Multi-label Classification of High Performance Computing Workload with Variable Transformation

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

***Abstract*—High Performance Computing (HPC) log analysis is an active research domain. The challenge is how to extract the useful information from the HPC log file because the information resulting from the analysis can be used as a new knowledge to re-configure the HPC system for improving its efficiency. The traditional manner of HPC log analysis is considered inefficient in the sense that it is time-consuming and requires specific knowledge and skills of system administrator. In this research, we empirical study the application of machine learning techniques to perform an HPC log analysis task. We apply machine learning techniques that are different in their learning schemes including C5.0, Support Vector Machine (SVM), and Artificial Neuron Network (ANN) to analyze and predict the job status on the HPC system. We also propose a novel technique, which is called "Grouping & Combining". Grouping means reducing the class labels of the target variable. Doing so the time-consuming for analyzing is reduced. Then, the class labels of the target variable are combined with another variable such that the efficiency of the interpretability could be increased. The dataset used in our experiment is the real-world data obtained from the HPC system of the National Electronics and Computer Technology Center, or NECTEC, Thailand. According to the experimental results, the C5.0 model has the highest prediction accuracy at 88.74%. In contrast, the ANN model shows the best robustness. In addition, the experimental results show that the proposed Grouping & Combining technique can be efficiently used for handling the multi-label classification as it helps increasing the accuracy, consuming less time, and improving interpretability of the learned model.**

***Index Terms*—High performance computing workload, log analysis, multi-label classification, performance evaluation.**

I. Introduction

In the last decade, the computer technologies including hardware, software, and data storage are rapidly growing. Nowadays, the price of a computer is inverse with the performance of the equipment. In other words, today we can buy a cheaper computer equipment with higher performance than in the past. For this reason, the high performance computing (HPC) [1] or super computer is wildly used, and the performance of HPC is scaling up very fast. However, operating the HPC system is highly electricity consumption. Therefore, many researchers pay attention to the issue of improving performance of HPC [2]-[4].

The HPC log analysis is one effective way to remedy the power consumption problem. In general, data in a log file is the time series of the events that occur in the system. Currently, many researchers attempt to analyze the log, particularly the HPC log. The results from HPC log analysis may reveal characteristics of the system or uncover some useful information that can be used to improve the efficiency of the system. The traditional manner of the log analysis that manually performed by a human is inefficient. It takes a lot of time and requires the expert knowledge and high level of skills. Therefore, many researchers apply the data mining techniques for analyzing the log [5]-[7].

In this research, the data mining techniques are utilized to develop a model based on the workload dataset for predicting the job finish status of the HPC system. The job finish status is an important information. It can be used to fine-tune the system leading to the increase in the efficiency of the system. For the comparative study of the classification method, we select the three popular classifiers: C5.0 [8]-[10], support vector machine (SVM) [11]-[13], and artificial neural network (ANN) [14], [15]. The performance of the models is evaluated and compared based on the HPC-workload dataset, which is collected from the production HPC system of National Electronics and Computer Technology Center of Thailand (NECTEC). The log file is created by the PBS scheduler software that recorded the job information such as job wall time, job computation time, job finish status, and so on. The dataset consists of approximately 421,459 records with 27 variables. Besides performing log analysis with the job finish status as a single main target, we also consider enhancing ability of the model by applying the multi-label classification technique.

Multi-label classification is one of the challenges to many researchers in the machine learning (ML) domain [16]-[18]. The major difficulty is the multi-class value of the target variable. It can affect the classification performance. Indeed, the target variable of the dataset in this research contains a high multi-class value. We thus propose a technique to handle the problem of multi-label classification by transforming the target variable. This technique is called "Grouping & Combining".

The contributions of this research are as follows.

1) We demonstrate empirically the efficiency on analyzing the HPC-workload log file of the three famous ML techniques including C5.0, SVM, and ANN.

2) We propose a novel technique that can be efficiently used to handle high multi-label classification problem.

In the next section, we illustrate background knowledge and the existing work, which are related to this research. In Section III, we describe the classifier methods and the proposed technique. The HPC-workload dataset is described in Section IV. Section V demonstrates the experimentation. The last two sections (VI and VII) are the experimental

Manuscript received August 23, 2018; revised October 20, 2018.

