

# An Empirical Analysis of Software Effort Estimation with Outlier Elimination \*

Yeong-Seok Seo, Kyung-A Yoon, Doo-Hwan Bae  
Software engineering laboratory, Division of Computer Science, EECS  
KAIST, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, South Korea  
{ysseo, kayoon, bae}@se.kaist.ac.kr

## ABSTRACT

Accurate software effort estimation has always been challenge for software engineering communities. To improve the estimation accuracy of software effort, many studies have focused on effort estimation methods without any consideration of data quality, although data quality is one of important factors to impact to the estimation accuracy. In this paper, we investigate the influence of outlier elimination upon the accuracy of software effort estimation through empirical studies applying two outlier elimination methods(Least trimmed square and K-means clustering) and three effort estimation methods(Least squares, Neural network and Bayesian network) associatively. The empirical studies are performed using two industry data sets(the ISBSG Release 9 and the Bank data set which consists of the project data performed in a bank in Korea) with or without outlier elimination.

## Categories and Subject Descriptors

D.2 [Software Engineering]: Management; K.6.3 [Management of Computing and Information Systems]: Software Management—*Software process*

## General Terms

Experimentation, Management, Measurement

## Keywords

Effort estimation, Outlier elimination, Software data quality

---

\*This research was supported by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement) (IITA-2008-(C1090-0801-0032)) and was partially supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

PROMISE'08, May 12–13, 2008, Leipzig, Germany.

Copyright 2008 ACM 978-1-60558-036-4/08/05 ...\$5.00.

## 1. INTRODUCTION

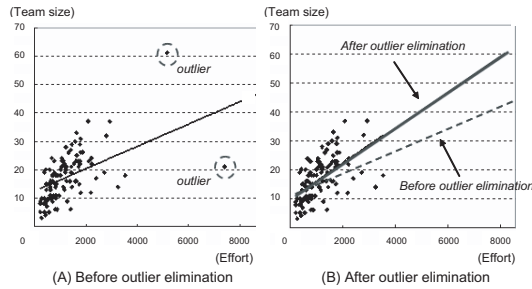
As the variety of software development environment and the software complexity increase, the importance of software project management is more emphasized. For effective software project management, accurate and reliable software cost estimation is essential. In particular, it has an influence on the project success or failure. For example, an underestimation of software project cost causes a schedule delay and over-budget, which finally can lead to the project failure; its overestimation causes a waste of cost owing by over-allocation of development resources.

To estimate an accurate software cost, many organizations use a software effort estimation model built upon their project history data[6, 12, 15, 17, 21]. However, these project history data contain outliers which can degrade project data quality. The outlier is defined as a set of data to be an observation which appears to be inconsistent with the remainder of the set of data[2]. It is caused by (1)the instable project environment such as frequent turnover of developers, (2)the rare event such as performance of large-scaled project in the software organization which mainly performed small-scaled projects, (3)the measurement mistake such as human collector's confusion between LOC and KLOC.

Like the maxim "Garbage In, Garbage Out", if a software effort estimation model is built on the history data containing outliers, it is difficult to obtain the accurate effort estimation results because of the distortion of result by outliers. For example, Figure 1 shows the quantitative relation between team size and development effort using linear regression model. The slope difference of each effort estimation model created before(Figure A) and after(Figure B) eliminating of two outliers is remarkable. Consequently, the outlier elimination is necessary for reliable and accurate software effort estimation based on the history data which reflect the general characteristics of past projects.

Although software data quality is one of important factors which affect the accuracy of software effort estimation, many studies have focused on the development of effort estimation method without any consideration of data quality[6, 12, 15, 17, 21]. Except that few studies mentioned outlier elimination as one of the preprocessing tasks, it is empirically unknown well that the effect of outlier elimination to the accuracy of software effort estimation.

In this paper, we investigate the influence of outlier elimination upon the accuracy of software effort estimation through empirical studies applying two outlier elimination methods and three effort estimation methods associatively. The Least squares, Neural network and Bayesian network are used to



**Figure 1: A variation of effort estimation model by outlier elimination**

software effort estimation because they are commonly used effort estimation methods based on statistics, machine learning and probability respectively. The Least trimmed squares and K-means clustering are used to detect outliers as the representative methods of statistics and data mining respectively. We perform the empirical studies using the following two industry data sets with or without outlier elimination: the ISBSG Release 9[1] and the Bank data set which consists of the project data performed in a bank in Korea. Finally, through the comparison with accuracy of each effort estimation result, we discuss the effect of outlier elimination to the accuracy of software effort estimation with respect to the characteristics of target data set and effort estimation method.

The remainder of this paper is organized as follows. Section 2 briefly introduces three software estimation methods and two outlier elimination methods used in our work as the background. Section 3 describes the related work. In Section 4, the overall approach of our empirical experiment is presented and the detail explanation of tasks and completion results of each step is described. We provide the final experimental result and discussion in Section 5 and conclude this paper with future work in Section 6.

