, R. Chatterjee, and S. Senapati, "Psoriasis skin biopsy image segmentation using deep convolutional neural network," *Computer Methods and Programs in Biomedicine*, vol. 159, pp. 59–69, Jun. 2018.

[20] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, Feb. 2018.

[21] C. Zhang, J. Yan, C. Li, H. Wu, and R. Bie, "End-to-end learning for image-based air quality level estimation," *Machine Vision and Applications*, vol. 29, no. 4, pp. 601–615, May 2018.

[22] N. Kondo, W. Chinsatit, and T. Saitoh, "Pupil center detection for infrared irradiation eye image using CNN," in *Proc. 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, Kanazawa, 2017, pp. 100–105.

[23] Q. Le, O. Boydell, B. Mac Namee, and M. Scanlon, "Deep learning at the shallow end: Malware classification for non-domain experts," *Digital Investigation*, vol. 26, pp. S118–S126, Jul. 2018.

[24] C. Boufenar, A. Kerboua, and M. Batouche, "Investigation on deep learning for off-line handwritten Arabic character recognition," *Cognitive Systems Research*, vol. 50, pp. 180-195, Aug. 2018.

[25] D. Han, Q. Liu, and W. Fan, "A new image classification method using CNN transfer learning and web data augmentation," *Expert Systems with Applications*, vol 95, pp. 43-56, Apr. 2018.

[26] B. Choubin, H. Darabi, O. Rahmati, F. Sajedi-Hosseini, and B. Kløve, "River suspended sediment modeling using the CART model: A comparative study of machine learning techniques," *Science of The Total Environment*, vol. 615, pp. 272–281, Feb. 2018.

[27] H. Li, F. Chung, and S. Wang, "An SVM based classification method for homogeneous data," *Applied Soft Computing*, vol. 36, pp. 228–235, Nov. 2015.

[28] R. Jafari-Marandi, S. Davarzani, M. S. Gharibdousti, and B. K. Smith, "An optimum ANN-based breast cancer diagnosis: Bridging gaps between ANN learning and decision-making goals," *Applied Soft Computing*, vol. 72, pp. 108–120, Nov. 2018.

Anupong Banjongkan is a Ph.D. student in computer engineering program with the School of computer engineering, Suranaree University of Technology (SUT), Thailand. He graduated with B.S. of computer science and master of engineering in electrical engineering at the King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand, in 2007 and 2011,

respectively. His current research of interest includes high-performance computing, machine learning, and knowledge discovery.

Watthana Pongsana is a Ph.D. student in the School of Computer Engineering, Suranaree University of Technology (SUT), Thailand. He received his B.E. and M.E. in computer engineering from Suranaree University of Technology, Thailand, in 2008 and 2012. His research of interest includes software engineering, data mining, artificial intelligence, and human-computer interaction.

Nittaya Kerdprasop is an associate professor and the head of Data Engineering Research Unit, School of Computer Engineering, SUT, Thailand. She received her B.S. in radiation techniques from Mahidol University, Thailand, in 1985, M.S. in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, U.S.A., in 1999. Her research of interest includes data mining, logic and constraint programming.

Kittisak Kerdprasop is an associate professor at the School of Computer Engineering, SUT, and a chair of the School. He received his bachelor degree in mathematics from Srinakarinwirot University, Thailand, in 1986, MS in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, U.S.A., in 1999. His current research includes machine learning and artificial intelligence.