CHAPTER IV
RESULTS AND DISCUSSION

This section explains performance evaluation, setting-up parameters of neural networks and other estimation techniques, results of feature selection, the accuracy of effort estimation obtained by the proposed approach, and discussion of estimation results.

4.1 Performance Evaluation

There have been many performance evaluation methods being proposed to measure the estimation accuracy of the estimation models. The methods and related metrics used in this study are described below.

4.1.1 Error Metrics

Median Magnitude of Relative Error (MdMRE) is based on median of Magnitude of Relative Error (MRE) [49]. The MRE measures the effort estimation accuracy as follows.

$$MRE^{(i)} = \frac{|y^{(i)} - \hat{y}^{(i)}|}{y^{(i)}} \tag{26}$$

where $y^{(i)}$ is the actual effort and $\hat{y}^{(i)}$ is the estimated effort of project i. It is criticized that the measure will penalize over-estimation.

Prediction at Level a (PRED(a)) is also based on MRE. The value of PRED(a) which is the percentage of prediction within a percent of the actual value. Conte et al. [49] suggested an acceptable value of a = 0.25 at which PRED(0.25) should be equal to or greater than 0.75.

Median Magnitude of Error Relative (MdMER) is based on median of Magnitude of Error Relative (MER) [52]. The MER measures the effort estimation accuracy as follows.

$$MER^{(i)} = \frac{|y^{(i)} - \hat{y}^{(i)}|}{\hat{y}^{(i)}} \tag{27}$$

Since the absolute of residual ($|y^{(i)} - \hat{y}^{(i)}|$) from MER is relative to the predicted value $\hat{y}^{(i)}$, MER penalizes under-estimation but is inclined to over-estimation because the predicted value is a divisor, where more over-estimation brings to low MER [53].

Median of Balanced Relative Error (MdBRE) is based on median of Balanced Relative Error (BRE) [54]. The BRE measures the effort estimation accuracy as follows.

$$BRE^{(i)} = \frac{| y^{(i)} - \hat{y}^{(i)} |}{\min(y^{(i)}, \hat{y}^{(i)})} \tag{28}$$

Median of Inverted Balanced Relative Error (MdIBRE) is based on median of Inverted Balanced Relative Error (BRE) [54]. The IBRE measures the effort estimation accuracy as follows.

$$IBRE^{(i)} = \frac{| y^{(i)} - \hat{y}^{(i)} |}{\max(y^{(i)}, \hat{y}^{(i)})} \tag{29}$$

From the Equation (28) and (29), value of BRE is equal to MRE for over estimation while is equal to MER for under estimation. In contrast, value of IBRE is equal to MER for over estimation while is equal to MRE for under estimation.

As above mentions, there is no the best criteria of performance evaluation. Hence, several error metrics were used to evaluate estimation models for this experiment, where median value was concerned. Since MdMRE, MdMER, MdBRE, and MdIBRE intend to measure an estimation error, low value of them indicates high performance of an estimation. In contrast, PRED (0.25) intends to measure an accuracy, so its high value indicates high performance.

MdMRE, MdMER, MdBRE, and MdIBRE were used to evaluate an estimation model because they are based on median value that pinpoints the middle position of all the observations arranged in an ascending or descending order. As a result, it doesn't affect to outliers or skewed data. For example, if MRE values of project 1 to 5 are 0.20, 0.21, 0.25, 0.27, and 1.75, then MdMRE is equal 0.25 while Mean of MRE (MMRE) based on mean value is equal 0.54. Consequently, MMRE can wrong conclude about performance of the model. Additionally, Foss and Stensrud [55] investigated criterion of performance evaluation. They concluded that the MMRE doesn't always choose the best model.

4.1.2 Statistical Hypothesis Tests

Statistical hypothesis test is used to statistically compare different models based on error metrics.

1313607376

Wilcoxon Signed-Rank test [56] is a non-parametric hypothesis test that is an alternative to parametric Paired Sample t-test. This test is used to find the significant difference between a pair of samples.

The *p*-value is the probability of obtaining the value of the hypothesis test. If the p-value is less than the significance level (α= 0.05), there is a significant difference, where the probability distribution of the value can be approximated by normal distribution.

4.1.3 Cross-Validation

Cross-validation [57] is to evaluate and compare the performance of learning methods. It splits the available data into two non-overlapped parts, namely, training and test data sets. There are several cross-validation methods (e.g., repeated random sub-sampling cross-validation and leave-one-out cross-validation). However, k-fold cross-validation (k-fold) where k=6 was opted for this study as the cross validation technique. Figure 14 shows an example of 6-fold cross-validation. It splits a data set into non-overlapped six folds. The first iteration is to use the fist fold as a test set and remaining folds as a training set. The second iteration is to use the second fold as a test set and remaining folds as a training set. The other iterations are the same process. The cross-validation guarantees all software projects will used to evaluate an estimation model.

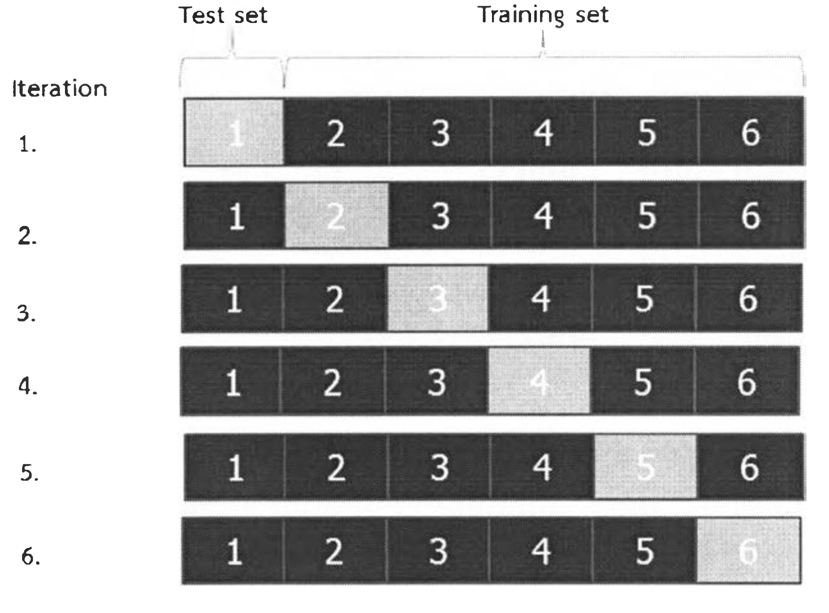

Figure 14: An example of 6-fold cross-validation.

4.2 Parameters Set-up of Neural Networks and Other Estimation Techniques.

Fifteen estimation techniques composed of neural networks (i.e., multilayer perceptron (MLP), fuzzy neural network with subtractive clustering (FnnSUB), fuzzy neural network with fuzzy *c*-mean clustering (FnnFCM), and radial basis function (RBF) network)), fuzzy inference system (i.e., with subtractive clustering (FisSUB) and with fuzzy *c*-mean (FisFCM)), support vector regression (SVR), regression analysis (i.e., ordinary least square (OLS) and robust regression analysis (RoReg)), classification and regression tree (CART), and *k*-nearest neighbor (i.e., KNN-1, KNN-2, KNN-3, KNN-4, and KNN-5). Note that the numbers after "KNN-" refers to the number of training software projects which there is the closet distance to a testing project.

For MLP, the number of input neurons is equal to the number of features from the proposed feature selection algorithm, and a single output neuron was used representing software effort. Warren [58] suggested a number of hidden neurons to be (number of input neurons + number of output neurons) * (2 / 3), which was adopted in this study. The hyperbolic and logistic sigmoid functions were used as an activation functions in the hidden and output neurons, respectively. The weight and bias were initialized with uniformly distributed random numbers corresponding to the interval of the activation function, i.e., interval [-1,1] and [0,1] for hyperbolic and logistic sigmoid functions, respectively. Levenberg-Marquardt was employed as backpropagation training algorithms. The learning rate and momentum were set to 0.001 and 0.9, respectively. Three conditions are used as the stopping rule to terminate network training as follows: (1) the maximum number of training iterations (i.e., 1,000 epochs) have been reached, (2) training error does not change, and (3) training error reduces to the error threshold. The training error is calculated in each iteration by Mean Square Error (MSE). To set the error threshold, assume that each software project has five percentage of residual error. Then, the error threshold is mean square of five percentage of residual error from all training projects.