The authors are with the School of Computer Engineering, Suranaree University of Technology (SUT), Thailand (e-mail: banjongkan@gmail.com, watthana.p@sskru.ac.th, arc_angle@hotmail.com, nittaya@sut.ac.th, kerdpras@sut.ac.th).

results and conclusions, respectively.

II. Background and Related Works

Log analysis is a popular research topic since the log file contains much information about hardware or software which is ordered in the time series fashion. Data mining is the efficient technique for extracting useful information from the log file. This research uses the data mining for conducting the comparative study of the three classifiers that can be used for analyzing and predicting the HPC-workload dataset. We focus on high multi-label classification issue.

A. Log Analysis

Q. Cao *et al.* [19] uses machine learning techniques to detect the abnormal sign from a cyber attack on web services. The decision tree (DT) and hidden Markov model (HMM) work together in this research. The dataset is collected from the industry sector. It contains around 4.6M records. The proposed method gives high accuracy (93.54%) for detecting the abnormal sign from cyber attack.

The existing work in [20] concerns about the quality of service (QoS). In order to keep high availability and reliability of service, the maintenance operation is very important. The problem is a high frequency of maintenance operation that leads to a high cost. Meanwhile, low frequency of maintenance operation is a risk. Therefore, this research fine-tunes optimum of the frequency of the maintenance operation. They use the classification-based ML technique to predict the system failure based on the data log file.

B. High Performance Computing Log Analysis

The research in [21] uses HMM method with frequency based strategy to focus only important log messages to predict the job remaining time in the supercomputing system. The maximum accuracy of the prediction is as high as 75%, and the error on job remaining time prediction is less than 200 seconds.

Y. Liang *et al.* [22] use the tagged logs from the BlueGene machine to discover the correlations between the fatal and non-fatal events. Then, they use these correlations for predicting the failures.

B. H. Park *et al.* [23] develop a framework for a deep monitoring of the HPC system. The framework composes of many tools, such as Cassandra (a highly scalable), the NoSQL distributed database (high performance column-oriented), and the Apache Spark (a real-time distributed in-memory analytics engine). The root cause analysis of the system failure is the focus of their research.

Yoo *et al.* [24] conduct a comparative study of classification methods including DT, Random Forest, Naive Bayes, and SVM with HPC-workload dataset from the Genepool scientific cluster at NERSC. Their research aims to find the patterns of unsuccessful job status. The result of the comparative study shows that the Random Forest method is the best with 99.8% accuracy, assessed with the 5-fold cross-validation method.

Although the existing research has widely studied on the log analysis, there is still a lack of research that applies the machine learning technique with the HPC-workload dataset. In addition, the existing research described in the literature review attempt to address the binary-class labels where the job status can be either success, or unsuccess. However, in the real world, the job finish status of the HPC system can be varied with miscellaneous multi-class values. Hence, the previous research may not have the ability for defining or expecting the root cause when the job finish status is in unsuccess state. This is a gap that our research aims to address.

III. Methods

This research uses the classification data mining technique to analyze and predict the job finish status in HPC-workload dataset. The classification is a supervised machine learning technique. There are so many algorithms available for the classification task. Those algorithms can be separated into three groups: 1) Linear, such as linear discriminant analysis (LDA) method, 2) Non-linear, such as SVM and ANN, 3) Rule-based or logic-based, such as DT. In this research, we study the non-linear and rule-based classifiers using C5.0, SVM, and ANN to conduct the comparative study.

In this work, we also propose a technique called "Grouping & Combining" that can be used to handle the high multi-label classification by transforming a single target variable to be a group of variables. This research uses IBM SPSS MODELER software for learning dataset and constructing the models. The simulation software runs on the machine with Intel Core-i5, CPU speed is 1.6 GHz and memory capacity is 8 GB.

A. Variable Transformation

We propose a "Grouping & Combining" technique to handle high multi-label classification as a main idea to transform the target variable in order to improve the prediction accuracy, and also to enhance the interpretability of the result. The "Grouping & Combining" technique composes of two steps described as follows.

The first step is called the grouping. This step reduces the wide range of multi-class values of the target variable in order to improve the prediction accuracy. We group the many possible values of target variable into the binary-class where the target class values consist of only "SUCCESS" and "UNSUCCESS". The advantage of this grouping is time reduction. However, the interpretability of the final result is also reduced. For example, if the model predicts the job finish status as "UNSUCCESS", we only know that the job is an error. But the root cause of that error cannot be identified by such result. Therefore, we have to perform the next step.