## 2. BACKGROUND

This section introduces the three effort estimation methods and two outlier elimination methods.

### 2.1 Effort estimation methods

The effort estimation methods applied in this paper are Least squares, Neural network and Bayesian network. These methods have entirely different theoretical background like statistics, machine learning and probability respectively. This section describes the essential concepts and contents for effort estimation.

#### 2.1.1 Least squares(LS)

Least squares which generates regression model based on statistic minimizes the sum of squared errors to determine the best estimates for coefficients[9]. This method is the most commonly used method for developing software estimation models[6, 12, 15] due to its simplicity, compared with other effort estimation method, and its availability in most statistical software packages[8].

#### 2.1.2 Neural network(NN)

Neural network is an information processing technique based on machine learning which is composed of the nodes

correspond to neurons and the arcs correspond to synaptic connections in the biological metaphor. NN is initialized with random weights for its arcs and gradually trained using the data sample by adjusting its weights to reduce the distance between actual data and predicted data of the model[11]. Among many different types of NN, the most commonly used type in our research area is feed-forward NN trained by the back-propagation algorithm[8, 6]. Feed-forward NN is designed that each layer contains connections to the next layer without backward connections. Back-propagation algorithm calculates error and adjusts weights of the layers backwards from output layer to input layer[11]. In this paper, we use this feed-forward back-propagation type for software effort estimation using NN.

#### 2.1.3 Bayesian network(BN)

Bayesian network is a probabilistic graphical model that represents a set of variables and their probabilistic dependencies. It consists of an associated set of probability tables and a causal model which describes the cause-effect relationships between variables. The variables have discrete interval values as their states and presented by node. The probability table for each node specifies how the probability of each state of the variable depends on the states of its parents using Bayes' theorem. Consequently, BN represents the complete joint probability distribution. Bayesian analysis using BN is a well-defined and rigorous process of inductive reasoning that has been used in many scientific disciplines[4]. Recently, BN has been used for software effort estimation[4, 17, 22] because traditional software effort estimation models do not provide any support for risk assessment and mitigation using 'what-if' analysis.

## 2.2 Outlier elimination methods

The outlier elimination methods applied in this paper are Least trimmed squares and K-means clustering. These methods have entirely different theoretical background as statistics and data mining respectively. This section introduces not only the fundamentals of these methods but also the concept of outlier from a viewpoint of them.

#### 2.2.1 Least trimmed squares(LTS)

While LS described in Section 2.1.1 minimizes the sum of the squared residuals, Least trimmed squares minimizes the sum of  $h$ , trimming constant, smallest squared residuals[19]. Instead of adding all the squared residuals as in LS, LTS can limit to a trimmed sum of squares by ordering the residuals ( $\epsilon_1^2 \leq \epsilon_2^2 \leq \dots \leq \epsilon_n^2$ ). Therefore, outlier elimination method using LTS regards the posterior trimmed values of  $h$  as outliers. Figure 2 shows an example of outliers identified by LTS. The data having large residual is identified as outlier while the data which is close to the line(linear regression model) that minimizes the sum of the squared residuals is regarded as normal data.

#### 2.2.2 K-means clustering(K-means)

The process of grouping a set of data into classes of similar data is called clustering. A cluster is a collection of data that are similar to one another within the same cluster and are dissimilar to the data in other clusters[10]. Contrary to supervised learning which learns how to perform a task, this method is unsupervised learning which groups together based on similarity by itself. Among many clustering

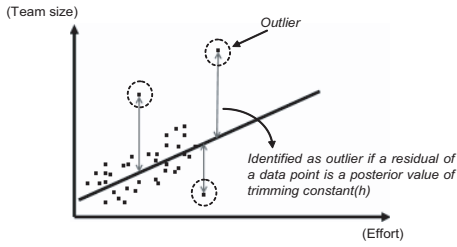


Figure 2: An example of outliers identified on LTS

methods, K-means clustering is popularly used because of its simplicity and efficiency. K-means sets  $K$  initial center points and partitions repeatedly the data into  $K$  mutually exclusive clusters until all clusters have the minimum total Euclidean distance between the data and their center point. To determine the important parameter of this method, the number of clusters( $K$ ), the mean silhouette value is used in this paper. The silhouette value for each data point measures the similarity of the data point with data of its own cluster compared to data of other clusters[14]. It ranges from  $-1$  to  $+1$ . The more close to the  $+1$ , the better clusters are formed for each data point. With a viewpoint of K-means, outlier is regarded as the data which has silhouette value less than  $0$  or which is a element of the cluster whose size is less than the minimum cluster size(the minimum number of data elements) to be meaningful cluster. Figure 3 presents an example of outliers identified by K-means. In our work, the minimum cluster size is  $3$ . Therefore, if data is a element of the cluster containing one or two data elements or has the silhouette value less than  $0$ , it is regarded as outliers.