For FnnSUB, fuzzy inference system (FIS) with subtractive clustering was used for creating rule of fuzzy inference. Input and output membership functions were Gaussian and linear function, respectively. Backpropagation gradient descent and least square method were for training the membership function parameters. The maximum number of training iterations was 1,000 epochs. Error threshold is calculated in Root Mean Square Error (RMSE). These parameters also were for FnnFCM except replacing subtractive clustering with fuzzy *c*-mean (FCM) clustering to separate sets of inputs and outputs. For RBF, error threshold was the same as MLP. Gaussian and linear functions were for hidden and output layers, respectively.

1313607376

For FisSUB and FisFCM, fuzzy inference system (FIS) with subtractive and fuzzy *c*-mean clustering was used for creating rule of fuzzy inference, respectively. Their Input and output membership functions were Gaussian and linear function, respectively, For SVR, Epsilon-insensitive loss function was used as margin-based loss for regression while Hyperbolic tangent was used as kernel function. For OLS, the output function was a linear function. For RoReg, logistic function was used for weighting function. For CART, Tolerance value of each node was set to 0.000001 in Euclidian norm. For KNN, Euclidean distance was applied.

4.3 Results of Features Selection Process

The proposed feature selection process can decrease the number of features, but improve the performance of an estimation model as shown in Table 14-18. There were three estimation models investigated, i.e., phase-wise, overall, adaptive phase-wise effort estimation models. For the first two models, software size/cost drivers/effort drivers were defined as features for estimating a software effort. Although there were 44 features collected, only 23 features of COCOMO II model were used for this experiment for a comparative reason. The proposed feature selection process is to find only necessary features from 23 features for estimating a software effort.

Figure 15 shows the number of features selected by feature selection process for phase-wise and overall effort estimation models averaged from fifteen estimation techniques, where the number of features was required for each estimation technique as shown in Table 13.

1313607376

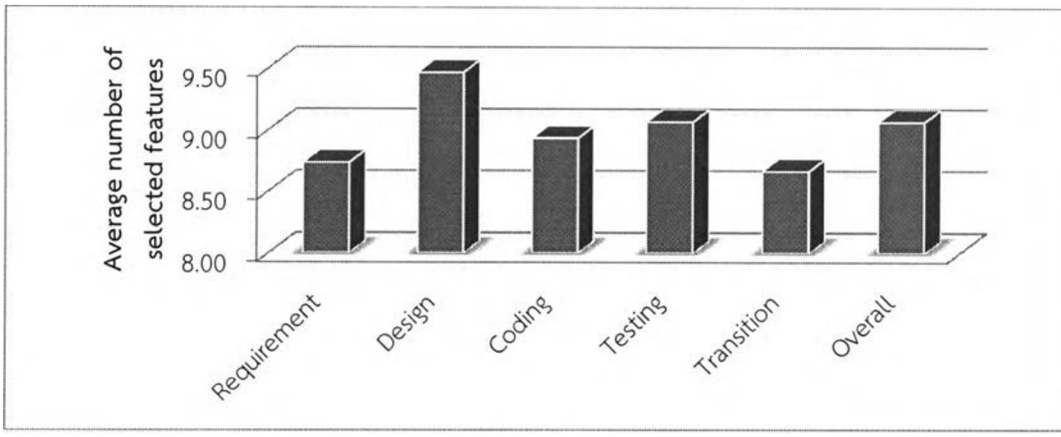

Figure 15: Average number of features required for phase-wise and overall effort estimations.

From Figure 15, it indicates there is slightly different number of features required for across life cycle and overall estimation. The number ranged from 8.67 to 9.47.

Table 13: A number of features required for estimation techniques.

Phase/Method	MLP	FnnSUB	FnnFC	RBF	FisSUB	FisFCM	SVR	OLS	RoReg	CART	KNN-1	KNN-2	KNN-3	KNN-4	KNN-5
Requirement	9	6	10	11	9	6	7	7	7	6	10	10	10	11	12
Design	11	7	10	9	12	6	10	11	7	12	10	9	10	9	9
Coding	11	9	8	11	12	7	7	11	7	10	7	8	9	9	8
Testing	11	11	6	11	11	6	8	10	7	11	7	6	10	9	12
Transition	6	9	9	12	10	6	12	10	7	11	10	7	8	6	7
Overall	10	6	6	12	10	10	8	9	7	11	9	7	8	11	12

For adaptive phase-wise effort estimation model, efforts of prior phases were defined as features. Total number of features was different for different phases, i.e., one, two, three, and four for estimating at design, coding, testing, and transition phases, respectively. As a result, feature selection process was not carried out for predicting a software effort of design phase.

Figure 16 shows an average number of features selected by feature selection process for adaptive phase-wise and overall effort estimation models.

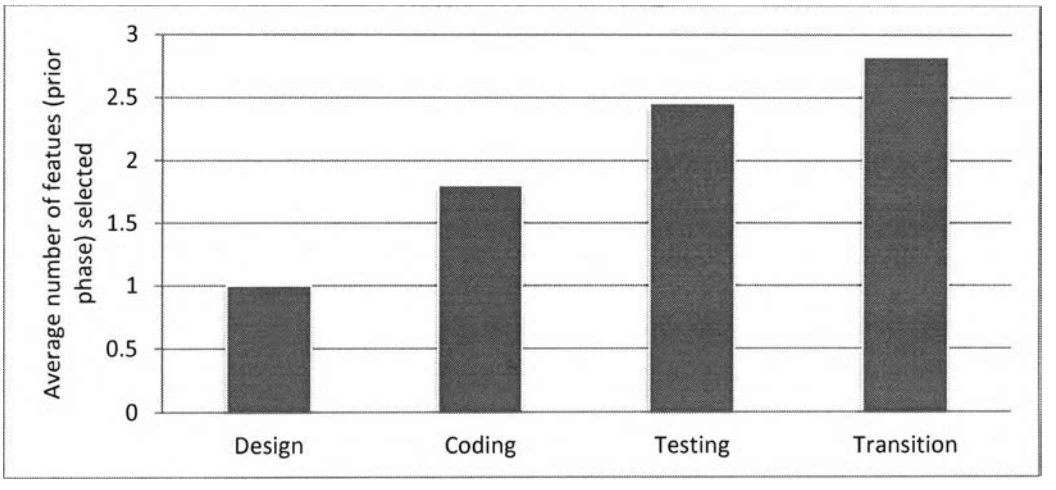

Figure 16: Average number of features (prior phase) required for adaptive phase-wise effort estimation.

A number of features (before taking feature selection process) for design, coding, testing, and transition phases were one, two, three, and four, reflectively. The process is to select some features or only relevant efforts of prior phases to estimate an effort of phase being prediction. From Figure 16, it is true that there was only one feature (an effort of planning phase) for design phase. The average number of features required was 1.8, 2.5, and 2.8 for coding, testing, and transition phases, respectively.

4.4 Results of Effort Estimation

This section describes the results of effort estimation from phase-wise, adaptive phase-wise, and overall estimation models based on all fifteen estimation techniques with the help of feature selection process, where thirty three software projects from a government organization (PNNE-1 data set) were used to create the estimation models cross-validated by 6-fold cross-validation.

Table 14-18 show performance of the estimation models in MdMRE, PRED (0.25), MdMER, MdBRE, and MdIBRE, respectively. For Table 14-18, there are three parts, namely, (1) the estimation results of phase-wise and overall estimation models with feature selection, (2) that without feature selection, and (3) the estimation results of adaptive phase-wise estimation model with feature selection. For the first and second part, the first five rows provide phase-wise estimation results (i.e., requirement, design, coding, testing, and transition). The sixth row (sum-up) is a sum of the phase-wise estimations. The last part provides adaptive phase-wise estimation results ranging from design phase to transition phase.