The second step is called the combining step. The objective of this step is to improve both accuracy and interpretability. The technique in this step is to create the new target variable and its class labels by combining the class labels of the original target variable with the class labels of another predictor variable. There are various ways to select the suitable predictor variable for the combination, such as making a choice based on the knowledge of expert, selecting from importance analysis of variables, or selecting from some statistical analysis techniques such as correlation coefficient analysis and factor component analysis (FCA). In this research, we rely on the knowledge of expert to select a predictor variable. Consequently, the "QUEUE_TPYE" is

selected, then its values are combined with the values of the target variable as demonstrated in Table I.

TABLE I: CLASS VALUES OF TARGET VARIABLE AFTER APPLYING "GROUPING & COMBINING" TECHNIQUE

Queue Type	Finish Status	New Target Variable
SHORT	SUCCESS	SHORT_SUCCESS
	UNSUCCESS	SHORT_UNSUCCESS
MEDIUM	SUCCESS	MEDIUM_SUCCESS
	UNSUCCESS	MEDIUM_UNSUCCESS
LONG	SUCCESS	LONG_SUCCESS
	UNSUCCESS	LONG_UNSUCCESS
OTHER	SUCCESS	OTHER_SUCCESS
	UNSUCCESS	OTHER_UNSUCCESS

B. Classification Techniques

C5.0 is a classifier in a group of rule-based or logic-based machine learning method. It is inherited from the C4.5 algorithm. Thus, C5.0 has all functions of C4.5. Moreover, it includes the new useful functions, such as boosting and cost-sensitive tree. The C5.0 uses information-gain as a criterion for splitting the tree branch. The advantage of the C5.0 is that it works very well with a big dataset because the nature of the algorithm uses less memory and it has high tolerance against the missing values. However, the disadvantages of this technique are that it supports only the categorical target variable and it does not work well with high multi-label classification.

Support vector machine (SVM) is an algorithm that builds classifier by searching for an optimum plane that can separate data with different class labels of target variable. The optimal plane is found from the applying the proper kernel function to transform data from the regular plane to the hyperplane. The popular kernel function of SVM is linear, radial basis function (RBF), polynomial and sigmoid. The original design of SVM is for the binary-label classification. Currently, SVM was developed to handle multi- label classification. The advantage of this method is that it can handle high multi-label classification and outlier. Meanwhile, the drawback is that it is difficult to fine-tune the appropriate parameters to achieve the best performance of the SVM model. In this research, we use the RBF kernel function and set the Gamma=1.0, C=3.

Artificial neural network (ANN) is developed from the concept of a human brain functioning. Normally, the ANN is composed of three main layers and it is called "Multilayer Perceptron". The first layer is the input layer. The second layer is the hidden layer. The hidden layer may contain more than one layer. The last layer is the output layer. The ANN works by propagating data into the input layer through the hidden layer. This process does multiply the input value with the weight, then, plus with the bias value of the hidden layer. The result is called "net value". Next process brings a net value into a transfer function for computing the final result, which is the output. This research configures multilayer perceptron as one input layer with 9 nodes, one hidden layer with 10 nodes, and one output layer with 1 node.

C. Quality Assessments

The accuracy is used to evaluate the performance of the classifier. Accuracy can be calculated from the confusion matrix as showed in Fig. 1. Besides the accuracy, we also consider another two important aspects of the performance including time-consuming and interpretability.

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

Fig. 1. Confusion matrix of a two-class problem.

Generally, the accuracy is used to evaluate the overall predictive performance of the model. The accuracy is a ratio of the number of objects that the model can predict correctly divided by the number of totals objects as demonstrated in equation (1). The value of accuracy stays in the range between 0 and 1. The value nearly 1 means that the model has high accuracy performance, while the value converges to 0 means that the model has poor predictive performance.

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

where: *TP* = Number of objects that the model predicts as "True" and real class is "True".

TN = Number of objects that the model predicts as "False" and real class is "False".

FP = Number of objects that the model predicts as "True" but real class is "False".

FN = Number of objects that the model predicts as "False" but real class is "True".