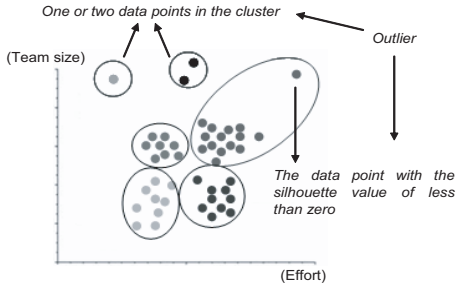


Figure 3: An example of outliers identified on K-means

### 3. RELATED WORK

Until now, the effect of outlier elimination to the software effort estimation has not been investigated much. However, the necessity of the research on this issue has been emphasized as the research interest in the performance improvement of quantitative analysis model based on the quality of measurement data increases. As the recent work, Chan et al.[3] have proposed a methodology to detect and eliminate outliers using Least Median Squares(LMS)[18] before software effort estimation based on the ISBSG Release 6[1]. Although Chan et al. show the outlier elimination is necessary to build an accurate effort estimation model, their work has the following limitations in terms of research scope and experimentation:

- Because this work only used statistical methods for outlier elimination and effort estimation, it cannot show the effect of outlier elimination to the accuracy of software effort estimation on the inappropriate data set to be applied by the statistical method; for example, the data whose distribution is unknown. Therefore, the application of various methods of outlier elimination and effort estimation having different theoretical background also needs to be investigated for our issue.
  - The evaluation of effort estimation accuracy was not insufficient in this work. Although various evaluation criteria are provided to measure the accuracy of prediction model, this work used only MMRE(The Mean Magnitude of Relative Error) which has the following weakness: the value may be strongly influenced by a few predictions with large MREs [12, 13]. Therefore, other evaluation criteria should be used to more reliable evaluation of the effort estimation accuracy.
  - The data characteristics were not considered to build the effort estimation model in this work. The ISBSG data contains many software project data with different characteristics collected from worldwide organizations in various business domains such as financial, insurance, manufacturing, and so forth. To obtain more accurate estimation result, it is better to use the data set after categorization according to similar characteristics such as business domain.
- Instead of LMS, LTS is used in this paper, because LTS build more robust estimation model for outlier than LMS theoretically[19].

## 4. EMPIRICAL EXPERIMENT

This section describes the overall approach of our empirical experiment. As shown in Figure 4, the overall process consists of four steps. In Step 1, a data preprocessing is performed to prepare the appropriate data set for the purpose of experiment by accomplishing proper attributes and data selection, missing data handling, normality and correlation test, and data normalization. In Step 2, the training and testing data set are prepared according to the concept of 10-fold cross validation, and the outlier detection and elimination are performed by outlier elimination method. In Step 3, effort estimation models are built using the prepared data set with and without outlier elimination. The accuracy of software effort estimation result is compared and analyzed in terms of outlier elimination method, software effort estimation method and data characteristics in Step 4. The following sections present the description of target data set and experimental environment and the detailed explanation of tasks and interim results of each step.

### 4.1 Description of target data set and experimental environment

To investigate the effect of outlier elimination to the accuracy of software effort estimation, the ISBSG data set and the Bank data set are used to our work.

The ISBSG Release 9[1] is publicly available multi-company data set which contains software project data collected from various organizations around the world from 1989 to 2004. This data set has been used by many studies focused on the issue of software effort estimation in spite of the diversity of its data elements.

The Bank data set collected from 2005 to 2006 in a financial company in Korea. Since this company acquired the

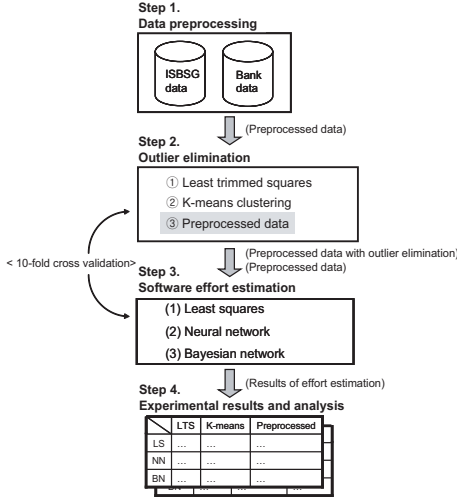


Figure 4: The overall process of our experiment

SW-CMM level 3 certification, the software project information has been measured continuously. This data set is managed using a measurement supporting tool, which provides simple data management functions such as the record and basic analysis of data. Because this tool can't automatically measure or collect software project data, user has to insert data manually using this tool.

With these multi-company and single-company data set, all outlier elimination methods and two software effort estimation methods using LS and NN are performed using MATLAB v7.3.0.267. The software effort estimation using BN is carried out using BayesiaLAB v4.3.1. Statistical significance was set at 0.05.