1313607376

Table 14: Results of effort estimation in MdMRE.

Phase	MLP	FnnSUB	FnnFCM	RBF	FisSUB	FisFCM	SVR	OLS	RoReg	CART	KNN-1	KNN-2	KNN-3	KNN-4	KNN-5
							Phase-wise and overall estimation with feature selection								
Require	0.36	0.43	0.36	0.56	0.47	0.53	0.57	0.5	0.42	0.51	0.69	0.43	0.38	0.34	0.33
Design	0.24	0.22	0.19	0.4	0.27	0.36	0.46	0.46	0.36	0.56	0.42	0.31	0.31	0.32	0.33
Coding	0.38	0.58	0.43	0.42	0.48	0.41	0.45	0.4	0.43	0.3	0.43	0.33	0.34	0.37	0.45
Testing	0.44	0.64	0.53	0.33	0.48	0.48	0.61	0.49	0.42	0.23	0.44	0.5	0.49	0.43	0.42
Trans.	0.64	0.66	0.59	0.68	0.75	0.92	0.61	0.71	0.78	0.68	0.55	0.57	0.59	0.59	0.62
Sum-up	0.31	0.32	0.3	0.45	0.38	0.32	0.43	0.34	0.32	0.41	0.28	0.4	0.34	0.4	0.44
Overall	0.27	0.32	0.41	0.42	0.42	0.38	0.44	0.38	0.31	0.44	0.28	0.44	0.26	0.37	0.39
							Phase-wise and overall estimation without feature selection								
Require	0.53	0.58	0.66	0.47	0.61	0.51	0.49	0.51	0.52	0.51	0.67	0.45	0.38	0.35	0.33
Design	0.25	0.33	0.22	0.52	0.41	0.47	0.48	0.47	0.37	0.56	0.33	0.30	0.30	0.32	0.33
Coding	0.44	0.48	0.44	0.45	0.61	0.46	0.44	0.46	0.33	0.30	0.40	0.41	0.36	0.37	0.33
Testing	0.35	0.63	0.51	0.38	0.50	0.43	0.51	0.43	0.45	0.23	0.48	0.50	0.44	0.42	0.48
Trans.	0.86	0.63	0.59	0.69	0.58	0.90	0.73	0.90	0.78	0.68	0.55	0.63	0.60	0.61	0.64
Sum-up	0.38	0.38	0.33	0.42	0.40	0.33	0.48	0.33	0.33	0.37	0.28	0.35	0.29	0.33	0.33
Overall	0.37	0.37	0.34	0.44	0.38	0.30	0.47	0.30	0.41	0.46	0.31	0.32	0.35	0.31	0.34
							Adaptive phase-wise with feature selection								
Design	0.39	0.45	0.52	0.5	0.46	0.41	0.44	0.41	0.46	0.34	0.63	0.58	0.6	0.58	0.54
Coding	0.26	0.24	0.27	0.19	0.24	0.22	0.33	0.28	0.19	0.22	0.25	0.25	0.19	0.21	0.2
Testing	0.56	0.33	0.45	0.45	0.3	0.28	0.43	0.28	0.28	0.52	0.44	0.29	0.31	0.39	0.38
Trans.	0.67	0.56	0.62	0.65	0.44	0.74	0.64	0.8	0.55	0.67	0.4	0.61	0.59	0.56	0.47

Table 15: Results of effort estimation in PRED (0.25)

Phase	MLP	FnnSUB	FnnFCM	RBF	FisSUB	FisFCM	SVR	OLS	RoReg	CART	KNN-1	KNN-2	KNN-3	KNN-4	KNN-5
							Phase-wise and overall estimation with feature selection								

Require	0.39	0.27	0.39	0.21	0.24	0.27	0.21	0.24	0.36	0.18	0.30	0.39	0.30	0.36	0.42
Design	0.58	0.55	0.52	0.27	0.49	0.39	0.24	0.24	0.36	0.39	0.42	0.42	0.46	0.33	0.24
Coding	0.24	0.15	0.27	0.30	0.18	0.30	0.21	0.27	0.30	0.46	0.21	0.39	0.33	0.30	0.30
Testing	0.39	0.18	0.21	0.39	0.27	0.21	0.24	0.36	0.27	0.52	0.30	0.30	0.30	0.39	0.33
Trans.	0.30	0.21	0.30	0.24	0.27	0.03	0.33	0.09	0.18	0.21	0.39	0.30	0.33	0.21	0.21
Sum-up	0.36	0.42	0.49	0.36	0.27	0.30	0.27	0.30	0.33	0.30	0.46	0.36	0.39	0.36	0.36
Overall	0.49	0.39	0.36	0.33	0.27	0.27	0.24	0.21	0.30	0.24	0.42	0.30	0.46	0.33	0.33
Phase-wise and overall estimation without feature selection															
Require	0.15	0.18	0.30	0.27	0.24	0.18	0.27	0.18	0.15	0.24	0.27	0.27	0.18	0.39	0.15
Design	0.52	0.46	0.52	0.36	0.33	0.33	0.21	0.33	0.27	0.39	0.49	0.42	0.46	0.30	0.52
Coding	0.27	0.18	0.21	0.24	0.12	0.27	0.18	0.27	0.30	0.36	0.21	0.21	0.36	0.30	0.27
Testing	0.39	0.27	0.15	0.33	0.27	0.30	0.24	0.30	0.27	0.52	0.24	0.33	0.36	0.39	0.39
Trans.	0.06	0.24	0.18	0.21	0.27	0.06	0.27	0.06	0.15	0.21	0.36	0.33	0.24	0.12	0.06
Sum-up	0.24	0.39	0.24	0.30	0.27	0.36	0.24	0.36	0.33	0.27	0.46	0.39	0.36	0.39	0.24
Overall	0.33	0.42	0.27	0.39	0.33	0.39	0.21	0.39	0.30	0.27	0.42	0.33	0.36	0.36	0.33
Adaptive phase-wise with feature selection															
Design	0.27	0.24	0.24	0.3	0.24	0.27	0.24	0.27	0.36	0.27	0.12	0.21	0.21	0.18	0.18
Coding	0.49	0.55	0.46	0.58	0.55	0.55	0.33	0.49	0.58	0.55	0.49	0.49	0.64	0.55	0.61
Testing	0.33	0.3	0.39	0.33	0.39	0.46	0.33	0.46	0.42	0.21	0.39	0.39	0.36	0.39	0.36
Trans.	0.18	0.15	0.12	0.24	0.27	0.21	0.24	0.24	0.27	0.27	0.3	0.15	0.24	0.21	0.27

1313607376

Table 16: Results of effort estimation in MdMER.