The time-consuming is the time that spends for building the model and the time used in the predicting process. The less value is defined as better performance. In the perspective of interpretability, we define three levels including poor, neutral, and good. The description of each interpretability level is explained in Table II.

TABLE II: INTERPRETABILITY CRITERION

Level	Description
Good	Receive the specific information, can track to the root cause
Neutral	Receive the scope of information, can expect to the root cause
Poor	Receive the general information, cannot track to the root cause

IV. HIGH PERFORMANCE COMPUTING DATASET

There are many log files in the HPC system since the system consists of many hardware and software modules, such as a log file of compute node, a log file of network equipment, the HPC-workload log from scheduler software, and others. This research pays attention to the job success rate in the HPC system. Thus, the HPC-workload log is appropriate to be a dataset for our experimentation.

A. Data Collection

The HPC-workload dataset in this research is collected from the production HPC system of NECTEC. The name of this HPC system is "Atom cluster computer". It is a medium size HPC system that has totally 580 CPUs, 2.7 terabytes of memory, and 50 terabytes of disk storages. The Atom cluster computer has provided high computing power for many researchers in Thailand since 2012 to present. The total size

of HPC-workload dataset is 421,659 records with 27 variables. Figure 2 shows the raw form of the HPC-workload log. In the figure, only the three records of HPC-workload log are illustrated.

```
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. 2. Example of record in realistic hpc workload dataset.

B. Data pre-Processing

Based on the raw data of the HPC-workload dataset, we select only data records during the year 2017. The dataset consists of 17,018 records. In addition, we select 9 variables from 27 variables for experimentation in this research. These selections follow the advice given by the knowledge expert who is the administrator of this system. The selected data attributes are 8 predictor variables and one target variable. The target field contains the number which has 32 possible values of exit code of job running in the system. Table III shows the details of these 9 variables.

Then, we split the dataset into four subsets. Each subset corresponds to each quarter of the year 2017. Therefore, the sizes of the four subsets (called quarter1, quarter2, quarter3, and quarter4) are 4,045, 5,007, 3,452 and 4,577 records, respectively. This data separation is a simple way to cross-check and validate the performance of the models.

TABLE III: DETAILS OF NINE VARIABLES IN THE DATASET

Feature	Description	Data Type
Queue Type	Queue system type in HPC	Categorical
Execute Host	Compute node that job running	Categorical
Finish Status	Exit code when job ending	Categorical
CPU Usage	Number of CPU that job requires	Numeric
Memory Usage	Memory space that job requires	Numeric
VMemory Usage	Memory space while job running	Numeric
Queueing Time	Time when job waiting in a queue	Numeric
Execute Time	The computation time of job	Numeric
CPU Time	Computation time x Number of CPU	Numeric

V. EXPERIMENTATIONS

We repeatedly perform the same set of experimentation steps on each of the four data-subsets. These steps are graphically shown in Fig. 3. In addition, we generate the three different scenarios:

1) "Raw" is a test case that target variable has not been transformed. The number of possible class values are 32.

2) "Grouping" is a test case that the target variable has been transformed by grouping the related class values into one group. The number of possible class values are 2.

3) "Grouping & Combining" is a test case that target variable has been transformed by combining related class values into one group and also grouping the class value with another predictor variable. The number of possible class values are 8.

Then, each test case is used for building and testing the performance of the models. In this process, we use 70% of data to build three different classifiers based on the three ML techniques including C5.0, SVM, and ANN. Then, the next process is class labeling or predicting process using the rest 30% of data. We evaluate the performance of models by the three measurements: the accuracy, time-consuming, and interpretability. Finally, the results based on each test case are analyzed.