## 4.2 Data preprocessing

In this step, the ISBSG and the Bank data are preprocessed to obtain the appropriate data set for the purpose of experiment by performing the selection of attributes and data elements from two target data set, the handling of missing data, the normality and correlation test and the data normalization. At first, in the ISBSG data set, we selected 99 project data and 6 attributes as shown in Table 1 based on data quality rating and same business domain (bank domain) among 3,024 projects data. In the Bank data, 120 project data and 6 attributes were selected as shown in Table 1 by excluding the data having the zero or negative value on the attribute *Effort*.

After the selection of attributes and data elements, missing data handling was performed using imputation method. The ISBSG data set had 6% and 4% of missing values on the attribute *Duration* and *Lan* respectively and the Bank data set had 2.5% of missing values on the attribute *KAELOC*. We applied K-Nearest Neighbor(K-NN) imputation to these missing values[21].

Because we use two statistical methods, LS as effort estimation method and LTS as outlier elimination method, normality and correlation test are necessary to know whether the target data are under the assumption of statistical method. To confirm normality of ratio-scaled attributes, Shapiro-Wilks test was used[16] after log transformation of attributes. Also, dummy variables[9] were generated for the nominal-

scaled attributes. Thus, two-tailed Pearson's correlation test and one-way ANOVA test were used to assess the relationship between attributes. After completion of all normality and correlation tests, the data were normalized to have values ranging from 0 to 1. The data normalization is important to K-means because it prevents the distortion of clustering result caused by differences in scale across different attributes.

## 4.3 Outlier elimination

In this step, two outlier elimination methods are applied to the preprocessed data. In Step 2 and 3, 10-fold cross validation was applied to the data sets to obtain reliable results. The approach divides the whole data set into 10 folds, and then 9 folds are used for the training set and the remaining fold is used for the testing set. After building the effort estimation model using 9 folds, compute an estimation accuracy by the evaluation criteria(Section 4.5) using the remaining one testing fold. Finally, the accuracy results across all the folds are aggregated by average. Outlier elimination methods were applied to the 10 training set on the two data sets. Figure 5 shows an example of applying LTS and K-means to the 10 training set on the ISBSG data using 10-fold cross validation.

The trimming constant( $h$ ) for LTS can be chosen between  $\frac{n}{2}$  and  $n$  ( $n$  is the total number of data). The default value which is roughly  $0.75n$ [20] was used in this paper. The  $K$  value for K-means was chosen by comparing the mean silhouette values. After the K-means clustering was applied using the  $K$  value, the data point which has the silhouette value less than zero and which is included in the cluster whose size is one or two was identified as outlier and then eliminated. Table 2 shows the number of outlier detected by LTS and K-means on the two data sets. A different number of outliers were detected by the outlier elimination methods, and more small number of outliers were detected in the Bank data set than the ISBSG data set.

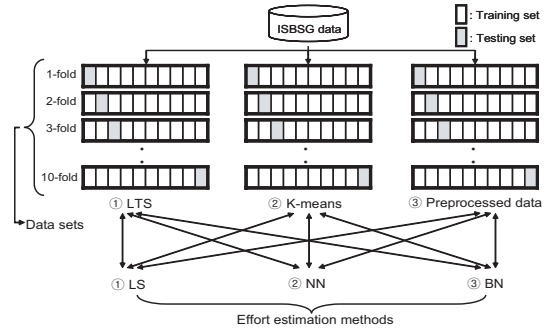


Figure 5: Applying effort estimation methods with outlier elimination methods using 10-fold cross validation

## 4.4 Software effort estimation

As shown in Figure 5, effort estimation method were applied to the 10 training sets with outlier elimination by LTS, and the 10 training sets with outlier elimination by K-means, and the preprocessed data without outlier elimination. Average value of the 10 effort estimation accuracies using each 10 testing set is used as the accuracy of the effort estimation