Phase	MLP	FnnSUB	FnnFCM	RBF	FisSUB	FisFCM	SVR	OLS	RoReg	CART	KNN-1	KNN-2	KNN-3	KNN-4	KNN-5
Phase-wise and overall estimation with feature selection															
Require	0.36	0.55	0.45	0.54	0.47	0.54	0.58	0.46	0.42	0.55	0.50	0.42	0.39	0.41	0.39
Design	0.20	0.18	0.23	0.43	0.33	0.30	0.45	0.49	0.37	0.40	0.34	0.28	0.31	0.35	0.35
Coding	0.49	0.56	0.38	0.35	0.53	0.40	0.44	0.51	0.44	0.28	0.35	0.37	0.37	0.41	0.40
Testing	0.39	0.50	0.45	0.37	0.53	0.41	0.45	0.45	0.48	0.30	0.50	0.39	0.41	0.44	0.40
Trans.	0.48	0.55	0.54	0.54	0.59	0.70	0.54	0.58	0.66	0.53	0.55	0.54	0.50	0.49	0.62
Sum-up	0.34	0.33	0.26	0.38	0.41	0.38	0.46	0.40	0.35	0.37	0.27	0.37	0.31	0.33	0.37
Overall	0.27	0.32	0.42	0.37	0.47	0.41	0.49	0.43	0.36	0.39	0.28	0.38	0.28	0.34	0.33
Phase-wise and overall estimation without feature selection															
Require	0.58	0.58	0.43	0.46	0.58	0.50	0.53	0.50	0.45	0.50	0.58	0.53	0.47	0.40	0.46
Design	0.25	0.27	0.21	0.34	0.44	0.49	0.44	0.49	0.34	0.40	0.33	0.25	0.31	0.32	0.31
Coding	0.43	0.56	0.42	0.37	0.57	0.44	0.47	0.44	0.39	0.32	0.40	0.37	0.37	0.37	0.30
Testing	0.38	0.46	0.47	0.36	0.47	0.46	0.44	0.46	0.44	0.30	0.50	0.40	0.34	0.34	0.43
Trans.	0.89	0.52	0.58	0.53	0.74	0.65	0.58	0.65	0.53	0.52	0.55	0.50	0.44	0.49	0.46

Sum-up	0.36	0.38	0.37	0.38	0.45	0.36	0.47	0.36	0.35	0.34	0.28	0.30	0.34	0.30	0.31
Overall	0.42	0.33	0.38	0.36	0.38	0.36	0.49	0.36	0.39	0.34	0.29	0.30	0.32	0.31	0.32
	Adaptive phase-wise with feature selection														
Design	0.39	0.4	0.47	0.47	0.45	0.4	0.41	0.4	0.34	0.4	0.63	0.6	0.6	0.6	0.6
Coding	0.23	0.21	0.25	0.22	0.24	0.26	0.33	0.27	0.23	0.22	0.26	0.25	0.17	0.2	0.2
Testing	0.41	0.31	0.36	0.43	0.28	0.32	0.39	0.3	0.34	0.43	0.39	0.38	0.38	0.35	0.31
Trans.	0.64	0.55	0.5	0.65	0.41	0.62	0.55	0.62	0.55	0.6	0.5	0.52	0.47	0.51	0.47

Table 17: Results of effort estimation in MdBRE.

Phase	MLP	FnnSUB	FnnFCM	RBF	FisSUB	FisFCM	SVR	OLS	RoReg	CART	KNN-1	KNN-2	KNN-3	KNN-4	KNN-5
	Phase-wise and overall estimation with feature selection														
Require	0.42	0.55	0.48	0.91	0.62	0.73	0.83	0.78	0.58	0.73	0.82	0.67	0.59	0.45	0.48
Design	0.25	0.22	0.23	0.52	0.33	0.37	0.57	0.63	0.44	0.63	0.50	0.32	0.31	0.38	0.45
Coding	0.58	0.91	0.50	0.54	0.68	0.50	0.72	0.67	0.63	0.35	0.48	0.49	0.48	0.50	0.52
Testing	0.52	0.83	0.78	0.46	0.65	0.62	0.76	0.80	0.68	0.30	0.74	0.62	0.65	0.68	0.67
Trans.	0.75	0.75	0.99	1.14	1.24	1.67	0.84	0.99	1.42	0.82	0.75	0.68	0.69	0.75	0.73
Sum-up	0.38	0.39	0.32	0.51	0.54	0.46	0.58	0.42	0.39	0.51	0.28	0.48	0.38	0.42	0.51
Overall	0.31	0.36	0.48	0.58	0.60	0.57	0.67	0.46	0.43	0.55	0.31	0.57	0.34	0.45	0.47
	Phase-wise and overall estimation without feature selection														
Require	0.60	0.91	0.68	0.64	1.00	0.78	0.88	0.78	0.68	0.70	0.82	0.75	0.61	0.46	0.46
Design	0.27	0.37	0.22	0.52	0.60	0.65	0.58	0.65	0.39	0.63	0.50	0.32	0.42	0.33	0.43
Coding	0.58	0.64	0.50	0.54	0.77	0.66	0.69	0.66	0.41	0.37	0.57	0.55	0.54	0.54	0.36
Testing	0.54	0.84	0.64	0.54	0.84	0.71	0.72	0.71	0.65	0.30	0.73	0.61	0.46	0.51	0.72
Trans.	3.48	0.70	1.14	0.94	1.00	1.19	0.85	1.19	1.02	0.82	0.75	0.68	0.75	0.73	0.67
Sum-up	0.50	0.50	0.43	0.56	0.54	0.36	0.59	0.36	0.45	0.51	0.28	0.38	0.41	0.35	0.43
Overall	0.46	0.45	0.46	0.48	0.45	0.42	0.61	0.42	0.55	0.51	0.31	0.37	0.42	0.36	0.45
	Adaptive phase-wise with feature selection														
Design	0.52	0.54	0.69	0.63	0.66	0.57	0.57	0.57	0.48	0.5	1.04	0.83	0.85	0.83	0.83
Coding	0.29	0.27	0.32	0.23	0.29	0.27	0.42	0.29	0.23	0.27	0.26	0.32	0.2	0.21	0.24
Testing	0.7	0.4	0.57	0.6	0.31	0.38	0.55	0.38	0.38	0.71	0.55	0.38	0.44	0.5	0.39
Trans.	0.78	1.05	0.91	0.99	0.68	0.97	1.08	1.1	0.62	1.07	0.67	0.95	0.81	0.73	0.77

Table 18: Results of effort estimation in MdIBRE.

Phase	MLP	FnnSUB	FnnFCM	RBF	FisSUB	FisFCM	SVR	OLS	RoReg	CART	KNN-1	KNN-2	KNN-3	KNN-4	KNN-5
	Phase-wise and overall estimation with feature selection														

1313607376

Require	0.29	0.36	0.32	0.48	0.38	0.42	0.45	0.44	0.37	0.42	0.45	0.40	0.37	0.31	0.33
Design	0.20	0.18	0.19	0.34	0.25	0.27	0.36	0.39	0.30	0.39	0.33	0.24	0.24	0.28	0.31
Coding	0.37	0.48	0.33	0.35	0.40	0.33	0.42	0.40	0.39	0.26	0.32	0.33	0.32	0.33	0.34
Testing	0.34	0.45	0.44	0.31	0.40	0.38	0.43	0.45	0.41	0.23	0.43	0.38	0.39	0.40	0.40
Trans.	0.43	0.43	0.50	0.53	0.55	0.63	0.46	0.50	0.59	0.45	0.43	0.40	0.41	0.43	0.42
Sum-up	0.28	0.28	0.24	0.34	0.35	0.31	0.37	0.30	0.28	0.34	0.22	0.32	0.28	0.30	0.34
Overall	0.24	0.27	0.32	0.37	0.38	0.36	0.40	0.31	0.30	0.35	0.24	0.36	0.26	0.31	0.32
Phase-wise and overall estimation without feature selection															
Require	0.37	0.48	0.41	0.39	0.50	0.44	0.47	0.44	0.40	0.41	0.45	0.43	0.38	0.32	0.32
Design	0.21	0.27	0.18	0.34	0.38	0.40	0.37	0.40	0.28	0.39	0.33	0.24	0.29	0.25	0.30
Coding	0.37	0.39	0.33	0.35	0.43	0.40	0.41	0.40	0.29	0.27	0.36	0.36	0.35	0.35	0.26
Testing	0.35	0.46	0.39	0.35	0.46	0.42	0.42	0.42	0.40	0.23	0.42	0.38	0.31	0.34	0.42
Trans.	0.78	0.41	0.53	0.49	0.50	0.54	0.46	0.54	0.50	0.45	0.43	0.41	0.43	0.42	0.40
Sum-up	0.33	0.34	0.30	0.36	0.35	0.27	0.37	0.27	0.31	0.34	0.22	0.27	0.29	0.26	0.30
Overall	0.31	0.31	0.31	0.32	0.31	0.29	0.38	0.29	0.36	0.34	0.24	0.27	0.30	0.27	0.31
Adaptive phase-wise with feature selection															
Design	0.34	0.35	0.41	0.39	0.4	0.37	0.36	0.37	0.33	0.33	0.51	0.45	0.46	0.45	0.45
Coding	0.23	0.21	0.24	0.19	0.22	0.21	0.29	0.23	0.19	0.21	0.2	0.24	0.17	0.17	0.19
Testing	0.41	0.28	0.36	0.37	0.24	0.27	0.35	0.27	0.28	0.42	0.36	0.28	0.31	0.34	0.28
Trans.	0.44	0.51	0.48	0.5	0.41	0.49	0.52	0.52	0.38	0.52	0.4	0.49	0.45	0.42	0.44