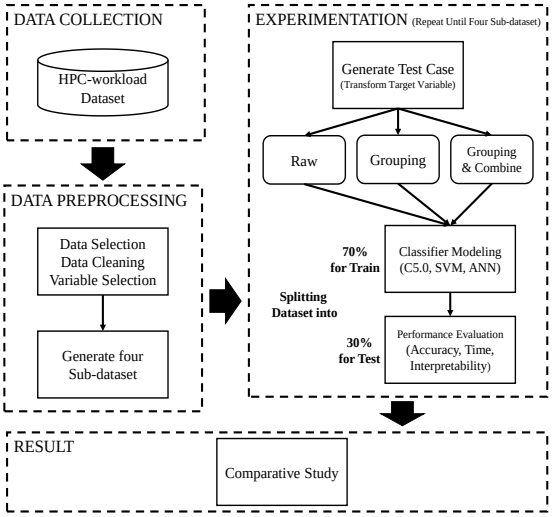


Fig. 3. The research workflow.

VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. Experimental Results

The results on the first data subset, which is the Quarter1 in Table IV, show that the accuracy of C5.0 evaluated on the "Grouping & Combining" test case shows the highest accuracy performance at 88.74%. However, in the "Raw" test case, C5.0 cannot make the rule set. For the "Grouping" test case, the accuracy of C5.0 is 87.16%.

The SVM models yield the accuracy at 71.85%, 67.35% and 74.88% in "Raw", "Grouping" and "Grouping & Combining" test case, respectively. For the ANN classifier, the accuracy results are 77.49%, 75.91% and 80.26% in "Raw", "Grouping" and "Grouping & Combining" test case, respectively.

The results of the second data subset, which is the Quarter 2 in Table IV, show that the accuracy of C5.0 with "Grouping & Combining" test case is the best at 88.38%, while in "Raw" test case, C5.0 also cannot make the rule set. For the "Grouping" test case, the accuracy of C5.0 is 87.46%. SVM perform poorly at 69.88%, 70.01% and 72.63% of accuracy in the "Raw", "Grouping" and "Grouping & Combining" test case, respectively. For the ANN, the predicting accuracy is 77.08%, 73.16% and 76.05% accuracy in "Raw", "Grouping" and "Grouping & Combining" test case, respectively.

The results of the third data subset, which is the Quarter 3 in Table IV, show that the accuracy of C5.0 with "Grouping & Combining" test case is the best at 85.19%, while in the

"Raw" test cast, C5.0 cannot make the rule set. For the "Grouping" test case, the accuracy of C5.0 is 84.46%. The SVM models predict with the accuracy at 65.99%, 66.27% and 75.96% in "Raw", "Grouping" and "Grouping & Combining" test case, respectively. For the ANN, the accuracy results are 72.94%, 71.75% and 79.52% in the "Raw", "Grouping" and "Grouping & Combining" test case, respectively.

The results of the fourth data subset, which is the Quarter 4 in Table IV, show that the accuracy of C5.0 with the "Grouping & Combining" test case is the best at 84.21%, while in the "Raw" test cast, C5.0 also cannot make the rule set. For the "Grouping" test case, the accuracy of C5.0 is 83.14%. The SVM models give the prediction accuracy approximately 77.78%, 72.52% and 73.31% in the "Raw", "Grouping" and "Grouping & Combining" test case, respectively. For the ANN, the results are 79.66%, 72.81% and 74.03% of accuracy in the "Raw", "Grouping" and "Grouping & Combining" test case, respectively.

TABLE IV: ACCURACY OF THE MODELS ON EACH QUARTER

Classifier	Target Variable Class Label Transformation		
	Raw	Grouping	Grouping & Combining
QUARTER 1			
C5.0	n/a	87.163%	88.748%
SVM	71.857%	67.353%	74.881%
ANN	77.493%	75.911%	80.269%
QUARTER 2			
C5.0	n/a	87.467%	88.386%
SVM	69.886%	70.013%	72.638%
ANN	77.088%	73.163%	76.051%
QUARTER 3			
C5.0	n/a	84.461%	85.192%
SVM	65.998%	66.271%	75.961%
ANN	72.939%	71.755%	79.525%
QUARTER 4			
C5.0	n/a	83.142%	84.218%
SVM	77.788%	66.271%	73.314%
ANN	79.659%	71.755%	74.032%

TABLE V: TIME-CONSUMING OF THE THREE CLASSIFIERS

Dataset (Size)	Target Variable Class Label Transformation		
	Raw	Grouping	Grouping & Combining
Q1 (4,045)	14 Mins 49 Secs	8 Secs	8 Secs
Q2 (5,007)	15 Mins 5 Secs	9 Secs	11 Secs
Q3 (3,452)	15 Mins 23 Secs	6 Secs	7 Secs
Q4 (4,577)	14 Mins 39 Secs	7 Secs	7 Secs

TABLE VI: INTERPRETABILITY PERFORMANCE OF THE PROPOSED TRANSFORMATION TECHNIQUES

Transformation Technique	Accuracy	Time	Interpretability
Grouping & Combining	Good	Good	Neutral
Grouping	Good	Good	Poor
Raw	Poor	Neutral	Good

Meanwhile, the time-consuming of the machine learning process including the time for building the models and predicting the results has been observed. The result shows that the "Raw" test case takes longer time than the other test cases. The average time-consuming is around 15 minutes. For the "Grouping" and "Grouping & Combining" test cases, the average time-consuming is around 9 seconds as shown in Table V. The unit of dataset size is the number of records.