**Table 1: Variable description of the ISBSG data and Bank data used in our work**

	Name of variable	Description	Mean	Std.dev	Type	
ISBSG data(99)	Normalized work effort	Total project effort in person hours	3745.85	3859.41	Dependent variable	
	Adjusted function points (AFP)	The adjusted function point count	443.73	1493.76	Continuous independent variables	
	Project elapsed time (Duration)	Duration for the project in calendar months	7.77	7.37		
	Development type (DT)	Main development type used. Dummy variable where 'enhancement' is coded as 1 and 'new development' is coded as 0			Categorical independent variables	
	Development platform (DP)	Main development platform used				
		DP_Multi	Dummy variable where 'Multi' platform is coded as 1 and others are coded as 0			
		DP_MR	Dummy variable where 'Midrange' platform is coded as 1 and others are coded as 0			
	Language (Lan)	Main language type used				
		Lan_3GL	Dummy variable where '3GL' language is coded as 1 and others are coded as 0			
Lan_4GL		Dummy variable where '4GL' language is coded as 1 and others are coded as 0				
Bank data(120)	Effort	Total project effort in person hours	1935.57	3769.25	Dependent variable	
	KAELOC	Total thousand assembly equivalent lines of code	340.65	1327.61	Continuous independent variables	
	Project elapsed time (Duration)	Duration for the project in calendar days	68.71	57		
	Max team size (MTS)	Maximum number of members	20.76	17.26		
	Development platform (DP)	Main development platform used				Categorical independent variables
		DP_Host	Dummy variable where 'Host' platform is coded as 1 and others are coded as 0			
		DP_Unix	Dummy variable where 'Unix' platform is coded as 1 and others are coded as 0			
		Life cycle model (Model)	The life cycle model on the project			
	Model_Inc	Dummy variable where 'Incremental' model is coded as 1 and others are coded as 0				
Model_V		Dummy variable where 'V' model is coded as 1 and others are coded as 0				

**Table 2: The number of detected outlier**

	# of detected outlier			
	ISBSG		Bank	
	LTS	K-means	LTS	K-means
1-fold	12	7	11	2
2-fold	15	6	11	7
3-fold	16	6	10	7
4-fold	14	8	11	5
5-fold	15	5	11	8
6-fold	14	6	12	5
7-fold	12	4	11	6
8-fold	12	2	13	5
9-fold	14	0	11	7
10-fold	14	6	10	6
Average	13.8	5.0	11.1	5.8
Average %	13.939%	5.051%(7.6)	9.25%	4.833%(5.8)
( ) : Average number of clusters for each fold				

model depending on each outlier elimination method and preprocessed data.

The best fitting software effort models using LS presented as the following equations:

• ISBSG data:

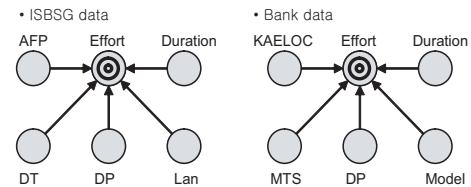
$$\log(Effort) = 1.7156 + 0.6186 * \log(AFP) + 0.3063 * \log(Duration) + (-0.21) * \log(DT) + (-0.0833) * DP\_Multi + 0.1341 * DP\_MR + 0.1178 * Lan\_3GL + (-0.1009) * Lan\_4GL$$

• Bank data:

$$\log(Effort) = 0.9222 + 0.1725 * \log(KAELOC) + 0.5314 * \log(Duration) + 0.6823 * \log(MTS) + 0.0398 * DP\_Host + 0.0571 * DP\_Unix + (-0.0093) * Model\_V + (-0.0424) * Model\_Inc$$

In NN, all steps to build effort estimation model operate as black box, thus so we only know the results of effort estimation.

In the case of BN, the causal models as shown in Figure 6 were used to create the probability table of BN for effort estimation.



**Figure 6: Causal models for effort estimation**

## 4.5 Experimental results and analysis

The effort estimation accuracy was measured using MMRE [8, 12], MdMRE[12], Pred(0.25)[12, 13], and Pred(0.5)[12, 13]. MMRE is one of the most widely used criteria. It is, however, sensitive to a few high MRE[12, 13] and to small

actual values[7]. MdmRE is included to make up for the weak points of MMRE. It is less sensitive to extreme values than MMRE. The small value of MMRE and MdmRE and the large value of Pred(0.25) and Pred(0.5) indicate that the estimation model has good accuracy. The estimation accuracy of  $MMRE \leq 0.25$  and  $Pred(0.25) \geq 0.75$  can be considered as acceptable levels of estimation accuracy[5]. Our experimental results are provided in the next section.

## 5. EXPERIMENTAL RESULT AND DISCUSSION

This section describes the final experimental results with the discussion.

### 5.1 Results on the ISBSG data

Table 3 shows the estimation accuracy of LS, NN, and BN using the ISBSG data set with outlier elimination by LTS and K-means and without outlier elimination. Each result is calculated by averaging out 10 estimation accuracy values which were generated by 10-fold cross validation. The value presented by bold font in the Table 3 indicates the best estimation accuracy for each effort estimation method.