4.4.1 Comparison of Estimation with and without Feature Selection

Feature selection algorithm was applied to select only relevant features to create phase-wise, adaptive phase-wise, and overall estimation models. The algorithm as explained in Section 3.3 is based on mathematical function relation, outlier detection, correlation, greedy backward search, estimation techniques, and cross-validation. Figure 17-21 show estimation results between estimation with feature selection and without feature selection in MdMRE, PRED (0.25), MdMER, MdBRE, and MdIBRE, respectively. For each phase with or without feature selection, a value of MdMRE was a median of values of MRE across 33 software projects. Since there were fifteen estimation techniques were taken, there were fifteen values of MdMRE were averaged as shown in Figure 17. This computation was applied to other error metrics that their results were shown in Figure 18-21.

1313607376

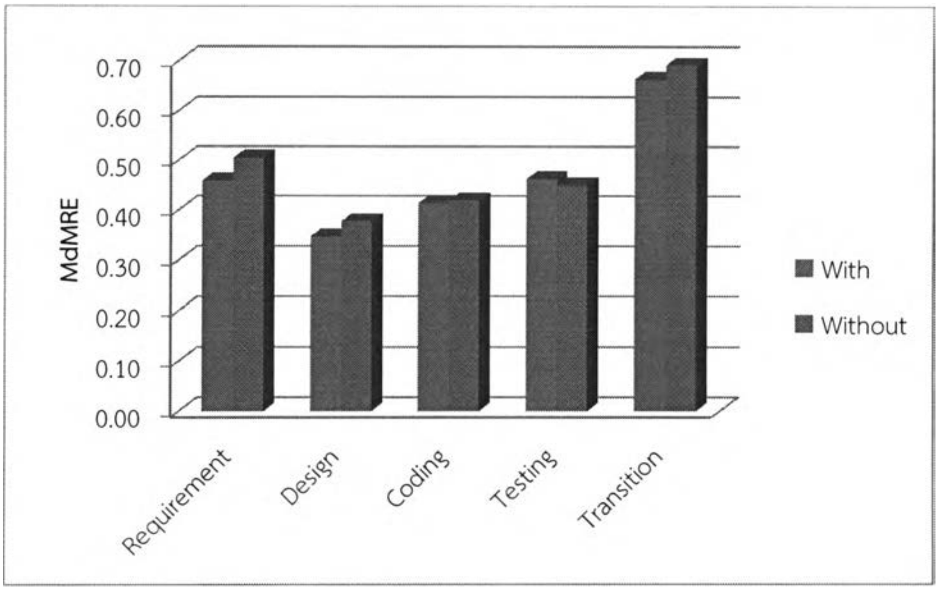

Figure 17: Comparison of estimation with and without feature selection in MdMRE.

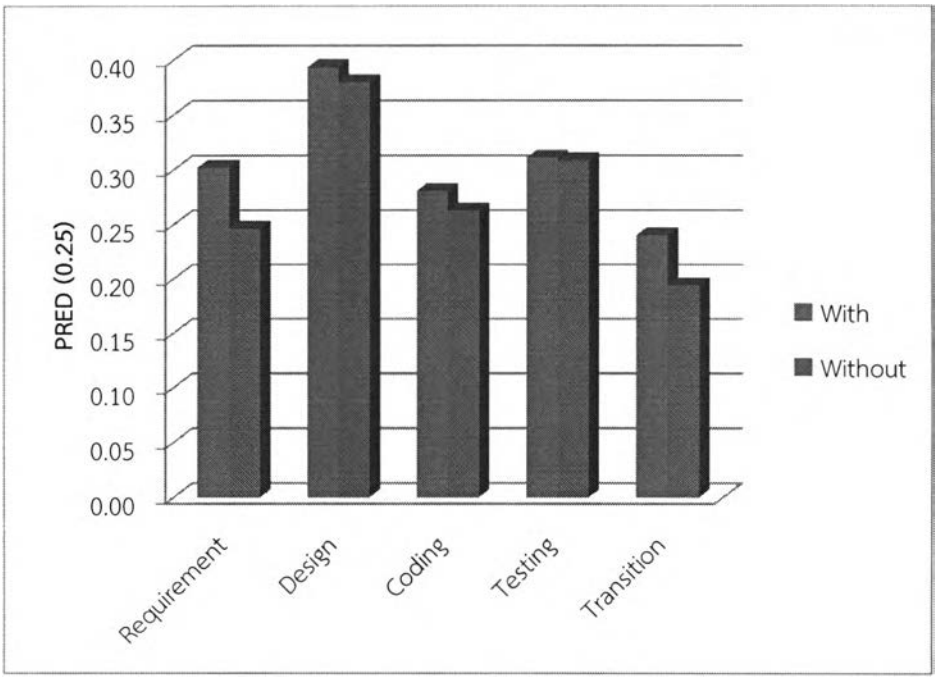

Figure 18: Comparison of estimation with and without feature selection in PRED (0.25).

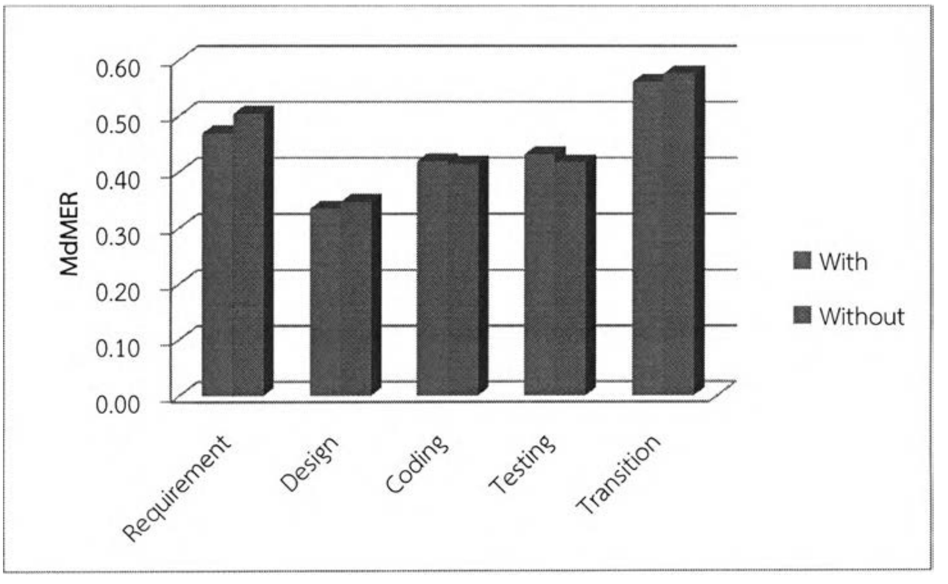

Figure 19: Comparison of estimation with and without feature selection in in MdMRE.