The interpretability of the results obtained from different kinds of models using various variable transformation techniques is summarized and shown in Table VI. In terms of all three criteria which are accuracy, time-consuming, and interpretability, it can be noticed that the "Grouping & Combining" transformation technique yields the best performance.

B. Discussions

One of the main contributions of this research is the proposal of a technique that provides classifiers the ability for handling the multi-label classification task. This research is a comparative study through the three classifiers which are C5.0, SVM, and ANN based on the realistic HPC-workload dataset. The result shows that C5.0 is the best classifiers in the "Grouping" and "Grouping & Combining" test cases. However, this classifier cannot return the result in the "Raw" test case. This means that the C5.0 cannot be used to handle the high multi-label classification data because the nature of the C5.0 algorithm supports only binary-label classification. Therefore, it is not suitable for the large dataset, which often contains a complex rule set or a deep tree. However, the C5.0 performs very well in the normal or low multi-class test cases.

For the non-linear classifiers including the SVM and ANN, the ANN is better than the SVM in terms of the predicting accuracy for all test cases. The performance of the SVM model is increased when choosing the proper kernel function and properly fine-tuning the gamma and C parameters according to the dataset characteristics.

VII. CONCLUSIONS

This research demonstrates the technique based on the data mining approach to analyze the HPC-workload dataset from production HPC system of the NECTEC. The target variable is the job finish status. It contains a wide range of multi-class values. The terms multi-class means the target variable has more than two values, and normally the values are much more than two. In this research, we propose a novel variable transformation technique called "Grouping & Combining", which can be effectively used for solving the high multi-label classification problem. The experimental results show that the proposed technique is sufficient to handle the high multi-label classification as it performs good results in terms of the predicting accuracy, time-consuming, and interpretability of the model. Furthermore, the results of the comparative study performed on different kinds of classifiers including the C5.0, SVM, and ANN show that the C5.0 model is potentially to be the best classifier as it shows the predicting accuracy as high as 88.74%. Meanwhile, the ANN is a classifier that is more likely to be the most robustness when considering from all kinds of test cases. For the future work, we plan to scale the experimentation to cover more classification methods and used other data domains.

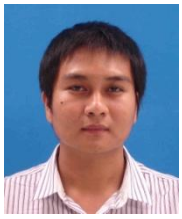
ACKNOWLEDGMENT

The authors would like to acknowledge the "National e-Science Infrastructure Consortium" of NECTEC for providing the HPC-workload as a dataset used in this research (URL: http://www.escience.in.th). The first author has been supported by scholarship from Suranaree University

of Technology (SUT). The second author has been supported by 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 research grant from SUT.

REFERENCES

[1] P. Uthayopas, T. Angskun, and J. Maneesilp, "Building a Parallel computer from cheap PCs: SMILE cluster experiences," in *Proc. the Second Annual National Symposium on Computational Science and Engineering*, p. 10.
[2] C.-H. Hsu and W.-C. Feng, "A power-aware run-time system for high performance computing," in *Proc. the 2005 ACM/IEEE Conference on Supercomputing*, 2005, p. 1.
[3] B. Schroeder and G. A. Gibson, "A large-scale study of failures in high performance computing systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 4, pp. 337–350, Oct. 2010.
[4] W. Huang, J. Liu, B. Abali, and D. K. Panda, "A case for high performance computing with virtual machines," A*CM Digital Library*, 2006, p. 125.
[5] 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.
[6] E. Chuah *et al.*, "Enabling dependability-driven resource use and message log-analysis for cluster system diagnosis," in *Proc. IEEE 24th International Conference on High Performance Computing*, 2017, pp. 317–327.
[7] 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*, pp. 766–773.