**Table 3: Results for effort estimation models with outlier elimination methods on the ISBSG data**

		ISBSG		
		LTS	K-means	Preprocessed
		Averages from 10 fold cross validation		
LS	MMRE	<b>0.7301</b>	0.7447	0.7417
	MdmRE	<b>0.4485</b>	0.4328	0.4381
	Pred(0.25)	<b>0.2822</b>	0.2611	0.2711
	Pred(0.5)	<b>0.5844</b>	0.5744	0.5744
NN	MMRE	<b>0.6232</b>	0.7164	0.7181
	MdmRE	<b>0.3644</b>	0.3818	0.4378
	Pred(0.25)	<b>0.3556</b>	0.3322	0.2822
	Pred(0.5)	<b>0.6167</b>	0.5967	0.5878
BN	MMRE	0.9941	<b>1.0468</b>	0.9848
	MdmRE	0.6054	<b>0.5537</b>	0.6044
	Pred(0.25)	0.2222	<b>0.2844</b>	0.2433
	Pred(0.5)	0.4133	<b>0.4656</b>	0.4556

The ISBSG data is collected from various worldwide organizations. As the data points which have diverse characteristics are widely scattered, it is hard to build accurate effort estimation models. The results are not favorable because the minimum MMRE is large as 0.6232 and the maximum Pred(0.25) is small as 0.3556. However, though the data with these multi-company characteristics, the accuracy of effort estimation model is improved by outlier elimination. Therefore, outlier elimination is important factor to build the accurate effort estimation model.

LS shows that applied outlier elimination methods do not support the remarkable improvement of effort estimation accuracy compared with NN and BN. The variation of four evaluation criteria is approximately equal. In our result, the most accurate outlier elimination method for LS is LTS because of the smallest MMRE, the largest Pred(0.25) and Pred(0.5). However, the estimation accuracy is not much improved.

NN presents the best estimation accuracy results for each outlier elimination method in terms of all evaluation criteria. Although the estimation accuracy is not enough to be

accurate according to the acceptable levels of estimation accuracy described in Section 4.5, NN performs better than other effort estimation methods with and without outlier elimination. In our result, the most accurate outlier elimination method for NN is LTS because of the smallest MMRE and MdmRE, the largest Pred(0.25) and Pred(0.5).

BN presents the worst estimation accuracy results compared with the results of LS and NN in terms of all evaluation criteria. The result is dropped even though LTS is applied to eliminate outliers. None of the three results for LTS, K-means, and preprocessed data are better than the estimation accuracies of the other effort estimation methods. In our result, the most accurate outlier elimination method for BN is K-means because of the smallest MdmRE, the largest Pred(0.25) and Pred(0.5).

Our experimental result shows that NN with LTS presents the most accurate effort estimation result on the ISBSG data.

### 5.2 Results on the Bank data

Table 4 shows the same type of the estimation accuracy results using the Bank data set.

**Table 4: Results for effort estimation models with outlier elimination methods on the Bank data**

		Bank		
		LTS	K-means	Preprocessed
		Averages from 10 fold cross validation		
LS	MMRE	0.3183	<b>0.3161</b>	0.3291
	MdmRE	0.1933	<b>0.2051</b>	0.2123
	Pred(0.25)	0.5750	<b>0.5917</b>	0.5500
	Pred(0.5)	0.8333	<b>0.8583</b>	0.8417
NN	MMRE	0.2962	<b>0.2772</b>	0.3052
	MdmRE	0.1774	<b>0.1523</b>	0.1840
	Pred(0.25)	0.6250	<b>0.7083</b>	0.6333
	Pred(0.5)	0.8417	<b>0.8500</b>	0.8417
BN	MMRE	0.8203	<b>0.5569</b>	0.7584
	MdmRE	0.3811	<b>0.3132</b>	0.3950
	Pred(0.25)	0.3750	<b>0.4167</b>	0.3583
	Pred(0.5)	0.6000	<b>0.6750</b>	0.5667

Contrary to the ISBSG data, the Bank data is collected from single financial company. Therefore, it contains relatively many data with similar characteristics and less outliers than the ISBSG data as shown in Table 2. Consequently, more accurate estimation can be expectable using the historical data. Actually, the experimental results are favorable because the minimum MMRE is 0.2772 and the maximum Pred(0.25) is 0.7083. The accuracies of effort estimation models with outlier elimination are improved, like the results of the ISBSG data. Therefore, outlier elimination is also important factor on the Bank data to build the accurate effort estimation model.

Similar to the LS results on the ISBSG data, the LS results on the Bank data also present a small improvement of effort estimation accuracy compared with the result of NN and BN after outlier elimination. Although the most accurate outlier elimination method for LS is K-means in our result because of the smallest MMRE, the largest Pred(0.25) and Pred(0.5), the estimation accuracy is not much improved.

NN presents the best estimation accuracy results for each outlier elimination method in terms of all evaluation criteria.

ria. Although the estimation accuracy is quite accurate, the accuracy is improved when the outlier elimination methods are applied. In our result, the most accurate outlier elimination method for NN is K-means because of the smallest MMRE and MdmRE, the largest Pred(0.25) and Pred(0.5).

BN presents the worst estimation accuracy results for each outlier elimination method in terms of all evaluation criteria. However, the results of effort estimation accuracy show the largest variation in our experimental results. In our result, the most accurate outlier elimination method for BN is K-means because of the smallest MMRE and MdmRE, the largest Pred(0.25) and Pred(0.5).