1313607376

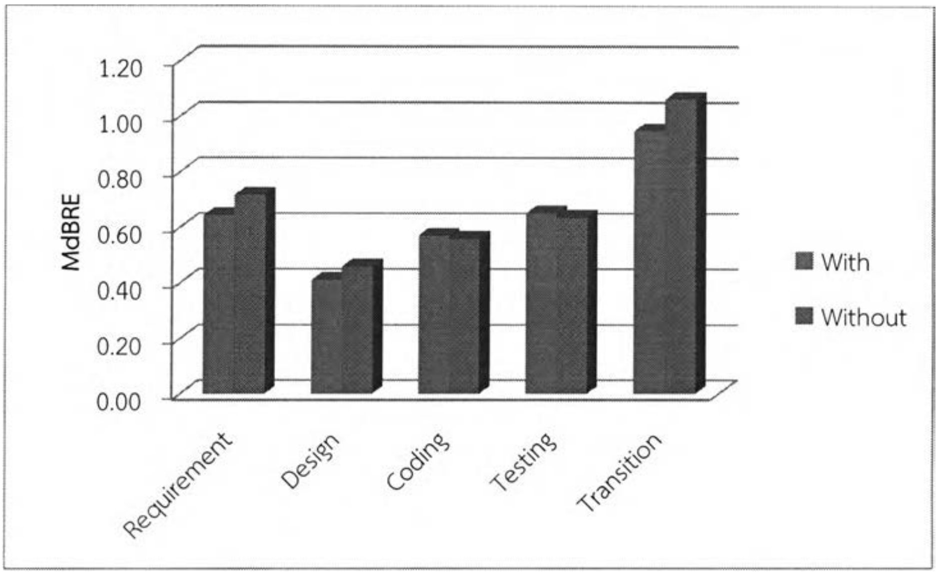

Figure 20: Comparison of estimation with and without feature selection in MdBRE.

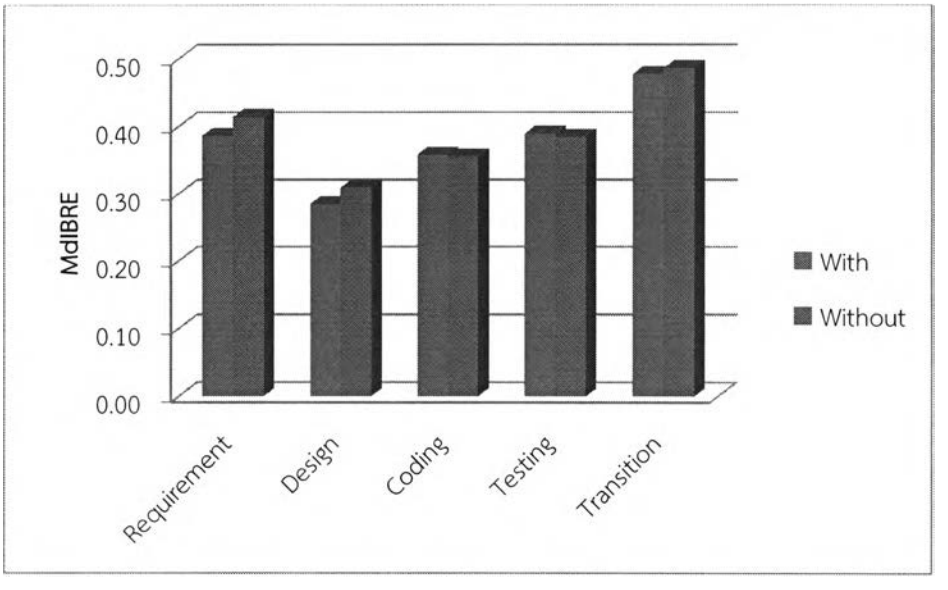

Figure 21: Comparison of estimation with and without feature selection in MdIBRE.

From Figure 17-21, they indicated that the estimation with feature selection can slightly outperform that without feature selection.

To statistically compare the estimation results as shown in part 1-2 of Table 14-17 between estimation with feature selection and without feature selection, 105 pairs of comparisons (i.e., 7 rows x 15 columns) were tested with Wilcoxon. Hypothesis test results indicated that there was no significant different between the estimation with and without feature selection for all pairs in terms of MdMRE, PRED (0.25), MdMER, MdBRE, and MdIBRE. However, the estimation with feature selection required smaller number of features that a project manager spends less time to collect them.

4.4.2 Comparison of Phase-Wise and Overall Effort Estimations

Phase-wise effort estimation tends to estimate an effort of each development phase of a project, where estimated effort of all the phases was sum to indirectly provide overall effort of the project. This effort was compared with estimated effort was directly derived from overall effort estimation. Figure 22 compares the estimation results of phase-wise and overall effort estimations in (a) MdMRE, (b) PRED(0.25), (c) MdMER, and (d) MdBRE, and (e) MdIBRE. To compute value of MdMRE for sum-up or overall estimation, the value of MdMRE was a median of values of MRE across 33 software projects. There were fifteen estimation techniques were taken. So there were fifteen values of MdMRE were averaged. This computation was applied to other error metrics.

1313607376

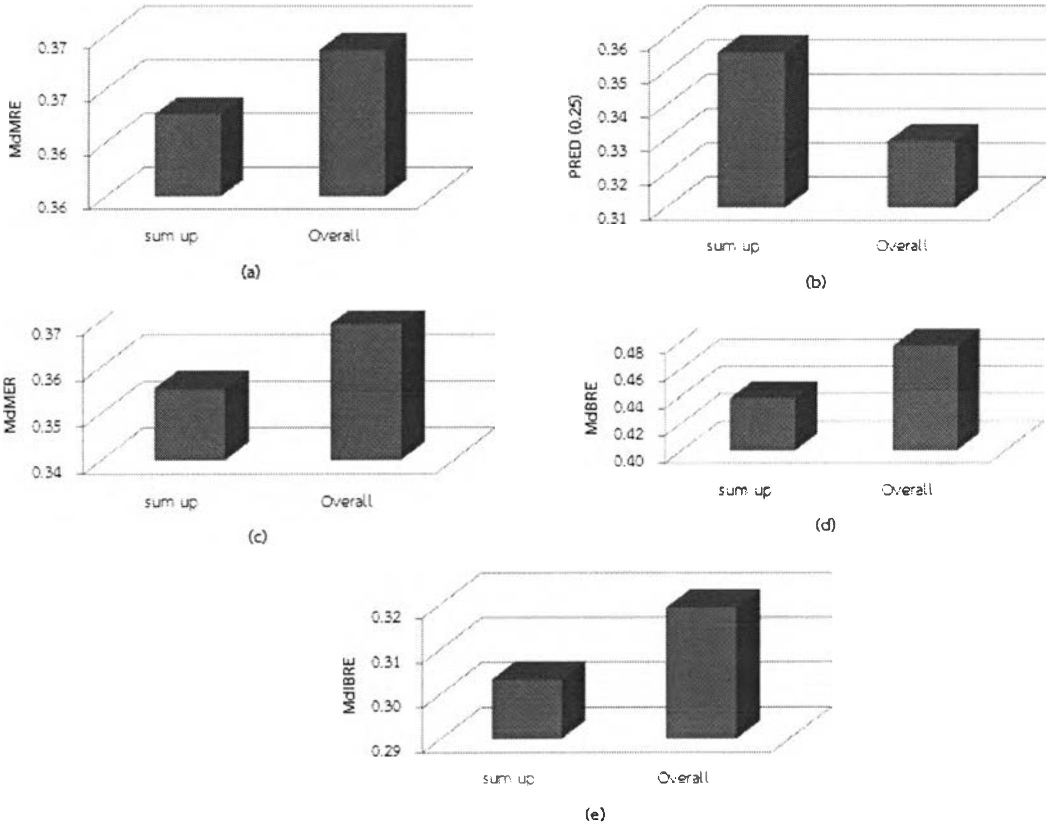

Figure 22: Comparison of phase-wise and overall effort estimations.

From Figure 22, it indicated that the estimation results of sum of all phases derived from phase-wise estimation outperformed that overall estimation for all error metrics. Additionally, estimation of sum of all phases and overall estimation with feature selection outperformed that without feature selection.

Figure 23 shows comparison of phase-wise, sum of phase-wise, and overall estimations across neural networks (Nnet), fuzzy inference (FIS), support vector regression (SVR), regression analysis (RA), and k-nearest neighbor (KNN) in MdMRE.