[8] Z. Sun, P. Leinenkugel, H. Guo, C. Huang, and C. Kuenzer, "Extracting distribution and expansion of rubber plantations from Landsat imagery using the C5.0 decision tree method," *Journal of Applied Remote Sensing*, vol. 11, no. 2, p. 026011, May 2017.
[9] 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*, 2012, pp. 237–241.
[10] S. Pang and J. Gong, "C5.0 classification algorithm and application on individual credit evaluation of banks," *Systems Engineering-Theory & Practice*, vol. 29, no. 12, pp. 94–104, Dec. 2009.
[11] M. R. Kapadia and D. C. N. Paunwala, "Analysis of SVM kernels for content based image retrieval system," in *Proc. 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing*, p. 6.
[12] S. Porwal, D. S. A. Akbar, and D. S. C. Jain, "Leakage detection and prediction of location in a smart water grid using SVM classification," in *Proc. 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing*, p. 5.
[13] D. P. Kaucha, P. W. C. Prasad, A. Alsadoon, A. Elchouemi, and S. Sreedharan, "Early detection of lung cancer using SVM classifier in biomedical image processing," in *Proc. 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering*, p. 6.
[14] G. C. Jaiswal, M. S. Ballal, and D. R. Tutakne, "ANN Based Methodology for Determination of Distribution Transformer Health Status," in *Proc. 2017 7th International Conference on Power Systems*, pp. 133-138.
[15] S. A. Chandran and M. J. R. Panicker, "An Efficient Multi-Label Classification System Using Ensemble of Classifiers," in *Proc. 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies*, 2017, p. 4.
[16] Y. Feng, J. Jones, Z. Chen, and C. Fang, "An empirical study on software failure classification with multi-label and problem-transformation techniques," in *Proc. the IEEE 11th International Conference on Software Testing, Verification and Validation*, 2018, pp. 320–330.
[17] K.-H. Lo and H.-T. Lin, "Cost-sensitive encoding for label space dimension reduction algorithms on multi-label classification," in *Proc. 2017 Conference on Technologies and Applications of Artificial Intelligence*, p. 6.
[18] R. Kiran, B. R. Lakshmikantha, and R. V. Parimala, "Optimal placement of PMUs and analytics on PMU data using ANN technique," in *Proc. 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing*, 2017, pp. 922-926.
[19] 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*, p. 5.
[20] J. Wang, C. Li, S. Han, and X. Zhou, "Predictive maintenance based on even-log analysis: A case study," *IBM Journal of Research and Development*, pp. 121–132, 2017.
[21] X. Chen, C.-D. Lu, and K. Pattabiraman, "Predicting job completion times using system logs in supercomputing clusters," in *Proc. 2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop*, 2013, pp. 1–8.
[22] Yinglung Liang, Yanyong Zhang, A. Sivasubramaniam, M. Jette, and R. Sahoo, "BlueGene/L failure analysis and prediction models," in *Proc. International Conference on Dependable Systems and Networks*, 2006, pp. 425–434.
[23] B. H. Park, S. Hukerikar, R. Adamson, and C. Engelmann, "Big data meets HPC log analytics: Scalable approach to understanding systems at extreme scale," in *Proc. 2017 IEEE International Conference on Cluster Computing*, Aug. 2017.
[24] W. Yoo, A. Sim, and K. Wu, "Machine learning based job status prediction in scientific clusters," in *Proc. 2016 SAI Computing Conference*, 2016, pp. 44–53.



Anupong Banjongkan is a Ph.D. student in computer engineering program with School of computer engineering, Suranaree University of Technology (SUT), Thailand. He graduated with a B.S. of computer science and master of engineering in electrical engineering from 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 Pongsena is a Ph.D. student in 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, respectively. His research of interest includes software engineering, data mining, artificial intelligence, and human-computer interaction.

Ratiporn Chanklan is currently a researcher with the Data and Knowledge Engineering Research Unit at Suranaree University of Technology (SUT), Thailand. She received his bachelor degree in computer engineering from SUT in 2013, the master degree in computer engineering from SUT in 2014, and a doctoral degree in computer engineering from SUT in 2018. Her current research of interest includes classification, data mining, artificial intelligence.

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.