Our experimental results show that NN with K-means presents the most accurate effort estimation result on the Bank data.

### 5.3 Discussion

As shown in Table 3 and Table 4, the different results are observed on the ISBSG and the Bank data set.

The outlier elimination using K-means does not have good performance on the ISBSG data set because of the large standard deviation of this data set. The large variation of data distribution is one of the characteristics of the data collected from multi-company or single-company which has instable software process. This characteristic causes the poor clustering results. Table 5 presents the silhouette values of the ISBSG and the Bank data set. The mean silhouette value of the ISBSG data set is smaller than that of the Bank data set. It indicates that the quality of clustering on the ISBSG data set is poorer than that on the Bank data set. Therefore, there is the possibility that some outliers may not be detected because they may be assigned incorrectly to normal cluster. Consequently, these outliers can be included to build effort estimation models as normal data and degrade the estimation accuracy of the effort estimation model. Our experimental results show that LTS is more effective outlier elimination method than K-means on the ISBSG data set in terms of the estimation accuracy. Because LTS regards the data as outlier when the data is over the trimming constant, widely scattered data are identified as outliers and the relatively similar data are remained. This influences on the accurate effort estimation. However, the effort estimation using BN with K-means has more accurate result than the associative application of BN with LTS. BN does not seem to be influenced above characteristics.

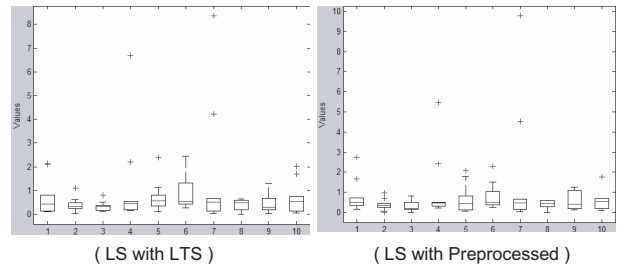
**Table 5: Mean silhouette values on both data sets**

	Used 10 mean silhouette values	
	ISBSG data	Bank data
1-fold	0.5261	0.7336
2-fold	0.5294	0.7525
3-fold	0.5162	0.7599
4-fold	0.5609	0.7725
5-fold	0.5338	0.749
6-fold	0.5231	0.7424
7-fold	0.5354	0.7673
8-fold	0.5754	0.7581
9-fold	0.5353	0.7567
10-fold	0.522	0.7873

However, K-means is better than LTS for three effort estimation method on the Bank data. The Bank data are

organized with the project data with similar characteristic and smaller standard deviation than ISBSG data. Therefore, clusters are well separated from the other clusters, and the outliers can be identified more clearly by K-means. Table 5 also shows the better clusters are built on the Bank data. This has influences on the accurate effort estimation. However, the performance of outlier elimination by LTS is decreased. Although few outliers are in the the data set, the data point over the trimming constant is always identified as outlier. Therefore, though the data point is not outlier, the point can be identified as outlier and eliminated. As a result, effort estimation accuracy can be degraded though the outliers are eliminated.

It is noticeable that LS with outlier elimination methods is not significantly more accurate than LS without outlier elimination. LS is the well-known model which is sensitive to outliers[9]. However, LS with outlier elimination is not performed better in our study. This is caused by the outliers in the testing set. The estimation accuracy of the training set for LS with outlier elimination is more improved, because the model is more fitted to the training set with outlier elimination. Therefore, the estimation accuracy of the testing set is more degraded than that of LS without outlier elimination. For example, Figure 7 shows box plots of the MREs in 10-fold testing sets for LS with LTS and LS with preprocessed data based on the ISBSG data. The cross symbols present the outliers in each fold. As shown in Table 6, mean and median MREs of the all outliers in 10-fold testing set of LS with LTS are more inaccurate than LS with preprocessed data. In our study, effort estimation accuracies are average values from 10-fold cross validation. Therefore, the estimation accuracies of LS with and without outlier elimination are not significantly different.



**Figure 7: Box plots of the MREs in 10 testing sets based on the ISBSG data**

**Table 6: Mean and median MREs of the outliers in 10 testing sets based on the ISBSG data**

	LS with LTS	LS with preprocessed
Mean	3.0672	2.7770
Median	2.163	2.08

NN provides the best estimation accuracy on both the ISBSG and the Bank data. However, the outlier elimination method applied to NN for the best estimation accuracy is different. The best choice is NN with LTS on the ISBSG data and NN with the K-means on the Bank data. Neural network performs well when the training set contains similar or redundant data point. In our study, as stated above, the training set with LTS on the ISBSG data and with K-means

on the Bank data include more similar or redundant data relatively. This characteristic may have a great influence to the estimation accuracy of NN

The estimation accuracy of BN is the worst in three estimation models. None of the three average results for LTS, the K-means and preprocessed data were better than other results. BN is not largely affected by outlier elimination method. However, BN with K-means is more accurate than BN with LTS on both data sets.