1313607376

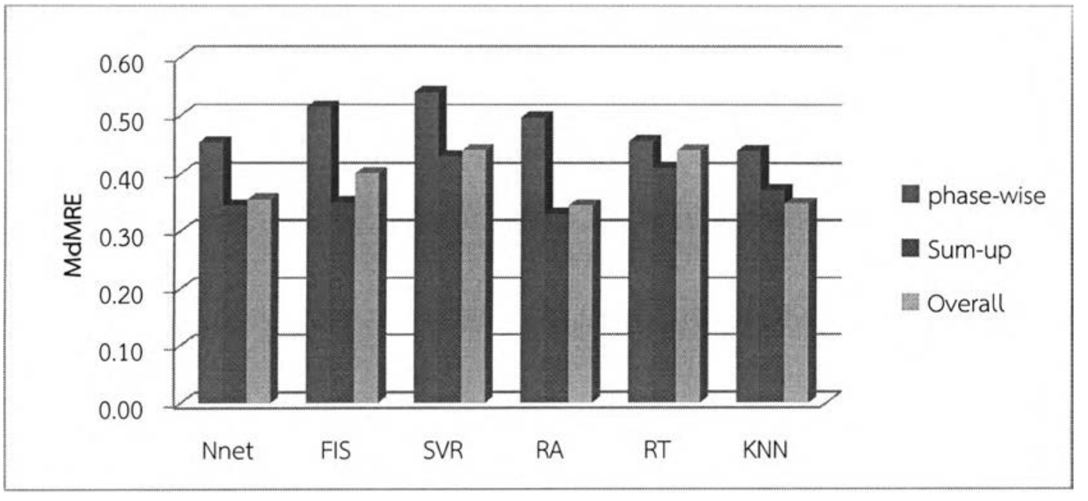

Figure 23: Comparison of phase-wise, sum of phase-wise, and overall estimations in MdMRE.

From Figure 23, it indicated that sum of efforts from phase-wise estimation outperformed overall estimation. Even though phase-wise estimation provided higher error estimation compared with other estimations, most errors came from transition phase having small proposition of effort to overall effort (i.e., 9.8% for PNNE-1 data set).

4.4.3 Comparison of Phase-Wise and Adaptive Phase-Wise Estimations

Adaptive phase-wise effort estimation provided the estimation result starting at second phase (design) because it requires an effort of prior phase to be an input for estimating an effort at predicting phase. Figure 24 compares the estimation results of adaptive phase-wise and phase-wise estimations in (a) MdMRE, (b) PRED(0.25), (c) MdMER, and (d) MdBRE, and (e) MdIBRE. For each phase for phase-wise estimation or adaptive phase-wise estimation, a value of MdMRE was a median of values of MRE across 33 software projects. Since there were fifteen estimation techniques were taken, there were fifteen values of MdMRE were averaged. This computation was applied to other error metrics.

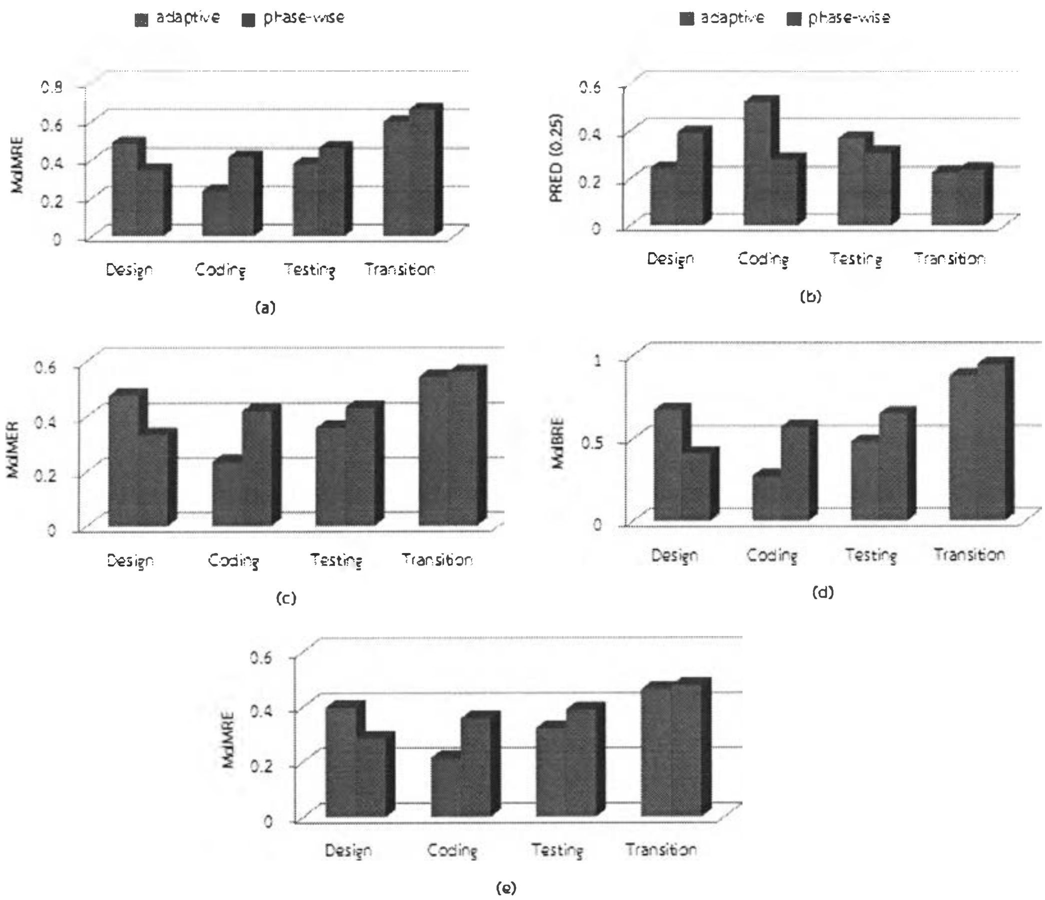

Figure 24: Comparison of phase-wise and adaptive phase-wise estimations.

From Figure 24, it indicated that adaptive phase-wise estimation more outperformed phase-wise estimation in coding and testing phases. For design and transition phases, both estimations provided similar results of estimation.

4.4.4 Comparison of Phase-Wise and COCOMO II Models

Proposed phase-wise estimation model was compared with COCOMO II Post-Architecture model [6]. Figure 25 shows approaches to estimate software effort for the proposed and the COCOMO II model. The proposed phase-wise model is to directly estimate an effort of each phase where the effort of all phases can be combined as overall effort while the COCOMO II model is to estimate overall effort and uses known proportion of an effort of each phase to overall effort (see Section 2.3 for an example of estimating an effort of individual phase).

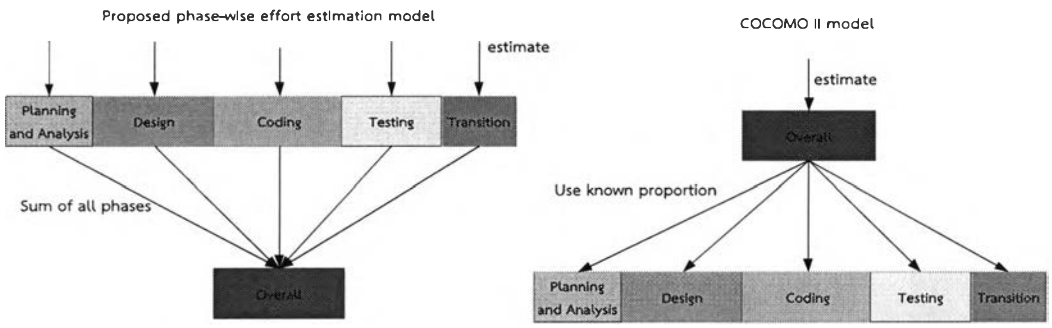

Figure 25: Proposed and COCOMO II phase-wise estimation.