## 6. CONCLUSIONS

There are inevitably a few outliers in the software project data. When software effort estimation models are built using the data samples with outliers, these models degrade the effort estimation accuracy for future projects. Therefore, in this paper, we examined the estimation accuracy of effort estimation models when applying outlier elimination methods on multiple- and single-company data sets. We used two outlier elimination methods and three effort estimation methods which have different theoretic background. Our empirical study shows that the applied outlier elimination methods improve the estimation accuracy of software effort estimation models. On the other hand, the effects of outlier elimination to the accuracy of effort estimation are different depending on the characteristics of data set, effort estimation method and outlier elimination method. In our result, the application of NN and LTS on the ISBSG data set and NN and K-means on the Bank data set as the effort estimation method and outlier elimination method respectively present the most accurate software estimation results. However, our result only shows the small part of unanswered questions. There are a number of work to be remained belongs to the research issue of our work. In order to derive more general results, more empirical study is required with other data sets. Recently, we take another software project data set from a single financial company. This data set consists of the same attributes with the Bank data set. Therefore we plan to perform the same experiment of this work on two single company data sets having the same domain and attributes. Additionally, the application of other outlier elimination methods such as CART(Classification and Regression Trees) are planned. Finally, we expect to find more valuable investigation results by strict analysis and interpretation in this future work.

## 7. REFERENCES

- [1] International software benchmarking standards group. <http://www.isbsg.org>, 2005.
- [2] V. Barret and T. Lewis. *Outliers in Statistical Data*. Wiley Series in Probability and Statistics, 1994.
- [3] V. Chan and W. Wong. Outlier elimination in construction of software metric models. *Proceedings of the 22nd ACM Symposium on Applied Computing*, pages 1484–1488, 2007.
- [4] S. Chulani, B. Boehm, and B. Steece. Bayesian analysis of empirical software engineering cost models. *IEEE Transactions on Software Engineering*, 25(4):573–583, 1999.
- [5] S. Conte, H. Dunsmore, and V. Shen. *Software Eng. Metrics and Models*. Benjamin/Cummings Publishing Company, 1986.
- [6] I. de Barcelos Tronto, J. da Silva, and N. Sant’Anna. Comparison of artificial neural network and regression models in software effort estimation. *International Joint Conference on Neural Networks*, pages 771–776, 2007.
- [7] T. Foss, E. Stensrud, B. Kitchenham, and I. Myrteit. A simulation study of the model evaluation criterion mmre. *IEEE Transactions on Software Engineering*, 29(11):985–995, 2003.
- [8] A. Gray and S. MacDonell. A comparison of techniques for developing predictive models of software metrics. *Information and Software Technology*, 39(6):425–437, 1997.
- [9] L. Hamilton. *Regression with Graphics, A Second Course in Applied Statistics*. Duxbury Press, 1992.
- [10] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2006.
- [11] J. Heaton. *Introduction to Neural Networks with Java*. Heaton Research, Inc, 2005.
- [12] M. Jorgensen. Experience with the accuracy of software maintenance task effort prediction models. *IEEE Transactions on Software Engineering*, 21(8):674–681, 1995.
- [13] B. Kitchenham, S. MacDonell, L. Pickard, and M. Shepperd. Assessing prediction systems. *The Information Science Discussion Paper Series, University of Otago*, 1999.
- [14] S. Lamrous and M. Taileb. Divisive hierarchical k-means. *International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, page 18, 2006.
- [15] E. Mendes, C. Lokan, R. Harrison, and C. Triggs. A replicated comparison of cross-company and within-company effort estimation models using the isbsg database. *11th IEEE International Software Metrics Symposium*, page 36, 2005.
- [16] M. Mendes and A. Pala. Type i error rate and power of three normality tests. *Pakistan Journal of Information and Technology*, 2(2):135–139, 2003.
- [17] P. Pendharkar, G. Subramanian, and J. Rodger. A probabilistic model for predicting software development effort. *IEEE Transactions on Software Engineering*, 31(7):615–624, 2005.
- [18] P. Rousseeuw. Least median of squares regression. *Journal of American Statistical Association*, 79(388):871–880, 1984.
- [19] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons, Inc, 1987.
- [20] P. Rousseeuw and K. van Driessen. Computing lts regression for large data sets. *Data Mining and Knowledge Discovery*, 12(1):29–45, 2006.
- [21] Q. Song and M. Shepperd. A new imputation method for small software project data sets. *Journal of Systems and Software*, 80(1):51–62, 2007.
- [22] T. H. Song, K. A. Yoon, and D. H. Bae. An approach to probabilistic effort estimation for military avionics software maintenance by considering structural characteristics. *Asia-Pacific Software Engineering Conference*, pages 406–413, 2007.