Table 19 shows estimation results of phase-wise model and COCOMO II model for localization and globalization estimations. The "localization estimation" is an effort estimation that software project data for estimating and creating estimation model come from the same organization (i.e., PNNE-1 data set was used to create and test the model). In contrast, "globalization estimation" is that software project data come from the different organizations (i.e., instead of PNNE-2 data set to test the model). The phase-wise model is based on multilayer perceptron (MLP) was used for a comparison. To fairly compare, COCOMO II model was calibrated with local project data (PNNE-1 data set) to adjust its parameters. The calibration was carefully carried out according to given guideline [6].

1313607376

Table 19: Estimation results of phase-wise and COCOMO II models for localization and globalization estimations in MdMRE.

Phase	Localization			Globalization		
	% of effort distribution (PNNE -1)	COCOMO II	MLP-based	% of effort distribution (PNNE -2)	COCOMO II	MLP-based
Planning & Requirement	10.22%	0.46	0.36	8.72%	0.85	0.46
Design	20.07%	0.39	0.24	43.68%	0.69	0.67

Coding	53.03%	0.41	0.38	30.36%	0.39	0.54
Testing	6.80%	0.45	0.44	7.28%	0.89	0.48
Transition	9.87%	0.51	0.64	9.95%	0.85	0.15
Sum-up	100%	0.39	0.31	100%	0.54	0.17

From Table 19, underlined values of MdMRE indicated that phase-wise estimation model based on MLP outperforms COCOMO II model for localization estimation in all development phases and whole project except transition phase and for globalization estimation in all phases and whole project. Although the MLP-based proposed model provided high MdMRE in transition phase of the localization, effort of this phase contained only 9.87% of whole project. Both the proposed and the COCOMO II model often provided better localization estimation results than globalization estimation. Nevertheless, they yielded systematic approach to re-create or calibrate to local environment. For the proposed phase-wise model, four processes to create the model were reported in this dissertation. Twenty one out of 44 features were not used for this experiment. However, they can be added to re-create the model if a project manager considers them as importance features.

Table 20 shows lists of features required for proposed phase-wise estimation model. A gray cell refers to a feature required for each phase.

Table 20: Required features of proposed phase-wise model.

Feature	KSLOC	PREC	FLEX	RESL	TEAM	RELY	CPLX	RUSE	DOCU	PCAP	APEX	PLEX	LTEX
Require													
Design													
Coding													
Testing													
Transition													

From Table 20, thirteen features were required for the phase-wise model while twenty three features were required for COCOMO II model. This means there was 24.5% of decreased number of features. Only three features (i.e., PREC, PLEX, and LTEX) were common features required for all phases. Even though software size was considered as important feature for an effort estimation [6] [7] [8] [9] [10], KSLOC was not required for all phases but required for only design and coding phase.

COCOMO II model supports Waterfall and RUP/MBASE while phase-wise model appropriates to Waterfall and V-model process which can be applied to create an estimation model for iteration software processes such as agile development, where each iteration of development are composed of Waterfall phase.

Summary of the comparison, the phase-wise estimation model was not claimed to be better than COCOMO II model at all since the phase-wise model was investigated with low to median scale of software projects, i.e., maximum team size ranged from three to fifteen persons, software size ranged from 0.27 to 112.28 thousand of SLOC, and software effort ranged from 92 to 1,278 man-days.

1313607376

4.5 Discussion

This section discusses use and limitations of proposed estimation models, effect of programming language, and globalization estimation.

4.5.1 Likely Used and Coverage Estimation Ranges of Estimation Models.

There were three proposed estimation models including phase-wise, adaptive phase-wise, and overall effort estimation models. Figure 26 shows likely used and coverage estimation ranges of all the models. Estimation range of phase-wise and overall estimation models ranges from planning phase to transition phase while that of adaptive phase-wise estimation model ranges from design phase to transition phase. Likely used range of overall and phase-wise estimation models range from planning phase to testing phase while that of adaptive phase-wise estimation model ranges from design phase to testing phase because an effort estimation at design phase requires the effort of planning phase. Note that adaptive phase-wise estimation model can more outperform phase-wise estimation for coding and testing phase as shown in Figure 24.

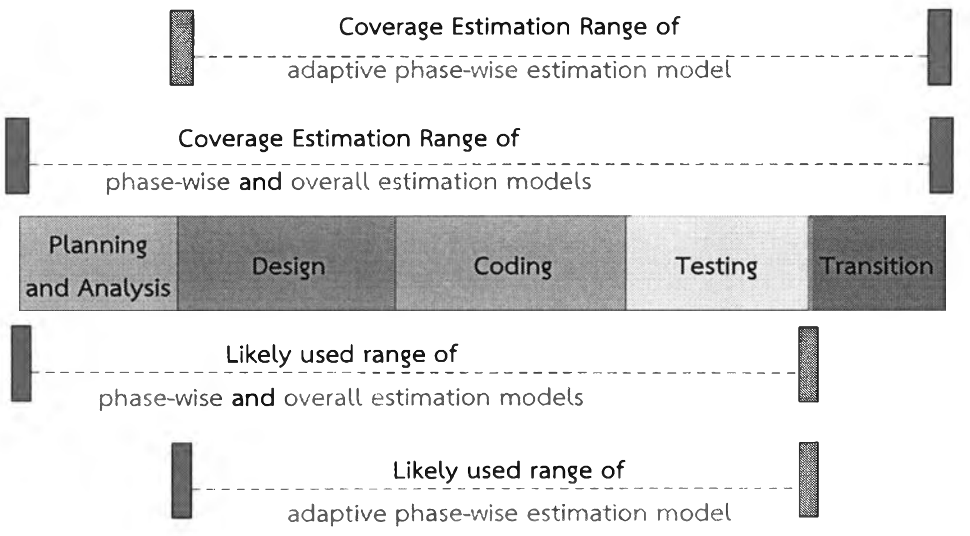

Figure 26: Likely used and coverage estimation ranges of all the proposed models.

4.5.2 Limitations of Estimation Models.

There are some limitations to use these estimation models. Firstly, phase-wise and adaptive phase-wise models are for sequential process models (i.e., Waterfall and V-model) ranging from planning to transition phase. Nevertheless, the proposed approach can be applied to create an estimation model for iteration software processes such as agile development. Secondly, since software size was up to 112.28 thousand of SLOC, the estimation model will be not appreciate for a project that its size is greater than 112.28 thousand of SLOC. However, project manager can apply the proposed approach for high SLOC. Lastly, the proposed model required lines of code as an input like COCOMO II model for predicting an effort. Actually, real lines of code cannot be used to estimate at initial phase. To solve this issue, the first is to use PERT technique or other expert judgment techniques for estimating lines of code. The second is to use function points, use case points, object points, or story points instead of lines of code. Note that backfiring table [6] can be used for conversion between function points and lines of code.

4.5.3 Effect of Programming Language

Since programming langue can effect estimating an effort, they are also indeed investigated in this experiment. As shown in Table 2, there was only one project written java android (project id 27) while most of projects were written by C#. As a result, from 6-fold cross-validation, if project 27 was tested, there was no project written by java android was trained to create an estimation model. Table 21 shows estimation results of projects for investigating effects of programming language. The results were derived from phase-wise and overall estimation models based on MLP. Project 27 can provide good estimation results. Note that platform experience (PLEX) and language and tool experience (LTEX) are required for all phases as shown in Table 20. These features directly affect the differences of programming languages.

1313607376

Table 21: Estimation results for a project written by java android.

Effort in man-days	Phase-wise estimation						Overall estimation
	Plan&Req.	Design	Coding	Testing	Transition	Sum-up	
Estimated	3.00	51.01	180.61	5.00	3.00	242.62	199.93
Actual	2.00	52.00	140.00	10.00	2.00	206	206
Residual	1.00	-0.99	40.61	-5.00	1.00	36.62	-6.07
MRE	0.50	0.02	0.29	0.50	0.50	0.18	0.03